The Logical Structure of Emotions
This work has been supported by SenterNovem, Dutch Companion project grant nr: IS053013.

SIKS Dissertation Series No. 2010-23

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

© 2010 Bas R. Steunebrink
Typeset by the author using \LaTeX\ 2ε.
Printed by GVO drukkers & vormgevers B.V. | Ponsen & Looijen, Ede.

The Logical Structure of Emotions

De Logische Structuur van Emoties

(met een samenvatting in het Nederlands)

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de rector magnificus, prof.dr. J.C. Stoof, ingevolge het besluit van het college voor promoties in het openbaar te verdedigen op dinsdag 27 april 2010 des ochtends te 10.30 uur

door

Bastiaan Reinier Steunebrink

geboren op 27 mei 1983 te Leidschendam.
Promotor:        Prof.dr. J.-J.Ch. Meyer
Co-promotor:     Dr. M.M. Dastani
# Contents

1 Introduction ................................................................. 1
   1.1 A Brief History of Emotion ........................................... 1
   1.2 What is Emotion Really? .............................................. 3
      1.2.1 A Classification .................................................... 4
      1.2.2 Duration as Distinguishing Factor ............................. 4
      1.2.3 A Procedural View of Emotion ................................. 6
   1.3 Emotions for Artificial Agents ...................................... 9
   1.4 Research Questions .................................................. 12
   1.5 Overview of this Dissertation ..................................... 14

2 Emotion in Psychology .................................................... 17
   2.1 From Cognitive Psychology to Artificial Agents .................. 18
   2.2 Overview of the OCC Model ......................................... 19
   2.3 The Logical Structure of the OCC Model .......................... 23
      2.3.1 General Emotion Types .......................................... 23
      2.3.2 Consequences of Events ......................................... 24
      2.3.3 Actions of Agents ................................................ 25
      2.3.4 Compound Emotion Types ....................................... 26
      2.3.5 Fortunes-of-others Emotion Types ............................. 27
      2.3.6 Prospect-based Emotion Types ................................. 28
   2.4 Concluding Remarks ................................................ 30

3 Emotion Elicitation I: The Logical Structure ......................... 37
   3.1 The Procedural View of Emotion Revisited ........................ 37
   3.2 Formalization of Emotion Triggers ................................. 39
      3.2.1 General Emotion Types .......................................... 40
      3.2.2 Concrete Emotion Types ....................................... 44
      3.2.3 Compounds ....................................................... 47
      3.2.4 Derived Emotion Types ........................................ 49
B.2.6  Some Final Remarks ................................. 207

References ................................................. 209

Acknowledgments ......................................... 215

Summary .................................................. 217

Samenvatting .............................................. 219

SIKS Dissertation Series ............................... 221
Chapter 1

Introduction

Everyone knows what an emotion is, until asked to give a definition.
– Fehr and Russell [1984]

In a nutshell, this dissertation builds a bridge between an informal psychological model of emotions on the one hand, and implementation of emotions in robots and virtual characters on the other hand. This is done by formalizing a psychological model of emotions. With that we\(^1\) mean translating the concepts used in the psychological model to a logical language with unambiguous semantics. The resulting formalization of emotions will give rise to theorems that can be proved to be true. These theorems are properties of the formal model and can be checked against the original psychological model of emotions. If they match, this will mean that the formalization is truthful with respect to the psychological model. The description of the model of emotions in this formal language can then be used as a foundation for implementing emotions in robots and virtual characters.

In this introduction, we provide a brief overview of historical attempts at understanding emotion, why we are interested in emotion, and what are the precise research questions addressed in this dissertation.

1.1 A Brief History of Emotion

Some of the most important ideas about emotion in contemporary cognitive psychology can already be found in ancient literature. For example, two ideas that are already present in the writings of Aristotle are that emotions derive from what one believes and that there is a relationship between emotion and action [Oatley and Jenkins, 1996]. Discussions on emotion by Aristotle can be found in his book *Rhetoric* and to a lesser extent also in his book *Poetics*. Aristotle was mainly concerned with the practical and ethical use of emotion elicitation in audiences of public speaking (*Rhetoric*) and tragic drama (*Poetics*). In *Rhetoric*, Aristotle provided an analysis of

\(^1\)I will use “we” throughout this dissertation, not as *pluralis maiestatis*, but as *pluralis modestiae*. This work has been conceived with the help of many people, so I will use “we” to speak for all of them.
several emotions by breaking them down into what beliefs they presuppose (e.g., anger requires the belief that oneself or one’s friends are subject to wrongdoing), their valence (e.g., anger is unpleasant), their associated actions (e.g., anger gives an urge to take revenge), and their cognitive effects (e.g., anger colors further judgments). This analysis was the first known cognitive approach to thinking about emotion in Western history [Oatley and Jenkins, 1996]. Furthermore, Aristotle’s idea of breaking different types of emotion down into parts will be an important theme in the formalization presented in this dissertation.

Another important historical account of emotion is provided by René Descartes in his book *The Passions of the Soul* [Descartes, 1649]. In this seventeenth century book, Descartes discusses in which way many different types of emotion can be produced by combining six fundamental emotions: wonder, desire, joy, love, hatred, and sadness. According to Descartes, emotions occur both in one’s soul and in one’s body. The soul is one’s thinking aspect, so from this perspective emotions arise from one’s cognitions regarding the outside world. On the other hand, emotions also impact one’s body (e.g., blushing with shame, trembling with fear, tears of joy). The title of Descartes’ book can then be explained as follows. Descartes defines “passion” as something suffered by a person, in contrast to “action” which is defined as something produced by a person. From this perspective, emotions are the passions of the soul, i.e., the things suffered by our thinking aspect. It should be noted, though, that despite this apparent “suffering,” Descartes recognized that emotions often serve a useful function by directing one’s thoughts to what is important. The insight that can be drawn from this view of emotion is neatly formulated by the psychologists Oatley and Jenkins [1996, page 16] as follows:

“Descartes’s [sic] fertile idea triggers the thought that just as perceptions tell us about what is important in the outside world, just as hunger and pain tell us about important events in the body, emotions tell us what is important in our souls – as we might now say, in our real selves.”

The first modern landmark book on emotion was written by Charles Darwin, called *The Expression of the Emotions in Man and Animals* [Darwin, 1872] and published thirteen years after *The Origin of Species*. As the title suggests, the emphasis of Darwin’s book is not on emotion as such, but more on emotional expressions. Although Darwin’s interest in emotional expressions came from his effort to gather evidence for the evolution of the human species, he did not present emotional expressions as beneficial for survival. On the contrary, Darwin’s idea about emotional expressions was that they are remnants of behaviors that were once useful in our evolutionary past. Darwin thus used emotional expressions much like fossils; that is, to show the continuity of adult human behavior with the behavior of infants and lower animals. Although Darwin recognized that certain emotional expressions may facilitate social communication, his main idea was that emotional expressions are involuntary and indicative of our primitive origins. Although this concept of emotional expressions as childish, disturbing, and irrational was not new (cf. Stoicism), Darwin provided it with a plausible biological foundation.

With the rise of cognitive psychology in the second half of the twentieth century, the idea of emotions as arising from perceptions and beliefs has become generally
accepted. A central concept here is that of appraisal: every emotion is the result of the evaluation of an event, action, or object [Ortony et al., 1988]. An emotion is thus always relative to something. But more importantly, and in contrast to Darwin’s ideas about the expression of emotions, neuropsychological research has begun to show the importance—even the necessity—of emotion in the lives of humans. The evidence for this comes mainly from investigations into individuals with damaged frontal lobes. Extensive damage to the frontal lobes has been shown to make one incapable of experiencing emotions; however, instead of becoming ‘super-rational’, sufferers of frontal lobe damage are characterized by a devastating incapability of making sensible social decisions and of deciding which issues are worth deliberating about. To explain these observations, Damasio [1994] proposed the somatic marker hypothesis, which states that mere thinking of possible decisions causes learned emotional reactions to indicate whether the outcome will be good or bad. The idea is thus that, overtly, emotions deliver cognitive information via ‘gut feelings’ and, more covertly, that emotions steer cognitive reasoning via neuromodulation. This way emotions effectively prune unpromising directions of reasoning and prevent the consideration of socially punishing courses of action. If emotion is indeed responsible for such information delivery and pruning, this would explain why the ability to make intelligent decisions is greatly impaired in persons incapable of experiencing emotions.2 Damasio does recognize that emotions can sometimes be detrimental to reasoning, but a life without emotion is a much worse fate:

“When emotion is entirely left out of the reasoning picture, as happens in certain neurological conditions, reason turns out to be even more flawed than when emotion plays bad tricks on our decisions.”

(Quoted from [Damasio, 1994, page xii].)

1.2 What is Emotion Really?

We have briefly discussed some of the most important historical views on emotion, but what is emotion according to the most recent insights? There are many words in the English language, and undoubtedly in most other languages as well, that appear to have something to do with emotion. For example, besides the word ‘emotion’ itself, there are words like ‘affect’, ‘mood’, ‘impulse’, ‘tendency’, and ‘temperament’ that appear to describe something emotion-like. All these terms carry different connotations, but how is emotion distinguished from them?

2 It may be worth noting that Sloman [2004] has pointed out that the reasoning steps leading to Damasio’s hypothesis are fallacious:

“Many wishful thinkers (…) fail to see the obvious flaw in Damasio’s widely quoted reasoning [Damasio, 1994] from the premise: Damage to frontal lobes impairs both intelligence and emotional capabilities to the conclusion Emotions are required for intelligence. A moment’s thought should show that two capabilities could presuppose some common mechanism without either capability being required for the other.”

(Italics in original.) Nevertheless, Sloman acknowledges that the conclusion that emotions are required for intelligent behavior in an environment with limited resources is probably true, despite this weakness in Damasio’s argument.
1.2.1 A Classification

It is widely recognized among psychologists that classifications distinguishing emotion from related affective phenomena (such as mood and temperament) invariably contain some ‘grey areas’. Nevertheless, this does not mean that making broad classifications is impossible or useless. According to a classification by Gross and Thompson [2007], emotions typically have specific objects and give rise to action tendencies relevant to these objects. For example, ‘anger’ is always directed to someone or something and gives the tendency to remove the irritant. Moreover, emotions can be both positive and negative. For example, anger about being reprimanded is a negative emotion, whereas happiness about winning a game is a positive emotion. Emotions are distinguished from moods, which are more diffuse and last longer than emotions. For example, ‘feeling down’ can be used to describe a long-term negative affective state that may not have a clear single cause. Other affective processes include stress, which arises in taxing circumstances and produces only negative responses; and impulses, which are related to hunger, sex, and pain and give rise to responses with limited flexibility. It is important to note that, of these four types of affective processes, we will focus on emotions in this dissertation.

In psychological literature, the word affect is normally used as an aggregate term for emotion, mood, stress, and impulse, as distinguished above [Gross and Thompson, 2007]. Affect thus refers to the broader concept of evaluating things with respect to one’s personal well-being. A further distinction can be made between affective dispositions and affective states, as done by Clore et al. [2001, page 29–30]:

“Analogous to the term cognitive, which refers to representations of knowledge (truth and falsity), the term affective refers to representations of value (goodness and badness). Affective thus designates a broad category of things with positive and negative personal value. These include preferences and attitudes, which are affective dispositions, and emotions and moods, which are affective states.”

(Italics in original.) So any particular emotion (e.g., anger) is an affective state. However, emotion as such can also be used to describe an affective process; namely, the process of evaluating events, actions, and objects as good or bad for oneself. In order to avoid potential confusion we will use the term appraisal to refer to this affective process.

A term that is sometimes used synonymously with emotion is feeling. Some authors, however, distinguish between emotions and feelings, often reserving feeling for describing the experiential part of an emotion. Because of a lack of consensus, we will only use the verb to feel and its gerund form, avoiding the noun feeling. In the rest of this dissertation, we will use “feeling an emotion” interchangeably with “experiencing an emotion.”

1.2.2 Duration as Distinguishing Factor

An interesting way of distinguishing emotion from other affective states and dispositions is by considering duration. Table 1.1 presents such a classification. In line with
1.2 WHAT IS EMOTION REALLY?

Table 1.1: Object salience and duration as factors that distinguish affective phenomena. Table taken from Clore et al. [2001, page 33].

<table>
<thead>
<tr>
<th>Object Salience</th>
<th>Duration of Affect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Salient</td>
<td>Temporary State</td>
</tr>
<tr>
<td>Object Not Salient</td>
<td>Enduring Disposition</td>
</tr>
<tr>
<td></td>
<td>Emotion</td>
</tr>
<tr>
<td></td>
<td>Attitude</td>
</tr>
<tr>
<td></td>
<td>Mood</td>
</tr>
<tr>
<td></td>
<td>Temperament</td>
</tr>
</tbody>
</table>

the aforementioned classification by Gross and Thompson [2007], emotion is distinguished from mood because an emotion is always relative to something, whereas a mood is a more diffuse affective state. However, both are considered to be of relatively short duration. The long-term counterparts of emotion and mood are attitude and temperament, respectively. In Table 1.1 we see again the distinction between affective states (emotion and mood) and affective dispositions (attitude and temperament).

When we say that emotions and moods are of relatively short duration, what time spans are we then actually talking about? As stated before, moods are considered to be longer lasting than emotions, but this is not apparent from Table 1.1. The typical difference in duration of several affective states and dispositions is shown in more detail in Figure 1.1.

Figure 1.1 shows that the more reflex-like emotional expressions and changes in the autonomic nervous system occur instantly after evaluation and last for seconds or minutes at most. Emotions are usually reported to last for seconds or minutes; in extreme cases up to several hours. Moods are in the range of hours to months. Extremely long-lasting moods are not considered ‘normal’; therefore, affective states lasting months on end are usually called emotional disorders and are a reason to seek professional help [Oatley and Jenkins, 1996]. Personality traits (including the attitudes and temperaments of Table 1.1) are affective dispositions that remain relatively constant during one’s lifetime. It should be noted that personality traits influence the short-term affective states such as emotions and their associated expressions and autonomic changes, such that there is consistency in an individual’s affective reactions over long periods of time.

What Figure 1.1 also nicely illustrates are the aforementioned ‘grey areas’ between different kinds of affective states. For example, consider Alice who has just learned that she has won the jackpot in a lottery. Since Alice greatly desires being rich, she will be very happy about winning the jackpot, and she will be cognitively aware of the intenseness of her happiness for minutes on end. This happiness would thus fall into the category of “Self-reported emotions.” Now at some point, Alice will start considering all the possibilities that the newly won money creates for her, which will reinforce and prolong her happiness. After some time, Alice’s happiness is not directed at anything in particular anymore; she has entered a state of just being happy, which would be considered a mood. Now, would it be possible in principle to

---

3Here “expressions” means the prominent bodily expressions such as a happy smile or a frightful yell, not persistent ‘background’ expressions such as a depressed person’s droopy face. Such a sad expression is usually not considered a single expression because it is continually blended with the person’s other expressions.
pinpoint the exact moment when Alice ceased to be happy about winning the jackpot and started to be just in a (diffuse) state of happiness? The process of an emotion or a chain of emotions turning into a mood is a gradual process and there may be a considerable grey area. (See also the overlap between “Self-reported emotions” and “Moods” in Figure 1.1.) Similarly, a chain of unhappiness-inducing events can turn into a mood of feeling down, which can turn into depression (which is usually considered an emotional disorder). Again, it may not be possible to pinpoint the exact moment where the negative mood turns into a depression disorder.

1.2.3 A Procedural View of Emotion

Having distinguished emotion from other affective phenomena, we can look more closely at how emotion works. With respect to the cognitive aspects of emotions, usually three phases are distinguished.

First, the perceived situation is appraised by an individual based on what he or she thinks is relevant and important. For example, Alice, who likes receiving presents, is given a necklace by Bob. Alice then judges receiving the necklace as desirable and Bob’s action as praiseworthy. Consequently, the appraisal of this action and its outcome causes gratitude towards Bob to be triggered for Alice. Note that different types of emotion may be triggered simultaneously by the same situation, some of which may even be seen as conflicting. For example, Alice may at the same time be disappointed because it was not the necklace she had hoped to receive. Emotion theories dealing with appraisal are for example [Frijda, 1987; Ortony et al., 1988; LeDoux, 1996; Oatley and Jenkins, 1996; Scherer et al., 2001].

Second, the appraisal of some situation can cause the triggered emotions—if they are strong enough—to create a conscious awareness of emotional feelings, leading to the experience of having emotions. For example, Alice’s gratitude towards Bob will have a certain intensity and will probably decrease over a certain amount of time. All this may depend on, e.g., the degree of desirability of receiving a necklace and Alice’s previous attitude towards Bob. Emotion theories dealing with these quantitative aspects of emotions are for example [Ortony et al., 1988; Frijda, 1987; Ekman and Davidson, 1994].
Third, emotional feelings need to be regulated. For example, Alice may want to organize her behavior such that positive emotions are triggered as often as possible and negative emotions are avoided or drowned by positive ones. She could do this by being nice to Bob so that he will give her more presents, or avoiding him altogether so that she will never again be confronted with his bad taste in jewelry. In fact, some emotion theories posit that the main purpose of emotion is to function as a heuristical mechanism for selecting behaviors [Damasio, 1994; Oatley and Jenkins, 1996; LeDoux, 1996]. Emotion theories dealing with behavioral consequences of emotions are for example [Frijda, 1987; Ekman and Davidson, 1994; Lazarus, 1994; Oatley and Jenkins, 1996; Gross, 2007].

This three-stage procedural view of emotion will be the backbone of this dissertation. Because this view is so important in understanding the approach taken in this dissertation, we will discuss it in more detail here. Figure 1.2 illustrates the idea from a computer science perspective by distinguishing processes (the rounded boxes) from data (the squared boxes). With respect to the three identified stages of emotion, appraisal and regulation are considered to be processes, and experienced emotions is considered to be a data set. In particular, appraisal is a process generating new emotions, and regulation is a process modifying existing emotions. Let us explain the reading of Figure 1.2 in more detail now.

Appraisal is seen as a process taking three kinds of data: percepts, concerns, and individual parameters. A percept is anything observed in the environment, typically an event, action, or object. Concerns is used as an aggregate term for goals, interests, standards, norms, ideals, attitudes, etc.; that is, any representation of personal values. Individual parameters (or “individual response propensities” [Sonnemans and Frijda, 1995]) are added to account for the fact that different individuals with similar concerns in similar situations can still have different emotional reactions. Thus the emotional ‘character’ of a person is supposed to be encoded in these individual parameters.

The appraisal process thus combines the percepts, concerns, and individual parameters, resulting in a set of elicited (triggered) emotions. After assigning intensities to these triggered emotions (indicating their strength), the output of the appraisal process is a set of new experienced emotions, which are added to the existing set of experienced emotions.

In many situations one may want to regulate one’s emotions in order to decrease the experience of negative emotions and prolong the experience of positive emotions [Gross, 2007]. Therefore Figure 1.2 contains a second process influencing the experienced emotions, called emotion regulation. There are several ways in which emotion can be regulated. For example, consider Alice who is watching a horror movie, which happens to be very distressing for her. In order to decrease her distress, she can apply a number of emotion regulation strategies. Gross and Thompson [2007] classify the following kinds of strategies: “situation modification” (e.g., mute the sound of the

---

4 Percepts from ‘inside’ the body (e.g., pain and hunger signals) are usually said to give rise to impulses, as discussed above. Since we are interested in emotions arising from cognitive evaluations, we will only consider cognitive percepts.

5 With respect to emotions, we use “triggered” and “elicited” interchangeably in this dissertation. “Eliciting conditions” is a phrase often encountered in psychological literature, but to be in line with computer science jargon we tend to substitute this phrase for “triggering conditions.”
Figure 1.2: A view of emotion in terms of data (square boxes) and processes (rounded boxes). The intended reading of this figure is explained in the text.

movie), “attentional deployment” (e.g., close her eyes during a gory scene), “cognitive change” (e.g., tell herself that it is not real blood), and “response modulation” (e.g., have a smoke to vent the distress). Of course, Alice could also have chosen not to watch the horror movie in the first place; this fifth emotion regulation strategy is called “situation selection” [Gross and Thompson, 2007].

Even without such ‘active’ emotion regulation, emotional experience is not constant over time; all other things being equal, the experience of an emotion usually fades over time. Furthermore, experienced emotions are intertwined with mood; the valence of each experienced emotion can ‘nudge’ the current mood in that direction (i.e., positive emotions affect mood positively and negative emotions affect mood negatively). On the other hand, mood can affect experienced emotions by functioning as a threshold. For example, a person in a sad mood may not experience happiness from a minor achievement; it would require a very strongly positive event to make the person experience happiness despite its sad mood. As with experienced emotions, mood may stabilize to ‘neutral’ over time, all other things being equal.6

6Extreme moods such as depression may not stabilize to neutral without intervention, but recall that such affective states are considered to be emotional disorders rather than moods.
In Figure 1.1, the effects of mood and time on emotional experience are hidden inside the processes of appraisal and regulation, respectively. This is done for clarity of presentation at this point; however, we will be more explicit about what affects emotional experience later on (cf. Figure 3.1).

To summarize this section, experienced emotions is viewed as a kind of data affected by two processes. First, new experienced emotions are created by a process called appraisal. Second, existing experienced emotions can be changed through a process called emotion regulation. It is important to note that the rest of this dissertation has been written from the perspective of this procedural view of emotion.

### 1.3 Emotions for Artificial Agents

There has recently been a great deal of interest in bringing emotions to Artificial Intelligence systems [Bates et al., 1994; Picard, 1997; Johns and Silverman, 2001; Sloman, 2001; Breazeal, 2002; Gratch and Marsella, 2004; Marinier and Laird, 2004; Meyer, 2006; Dastani and Meyer, 2006; Steunebrink et al., 2007, 2008a, 2009a; Adam et al., 2009]. The branch of computer science that is concerned with computational systems that somehow incorporate emotion is typically called “affective computing.” This term was coined by Rosalind Picard and served as the title of her seminal book [Picard, 1997] on the subject of emotion in computational systems. It should be noted that the term “affective computing” is intentionally ambiguous; it is used to describe systems that reason about the emotions of their human users as well as systems that display emotional expressions and exhibit emotional behavior themselves.

There are (at least) three reasons why emotions are important for Artificial Intelligence systems. First, an obvious application of emotions is to make robots and virtual characters more believable to human users. In particular, the behaviors of robots and virtual characters are expected to appear increasingly convincing, social, and intuitive if they seem to have an emotional state congruent with the situations they find themselves in [Picard, 1997; Breazeal, 2002; Gratch and Marsella, 2004; Marinier and Laird, 2004]. For example, humans are easily startled by sudden loud noises, so a believable robot or virtual character should also display startle behavior when subjected to a sudden loud noise (real or virtual).

Second, from a more theoretical perspective, it is investigated what the role of emotions is in models of human decision-making and how they may be employed to make these models more accurate and effective [Elster, 1996; Johns and Silverman, 2001; Coppin, 2008]. Classical models of economics assume humans behave completely rationally, whereas classical decision theory assumes humans try to maximize their expected utility. However, human decision making can be greatly influenced by affective processes such as emotions, mood, impulses, and stress. For example, the possibility of later regretting a decision makes one more risk-averse, whereas envisaging future emotional responses affects the current emotional state, which in turn influences decision making [Gratch and Marsella, 2004]. A person that is trying to ‘feel as good as possible’ will thus try to maximize emotional reward rather than just material or monetary gain. Now the concept of utility, which is central to decision theory, can be extended to include emotional rewards, but the question remains how
to design specific utility functions that incorporate emotion? Here investigations into the mechanisms underlying human emotion can give rise to heuristics that can guide the design of utility functions that are more in line with actual human behavior.

Third, there exists psychological [Frijda, 1987; LeDoux, 1996; Ekman and Davidson, 1994; Oatley and Jenkins, 1996] and neurological [Damasio, 1994] evidence that emotions are not only relevant but even necessary for intelligent behavior. Particularly, it has been shown that persons who cannot experience emotions (e.g., due to specific brain damage) have trouble distinguishing between important and irrelevant details, consistently make bad decisions, and do not display adequate social behavior necessary to function normally in society. A related, more philosophical argument posits that emotions are an inevitable consequence of mechanisms that allow for intelligent and rational behavior in complex environments with limited resources [Sloman, 2001; Gordon, 1987; Elster, 1996; Hoffman et al., 1991].

So besides a considerable body of literature on the subject of emotion in psychology, sociology, philosophy, biology, neurology, and economics, there is also a quickly growing body of literature on emotion in computer science (affective computing). However, as far as we are aware, all computer science literature focuses on either of the following three specific topics.

First, and most visibly, there is the work on emotional expressions for robots and virtual characters; for example, [Bates et al., 1994; Neal Reilly, 1996; Koda, 1996; Breazeal, 2002; Pelachaud, 2009]. The goal of this type of research on emotion is to attain as believable and natural as possible facial (and sometimes bodily) expressions of several types of emotion. The most popular emotion types to design facial expressions for are happiness, sadness, anger, fear, surprise, and disgust, because these emotion types have been shown relate to culturally-independent facial expression [Ekman, 1982]. On this topic, an interesting issue that has received considerable attention is the blending of facial expressions; for example, blending a happy expression with a surprised one to obtain an expression for pleasant surprise. However, there are many types of emotion that cannot be unambiguously mapped to unique facial expressions. For example, how does one distinguish between a happy face and a proud face? Or what is the facial expression for the emotion admiration? Although the outward appearance of robots and virtual characters is very important for the comfort of their human users, the outward appearance is usually all there is in this kind of research. That is, there is often no model of appraisal running on the robot or virtual character in question; the emotional expressions are just animations played at the experimenter’s command.

Second, there exist implementations of emotions with the aim of investigating how emotions can be calculated; for example, [Gratch and Marsella, 2004; Marinier and Laird, 2004]. Specifically, it is investigated how intense triggered emotions should be and how these intensities may be regulated in virtual characters. An important application for this kind of research is the modeling of civilians and terrorists in military training scenarios. However, a problem inherent in such work is that a complete implementation of emotions would require nothing less than a complete simulation of a human. Therefore current implementations of emotions are forced to simplify the model of emotion that they use and restrict the aspects of emotion that they investigate to the ones that can be meaningfully addressed in their application.
1.4 EMOTIONS FOR ARTIFICIAL AGENTS

domain. Interestingly, investigations into the simulation of emotions in artificial agents often do evoke discussions on whether or not artificial systems could ever ‘have’ emotions. More specifically, the question here is often whether an artificial system that perfectly simulates the workings of human emotions can in fact be said to *experience* emotions itself. Although this is a very fascinating topic, we will avoid this (philosophical) discussion in this dissertation and instead focus on investigating how emotions can be modeled and computed in the first place. The resulting analysis and formalization can then be used to build robots and virtual characters that display emotion-like feelings and behavior, based on human emotions, but regardless of whether or not we actually want to call them ‘emotional’.

Third, there exist (a few) attempts at formalizing particular psychological models of emotion in a complete and rigorous manner; for example [Meyer, 2006; Adam, 2007]. Such formalizations aim at capturing the logical structure underlying a chosen psychological model of emotion; the resulting logical specification of emotions can then be used to reason about properties of emotions in a formal manner, thus gaining more insight into the workings of human emotions. Moreover, a proper and comprehensive formalization of emotion will facilitate the construction of systems described in the first two points by providing a view of emotions from a computer science perspective. A specification of emotions in a formal logic will be much easier to implement in a robot or virtual character than an informal (psychological) specification. This is because formal logic is much closer to programming code—and thus easier to translate to programming code—than the natural language used in psychological descriptions of emotions. A formalization thus builds a bridge from psychological models of emotions to computational models of emotions.

It should be noted that in our research group there has been some work on emotions prior to this dissertation work. In a seminal article, Meyer [2006] first attempted to build a bridge between psychological specifications of emotions and specifications of emotions in a formal logic. In this formalization, an emotion arises as a label of a cognitive state when that cognitive state matches the specification of the triggering conditions of that emotion according to the psychological model. An heuristic is then associated with each emotion, specifying what to do when the emotion in question has arisen. This formalization was subsequently adapted for implementation in 3APL (An Abstract Agent Programming Language) [Dastani and Meyer, 2006]. In this implementation the heuristics associated with the emotions were used to enhance the deliberation of agents programmed in 3APL. The current dissertation continues in the spirit of this previous work [Meyer, 2006; Dastani and Meyer, 2006], although completely from scratch with respect to the approach to formalization. In the next section, then, we will formulate the research questions addressed in this dissertation.

---

7The process that determines when to perform which action is called the *deliberation cycle* in 3APL. The deliberation cycle thus determines how agent programs are executed. By incorporating emotions in 3APL’s deliberation cycle, nothing in the agent programming language itself was changed; agents would only behave more ‘emotionally’ due to changes under the hood of 3APL.
1.4 Research Questions

As mentioned at the very start of this chapter, the main aim of this dissertation is to build a bridge from psychology to computer science on the subject of emotion. The main device employed for this endeavor is formalization. The all-encompassing research question addressed by this dissertation is thus the following.

Main Research Question.  
How can emotions be formalized?

Note that we are interested here in describing the logical structure of emotions using a logical language with formal semantics, but we will not be concerned with axiomatization or soundness and completeness proofs.

In order to formalize emotion, it must first be recognized how emotions differ from related affective phenomena. Having done so in this introduction, we can turn to the structure of emotion itself. We have already provided a procedural view of emotion, but this tells us little about the objects of this process. Specifically, different types of emotion (e.g., joy, anger, pride) can be elicited, experienced, and regulated. We must find a suitable structure of emotion types to guide our formalization. So the first specific question that should naturally be asked when embarking on a formalization of emotions is the following.

Research Question 1.  
How can the structure of emotion types be represented such that it is suitable for formalization?

An answer to this question will be provided in Chapter 2 by discussing in detail a compositional structure of emotion types.

Formalizations of emotions in general, and the work presented in this dissertation in particular, can be used for multiple purposes. The following three purposes for which we formalize emotions give rise to three more research questions.

The first purpose is to facilitate the construction of systems that maintain a theory of mind about their human users (including their emotions). The idea here is that systems (particularly companion robots and virtual characters) that are capable of reacting to a human user’s emotional state in an appropriate way will be better accepted, and be found more pleasurable to interact with, than a system that is not sensitive to the user’s emotions. For this to work, the affective system must know when emotions can arise. This leads us to the following research question.

Research Question 2.  
How can the conditions that trigger emotions be formalized?

This question will be addressed in a top-down fashion in three stages. Chapters 3, 4, and 5 are devoted to one stage each.

The second purpose is to facilitate the construction of systems that display emotion-like feelings and behavior. The idea here is that systems that appear to have emotions themselves will be found to be more believable to human users [Breazeal, 2002]. But in order to be able to determine appropriate emotional expressions, it must
not only be determined when emotions are triggered, but also what is the structure and magnitude of their intensities. This leads us to the following research question.

**Research Question 3.**
*Which aspects determine the experience of an emotion and how can they be formalized?*

This question will be addressed in Chapter 6. It should be noted that we do not go into the issue of generating emotional expressions in this dissertation; our focus will be kept on determining what happens ‘inside’. As discussed above, systems that incorporate emotion-like processes may work more efficiently than those that do not [Damasio, 1994]. This is because emotions can work as heuristics that focus the attention (computational resources) to salient and important aspects of a situation. An important aspect of emotions is thus their behavioral consequences, which leads us to the following research question.

**Research Question 4.**
*How can the influence of emotions on decision making and behavior be formalized?*

This question will be addressed in Chapter 7 from two different perspectives; one approach takes a theoretical, psychological perspective and the other approach takes a practical, agent programming perspective.

Finally, the third purpose of our formalization is to study the concept of emotion in a more formal way. This application of formalization of emotions is of a more fundamental nature than the previous two. Generally speaking, the process of formalization can be used to catch ambiguities and contradictions in an informal specification, if there are any. Moreover, properties arising from the formalization can be checked for congruence with the original specification. Thus, if a formalization of a psychological model of emotions can be shown to be both in line with the psychological model and free of contradiction, then the formalization serves as a kind of logical validation of the psychological model. Because a formal logic is more precise than natural language, the process of formalization of a psychological model of emotions (which is specified in natural language) can be used to make the model more precise, thereby gaining more insight into its workings, and ultimately enriching our understanding of human nature.

This third purpose will be addressed throughout this dissertation by investigating what are the properties arising from our formalization and whether these properties are in line with psychological models of emotion. Naturally, the discussed properties will be accompanied by formal proofs.

Together these four research questions address a functional subset of the subject emotion. The first question forces us to search for and commit to a clear and integral model of emotions. The second question then makes us specify when emotions arise in the first place, according to the chosen model. The third question makes us specify when triggered emotions are experienced, what are the aspects of emotional experience, and how emotional experience changes over time. Finally, the fourth question makes us consider what is the use of emotions for artificial systems. It will be clear that this trajectory from triggering to experience to behavior constitutes a
complete functional approach to formalizing emotion, in line with the procedural view of emotion as depicted in Figure 1.2.

1.5 Overview of this Dissertation

A top-down approach is used throughout this dissertation. This means that we will first introduce and define aggregate concepts in terms of other concepts, and then define those concepts in terms of yet more basic concepts, and so on. This is done primarily for clarity of presentation; it prevents having to introduce and explain all sorts of small details when it is not yet clear what they will be used for. By using a top-down approach, a concept will only be introduced at the moment it is needed; it will thus always be immediately clear what a new concept will be used for.

A perhaps slightly unsettling consequence of a top-down approach is that at first it may seem that definitions are not properly grounded, i.e., it may seem that little progress is made whenever we define something in terms of other things that were themselves not yet defined. However, we assure the reader that eventually all definitions will be firmly grounded.

A second important thing to note is that all the work presented in this dissertation is cumulative. This means that each successive chapter builds on the previous chapter. Thus in each chapter it will be assumed that all definitions and constraints introduced in preceding chapters are in effect.

The rest of this dissertation is organized as follows.

Chapter 2 discusses psychological models of emotions. The particular model that is chosen for formalization will be discussed in greater detail. The emphasis of this discussion will be on the logical structure underlying the model, because it is on this structure that our formalization will be based. The formalization of the conditions that—according to the chosen psychological model—trigger emotions will proceed in three stages and is spread over the next three chapters.

Chapter 3 presents the first stage of formalization of the conditions that trigger emotions. This stage will constitute a semiformal specification of the logical structure of triggering conditions.

Chapter 4 presents the second stage of formalization of the conditions that trigger emotions. This stage will ground important concepts used in the psychological model of emotions in a specific type of logic, namely dynamic doxastic logic.

Chapter 5 presents the third stage of formalization of the conditions that trigger emotions. This stage will formalize the main appraisal concepts used in the psychological model of emotions in an existing agent specification framework. Because this framework is an extension of dynamic doxastic logic, it will firmly ground the preceding two stages.

Chapter 6 shows how the experience of emotions can be modeled on top of the preceding formalization. In particular, we will investigate the issues of what is emotional intensity and how can it be calculated, how emotions give one the tendency to perform certain actions, and how arbitrary types of emotion can be represented.

Chapter 7 discusses several ways in which emotions can be used to guide the behavior of artificial agents. In particular, we will show for several emotion types...
how they can be used to specify constraints on an agent’s deliberation in a principled way.

Chapter 8 concludes this dissertation.

Finally, there are two appendices, which provide the following additional material.

Appendix A summarizes the most important specifications of the chosen psychological model of emotions for easy reference.

Appendix B provides an outline of the project which served as the context in which this research has been performed.
Chapter 2

Emotion in Psychology

We build too many walls and not enough bridges.
– Isaac Newton

The aim of this chapter is to explain what properties of a model of emotions are most interesting to artificial intelligence researchers and the affective computing community in particular. Furthermore, this chapter provides a detailed description of the model of emotions that we have found to be most suitable for formalization, in the sense that it satisfies those properties that are important to artificial intelligence researchers.

To this end, we must first decide from which perspective we want to study emotions. For example, one can study emotion from a linguistic perspective by investigating which words describing emotions exist in a certain language and attempting to make classifications based on these findings, or from a cultural perspective by investigating which concepts related to emotion exist in a certain culture. A study by Clore, Ortony, and Foss [1987] found close to 600 words referring to emotions in the English language. Interestingly, this study also found that quite a few words that had been used by some psychologists as examples of emotions did not appear to refer to emotions or other affective states after all (e.g., distrustful, puzzled, receptive). What this shows is that a linguistic perspective can produce different classifications of emotions even within the same language, not to speak of comparing emotional words from different languages and emotional concepts from different cultures. However, what a linguistic perspective may not be able to fully address is where emotions come from; emotional words just refer to emotions, but they do not explain them.

Emotion can also be studied neurologically, by investigating which neuron firing patterns correspond to which emotions, and making classifications based on the findings. Moreover, emotions can be related to the release of hormones, changes in the autonomic nervous system, and other physiological phenomena. Models of emotion in terms of human physiology can be relevant for computer scientists attempting to build realistic simulations of bodily processes with the aim of learning how human physiology works. However, many current artificial intelligence researchers con-
cerned with problem solving abstract from bodily processes and focus on cognitive reasoning. Although we recognize that bodily processes are an essential component of emotions in humans, the artificial intelligence community could benefit more from a model of emotion that does not rely on human physiology. Of course, the model of emotion can incorporate components inspired by human physiology, but the point is that a formalization or implementation of the model should not require a complete simulation of a human body.

It is currently widely accepted among psychologists that emotions have an important cognitive component. Indeed, psychological models of emotion are built mostly in terms of cognitive states and processes such as beliefs and action selection. Such models of emotion are good candidates for use by artificial intelligence researchers. The concepts related to cognition that are often used in both cognitive psychology and artificial intelligence will be discussed in Section 2.1. Then in Section 2.2 we will give an overview of a particular psychological model of emotions which fits these concepts very well. A detailed analysis of the logical structure underlying the chosen psychological model is provided in Section 2.3. Importantly, the insights gained from this logical analysis will serve as a guide for our formalization of emotions in the following chapters.

2.1 From Cognitive Psychology to Artificial Agents

A particularly interesting connection between cognitive psychology and artificial intelligence is made by the BDI model [Bratman, 1987]. BDI (Belief, Desire, Intention) is based on the idea from folk psychology that agents choose their actions using the three concepts of beliefs, desires, and intentions. An agent’s beliefs constitute a representation of the situation which the agent finds itself in, i.e., its beliefs constitute the agent’s model of the world. An agent’s desires constitute a representation of how the agent would like the situation to be. Although beliefs and desires may seem enough information to determine which action to select at any point in time, it is usually held that something more is needed. This something extra is intention.

If every time an agent has to decide on a new action it would have to review its beliefs and desires, this may result in erratic and inefficient behavior. For example, assume Bob is an agent with just beliefs and desires, and that Bob has the desire visit Alice and the desire to buy milk. Now Bob may decide first to move several steps in the direction of Alice’s place, then move several steps in the direction of the supermarket, and so on. Obviously this will lead Bob to get stuck halfway between Alice’s place and the supermarket, whereas he could just have picked up some milk first on his way to Alice. What Bob is lacking is commitment to a desire, persistence to see to that desire becoming a reality, and a way of checking for inconsistent courses of action. A further problem for Bob is that he has to decide which desire to act on and how to act on that desire at each point in time, which is very inefficient. The solution to these problems is given through the concept of intention. Bob needs to form an intention to buy milk, which will force him to make a plan and commit to it. This plan will only be abandoned or changed in exceptional circumstances. Thus Bob can execute his plan step by step without having to reconsider often, and he will
not be tempted to try and visit Alice before having bought his milk.

The concepts of BDI have proven to be very fruitful for designing artificial intelligence system architectures [Cohen and Levesque, 1990; Rao and Georgeff, 1991; Hoek et al., 1999] and implementations [Winikoff, 2005; Pokahr et al., 2005; Bordini et al., 2007; Hindriks, 2008; Dastani, 2008]. Because the BDI model is well-studied and widely used, it will be interesting to find a model of emotion that can be described in terms of BDI. If emotions were formalized using BDI concepts, it would be straightforward for computer scientists to implement emotions in new and existing artificial intelligence systems. Such a BDI-based formalization of emotions could thus function as a bridge between the cognitive psychology of emotion on the one hand and artificial intelligence on the other hand.

A psychological model of emotions that fits this aim is one commonly referred to as “the OCC model,” after the initial letters of the authors’ last names. In their book *The Cognitive Structure of Emotions* [1988], Ortony, Clore, and Collins have proposed a very interesting model of emotions that provides specifications of the eliciting conditions of emotions and the variables that affect their intensities. We have chosen to formalize the OCC model because (1) it provides a clear classification of a broad range of emotion types, (2) it lists concise descriptions of the conditions that elicit emotions, and (3) for this it uses BDI-like concepts that are well-studied and relatively straightforward to formalize. In the following sections we will discuss the OCC model in detail, and in the following chapters we will show how the OCC model can be formalized in terms of BDI. In line with the affective computing literature, we will refer to the work of Ortony, Clore, and Collins [1988] as the OCC model, or simply “OCC,” henceforth.

### 2.2 Overview of the OCC Model

The OCC model classifies 22 emotion types (see Appendix A for a complete summation). This is done by considering on which kinds of aspects of a situation one can appraise. OCC consider a human can either appraise consequences of events, actions of agents, or aspects of objects. If one focuses on a consequence of an event, one can appraise this consequence as desirable or undesirable (or both, or neither) with respect to one’s goals. For example, joy about winning a lottery is an event-based emotion, because the satisfaction of the goal to become rich is a desirable consequence of the event of winning the lottery. If one focuses on an action of an agent, one can appraise this action as praiseworthy or blameworthy (or both, or neither) with respect to one’s standards. For example, pride about saving a child from drowning is an action-based emotion, because it is praiseworthy to perform an action which satisfies the standard that one should save a person’s life whenever (reasonably) possible. If one focuses on an aspect of an object, one can appraise this aspect as appealing or unappealing (or both, or neither) with respect to one’s attitudes. For example, love for an old car is an object-based emotion, because the car may have appealing aspects according to one’s attitudes. It should be noted that (gustatory, aesthetic, social)

---

1 On page 18 [Ortony et al., 1988], OCC say that “events are simply people’s construals about things that happen.” From a computational perspective, however, we would say that this is far from simple!
20  EMOTION IN PSYCHOLOGY

<table>
<thead>
<tr>
<th>Type of percept</th>
<th>Evaluated against</th>
<th>Central variable (positive/negative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consequence of event</td>
<td>Goals</td>
<td>Desirability/Undesirability</td>
</tr>
<tr>
<td>Action of agent</td>
<td>Standards</td>
<td>Praiseworthiness/Blameworthiness</td>
</tr>
<tr>
<td>Aspect of object</td>
<td>Attitudes</td>
<td>Appealingness/Unappealingness</td>
</tr>
</tbody>
</table>

Table 2.1: The kinds of aspects of a situation that can be appraised according to the OCC model.

tastes are included in the notion of attitudes. This ontology of the OCC model is summarized in Table 2.1.

The notions of consequence of event, action of agent, and aspect of object are thus used to distinguish three main categories of emotion types. Within these categories, the OCC model makes further differentiations based on, e.g., whether prospects are relevant (as in hope and fear), whether events apply to others (as in pity and gloating), or whether an action was performed by the self or someone else (to distinguish, e.g., pride from admiration). Additionally, some event-based and action-based emotion types are combined to form a category of emotions concerning consequences of events caused by actions of agents. For example, anger can arise when one focuses on both the blameworthy action of another agent and an undesirable event which has been (presumed to be) caused by it.

It should be emphasized that in the OCC model, emotions are never used to describe the entire cognitive state of an agent (as in “Alice is happy”); rather, emotions are always relative to individual events, actions, and objects. So Alice can be joyous about receiving her new furniture and at the same time be distressed about the height of the accompanying bill.

We should right away eliminate another possible source of confusion, namely with respect to the distinction between consequences and events. OCC often use the phrase “(un)desirable event” (see also Table 2.2); however, events are actually always appraised with respect to their consequences. For example, an earthquake in itself does not have a valence; only the consequences of this event (e.g., valuable lessons for seismologists, property damage, loss of life) are appraised as being desirable or undesirable. Because desirability only applies to consequences of events, every instance of the phrase “(un)desirable event” should actually be read as a shorthand for “(un)desirable consequence of an event.”

One important reason for the popularity of the OCC model is that throughout the book, specifications are given for each of the 22 emotion types. For example, below is the specification of the class of emotions labeled as ‘fear’ in the OCC model (copied from [Ortony et al., 1988, page 112]).
FEAR EMOTIONS
TYPE SPECIFICATION: (displeased about) the prospect of an undesirable event
TOKENS: apprehensive, anxious, cowering, dread, fear, fright, nervous, petrified, scared, terrified, timid, worried, etc.
VARIABLES AFFECTING INTENSITY:
(1) the degree to which the event is undesirable
(2) the likelihood of the event
EXAMPLE: The employee, suspecting he was no longer needed, feared that he would be fired.

All the specifications provided by the OCC model have the following five elements.

1. The type label (e.g., “fear” in “FEAR EMOTIONS”) identifies the most convenient word to describe emotions of the type under consideration. Here “fear” is chosen because of all words describing a state of fear (e.g., worried, frightened, terrified), “fear” is the most free of connotations and average in terms of typical intensity.

2. The type specification provides, in a concise sentence, the eliciting conditions of the emotion type in question. It should be noted that these eliciting conditions actually specify the type of emotion under consideration. Thus the word “fear” is only used as a convenient label for the type of emotion about “the prospect of an undesirable event.” Indeed, many emotion words fit this type, which is what the next item is about.

3. A list of tokens is provided, showing which emotion words can be classified as belonging to the emotion type in question. For example, ‘fright’, ‘scared’, and ‘terrified’ are all emotions with respect to “the prospect of an undesirable event.” (Of course, ‘fear’ is also among the tokens, and it is no more special than the other tokens, except for being regarded by OCC as the most convenient label for this type of emotion.) It may be interesting to note that some tokens include more than just having “the prospect of an undesirable event.” For example, ‘apprehensive’ signifies a much lower intensity of fear than ‘terrified’, while ‘cowering’ and ‘petrified’ also tell something about the behavioral responses of the fear experiencing individual.

4. For each emotion type, a list of variables affecting intensity is provided. The idea is that higher values for these variables result in higher emotional intensities. So high undesirability and high likelihood are likely to result in a high intensity fear. It should be noted that, if we have such additional information about intensities, we can usually pick a better token to describe the emotion in question. In this example, ‘dread’ may be a more appropriate term than the generic label ‘fear’. It should also be noted that only variables that are local to the emotion type in question are listed. There are also several global variables identified in the OCC model (such as arousal) that affect all emotion types. These global

2In this dissertation, we use “triggers” and “eliciting conditions” interchangeably.
Joy: (pleased about) a desirable event
Distress: (displeased about) an undesirable event
Happy-for: (pleased about) an event presumed to be desirable for someone else
Pity: (displeased about) an event presumed to be undesirable for someone else
Gloating: (pleased about) an event presumed to be undesirable for someone else
Resentment: (displeased about) an event presumed to be desirable for someone else
Hope: (pleased about) the prospect of a desirable event
Fear: (displeased about) the prospect of an undesirable event
Satisfaction: (pleased about) the confirmation of the prospect of a desirable event
Fears-confirmed: (displeased about) the confirmation of the prospect of an undesirable event
Relief: (pleased about) the disconfirmation of the prospect of an undesirable event
Disappointment: (displeased about) the disconfirmation of the prospect of a desirable event
Pride: (approving of) one’s own praiseworthy action
Shame: (disapproving of) one’s own blameworthy action
Admiration: (approving of) someone else’s praiseworthy action
Reproach: (disapproving of) someone else’s blameworthy action
Gratification: (approving of) one’s own praiseworthy action and (being pleased about) the related desirable event
Remorse: (disapproving of) one’s own blameworthy action and (being displeased about) the related undesirable event
Gratitude: (approving of) someone else’s praiseworthy action and (being pleased about) the related desirable event
Anger: (disapproving of) someone else’s blameworthy action and (being displeased about) the related undesirable event
Love: (liking) an appealing object
Hate: (disliking) an unappealing object

Table 2.2: An overview of the eliciting conditions of all 22 emotion types of the OCC model, copied from the book [Ortony, Clore, and Collins, 1988].

variables are not repeated for each emotion type (see Chapter 6 for a discussing of the global variables).

5. Finally, an example sentence is provided illustrating a situation in which the type of emotion under consideration might arise.

A complete listing of the specifications of all 22 emotion types distinguished in the OCC model is provided in Appendix A for easy reference. In Table 2.2 we have summarized the eliciting conditions (i.e., the second item) of all 22 emotion types provided in the book [Ortony et al., 1988]. Table 2.2 can thus be seen as a summary of all possible conditions that can trigger emotions. Because this is extremely interesting for our formalization, we will discuss these eliciting conditions in more detail in the next section. As can already be seen from a quick glance at Table 2.2, the formulations of the eliciting conditions clearly follow some patterns. For example, many of the descriptions look very much alike save for some “un-” and “dis-” prefixes. Surely, there must be a way to structure the described emotion types that explicates this pattern. In the next section we will describe such a structure.
2.3 The Logical Structure of the OCC Model

As discussed at the beginning of this chapter, emotions can be viewed from many different perspectives, which means that they can be structured in different ways, depending on which perspective one wants to emphasize. This observation also holds within the cognitive perspective of emotions; that is, from a cognitive perspective of emotion there are still multiple ways of structuring emotions. For example, on page 19 of their book [Ortony et al., 1988], OCC present a diagram which structures their emotion types based on focus of attention. From this perspective, the event-based emotions are subdivided based on what aspect of the event in question one is focusing. For example, ‘relief’ and ‘joy’ have been placed in different branches because with ‘relief’ one is “focusing on” the disconfirmation of an undesirable prospect and with ‘joy’ one is not “focusing on” prospects (see Figure A.1 on page 188). However, as Ortony and Clore themselves have noted [Ortony and Clore, 2009], it is possible to end up with wholly different arrangements of the emotion types, if instead one considers their motivational and behavioral aspects, or a mapping of feelings, or physiological antecedents or effects, etc. For our purposes, we want to find a compositional perspective on the emotion types of the OCC model, because our aim is to formalize the model in logic, and logic is compositional.

The aforementioned diagram ([Ortony et al., 1988, page 19]) is often reproduced when an overview of the OCC model is to be given. In this section we also give an overview of the OCC model, but we will illustrate the OCC model with a slightly different diagram, namely Figure 2.1 on page 31. Our goal is to provide a logical account of emotion triggers. However, OCC’s diagram, which is based on focus of attention, is not very well suited to guide our formalization, because it is not compositional with respect to the eliciting conditions of emotions. Therefore, we have created a new diagram illustrating the structure of the emotion types based on their eliciting conditions (as listed in Table 2.2). It should be noted that in personal communication, Ortony and Clore have confirmed Figure 2.1 to be an accurate compositional illustration of the logical structure underlying the eliciting conditions of their emotion types [Ortony and Clore, 2009]. The following paragraphs serve to explain Figure 2.1 part by part. This is important because this figure serves as a guide for the formalization that will be presented in the chapters to follow.

2.3.1 General Emotion Types

Figure 2.1 can be seen as an inheritance structure. This means that the depicted emotion types are specializations of those above them and generalizations of those below them. This inheritance-based perspective results in a compositional formulation of the eliciting conditions, as follows.

At the most general level, all emotions are valenced reactions (to something). Although valenced reactions can have different magnitudes, each one is at least either positive or negative. Therefore ‘positive’ and ‘negative’ have been placed at the top of the hierarchy. At the next level, the OCC model specifies that valenced reactions can be directed at either consequences of events, actions of agents, or aspects of personal states. Therefore, the second level of the hierarchy is further divided into three branches: consequences, actions, and personal states. Each of these branches is further divided into sub-branches, each representing a specific type of emotion.

3This section is largely based on a paper titled “The OCC Model Revisited” [Steunebrink et al., 2009b].
of objects. Thus the top of the compositional structure of the eliciting conditions can be illustrated as follows.

```
<table>
<thead>
<tr>
<th>VALENCED REACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive</td>
</tr>
<tr>
<td>negative</td>
</tr>
<tr>
<td>TO</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>CONSEQUENCE</td>
</tr>
<tr>
<td>(OF EVENT)</td>
</tr>
<tr>
<td>pleased</td>
</tr>
<tr>
<td>displeased</td>
</tr>
<tr>
<td>ACTION</td>
</tr>
<tr>
<td>(OF AGENT)</td>
</tr>
<tr>
<td>approving</td>
</tr>
<tr>
<td>disapproving</td>
</tr>
<tr>
<td>ASPECT</td>
</tr>
<tr>
<td>(OF OBJECT)</td>
</tr>
<tr>
<td>liking</td>
</tr>
<tr>
<td>disliking</td>
</tr>
</tbody>
</table>
```

‘Pleased’ and ‘displeased’ have been chosen by OCC to function as labels for the most general type of valenced reactions to consequences of events, because they are very neutral sounding words with respect to intensity of experience, focus of attention, motivational and behavioral effects, etc. For the same reasons, ‘approving’ and ‘disapproving’ are used as labels for the most general type of valenced reactions to actions of agents, and ‘liking’ and ‘disliking’ are used as labels for the most general type of valenced reactions to aspects of objects.

### 2.3.2 Consequences of Events

With respect to valenced reactions to consequences of events, a distinction is made based on whether the consequence in question is prospective or not. In the following, we will call an event *actual* if it is not prospective. Note that *actual* only applies to an agent’s perceptions; as soon as (a consequence of) an event is perceived, it is called *actual*, even though the event may have occurred some time in the past. For example, learning that tomorrow it will rain is an event, but it has an undesirable consequence (e.g., the undermining of the goal to have a dry picnic) that is prospective and not actual. But this event can also have a consequence which is actual (e.g., the achievement of the goal to know the weather forecast). This differentiation on prospects then results in distinguishing between the ‘hope’ and ‘fear’ types on the one hand (e.g., Alice fears tomorrow her picnic will get wet) and the ‘joy’ and ‘distress’ types on the other hand (e.g., Alice is glad to have learned the weather forecast, where ‘glad’ is one of the *tokens* of the ‘joy’ type). This distinction can be illustrated as follows.
It should be noted that the term *prospect* (used in, e.g., hope and fear) is intentionally ambiguous: it is used to refer to both *future* events and *uncertain* (past or current) events. For example, hoping that tomorrow will be a sunny day is future-directed, whereas hoping that a mailed package has safely reached its intended recipient is uncertainty-directed. Many formalizations appear to use OCC’s notion of prospect in only one of these senses. For example, Adam et al. [2009] and Gratch and Marsella [2004] only used uncertain prospects when formalizing hope and fear, whereas Steunebrink, Dastani, and Meyer [2007] only used future prospects.

### 2.3.3 Actions of Agents

With respect to valenced reactions to actions of agents, a distinction is made based on whether the action in question has been performed by the self or by someone else. This distinction can be illustrated as follows.
The distinction between “one’s own action” and “someone else’s action” is, however, not as simple as it may seem. For example, a mother may be proud of the achievements of her son, even though the actions of her son are, strictly speaking, not her own. To resolve this, the OCC model uses the concept of a cognitive unit: the mother can consider herself and her son as part of a single cognitive unit and then, when appraising her son’s actions as praiseworthy, feel proud of the actions performed by (an agent in) the cognitive unit. This differentiation on cognitive unit then results in distinguishing between the ‘pride’ and ‘shame’ types on the one hand, and the ‘admiration’ and ‘reproach’ types on the other hand.

At this point the reader may expect there to be a branch below liking and disliking, after seeing branches being added below pleased/displeased and approving/disapproving. Indeed, in the original diagram of the OCC model, a single branch appears below liking/disliking with the emotion types ‘love’ and ‘hate’ (cf. Figure A.1 on page 188). The idea of OCC was that ‘love’ and ‘hate’ are examples of emotions of the type ‘liking’ and ‘disliking’, respectively [Ortony and Clore, 2009]. However, this means that the distinction between love/hate and liking/disliking does not constitute a differentiation in terms of eliciting conditions, but merely that ‘love’ is a token for the type of emotions labeled ‘liking’ and that ‘hate’ is a token for the type of emotions labeled ‘disliking’. So in our inheritance-based perspective, no branch has to be added below liking/disliking, because ‘love’ and ‘hate’ are not specializations of ‘liking’ and ‘disliking’ with respect to eliciting conditions.

### 2.3.4 Compound Emotion Types

In addition to valenced reaction to either consequences of events or actions of agents, the OCC model also considers several types of emotion arising from observing relations between the two. Specifically, these emotion types correspond to valenced reactions to consequences of events caused by actions of agents. The eliciting conditions of these so-called compound emotion types are conjunctions of their constituent emotion types. This means that ‘adding up’ an event-based emotion and an action-based emotion will yield a compound emotion, as follows:

\[
\begin{align*}
& \text{‘joy’} + \text{‘pride’} = \text{‘gratification’} \\
& \text{‘joy’} + \text{‘admiration’} = \text{‘gratitude’} \\
& \text{‘distress’} + \text{‘shame’} = \text{‘remorse’} \\
& \text{‘distress’} + \text{‘reproach’} = \text{‘anger’}
\end{align*}
\]

One should be careful though, when interpreting these ‘equations’. ‘Joy’ is a label referring to the eliciting conditions “(pleased about) a desirable event,” and likewise for the other labels. Moreover, the +-signs contain an implicit assertion about the (presumably causal) relation between the action and consequence in question. Taking these considerations into account, the type specification for, e.g., ‘anger’ becomes “(disapproving of) someone else’s blameworthy action and (being displeased about) the related undesirable event” (see Table 2.2).

The eliciting conditions of these compound emotion types can be illustrated using multiple inheritance and an additional condition stating that the action and
consequence under consideration must be related, as follows.

This illustration may be slightly ambiguous with respect to the multiple inheritance. For example, it may appear to be possible that gratification is formed by combining joy, pride, and admiration, but this is of course not the intended reading. We assume it is understood that the line going down from pride only combines with joy and then goes down to gratification. Similarly, the line going down from admiration only combines with joy and then goes down to gratitude.

### 2.3.5 Fortunes-of-others Emotion Types

The OCC model distinguishes several emotion types that are specializations of ‘joy’ and ‘distress’. One such group of specializations is called the “fortunes-of-others” emotion types and contains the emotions ‘happy-for’, ‘resentment’, ‘gloating’, and ‘pity’. These are valenced reactions arising from presuming that events have consequences for others. ‘Happy-for’ and ‘pity’ are specializations of ‘joy’ and ‘distress’, respectively, where the presumed desirability for the other is congruent with the desirability for the self. ‘Gloating’ and ‘resentment’, on the other hand, can arise in the incongruent cases: if the desirable consequence for the self is presumed to be undesirable for another, ‘gloating’ may arise, whereas if the undesirable consequence for the self is presumed to be desirable for another, ‘resentment’ may arise. Note that the fact that there may be social tabus resting on feelings of gloating and resentment is not of any concern to a model of emotions. If such emotion types exist, they have to be taken into account by a model of emotions for the sake of comprehensiveness. Therefore the emotion types ‘gloating’ and ‘resentment’ are included in our treatment of emotions, in line with the OCC model.

The inheritance of the fortunes-of-others emotion types from ‘joy’ and ‘distress’ can be illustrated as follows.
It may be interesting to note that in OCC’s original hierarchy (see Figure A.1 on page 188) the fortunes-of-others emotion types were not placed below ‘joy’ and ‘distress’. The reason for this is that the structure of OCC’s diagram is based on focus of attention and not on the structure of eliciting conditions. The fact that, e.g., ‘happy-for’ is a specialization of ‘joy’ with respect to eliciting conditions can be demonstrated as follows. For there to be a ‘happy-for’ emotion, the consequence that is desirable for the other must also be desirable for oneself to some degree (probably because it satisfies an interest goal to wish other people well, as suggested on page 94 [Ortony et al., 1988]). But if a consequence of an event is appraised as being desirable for oneself, the eliciting conditions for ‘joy’ are satisfied (see Table 2.2). So logically speaking, the eliciting conditions of ‘happy-for’ (and ‘gloating’) entail those of ‘joy’, and by a similar argument the eliciting conditions of ‘resentment’ and ‘pity’ entail those of ‘distress’. This observation is also supported by OCC:

“[I]n the type specifications of the Fortunes-of-others emotions, we mean it to be understood that when the reaction is one of being pleased, the event about which one is pleased (i.e., the event presumed to be desirable for someone else) is by implication (necessarily) a desirable event for oneself.”

(Quoted from [Ortony et al., 1988, page 94].) Therefore, in an inheritance-based hierarchy of eliciting conditions, the fortunes-of-others emotion types must be placed below ‘joy’ and ‘distress’, because the latter two emotion types are generalizations of them.

### 2.3.6 Prospect-based Emotion Types

The other group of specializations of ‘joy’ and ‘distress’ contains the emotion types ‘satisfaction’, ‘fears-confirmed’, ‘relief’, and ‘disappointment’. Together with ‘hope’ and ‘fear’, these six emotion types are called “prospect-based” in the OCC model.
However, the eliciting conditions of the former four emotion types are not specializations of ‘hope’ and ‘fear’. ‘Satisfaction’, ‘fears-confirmed’, ‘relief’, and ‘disappointment’ are emotions in response to actual consequences of events, namely consequences signaling the confirmation or disconfirmation of a previously prospective consequence. The relation between, e.g., ‘hope’ and ‘disappointment’ is thus more of a temporal kind. For example, first Bob hopes Alice will show up for their date, but when she does not, his hope turns into disappointment. Thus ‘satisfaction’, ‘fears-confirmed’, ‘relief’, and ‘disappointment’ are not special kinds of ‘hope’ or ‘fear’, but more like continuations of ‘hope’ or ‘fear’, counting from the point when an event has been perceived that signals the confirmation or disconfirmation of the thing hoped for or feared.

The inheritance of these four prospect-based emotion types from ‘joy’ and ‘distress’ can be illustrated as follows.

![Inheritance Diagram](attachment:image.png)

It should be noted that the dashes accompanying ‘satisfaction’, ‘fears-confirmed’, ‘relief’, and ‘disappointment’ are intentional and do not indicate a problem. For example, ‘satisfaction’ is a label for a positively valenced reaction to the confirmation of a prospective desirable consequence, but the dashes below ‘satisfaction’ are a placeholder for a negatively valenced reaction to the confirmation of a prospective desirable consequence. Of course, in practice such a negative reaction to something positive never occurs, and thus does not have to be labeled.

We can now put the entire inheritance hierarchy together, the result of which is Figure 2.1. It should be noted that Figure 2.1 can actually be regarded as two superimposed hierarchies; namely, one hierarchy for the positively valenced emotion types (i.e., the top halves of the boxes) and one hierarchy for the negatively valenced emotion types (i.e., the bottom halves of the boxes). Such a separation also resolves the dashes. In the ‘positive’ hierarchy, there would be four boxes at the bottom (satisfaction, relief, happy-for, gloating), and in the ‘negative’ hierarchy, there would also be four boxes at the bottom (fears-confirmed, disappointment, resentment, pity). This distinction is illustrated in Figures 2.2 and 2.3, which do not contain any unlabeled types. Superimposing Figures 2.2 and 2.3 will result in Figure 2.1.
2.4 Concluding Remarks

Having now discussed all emotions types distinguished in the OCC model, it will be interesting to check the correctness of our inheritance-based perspective (as illustrated in Figure 2.1) with respect to the eliciting conditions as specified by OCC. This check can be performed as follows. Sentences can be constructed by reading the text on the links from the top of the hierarchy to any node. The resulting sentences should be comparable to OCC’s specifications of the eliciting conditions, as collected in Table 2.2. For example, according to Figure 2.1, ‘relief’ is a “valenced reaction to [a] consequence (of an event) which is actual and which disconfirms [a] prospective undesirable consequence.” Indeed, this is more verbose than “(pleased about) the disconfirmation of the prospect of an undesirable event,” but it is accurate granted some purely grammatical transformations. The ‘new’ specification for ‘relief’ is even more precise than the one given by OCC, because it explicates the distinction between consequences and events and the fact that ‘relief’ is a reaction to an actual consequence.

The eliciting conditions resulting from constructing sentences from Figure 2.1 (granted some minor grammatical transformations) have been collected in Table 2.3. Probably more insightful however, and certainly less wordy, is to construct sentences by reading Figure 2.1 only from child to parent node(s). The specifications of eliciting conditions resulting from such readings are listed in Table 2.4. For example, ‘relief’ then becomes simply “joy about the disconfirmation of a prospective undesirable consequence.” Table 2.4 will play an important role in guiding our formalization of emotions in the following chapters.

Finally, we emphasize again that there are many ways to structure the emotion types of the OCC model. What we have just done is structuring the emotion types of the OCC model in a purely logical and compositional way based on their eliciting conditions. In the book [Ortony et al., 1988], a slightly different structure is presented on page 19 (see Figure A.1 on page 188). That structure, however, is based on focus of attention and not on eliciting conditions. For example, it depicts ‘happy-for’ as an emotion where one is “focusing on” consequences for others and ‘joy’ as an emotion where one is “focusing on” consequences for the self. This differentiation makes it more intuitive to place the four fortunes-of-other emotions next to ‘joy’ and ‘distress’, instead of below ‘joy’ and ‘distress’ as their eliciting conditions would dictate. The emotion types can be structured in many other ways as well (e.g., based on motivational and behavioral aspects, a mapping of feelings, physiological antecedents or effects, etc.), depending on which perspective one wishes to emphasize. For our formalization of appraisal, however, we are interested in specifying when emotions are triggered, so the presented structure of their eliciting conditions will be our guide in the rest of this dissertation.
Figure 2.1: An inheritance-based view of the eliciting conditions of emotions of the OCC model.
**positive** is a valenced reaction

**negative** is a valenced reaction

**pleased** is a positive reaction to a consequence (of an event)

**displeased** is a negative reaction to a consequence (of an event)

**hope** is a positive reaction to a prospective consequence (of an event)

**fear** is a negative reaction to a prospective consequence (of an event)

**joy** is a positive reaction to an actual consequence (of an event)

**distress** is a negative reaction to an actual consequence (of an event)

**satisfaction** is a positive reaction to an actual consequence (of an event)

which confirms a prospective desirable consequence

**fears-confirmed** is a negative reaction to an actual consequence (of an event)

which confirms a prospective undesirable consequence

**relief** is a positive reaction to an actual consequence (of an event)

which disconfirms a prospective undesirable consequence

**disappointment** is a negative reaction to an actual consequence (of an event)

which disconfirms a prospective desirable consequence

**happy-for** is a positive reaction to an actual consequence (of an event)

which is presumed to be desirable for another

**resentment** is a negative reaction to an actual consequence (of an event)

which is presumed to be desirable for another

**gloating** is a positive reaction to an actual consequence (of an event)

which is presumed to be undesirable for another

**pity** is a negative reaction to an actual consequence (of an event)

which is presumed to be undesirable for another

**approving** is a positive reaction to an action (of an agent)

**disapproving** is a negative reaction to an action (of an agent)

**pride** is a positive reaction to an action of oneself

**shame** is a negative reaction to an action of oneself

**admiration** is a positive reaction to an action of another

**reproach** is a negative reaction to an action of another

**gratification** is a positive reaction to an action of oneself and a related consequence

**remorse** is a negative reaction to an action of oneself and a related consequence

**gratitude** is a positive reaction to an action of another and a related consequence

**anger** is a negative reaction to an action of another and a related consequence

**liking** is a positive reaction to an aspect (of an object)

**disliking** is a negative reaction to an aspect (of an object)

**Table 2.3**: The emotion type specifications corresponding directly to Figure 2.1.
positive and negative are valenced reactions (to “something”)

pleased is being positive about a consequence (of an event)
displeased is being negative about a consequence (of an event)

hope is being pleased about a prospective consequence (of an event)
fear is being displeased about a prospective consequence (of an event)
joy is being pleased about an actual consequence (of an event)
distress is being displeased about an actual consequence (of an event)
satisfaction is joy about the confirmation of a prospective desirable consequence

fears-confirmed is distress about the confirmation of a prospective undesirable consequence
relief is joy about the disconfirmation of a prospective undesirable consequence
disappointment is distress about the disconfirmation of a prospective desirable consequence

happy-for is joy about a consequence (of an event)

presumed to be desirable for someone else
resentment is distress about a consequence (of an event)

presumed to be desirable for someone else
gloating is joy about a consequence (of an event)

presumed to be undesirable for someone else
pity is distress about a consequence (of an event)

presumed to be undesirable for someone else
approving is being positive about an action (of an agent)

presumed to be one’s own action
pride is approving of one’s own action

shame is disapproving of one’s own action
admiration is approving of someone else’s action

reproach is disapproving of someone else’s action
gratification is pride about an action and joy about a related consequence

remorse is shame about an action and distress about a related consequence
gratitude is admiration about an action and joy about a related consequence

anger is reproach about an action and distress about a related consequence
liking is being positive about an aspect (of an object)

disliking is being negative about an aspect (of an object)

Table 2.4: The emotion type specifications corresponding to Figure 2.1, acquired by reading child to parent node(s).
Figure 2.2: An inheritance-based view of the eliciting conditions of the positively valenced emotion types of the OCC model.
Figure 2.3: An inheritance-based view of the eliciting conditions of the negatively valenced emotion types of the OCC model.
Chapter 3

Emotion Elicitation I:
The Logical Structure

In this chapter we will make a start with the formalization of the eliciting conditions of emotions according to the OCC model. The formalization of eliciting conditions will proceed in three stages, spread over Chapters 3, 4, and 5, respectively. The reason for this split is to provide different levels of commitment to formalisms. The first stage of formalization, presented in this chapter, is not oriented towards any logical framework in particular. As a result, all formulas presented in this chapter are only semiformal, since no semantics are yet given, only intuitive readings. We use logical connectives with their usual interpretation and some operators with suggestive names. The idea is that the presented formulas are formal enough to capture the logical structure of the psychological model in an adequate way, while remaining free from a commitment to an underlying formalism.

In later chapters we will increasingly commit to formalisms, eventually providing a firm grounding. However, it is our intention that it remains possible to plug in another formalism if desired. This way one can provide a different interpretation of the operators, without having to start from scratch regarding the overall logical structure of the model.

This chapter is organized as follows. First we will revisit the procedural view of emotion (see Figure 1.2) in order to explain in more detail what aspect of emotion is being formalized here. Then Section 3.2 presents the first stage of our formalization of eliciting conditions. Properties of this formalization are presented in Section 3.3 and related work is discussed in Section 3.4.

3.1 The Procedural View of Emotion Revisited

It is important to re-emphasize that a distinction is made between what triggers an emotion and how an emotion is experienced. This distinction was already touched
In this chapter (and the next two) we will be concerned with the formalization of the process of emotion elicitation. As can be seen in Figure 3.1, for this we will need to be able to represent percepts and concerns, which constitute the input for emotion elicitation, and triggered emotions, which constitute the output of emotion elicitation. These triggered emotions then represent the emotions that may be experienced by an agent having the percepts and concerns in question. Whether the triggered emotions are actually experienced is treated as a separate issue in this dissertation. Actual emotional experience depends on the assignment of emotional intensities, which again depends on individual parameters (one’s emotional ‘character’) and one’s mood. Because individual parameters are not yet taken into account in the process of emotion elicitation, the presented formalization of emotion elicitation will have a generic, ‘impersonal’ flavor.
In the rest of this dissertation, it is assumed that the following relation between emotion elicitation and emotion experience exists.

**Definition 3.1 (Emotional experience, informally)**

Let \( \text{emotion}_\text{type} \) stand for any emotion type relative to something (e.g., joy about winning, anger at being scolded, liking the dessert); then:

- \( \text{emotion}_\text{type} \) is experienced if and only if
  1. \( \text{emotion}_\text{type} \) has been triggered sometime in the past and
  2. overall felt intensity of \( \text{emotion}_\text{type} \) is positive.

The intensity at which an emotion is experienced is typically influenced by many factors. Moreover, emotional intensity is probably multidimensional [Frijda et al., 1992]. Nevertheless, it is also assumed that for any emotion an estimate can be made of its ‘overall felt intensity’. For example, a questionnaire about emotional feelings may include a question like “indicate how angry you were when hearing about the political murder on a scale from 1 to 10” and such questions are usually not difficult to answer.

With “... is positive” we mean having a value strictly greater than zero. Emotional intensity does not take on negative values. An emotion type for which the triggering conditions hold is not necessarily experienced, because its intensity may be too low (i.e., zero). Conversely, an emotion type which is being experienced does not have to have its triggering conditions to hold, because it may have been triggered some time in the past. In this chapter and the next two, we will focus on the triggering conditions of the emotion types of the OCC model. The treatment of emotional experience, as expressed in terms of triggering and intensity as above, will be deferred until Chapter 6. There we will also present a formal definition of emotional experience (namely Definition 6.9).

### 3.2 Formalization of Emotion Triggers

Having now distinguished emotion elicitation from emotional experience, we will focus on the formalization of emotion elicitation according to the OCC model. In the following subsections, we will formalize the triggering conditions of each of the 28 emotion types depicted in Figure 2.1. Usually the OCC model is said to distinguish 22 emotion types, but we arrive at 28 because we explicitly represent the “abstract” emotion types ‘positive’, ‘negative’, ‘pleased’, ‘displeased’, ‘approving’, and ‘disapproving’ as well. Formalizing these emotion types as well is important because of our compositional approach. For example, ‘joy’ and ‘hope’ are specializations of ‘pleased’ in the OCC model, so in order to derive formalizations for ‘joy’ and ‘hope’, we must first have a formalization of ‘pleased’.

The structure of the eliciting conditions of emotions was illustrated in Figure 2.1. This figure and its textual version, as presented in Table 2.4, will guide our formalization. The order in which the triggering conditions are formalized is from top to bottom with respect to Figure 2.1.
3.2.1 General Emotion Types

At the most abstract level, the OCC model considers every emotion as a valenced reaction, which can either be positive or negative. So ‘positive’ and ‘negative’ are regarded by OCC as the most general, undifferentiated emotion types. In order to know which one of ‘positive’ or ‘negative’ (or both, or neither) is triggered at some point, something must be perceived and valued, which is called appraisal. We should note that the term ‘appraisal’ can be used to mean perception plus valuation. In order to avoid confusion, we explicitly mention perception (of an event, action, or object) as a precondition for appraisal, and use the term appraisal strictly for valuation.

According to Figure 3.1, the triggering conditions of a ‘positive’ or ‘negative’ emotion must have two ingredients: a percept and a concern. Without specifying the kind of percept or concern we are talking about, we can do no better than say that the outcome of the valuation of the percept against the concern can be either ‘good’ or ‘bad’. We can thus specify the triggering conditions of the emotion types ‘positive’ and ‘negative’ as the perception of something good and bad, respectively, as follows.

Definition 3.2 (Eliciting conditions of ‘positive’ and ‘negative’)
A positive emotion is triggered for agent i with respect to X if\(\text{iff}\) X is perceived by i and valued as good by i, which is denoted as \(\text{Positive}_i^T(X)\). Likewise, a negative emotion is triggered for agent i with respect to X if\(\text{iff}\) X is perceived by i and valued as bad by i, which is denoted as \(\text{Negative}_i^T(X)\).

\[
\text{Positive}_i^T(X) \equiv \text{Perceive}_i(X) \land \text{Good}_i(X) \quad (3.1)
\]
\[
\text{Negative}_i^T(X) \equiv \text{Perceive}_i(X) \land \text{Bad}_i(X) \quad (3.2)
\]

Emotions are always relative to something, and here X stands for that “something.” \(\text{Positive}_i^T(X)\) is read as “a positively valenced reaction to X is triggered for agent i.” The superscript “T” (for trigger) indicates that we are talking about eliciting conditions, in order to avoid confusion with actual experience. It is crucial to note that \(\text{Positive}_i^T(X)\) is not the same as “agent i is positive about X.” The feeling of being positive about X may manifest itself gradually over time, if at all, and may not coincide with the satisfaction of its triggering conditions, which is what \(\text{Positive}_i^T(X)\) expresses. Emotional experience will be written without the superscript “T”, i.e., “agent i is positive about X” will be denoted as \(\text{Positive}_i(X)\) (see Chapter 6). In this chapter, all emotion formulas will have a superscript “T” to indicate that the eliciting conditions are what is being formalized.

\(\text{Perceive}_i(X)\) is read as “agent i perceives X.” \(\text{Good}_i(X)\) is read as “agent i appraises X as good,” and similarly for \(\text{Bad}_i(X)\).

With ‘positive’ and ‘negative’ at the top of the hierarchy, the first differentiation is with respect to the object of the emotion. As previously described, the OCC model considers three types: consequences of events, actions of agents, and aspects of objects. This was illustrated in Figure 2.1 as follows.

\(^1\)“Iff” is a (common) abbreviation for “if and only if.”
We can thus define ‘perceive’ as a disjunction of perceiving either of these three kinds of percepts, as follows.

**Definition 3.3 (Perceiving)**
The OCC model distinguishes three kinds of percepts, so the perception construct \( \text{Perceive}_i(X) \) is defined as a disjunction of three different percept constructs, as follows.

\[
\text{Perceive}_i(X) \equiv \text{PerceiveConseq}_i(X) \lor \text{PerceiveAction}_i(X) \lor \text{PerceiveObject}_i(X)
\]  
(3.3)

These three perception constructs will be clarified in the next three subsections, respectively.

If a consequence of an event is appraised as being good or bad, it is said to be ‘desirable’ or ‘undesirable’, respectively. If an action of an agent is appraised as being good or bad, it is said to be ‘praiseworthy’ or ‘blameworthy’, respectively. If an aspect of an object is appraised as being good or bad, it is said to be ‘appealing’ or ‘unappealing’, respectively. ‘Good’ and ‘bad’ can thus be defined in terms of these six notions as follows.

**Definition 3.4 (General concerns)**
The appraisal constructs \( \text{Good} \) and \( \text{Bad} \) are defined in terms of six operators:

\[
\text{Good}_i(X) \equiv \text{Des}_i(X) \lor \text{Praisew}_i(X) \lor \text{Appeal}_i(X)
\]  
(3.4)

\[
\text{Bad}_i(X) \equiv \text{Undes}_i(X) \lor \text{Blamew}_i(X) \lor \text{Unappeal}_i(X)
\]  
(3.5)

where \( \text{Des} \) stands for “desirable,” \( \text{Undes} \) for “undesirable,” \( \text{Praisew} \) for “praiseworthy,” \( \text{Blamew} \) for “blameworthy,” \( \text{Appeal} \) for “appealing,” and \( \text{Unappeal} \) for “unappealing.”

A note about the types of arguments is in order here. As noted previously, desirability is only applicable to consequences of events, praiseworthiness is only applicable to actions of agents, etc. However, the current construction says that, e.g., \( \text{PerceiveConseq}_i(X) \land \text{Praisew}_i(X) \rightarrow \text{Positive}_i(X) \) is valid. Of course, in this...
example, either $\text{PerceiveConseq}_i(X)$ or $\text{Praisew}_i(X)$ must be applied to the wrong type of $X$. Therefore it is assumed that all these constructs\(^2\) evaluate to false if they are applied to an argument of the wrong type. This way $\text{PerceiveConseq}_i(X) \land \text{Praisew}_i(X)$ is always false and the implication is still true.

It should also be noted that none of the desirable–undesirable, praiseworthy–blameworthy, and appealing–unappealing pairs are considered to be opposites, nor are they considered to be mutually exclusive. For example, a consequence which is not desirable is not necessarily undesirable; a lack of appeal does not make something unappealing; and the exact same action can be appraised as being both praiseworthy and blameworthy. Therefore, we really need six distinct appraisal constructs here. These three pairs of appraisal operators will be clarified in the next three subsections, respectively.

**General Event-based Emotion Types**

Let us consider the emotion types concerning consequences of events. At the top of this branch are placed the labels ‘pleased’ and ‘displeased’ (see Figure 2.1). OCC consider desirability as the central variable measuring how positive a consequence of an event is for an individual. A consequence of an event that is valued negatively is called undesirable. As noted previously, undesirability is neither the same as the absence of desirability, nor are desirability and undesirability assumed to exclude each other; they are seen as separate variables.\(^3\) In the following, we will treat desirability in a qualitative manner, i.e., something is either desirable or not. Of course, there can be degrees of desirability, so when we say that something is desirable, this may be read as “having strictly positive desirability,” and when we say “not desirable,” this may be read as “having zero desirability.” Analogous readings apply to undesirability.

Below is then a (semiformal) logical description of the eliciting conditions of ‘pleased’ and ‘displeased’. $\text{Pleased}_i^T(c)$ should be read as “pleased about consequence $c$ of an event is triggered for agent $i$,” and similarly for ‘displeased’.

**Definition 3.5 (Eliciting conditions of ‘pleased’ and ‘displeased’)***

‘Pleased’ is triggered for agent $i$ with respect to $c$ if $c$ is a consequence (of an event) perceived by $i$ and valued as desirable by $i$, which is denoted as $\text{Pleased}_i^T(c)$. Likewise, ‘displeased’ is triggered for agent $i$ with respect to $c$ if $c$ is a consequence (of an event) perceived by $i$ and valued as undesirable by $i$, which is denoted as $\text{Displeased}_i^T(c)$.

\[
\text{Pleased}_i^T(c) \overset{\text{def}}{=} \text{PerceiveConseq}_i(c) \land \text{Des}_i(c) \quad (3.6)
\]

\[
\text{Displeased}_i^T(c) \overset{\text{def}}{=} \text{PerceiveConseq}_i(c) \land \text{Undes}_i(c) \quad (3.7)
\]

\(^2\)That is, $\text{PerceiveConseq}$, $\text{PerceiveAction}$, $\text{PerceiveObject}$, $\text{Des}$, $\text{Undes}$, $\text{Praisew}$, $\text{Blamew}$, $\text{Appeal}$, and $\text{Unappeal}$.

\(^3\)OCC use undesirability as a kind of negative desirability, i.e., a consequence is undesirable if its desirability is strictly less than zero. Here we prefer to keep desirability and undesirability as separate measures, each ranging over non-negative values. But it will be clear that the two approaches do not conflict.
These formulas express that the eliciting conditions of being ‘pleased’ and being ‘displeased’ have two components; namely, the perception of a consequence of an event and the appraisal of that consequence as being (un)desirable. \textit{PerceiveConseq},(c) is read as “agent i perceives consequence c (of an event)” and \textit{Des},(c) is read as “agent i appraises consequence c as desirable (with respect to its goals),” or, less precisely, “i desires c.” It will be clear that ‘pleased’ and ‘displeased’ are undifferentiated event-based emotions, because nothing is assumed about what kind of consequence we are dealing with, nor anything about who or what caused the event, nor to whom (other than the appraising agent) the consequence applies.

**General Action-based Emotion Types**

At the top of the branch of emotion types concerning actions of agents are placed the labels ‘approving’ and ‘disapproving’, which are regarded by OCC as the most general action-based emotion types. The OCC model considers \textit{praiseworthiness} and \textit{blameworthiness} to be the central variables for valuating actions of agents. Analogously to ‘pleased’ and ‘displeased’ (see above), ‘approving’ and ‘disapproving’ can be specified as perceiving an action of an agent and appraising that action as praiseworthy or blameworthy (or both, if one has conflicting standards). Below, \textit{Approving},(j:a) is read as “approving of action a by agent j is triggered for agent i,” and similarly for ‘disapproving’. Note that i and j may refer to the same agent.

**Definition 3.6 (Eliciting conditions of ‘approving’ and ‘disapproving’)**

‘Approving’ is triggered for agent i with respect to action a of agent j iff a by j is perceived by i and valued as praiseworthy by i, which is denoted as \textit{Approving},(j:a). Likewise, ‘disapproving’ is triggered for agent i with respect to action a of agent j iff a by j is perceived by i and valued as blameworthy by i, which is denoted as \textit{Disapproving},(j:a).

\[
\text{Approving},(j:a) \overset{\text{def}}{=} \text{PerceiveAction},(j:a) \land \text{Praisew},(j:a) \tag{3.8}
\]

\[
\text{Disapproving},(j:a) \overset{\text{def}}{=} \text{PerceiveAction},(j:a) \land \text{Blamew},(j:a) \tag{3.9}
\]

where j:a represents action a of agent j.

\textit{PerceiveAction},(j:a) is read as “agent i perceives agent j has performed action a.” \textit{Praisew},(j:a) is read as “agent i appraises action a by agent j as praiseworthy (with respect to its standards),” and similarly for \textit{Blamew},(j:a) and ‘blameworthy’.

A note about the notation “j:a” is in order here. The constructs based on actions of agents obviously require three arguments, namely the agent for which an emotion may be triggered, and the agent and its action toward which the emotion may be triggered. So for example, we could have written \textit{Praisew},(i, j, a) instead of \textit{Praisew},(j:a). However, in order to make it clear who is the appraising agent and who is the appraised agent, we distinguish the appraising agent by writing it as a subscript. Furthermore, we use the notation j:a to be more in line with notation commonly used in logics of action. So the notation j:a is just syntactic sugar for the tuple (j,a).
General Object-based Emotion Types

At the top of the branch of emotion types concerning aspects of objects are placed the labels ‘liking’ and ‘disliking’, which are regarded by OCC as the most general object-based emotion types. The OCC model considers appealingness and unappealingness to be the central variables for valuating aspects of objects. Just as desirability only applies to consequences of events, the OCC model considers appealingness to apply to aspects of objects. In the rest of this paper, however, we will simplify slightly by not representing aspects explicitly. This is usually not problematic, because different aspects of an object can often be regarded as objects themselves. For example, when one appraises a car, different aspects of the car (e.g., headlights, doors, wheels) are objects themselves that can be liked or disliked. The appraisal of aspects that are not objects (e.g., the car’s color) is simply assumed to be handled implicitly by the constructs for appealingness and unappealingness when applied to the object in question.

Analogously to the event-based and action-based emotion types above, ‘liking’ and ‘disliking’ can be specified as perceiving an object and appraising that object as appealing or unappealing. Liking\(^T\)(x) should be read as “liking of object x is triggered for agent i,” and similarly for ‘disliking’.

**Definition 3.7 (Eliciting conditions of ‘liking’ and ‘disliking’)**

‘Liking’ is triggered for agent i with respect to object x iff x is perceived by i and valued as appealing by i, which is denoted as Liking\(^T\)_i(x). Likewise, ‘disliking’ is triggered for agent i with respect to object x iff x is perceived by i and valued as unappealing by i, which is denoted as Disliking\(^T\)_i(x).

\[
\begin{align*}
\text{Liking}^T_i(x) & \equiv \text{PerceiveObject}^i(x) \land \text{Appeal}^i(x) \\
\text{Disliking}^T_i(x) & \equiv \text{PerceiveObject}^i(x) \land \text{Unappeal}^i(x)
\end{align*}
\]

where x represents an object.

PerceiveObject\(^i\)(x) is read as “agent i perceives object x.” Appeal\(^i\)(x) is read as “agent i appraises object x as appealing (with respect to its attitudes),” and similarly for Unappeal\(^i\)(x) and ‘unappealing’.

**3.2.2 Concrete Emotion Types**

Having specified the eliciting conditions of the first two layers of Figure 2.1, let us proceed to the third layer.

**Event-based Emotion Types**

The first differentiation with respect to event-based emotion types is on whether the consequence in question is prospective or actual. The general emotion types ‘pleased’ and ‘displeased’ are then branched into ‘hope’ and ‘fear on the one hand, and ‘joy’ and ‘distress’ on the other hand, as follows.
First, we will treat the case of prospective consequences of events, leading to the emotion types labeled as ‘hope’ and ‘fear’.

**Definition 3.8 (Eliciting conditions of ‘hope’ and ‘fear’)**
According to Table 2.4, ‘hope’ is ‘pleased’ about a prospective consequence (of an event), whereas ‘fear’ is ‘displeased’ about a prospective consequence (of an event). \( \text{Hope}_i^T(c) \) and \( \text{Fear}_i^T(c) \) are then defined in terms of \( \text{Pleased}_i^T(c) \) and \( \text{Displeased}_i^T(c) \) as follows.

\[
\text{Hope}_i^T(c) \overset{\text{def}}{=} \text{Pleased}_i^T(c) \land \text{Prospective}_i(c) \tag{3.12}
\]
\[
\text{Fear}_i^T(c) \overset{\text{def}}{=} \text{Displeased}_i^T(c) \land \text{Prospective}_i(c) \tag{3.13}
\]

where \( \text{Prospective}_i(c) \) asserts that \( c \) is prospective.

\( \text{Prospective}_i(c) \) is read as “agent \( i \) considers \( c \) to be a prospective consequence (of an event).” \( \text{Hope}_i^T(c) \) is then read as “hope about consequence \( c \) (of an event) is triggered for agent \( i \),” and similarly for ‘fear’.

Next, we will treat the case of actual consequences, leading to the emotion types labeled as ‘joy’ and ‘distress’.

**Definition 3.9 (Eliciting conditions of ‘joy’ and ‘distress’)**
According to Table 2.4, ‘joy’ is ‘pleased’ about an actual consequence (of an event), whereas ‘distress’ is ‘displeased’ about an actual consequence (of an event). \( \text{Joy}_i^T(c) \) and \( \text{Distress}_i^T(c) \) are then defined in terms of \( \text{Pleased}_i^T(c) \) and \( \text{Displeased}_i^T(c) \) as follows.

\[
\text{Joy}_i^T(c) \overset{\text{def}}{=} \text{Pleased}_i^T(c) \land \text{Actual}_i(c) \tag{3.14}
\]
\[
\text{Distress}_i^T(c) \overset{\text{def}}{=} \text{Displeased}_i^T(c) \land \text{Actual}_i(c) \tag{3.15}
\]

where \( \text{Actual}_i(c) \) asserts that \( c \) is actual.
**Actual** \(_i(c)\) is read as “agent \(i\) considers \(c\) to be an actual consequence (of an event).”

**Joy** \(_i^T(c)\) is then read as “joy about consequence \(c\) (of an event) is triggered for agent \(i\),” and similarly for ‘distress’.

We emphasize again that ‘joy’ and ‘distress’ are considered as nothing more than convenient labels for these emotion types. Other labels are perfectly possible as well; for example, emotions of the type labeled as ‘joy’ include contentment, delight, being glad, happiness, cheerfulness, being ecstatic, and so on and so forth. Similarly, emotions of the type labeled as ‘distress’ include sadness, upset, being distraught, shock, etc. If one further differentiates the type of event towards which one is distressed, even more specific labels can be chosen. For example, being distressed about the loss of a loved one can be labeled as ‘grief’ and being distressed about the loss of an opportunity can be labeled as ‘regret’. The OCC model does not pursue further differentiation of ‘joy’ and ‘distress’ besides the emotion types shown at the bottom of Figure 2.1, but this is certainly an interesting direction for future research.

**Attribution Emotion Types**

The OCC model considers one differentiation in the action-based emotion types, namely in the actor. By differentiating with respect to the concept of cognitive unit (see page 26), the action-based emotion types can be captured as follows.

**Definition 3.10 (Eliciting conditions of ‘pride’, ‘shame’, ‘admiration’, and ‘reproach’)**

According to Table 2.4, ‘pride’ is ‘approving’ of one’s own action, ‘shame’ is ‘disapproving’ of one’s own action, ‘admiration’ is ‘approving’ of someone else’s action, and ‘reproach’ is ‘disapproving’ of someone else’s action. \(\text{Pride}^\_i(j:a)\), \(\text{Shame}^\_i(j:a)\), \(\text{Admiration}^\_i(j:a)\), and \(\text{Reproach}^\_i(j:a)\) are then defined in terms of \(\text{Approving}^\_i(j:a)\) and \(\text{Disapproving}^\_i(j:a)\) as follows.

\[
\begin{align*}
\text{Pride}^\_i(j:a) & \triangleq \text{Approving}^\_i(j:a) \land \text{CogUnit}^\_i(j) & (3.16) \\
\text{Shame}^\_i(j:a) & \triangleq \text{Disapproving}^\_i(j:a) \land \text{CogUnit}^\_i(j) & (3.17) \\
\text{Admiration}^\_i(j:a) & \triangleq \text{Approving}^\_i(j:a) \land \neg \text{CogUnit}^\_i(j) & (3.18) \\
\text{Reproach}^\_i(j:a) & \triangleq \text{Disapproving}^\_i(j:a) \land \neg \text{CogUnit}^\_i(j) & (3.19)
\end{align*}
\]

where \(\text{CogUnit}^\_i(j)\) is used to distinguish between the self and others.

\(\text{CogUnit}^\_i(j)\) is read as “agent \(i\) views agent \(j\) as being in a cognitive unit with itself.” \(\text{Pride}^\_i(j:a)\) is then read as “pride about action \(a\) of agent \(j\) is triggered for agent \(i\),” and similarly for ‘shame’, ‘admiration’, and ‘reproach’.

**Attraction Emotion Types**

The OCC model does not structure the valenced reactions to aspects of objects, even though OCC admit that momentary reactions of liking and disliking are among the
most salient experiences for humans. Interestingly, they do consider one variable affecting the intensity of liking and disliking reactions (besides appealingness), namely familiarity, but they have chosen not to differentiate based on this variable.

In contrast, below approving/disapproving, the variable strength of cognitive unit is used to differentiate between pride/shame on the one hand (i.e., the acting agent is in a cognitive unit with the self) and admiration/reproach on the other hand (i.e., the acting agent is distinct from the self). Likewise, below pleased/displeased, the variable likelihood is used to differentiate between hope/fear on the one hand (i.e., an event is possible but not certain) and joy/distress on the other hand (i.e., an event has actually happened). Analogously, one could use the variable familiarity to differentiate between ‘love’ and ‘hate’ types on the one hand (i.e., for familiar objects) and ‘attraction’ and ‘disgust’ types on the other hand (i.e., for unfamiliar objects).

Doing so would then result in definitions such as $\text{Love}_T^T(x) \overset{\text{def}}{=} \text{Liking}_T^T(x) \land \text{Familiar}_i(x)$, $\text{Hate}_T^T(x) \overset{\text{def}}{=} \text{Disliking}_T^T(x) \land \text{Familiar}_i(x)$, $\text{Attraction}_T^T(x) \overset{\text{def}}{=} \text{Liking}_T^T(x) \land \neg \text{Familiar}_i(x)$, and $\text{Disgust}_T^T(x) \overset{\text{def}}{=} \text{Disliking}_T^T(x) \land \neg \text{Familiar}_i(x)$. However, differentiating based on familiarity would not be correct because the relation between familiarity and overall liking or disliking is not monotonic [Ortony et al., 1988; Ortony and Clore, 2009; Ortony, 2009]. As is also suggested by the proverb “familiarity breeds contempt,” liking of an object can decrease when one is very familiar with it, even though initially, liking usually increases with familiarity. Indeed, in the OCC model it is suggested that the relation between familiarity and overall liking probably follows a bell shape.

### 3.2.3 Compounds

Two branches of Figure 2.1 combine to form the so-called compound emotion types. These emotions arise when one focuses on both the praiseworthiness of an action and the desirability of the related consequences. According to OCC, the eliciting conditions of the compound emotion types are a conjunction of the eliciting conditions of an event-based emotion (‘joy’ or ‘distress’) and an action-based emotion (‘pride’, ‘shame’, ‘admiration’, or ‘reproach’), together with an assertion about their relatedness. In our inheritance-based perspective, this was illustrated as follows.
However, care must be taken when formalizing the depicted multiple inheritance, because of temporal differences between the constituent event-based and action-based emotions. Specifically, realizing that an action and a consequence of an event are related may come at a later time than perceiving either the action or the consequence. Moreover, the perceptions of the action and its consequences may occur at different time points. In order to capture such temporal differences, we need to be able to look back in time when describing the eliciting conditions of the compound emotion types. To this end, we use the construct Past $\varphi$, which asserts that $\varphi$ was true sometime in the past, where, importantly, the past is understood to include the present. The compound emotion types can then be captured as follows.

**Definition 3.11 (Eliciting conditions of ‘gratification’, ‘remorse’, ‘gratitude’, and ‘anger’)**

According to Table 2.4, ‘gratification’ is ‘pride’ about an action and ‘joy’ about a related consequence, ‘remorse’ is ‘shame’ about an action and ‘distress’ about a related consequence, ‘gratitude’ is ‘admiration’ about an action and ‘joy’ about a related consequence, and ‘anger’ is ‘reproach’ about an action and ‘distress’ about a related consequence. Gratification$^T$($j$:$a$, $c$), Remorse$^T$($j$:$a$, $c$), Gratitude$^T$($j$:$a$, $c$), and Anger$^T$($j$:$a$, $c$) are then defined as follows.

\[
\text{Gratification}^T(j:a, c) \overset{\text{def}}{=} \text{Past Pride}^T(j:a) \land \text{Past Joy}^T(c) \\
\quad \land \text{PerceiveRelated}_i(j:a, c)
\]

\[
\text{Remorse}^T(j:a, c) \overset{\text{def}}{=} \text{Past Shame}^T(j:a) \land \text{Past Distress}^T(c) \\
\quad \land \text{PerceiveRelated}_i(j:a, c)
\]

\[
\text{Gratitude}^T(j:a, c) \overset{\text{def}}{=} \text{Past Admiration}^T(j:a) \land \text{Past Joy}^T(c) \\
\quad \land \text{PerceiveRelated}_i(j:a, c)
\]

\[
\text{Anger}^T(j:a, c) \overset{\text{def}}{=} \text{Past Reproach}^T(j:a) \land \text{Past Distress}^T(c) \\
\quad \land \text{PerceiveRelated}_i(j:a, c)
\]

where PerceiveRelated$^T_i(j:a, c)$ is used to express the perception of a relation between an action and a consequence.

PerceiveRelated$^T_i(j:a, c)$ is read as “agent $i$ perceives action $a$ of agent $j$ as being related to consequence $c$.” Gratification$^T(j:a, c)$ is then read as “gratification about action $a$ of agent $j$ and the related consequence $c$ is triggered for agent $i$,” and similarly for ‘remorse’, ‘gratitude’, and ‘anger’. We emphasize that each instance of Past $\varphi$ (read as “some time in the past, $\varphi$ was true”) refers to an arbitrary time in the past, including the present. Each of the four formulas above contains two instances of Past $\varphi$, each of which may thus refer to different times in the past.

In order to ensure that the action appearing twice in each of these definitions is really the same action, it is assumed that all actions are unique. This can be seen as each performed action being a unique instance of an action. For example, $j:a$ in Gratification$^T(j:a, c)$ appears in both Pride$^T(j:a)$ and PerceiveRelated$^T_i(j:a, c)$. With this assumption of uniqueness, the action that is the object of the perceived relation (i.e., $j:a$ in PerceiveRelated$^T_i(j:a, c)$) must really be the same action as the one which
is the object of the earlier action-based emotion (e.g., \(ja\) in \(\text{Pride}^j_i(j:a)\)). Being only semiformal in this chapter, we can do no better than simply stating this assumption in text; however, we will formalize this assumption as well in later chapters.

### 3.2.4 Derived Emotion Types

Finally, we come to the bottom layer of Figure 2.1, containing the prospect-based emotion types and the fortunes-of-others emotion types. All these emotion types are specializations of ‘joy’ and ‘distress’, as illustrated in the following hierarchy.

We will first treat the four prospect-based emotion types and then the four fortunes-of-others emotion types.

#### Prospect-based Emotion Types

Whereas ‘hope’ and ‘fear’ concern unconfirmed prospects of events, the OCC model also distinguishes emotion types concerning confirmed and disconfirmed prospects, namely ‘satisfaction’, ‘fears-confirmed’, ‘relief’, and ‘disappointment’. As explained in Section 2.3.6, a confirmation or disconfirmation is regarded as an actual consequence of an event, and therefore these emotion types are specializations of ‘joy’ and ‘distress’. However, they do depend on an earlier instance of ‘hope’ or ‘fear’, so the formalizations below use the \(\text{Past}\) operator to capture this temporal link. These four prospect-based emotion types can then be captured as follows.

**Definition 3.12 (Eliciting cond. of ‘satisfaction’, ‘fears-confirmed’, ‘relief’, and ‘disappointment’)**

According to Table 2.4, ‘satisfaction’ is ‘joy’ about the confirmation of a prospective desirable consequence, ‘fears-confirmed’ is ‘distress’ about the confirmation of a prospective undesirable consequence, ‘relief’ is ‘joy’ about the disconfirmation of a prospective undesirable consequence, and ‘disappointment’ is ‘distress’ about the disconfirmation of a prospective desirable consequence. \(\text{Satisfaction}^T_i(c, c')\), \(\text{Fears-confirmed}^T_i(c, c')\), \(\text{Relief}^T_i(c, c')\), and \(\text{Disappointment}^T_i(c, c')\) are then defined in terms of \(\text{Joy}^T_i(c)\) and \(\text{Distress}^T_i(c)\) as follows.

\[
\text{Satisfaction}^T_i(c, c') \equiv \text{Joy}^T_i(c) \land \text{Past Hope}^T_i(c') \land \text{Confirms}_i(c, c') \quad (3.24)
\]
EMOTION ELICITATION I: THE LOGICAL STRUCTURE

3.2

Fears-confirmed\(T_i(c, c')\) \(\text{def} = \text{Distress}^T_i(c) \land \text{Past Fear}^T_i(c') \land \text{Confirms}_i(c, c')\) \hspace{1cm} (3.25)

Relief\(T_i(c, c')\) \(\text{def} = \text{Joy}^T_i(c) \land \text{Past Fear}^T_i(c') \land \text{Disconfirms}_i(c, c')\) \hspace{1cm} (3.26)

Disappointment\(T_i(c, c')\) \(\text{def} = \text{Distress}^T_i(c) \land \text{Past Hope}^T_i(c') \land \text{Disconfirms}_i(c, c')\) \hspace{1cm} (3.27)

where Confirms\(_i(c, c')\) and Disconfirms\(_i(c, c')\) assert that one consequence confirms or disconfirms another consequence.

Confirms\(_i(c, c')\) is read as “agent \(i\) considers consequence \(c\) as (partially) confirming consequence \(c'\),” and likewise for ‘disconfirm’. Satisfaction\(_i(c, c')\) is then read as “satisfaction about consequence \(c\) confirming consequence \(c'\) is triggered for agent \(i\),” and similarly for ‘fears-confirmed’, ‘relief’, and ‘disappointment’.

It should be noted that defining the eliciting conditions of, e.g., ‘satisfaction’ as Satisfaction\(_i(c, c') = \text{Joy}^T_i(c) \land \text{Past Hope}^T_i(c')\) is not correct, because such a definition does not account for partial confirmations. For example, many government leaders were (mildly) satisfied when a weak climate deal was struck in Copenhagen in late 2009, even though they had hoped for a much more comprehensive deal. The weak deal then partially satisfied what they had hoped for, and thus they could be satisfied that at least something had been achieved. The expression \(\text{Joy}^T_i(c) \land \text{Past Hope}^T_i(c')\) would be too strong because it demands that nothing less than the comprehensive deal hoped for must come about for satisfaction to be triggered.

With these definitions of ‘satisfaction’, ‘fears-confirmed’, ‘relief’, and ‘disappointment’ it becomes apparent what the dashes accompanying these emotion types in Figure 2.1 stand for. For example, the specification of the eliciting conditions of the the empty spot below ‘satisfaction’ would read Distress\(_i(c) \land \text{Past Hope}^T_i(c') \land \text{Confirms}_i(c, c')\). Of course, in humans the confirmation of a prospective desirable consequence will never be appraised negatively, so this type of ‘distress’ can be discarded.

Fortunes-of-others Emotion Types

Finally, the four so-called fortunes-of-others emotion types are also specializations of ‘joy’ and ‘distress’, as explained in Section 2.3.5. These emotion types concern consequences of events presumed to be desirable or undesirable for someone else. In order to capture presumptions, we introduce the Presume operator; Presume\(_i\varphi\) is read as “agent \(i\) presumes \(\varphi\) (to be true).” When grounding these semiformal specifications in a BDI-based logic (as we will do later), the presume operator can easily be conflated with belief. However, in order to remain independent of any underlying formalism, we stick to OCC’s phrasing at this point and use ‘presume’ as the name for the operator. The fortunes-of-others emotion types can then be captured as follows.

Definition 3.13 (Eliciting conditions of ‘happy-for’, ‘pity’, ‘gloating’, and ‘resentment’)

According to Table 2.4, ‘happy-for’ is ‘joy’ about a consequence (of an event) presumed to be desirable for someone else, ‘resentment’ is ‘distress’ about a consequence (of an event) presumed to be desirable for someone else, ‘gloating’ is ‘joy’ about a consequence (of an event)
presumed to be undesirable for someone else, and ‘pity’ is ‘distress’ about a consequence (of an event) presumed to be undesirable for someone else. Happy_for_T(\(c, j\)), Pity_T(\(c, j\)), Gloating_T(\(c, j\)), and Resentment_T(\(c, j\)) are then defined in terms of Joy_T(\(c\)) and Distress_T(\(c\)) as follows.

\[
\begin{align*}
\text{Happy-for}_T(c, j) & \equiv \text{Joy}_T(c) \land \text{Presume}_i \text{Des}_j(c) \\
\text{Pity}_T(c, j) & \equiv \text{Distress}_T(c) \land \text{Presume}_i \text{Undes}_j(c) \\
\text{Gloating}_T(c, j) & \equiv \text{Joy}_T(c) \land \text{Presume}_i \text{Undes}_j(c) \\
\text{Resentment}_T(c, j) & \equiv \text{Distress}_T(c) \land \text{Presume}_i \text{Des}_j(c)
\end{align*}
\]

where \(\text{Presume}_i \varphi\) is used to capture presumptions of agents.

Happy_for_T(\(c, j\)) is read as “happy-for about consequence \(c\) (of an event) for agent \(j\) is triggered for agent \(i\),” and similarly for ‘pity’, ‘gloating’, and ‘resentment’.

It may be interesting to note that it is not required for agent \(i\) to presume that agent \(j\) is aware of the event in question as well. For example, assume Alice has just learned that she has won a magnificent cruise for two. She may feel very happy for her husband (who she intends to take the cruise with) without him being aware of the prize yet. Of course, Alice may feel inclined to tell her husband about the prize as soon as possible, but it would be unreasonable to argue that she cannot feel happy for him before having informed him.

### 3.3 Properties

This section presents some properties of the (semiformal) specifications presented in this chapter. In the following propositions, let \(\vdash_T\) (where \(T\) stands for trigger) be a classical propositional entailment relation with formulas (3.1)–(3.31) as axioms. Furthermore, \(\Gamma \vdash_T \varphi\) denotes that \(\varphi\) is derivable assuming \(\Gamma\). Although formal proofs of theorems appearing later in this dissertation are provided in the respective chapters, the derivations of the theorems below only involve manipulation of regular propositional connectives and therefore we deem it as unnecessary to spell out these derivations.

The following theorems read exactly as the type specifications for ‘pleased’, ‘displeased’, ‘approving’, ‘disapproving’, ‘liking’, and ‘disliking’ given in Table 2.4.

**Proposition 3.14 (‘Positive’ and ‘negative’)**

The following equivalences hold:

\[
\begin{align*}
\vdash_T \text{Pleased}_T(c) & \iff \text{Positive}_T(c) \\
\vdash_T \text{Displeased}_T(c) & \iff \text{Negative}_T(c) \\
\vdash_T \text{Approving}_T(i:a) & \iff \text{Positive}_T(i:a) \\
\vdash_T \text{Disapproving}_T(i:a) & \iff \text{Negative}_T(i:a)
\end{align*}
\]
⊢ \text{Liking}^T_i(x) \leftrightarrow \text{Positive}^T_i(x) \quad (3.36)

⊢ \text{Disliking}^T_i(x) \leftrightarrow \text{Negative}^T_i(x) \quad (3.37)

For example, Table 2.4 states that “pleased is being positive about a consequence (of an event).” So if we put a consequence \(c\) into ‘positive’ (i.e., \(\text{Positive}^T_i(c)\)), we should get ‘pleased’. And indeed, formula (3.32) states that \(\text{Positive}^T_i(c)\) is equivalent to \(\text{Pleased}^T_i(c)\). The other theorems follow the same pattern.

The following theorems state that each pair of ‘siblings’ in the third layer of Figure 2.1 completely subdivide their ‘parent’.

**Proposition 3.15 (Subdivision of abstract emotion types)**

The following equivalences hold:

\[ \Gamma \vdash \text{Pleased}^T_i(c) \leftrightarrow (\text{Hope}^T_i(c) \lor \text{Joy}^T_i(c)) \quad (3.38) \]

\[ \Gamma \vdash \text{Displeased}^T_i(c) \leftrightarrow (\text{Fear}^T_i(c) \lor \text{Distress}^T_i(c)) \quad (3.39) \]

\[ \vdash \text{Approving}^T_i(i:a) \leftrightarrow (\text{Pride}^T_i(j:a) \lor \text{Admiration}^T_i(j:a)) \quad (3.40) \]

\[ \vdash \text{Disapproving}^T_i(i:a) \leftrightarrow (\text{Shame}^T_i(j:a) \lor \text{Reproach}^T_i(j:a)) \quad (3.41) \]

where \(\Gamma = \text{PerceiveConseq}_i(c) \rightarrow (\text{Actual}_i(c) \lor \text{Prospective}_i(c))\).

The assumption \(\Gamma\) expresses that all consequences that can be perceived are either actual or prospective, in line with the OCC model. This assumption itself will become a theorem in the next chapter.

The following theorems state that ‘siblings’ in the third layer of Figure 2.1 exclude each other.

**Proposition 3.16 (Mutual exclusions)**

The following exclusions hold:

\[ \Gamma \vdash \neg(\text{Hope}^T_i(c) \land \text{Joy}^T_i(c)) \quad (3.42) \]

\[ \Gamma \vdash \neg(\text{Fear}^T_i(c) \land \text{Distress}^T_i(c)) \quad (3.43) \]

\[ \vdash \neg(\text{Pride}^T_i(j:a) \land \text{Admiration}^T_i(j:a)) \quad (3.44) \]

\[ \vdash \neg(\text{Shame}^T_i(j:a) \land \text{Reproach}^T_i(j:a)) \quad (3.45) \]

where \(\Gamma = \neg(\text{Actual}_i(c) \land \text{Prospective}_i(c))\).

Together with the previous set of theorems, this means that the differentiations directly below pleased/displeased and approving/disapproving are strict and complete. It should be noted that these theorems do not express that, e.g., an agent cannot experience fear and distress at the same time. The fact that \(\text{Fear}^T_i(c) \land \text{Distress}^T_i(c)\) is a contradiction means that the perception of one consequence \(c\) cannot trigger both fear and distress with respect to \(c\) (because the triggering conditions for the ‘fear’ and ‘distress’ emotion types exclude each other). The assumption \(\Gamma\) expresses that the same consequence cannot be found to be both actual and prospective, which should obviously be a property and not an assumption. Indeed, in the chapters to
follow Actual and Prospective will be defined such that $\text{Actual}_{i}(c) \land \text{Prospective}_{i}(c)$ becomes a contradiction.

The following theorems state that the cases in which (previously unconfirmed) consequences of events become fully realized will trigger ‘satisfaction’ or ‘fears-confirmed’ with respect to the complete consequence hoped for or feared.

**Proposition 3.17 (Full confirmation)**

The following implications hold:

$$
\Gamma \vdash T_{i}(\text{Joy}_{i}(c) \land \text{Past Hope}_{i}(c)) \rightarrow \text{Satisfaction}_{i}(c, c) \\
\Gamma \vdash T_{i}(\text{Distress}_{i}(c) \land \text{Past Fear}_{i}(c)) \rightarrow \text{Fears-confirmed}_{i}(c, c)
$$

where $\Gamma = \text{Confirms}_{i}(c, c)$.

The assumption that confirmation is reflexive will sound very natural. Indeed, in Chapter 5 Confirms will be defined such that $\text{Confirms}_{i}(c, c)$ becomes a theorem itself.

The following theorems state that ‘awareness’ of what one finds desirable and undesirable leads to ‘joy’ being equivalent to “happy-for-self” and ‘distress’ being equivalent to “self-pity.”

**Proposition 3.18 (Fortunes of the self)**

The following equivalences hold:

$$
\Gamma_{1} \vdash T_{i}(\text{Joy}_{i}(c) \leftrightarrow \text{Happy-for}_{i}(c, i)) \\
\Gamma_{2} \vdash T_{i}(\text{Distress}_{i}(c) \leftrightarrow \text{Pity}_{i}(c, i))
$$

where $\Gamma_{1} = \text{Des}_{i}(c) \rightarrow \text{Presume}_{i}\text{Des}_{i}(c)$ and $\Gamma_{2} = \text{Undes}_{i}(c) \rightarrow \text{Presume}_{i}\text{Undes}_{i}(c)$.

In the rest of this dissertation we will not pursue the idea of whether or not agents are ‘aware’ of their own desires any further.

The following theorems state that proper ‘pride’ and ‘shame’ (in the sense that the agent of the praiseworthy/blameworthy action in question is exactly the self) are equivalent to “self-approving” and “self-disapproving,” respectively.

**Proposition 3.19 (Actions of the self)**

The following equivalences hold:

$$
\Gamma \vdash T_{i}(\text{Pride}_{i}(i:a) \leftrightarrow \text{Approving}_{i}(i:a)) \\
\Gamma \vdash T_{i}(\text{Shame}_{i}(i:a) \leftrightarrow \text{Disapproving}_{i}(i:a))
$$

where $\Gamma = \text{CogUnit}_{i}(i)$.

Note that these theorems read exactly as the specifications for ‘pride’ and ‘shame’ in Table 2.4, e.g., “pride is approving of one’s own action.” The assumption that an agent will consider itself to be in a cognitive unit with itself may be violated in
some pathological cases, but for healthy individuals in normal circumstances this assumption will obviously hold.

The inheritance-based view of the eliciting conditions of emotions, as illustrated in Figure 2.1, raises the expectation that each depicted emotion type implies its parent. Indeed, chains of implications such as the one below can be made for all emotion types. For example:

\[ \vdash T \text{Gloating}_i^T(c, j) \rightarrow \text{Joy}_i^T(c) \]
\[ \vdash T \text{Joy}_i^T(c) \rightarrow \text{Pleased}_i^T(c) \]
\[ \vdash T \text{Pleased}_i^T(c) \rightarrow \text{Positive}_i^T(c) \]

However, each of the compound emotion types does inherit the eliciting conditions of its parents (e.g., ‘remorse’ inherits from ‘distress’ and ‘shame’), but they are preceded by a Past operator because the two inherited sets of conditions do not have to be satisfied at the same time (see Section 3.2.3). Still, a chain of implications can be made if one allows it to be “contaminated” by a Past operator. For example, we can form the following theorems:

\[ \vdash T \text{Displeased}_i^T(c) \rightarrow \text{Negative}_i^T(c) \]
\[ \vdash T \text{Distress}_i^T(c) \rightarrow \text{Displeased}_i^T(c) \]
\[ \vdash T \text{Remorse}_i^T(c, j,a) \rightarrow \text{Past Distress}_i^T(c) \]
\[ \vdash T \text{Remorse}_i^T(c, j,a) \rightarrow \text{Past Shame}_i^T(j,a) \]
\[ \vdash T \text{Shame}_i^T(j,a) \rightarrow \text{Disapproving}_i^T(j,a) \]
\[ \vdash T \text{Disapproving}_i^T(j,a) \rightarrow \text{Negative}_i^T(j,a) \]

These theorems have been outlined such that the multiple inheritance of the compound emotion types becomes apparent if the page is turned 90 degrees counterclockwise.

Finally, it is worth emphasizing that the OCC model does not require appraisal to be consistent. Indeed, the following propositions are not derivable.

\[ \not \vdash T \neg (\text{Des}_i(c) \land \text{Undes}_i(c)) \] (3.52)
\[ \not \vdash T \neg (\text{Praisew}_i(j,a) \land \text{Blamew}_i(j,a)) \] (3.53)
\[ \not \vdash T \neg (\text{Appeal}_i(x) \land \text{Unappeal}_i(x)) \] (3.54)

So it is not assumed that an agent’s goals, standards, and attitudes are consistent. This implies that ‘mixed feelings’ are possible; that is, formulas such as \( \text{Admiration}_i^T(j,a) \land \text{Reproach}_i^T(j,a) \) are satisfiable. In fact, for each pair of ‘opposing’ emotion types (i.e., those sharing a box in Figure 2.1), we have that their eliciting conditions do not exclude each other. With slight abuse of notation, this can be expressed as follows.

\[ \not \vdash T \neg (\text{Emotion}_i^T(X) \land \text{Emotion}_i^{-T}(X)) \] (3.55)

where, e.g., \( \text{Emotion}_i^T(X) \) stands for \( \text{Hope}_i^T(c) \) and \( \text{Emotion}_i^{-T}(X) \) stands for \( \text{Fear}_i^T(c) \).
3.4 Related Work

In this section we discuss several related attempts at adopting psychological models of emotions for modeling artificial agents. We will discuss similarities and differences with the presented approach.

3.4.1 Previous Work

In previous work, Meyer [2006] and Dastani and Meyer [2006] proposed a functional approach to describe the role of four basic emotions in practical reasoning. According to this functional approach, an agent is assumed to execute domain actions in order to reach its goals. The effects of these domain actions cause and/or influence the elicitation of emotions according to a human-inspired model. These emotions in turn influence the deliberation operations of the agent, functioning as heuristics for determining which domain actions have to be chosen next, which completes the cycle.

The specification and implementation of emotions carried out by Meyer [2006] and Dastani and Meyer [2006] follows the model of emotions by Oatley and Jenkins [1996]. In contrast to our approach of capturing a broad and complete range of emotion types, they consider only four emotions: happy, sad, angry, and fearful. Each emotion functions as a label of an aspect of an agent’s cognitive state. The deliberation of an agent then behaves in accordance with heuristics associated with these four emotions. Later we have extended this approach by showing how interaction between hope and fear can influence an agent’s deliberation [Steunebrink et al., 2007]. The present chapter, however, contains the first complete presentation of our formalization of the eliciting conditions of the emotion types of the OCC model.

3.4.2 A Related Formalization of the OCC Model

The construction of a complete formalization of the OCC model in agent logic has previously been attempted by Adam, Herzig, and Longin [2009]. Our approach is similar to Adam’s formalization in the sense that both use BDI-based logics (belief, desire, intention) to formalize the emotions of the OCC model and that both approaches are based on modal logic. Below we will briefly discuss major differences between the presented formalization of the OCC model and the one by Adam.

Just like us, Adam aims to be “as faithful as possible” to the OCC model. However, Adam’s formalization of OCC’s emotion types has been tailored to their BDI-based logical framework. In contrast, our formalization proceeds in three stages, where the first stage, as presented in this chapter, captures the logical structure of the OCC model. Only the last stage, detailed in Chapter 5, commits to BDI. Furthermore, Adam’s logical framework incorporates several very strong assumptions. For example, desires and ideals are assumed never to change and to be free of contradictions (thus excluding many forms of ‘mixed feelings’); agents are assumed to have complete introspection with respect to their desires; and all actions are assumed to be

\(^4\)That is, complete with respect to one psychological model of emotions, namely the OCC model.

\(^5\)In the following, we simply use “Adam” to refer to Adam et al. [2009].
deterministic, public, and accordant (i.e., no forgetting of effects). By refraining from making such assumptions, we believe our formalization is able to account for more situations in which emotions can arise (according to psychology).

Some of Adam’s definitions of emotions do not capture all aspects of what is supposed to be formalized. For example, Adam’s formalization of hope and fear does not account for future-directed prospects (only current uncertainty); ‘easy’ actions preclude pride and shame; and partial (dis)confirmations cannot trigger satisfaction, fears-confirmed, relief, or disappointment. Admittedly, the OCC model may be implicit or ambiguous with respect to these and other aspects, but ideally, the process of formalization should explicate such issues and offer clarifications.

There is some confusion in Adam’s formalization between emotion elicitation and experience. Adam claims to formalize the eliciting conditions of emotions (as do we), and the action-based emotions indeed appear to incorporate a trigger, namely in the form of the perception of an action. However, Adam’s formalizations of the event-based emotions do not incorporate any triggers. For example, joy is defined as $\text{Joy}_i\varphi \overset{\text{def}}{=} \text{Bel}_i\varphi \land \text{Des}_i\varphi$, but this expresses a ‘state of joy’ more than a trigger for joy. Indeed, in the text Adam often identifies the satisfaction of an emotion formula with feeling the emotion in question. When Adam defines the compound emotions simply as conjunctions (e.g., $\text{Gratification}_i(i;\alpha, \varphi) \overset{\text{def}}{=} \text{Pride}_i(i;\alpha, \varphi) \land \text{Joy}_i\varphi$), it is then unclear what $\text{Gratification}_i(i;\alpha, \varphi)$ actually represents because it mixes triggering ($\text{Pride}_i(i;\alpha, \varphi)$) and experience ($\text{Joy}_i\varphi$). In our approach, we have made a clear distinction between emotion elicitation and experience in order to avoid such confusion.

Finally, Adam’s formalization renders a number of properties of emotions that we find too strong. For example, Adam proves that $\vdash \neg (\text{Joy}_i\varphi \land \text{Distress}_i\varphi)$ and similarly for all pairs of opposing emotions applied to the same argument(s). Such formulas are not valid in our formalization because we allow goals, standards, and attitudes to be inconsistent. However, if their consistency would be adopted as a constraint, it would indeed be provable in our framework that opposing emotion triggers contradict.

### 3.4.3 A Computational Model of Emotions

Gratch and Marsella [2004] have been working on a computational framework for modeling emotions. The framework is claimed to be domain-independent and they have implemented a process model, called EMA after the title of [Lazarus, 1994], for social training applications. The appraisal process used in EMA is inspired by the OCC model. As with our approach, the cognitive reasoning aspects of EMA are represented using BDI concepts and the emphasis of appraisal is on goal attainment.

In contrast to our approach, Gratch and Marsella take a computational, quantitative approach towards modeling appraisal. Specifically, the eliciting conditions of emotions modeled in EMA are based on quantitative measures of, e.g., desirability and likelihood. The calculation of these quantitative measures is facilitated by the usage of subjective probabilities for beliefs and assignment of utilities to states. However, precise triggering conditions for all emotions are not provided, so it is hard to judge how strictly Gratch and Marsella follow psychological models of emotions.
and how they deviate from or extend these.

3.5 Concluding Remarks

In this chapter we have given semiformal specifications of the eliciting conditions of the emotions described in the psychological OCC model. So far we have ‘reduced’ these eliciting conditions to formulas involving some standard logical connectives and the following seventeen constructs.

<table>
<thead>
<tr>
<th>PerceiveConseq</th>
<th>Des</th>
<th>Prospective</th>
<th>Past</th>
</tr>
</thead>
<tbody>
<tr>
<td>PerceiveAction</td>
<td>Undes</td>
<td>Actual</td>
<td>Presume</td>
</tr>
<tr>
<td>PerceiveObject</td>
<td>Praisew</td>
<td>CogUnit</td>
<td></td>
</tr>
<tr>
<td>PerceiveRelated</td>
<td>Blamew</td>
<td>Confirms</td>
<td></td>
</tr>
<tr>
<td>Appeal</td>
<td>Disconfirms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unappeal</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If the specifications presented in this chapter are accurate, then the eliciting conditions of the emotion types of the OCC model are constructed around no more than seventeen\(^6\) notions, represented in our formalization by the seventeen constructs above. Eight of these constructs will be grounded in dynamic doxastic logic in the next chapter, whereas the remaining nine constructs will be grounded in KARO (which is a BDI-based extension of dynamic doxastic logic) in the chapter after that.

---

\(^6\)Not counting the propositional connectives.
In this chapter we introduce a formalism that grounds many of the constructs used in the previous chapter to (semiformally) specify the eliciting conditions of the emotion types of the OCC model. For this purpose, we have chosen to use dynamic doxastic logic, because this is a well-understood formalism which readily provides ways for reasoning about agents and their actions (because it is dynamic) and beliefs (because it is doxastic). Furthermore, this dynamic perspective allows for a straightforward representation of events and their consequences. Although dynamic doxastic logic is not concerned with objects, we will introduce a reasonable way of representing them as well. For a deep introduction in the field dynamic doxastic (or epistemic) logic, we refer the reader to the book by van Ditmarsch, van der Hoek, and Kooi [2007].

In particular, the following constructs used in the previous chapter will be defined in this chapter.

- `PerceiveConseq` \(\text{Prospective}\)
- `PerceiveAction` \(\text{Actual}\)
- `PerceiveObject` \(\text{Past}\)
- `PerceiveRelated` \(\text{Presume}\)

In line with our top-down approach, these constructs will be defined as abbreviations (i.e., using \(\text{def}\)), just like in the previous chapter. Formal semantics to ground all constructs used in this chapter will be presented in the next chapter. This approach makes it possible to choose another formalism and semantics than we did, as long as it supports the representation and interpretation of events and desirability, ac-
tions and praiseworthiness, and objects and appealingness, as well as some temporal constructs for the prospect-based emotions.

Some constructs used in Chapter 3, however, will be left undefined even here. These are the appraisal operators (i.e., Des, Undes, Praisew, Blamew, Appeal, and Unappeal), confirmation operators (Confirms, Disconfirms), and the cognitive unit operator (CogUnit). There are separate reasons for this, which will be explained in more detail in Section 4.6. We cannot define the appraisal constructs (desirability, praiseworthiness, appealingness) in pure dynamic doxastic logic in this chapter, because it lacks ways of representing goals, standards, and attitudes. In Chapter 5, then, we add BDI-based constructs and finish the grounding of the specification of eliciting conditions of emotions. Also, we will not be concerned with formal semantics until Chapter 5.

This chapter is organized as follows. First of all, the basic operators of dynamic doxastic logic will be introduced in Section 4.1. With the use of these operators, the aforementioned constructs are defined in Sections 4.2–4.4. Related work is discussed in Section 4.5 and a conclusion discussing what has been done and what is left for the next chapter is presented in Section 4.6. Finally, proofs of propositions presented throughout this chapter are summarized in Section 4.7.

4.1 Basic Operators

In this section we introduce the basic language and operators lying at the heart of our formalization. We typically use \( p \) to denote atomic propositions, which are assumed to be drawn from the set \( \text{atm} \) of atomic propositions. Furthermore, we use the propositional connectives \( \neg, \land, \lor, \rightarrow, \text{and} \leftrightarrow \) with their usual interpretation, as well as \( \bot \) for falsum and \( \top \) for verum. To ease notation, we will often omit parentheses, with the common rule that unary operators (including \( \neg \)) bind strongest, and that \( \land \) and \( \lor \) bind stronger than \( \rightarrow \) and \( \leftrightarrow \). We then typically use \( \varphi \) and \( \psi \) to denote arbitrary formulas.

In dynamic doxastic logic there are of course two modal operators, namely for belief and action. They are expressed and read as follows.

**Definition 4.1 (Modal operators of dynamic doxastic logic)**

There are two modal operators, namely for belief and action.

\[
\begin{align*}
\mathbb{B}_i \varphi & : \text{Agent } i \text{ believes } \varphi \text{ (to be true).} \\
[i: \alpha] \varphi & : \text{After the execution of action } \alpha \text{ by agent } i, \varphi \text{ holds.}
\end{align*}
\]

With respect to the action operator, it is commonplace to denote its dual using angled brackets; that is, \( \langle i: \alpha \rangle \varphi = \neg [i: \alpha] \neg \varphi \). So whereas \( [i: \alpha] \varphi \) expresses that \( \varphi \) is a necessary result of the execution of action \( \alpha \) by agent \( i \), \( (i: \alpha) \varphi \) expresses that \( \varphi \) is a possible result of the execution of action \( \alpha \) by agent \( i \). Likewise, the expression \( \neg \mathbb{B}_i \neg \varphi \) can be read as agent \( i \) holding \( \varphi \) as possible. Note that these operators can be arbitrarily nested. For example, the proposition \( \mathbb{B}_{bob} [\text{Bob:drink}] \text{can\_drive\_safely} \) states that Bob believes that after drinking he will be able to drive safely, whereas the
proposition \([Bob:d_r_i_n_k]_{Bob}can\_drive\_safely\) states that after Bob has been drinking, he will believe that he can drive safely. (Either one does not have to imply the other!)

In dynamic logic, actions are used as an abstraction of time. Time is thus not explicitly represented in dynamic logic; instead, it is assumed that each action takes (some undefined amount of) time. This means that temporal constructs can be interpreted over actions, because any succession of actions implicitly models a passage of time.

In the following, we will be using three basic temporal operators, namely for representing previous, past, and future states of affairs. They are expressed and read as follows.

**Definition 4.2 (Temporal operators)**

There are three basic temporal operators, namely for making expressions regarding previous, past, and future states of affairs.

- **Prev** \(\varphi\): In the previous state, \(\varphi\) was true.
- **Past** \(\varphi\): Some time in the past, \(\varphi\) was true.
- **Fut** \(\varphi\): Some time in the future, \(\varphi\) may be true.

**Prev** is used to refer to the state before the execution of the last action. With respect to the reading of **Prev** \(\varphi\), we say “the previous state,” because it is assumed there is a linear history (and a branching future). For the rest of this chapter we will just take the linearity of history for granted; when formal semantics are introduced in Chapter 5, however, we will be in a position to investigate this assumption more deeply and formalize it.

Because actions are used as an abstraction of time, actions effectively discretize the time. This means that **Past** \(\varphi\) can intuitively be seen as the (infinite) expression \(\varphi \lor \text{Prev} \varphi \lor \text{Prev Prev} \varphi \lor \ldots\). However, because infinite formulas are not allowed, the past operator cannot be defined as an abbreviation in terms of the previous operator.

The future operator can intuitively be seen as an existential quantification over agents and actions. That is, **Fut** \(\varphi\) holds iff there exists a number of agents and actions such that at least one possible execution of these actions by these agents results in a state where \(\varphi\) holds (cf. formula (4.8)). It should be noted that both the past (as captured by the **Past** operator) and the future (as captured by the **Fut** operator) are understood to include the present, as is usual for such temporal operators. However, in the following we will mostly be using the future operator in situations that exclude the present. For convenience, then, we define a strict version of the future operator as follows.

**Definition 4.3 (Strict future operator)**

\[
\text{Fut}^+ \varphi \overset{\text{def}}{=} \neg \varphi \land \text{Fut} \varphi
\]  

(4.1)

**Fut**\(^+\) \(\varphi\) is then read as “some time in the future, but not presently, \(\varphi\) may be true.”

Below are several propositions showing how the dynamic and temporal operators just introduced interact. Because we have not yet introduced formal semantics, they cannot be called theorems yet. Nevertheless, Section 4.7 offers formal proofs using
the semantics introduced in Chapter 5. The propositions are shown here to get a feel for the properties of the operators.

**Proposition 4.4 (Properties of the temporal operators)**
Given the semantics to be introduced in Chapter 5, the following propositions are valid:

\[
\begin{align*}
\varphi & \rightarrow \text{Past} \varphi \land \text{Fut} \varphi & (4.2) \\
\varphi & \rightarrow [i: \alpha] \text{Prev} \varphi & (4.3) \\
\varphi & \rightarrow \neg \text{Fut} \neg \text{Past} \varphi & (4.4) \\
\varphi & \rightarrow \neg \text{Past} \neg \text{Fut} \varphi & (4.5) \\
\text{Prev Past} \varphi & \leftrightarrow \text{Past Prev} \varphi & (4.6) \\
\text{Past} \varphi & \leftrightarrow \varphi \lor \text{Prev Past} \varphi & (4.7) \\
\text{Fut} \varphi & \leftrightarrow \varphi \lor \langle i_1: \alpha_1 \rangle \cdots \langle i_n: \alpha_n \rangle \varphi & (\exists i_1, \ldots, i_n, \exists \alpha_1, \ldots, \alpha_n) (4.8) \\
\text{Prev} \varphi & \leftrightarrow \neg \text{Prev} \neg \varphi \land \text{Prev} \top & (4.9)
\end{align*}
\]

The first proposition states that the past and the future both include the present. The second proposition states that what is true now will be a previous truth after the execution of an action. The third proposition states that what is true now will have been true in the past in all possible futures. Conversely, the fourth proposition states that what is true now must have been a future possibility in all of the past. The fifth proposition states that the previous and past operators can freely be swapped. This is because Prev looks one ‘step’ into the past, whereas Past looks zero or more ‘steps’ into the past. So both the Prev Past and Past Prev combination will result in looking one or more ‘steps’ into the past. The sixth proposition states that the past can be built inductively from the present and the previous past. The seventh proposition states that the past can be built inductively from the present and the previous past. The seventh proposition states that, as explained above, Fut can be seen as an existential quantification over agents and actions. Note that we slightly abuse notation here, because strictly speaking we do not have quantification in our object language. Finally, the eighth proposition states that Prev is its own dual, provided that there exists a previous state, which is expressed by Prev T.

The next three sections present definitions of the constructs listed in the introduction of this chapter in terms of the operators introduced above. We will follow Figure 2.1 from left to right; that is, first we will define the constructs used for the event-based emotion types in Section 4.2, then those for the action-based emotion types in Section 4.3, and then those for the object-based emotion types in Section 4.4.

### 4.2 Events and their Consequences

Let us start simple. In the previous chapter we saw that the fortunes-of-others emotion types depended on the notion of presuming. For example, ‘pity’ was specified as ‘distress’ about a consequence (of an event) presumed to be undesirable for someone else, leading to the definition \( \text{Pity}_T(c, j) \defeq \text{Distress}_T(c) \land \text{Presume, Undes}_j(c) \). Now that we have chosen to use doxastic logic, we can simply conflate presuming with believing, as follows.
Definition 4.5 (Presuming) Presuming is conflated with believing:
\[ \text{Presume}_i \varphi \overset{\text{def}}{=} B_i \varphi \] (4.10)

The largest branch of the OCC model is concerned with valenced reactions to events; however, events are said to always be appraised with respect to their consequences. Now let us consider the distinction between consequences and events in more detail. The usual view in dynamic logic is that the execution of an action is regarded as an event. This makes sense because in dynamic logic, time passes only through the execution of actions, i.e., through a succession of events. Here we will follow this view and only regard executions of actions as events. We then consider a consequence of an event to be anything that was not true directly before the event, but is true directly after the event. This idea can be illustrated as follows.

This figure illustrates that state \( w_2 \) is the result of event \( i : \alpha \); that is, the execution of action \( \alpha \) by agent \( i \). Now any formula \( \varphi \) that is true in state \( w_2 \) (i.e., \( w_2 \models \varphi \)) but was not true in the previous state called \( w_1 \) (i.e., \( w_1 \models \neg \varphi \)) is considered to be a consequence of the event \( i : \alpha \). It will be clear that an event can also have multiple or no consequences.

For convenience, then, we introduce the following construct to capture consequences of events.

Definition 4.6 (Consequences (of events))
A formula \( \varphi \) is a consequence iff \( \varphi \) holds now but not previously:
\[ \text{New} \varphi \overset{\text{def}}{=} \varphi \land \neg \text{Prev} \varphi \] (4.11)

\( \text{New} \varphi \) is read as “\( \varphi \) was not true in the previous state but \( \varphi \) is true in the current state.”

Note that \( \text{Prev} \psi \) expresses that \( \psi \) was true before the execution of the latest action, i.e., before the latest event. Therefore, if \( \text{New} \varphi \) holds for some formula \( \varphi \), then \( \varphi \) can be regarded as a consequence of an event. Indeed, we have that \( w_2 \models \text{New} \varphi \) in the illustration above.

Because the \( \text{New} \) construct plays such a central role in the definitions presented in the remainder of this chapter, we list several properties if the construct.

Proposition 4.7 (Properties of New)
The following propositions express properties of \( \text{New} \):
\[ \text{New} (\varphi \land \psi) \rightarrow \text{New} \varphi \lor \text{New} \psi \] (4.12)

\(^1\)One may wonder whether \( \text{New} \varphi \) could also have been defined as \( \varphi \land \text{Prev} \neg \varphi \). Indeed, the difference is subtle and depends on the semantics of the \( \text{Prev} \) operator. With the semantics that will be given in Chapter 5, if \( \text{New} \varphi = \varphi \land \text{Prev} \neg \varphi \), then \( \text{New} \varphi \) will evaluate to false for all \( \varphi \) if there exists no previous state. However, we believe \( \text{New} \varphi \) should evaluate to true if \( \varphi \) is true and there exists no previous state, which formula (4.11) ensures. See also formula (4.16).
\[
\text{New } \varphi \land \text{New } \psi \rightarrow \text{New } (\varphi \land \psi) \quad (4.13)
\]
\[
(\text{New } \varphi) \land \psi \rightarrow \text{New } (\varphi \land \psi) \quad (4.14)
\]
\[
\text{New } \varphi \leftrightarrow \text{New } \text{New } \varphi \quad (4.15)
\]
\[
\varphi \land \text{Initial } \rightarrow \text{New } \varphi \quad (4.16)
\]

where \textbf{Initial} \( \overset{\text{def}}{=} \neg \text{Prev } \top \), i.e., \textbf{Initial} expresses that there is no history.

The first proposition states that if the conjunction of \( \varphi \) and \( \psi \) is a consequence, then \( \varphi \) is a consequence or \( \psi \) is a consequence. The second proposition states that \textbf{New} can be moved outside a conjunction. The third proposition states that if \( \varphi \) is a consequence and \( \psi \) is currently true, then \( \varphi \) and \( \psi \) together are a consequence. Note that this proposition directly implies the second proposition. The fourth proposition states that any row of \textbf{New}'s can be reduced to just one \textbf{New}. The fifth proposition states that, if there exists no previous state, then everything that is true now is also ‘new’.

Even though \textbf{New} \( \varphi \) expresses that \( \varphi \) is a consequence of an event, it may very well be that no agent is aware of this consequence. There must be a change in an agent’s beliefs before we can say that it perceives a consequence of an event (or anything in general). Belief changes can easily be defined using the \textbf{New} construct, as follows.

**Definition 4.8 (Belief update)**

A belief update is defined as a ‘new’ belief.

\[
\text{BelUpd}_i(\varphi) \overset{\text{def}}{=} \text{New } B_i \varphi \quad (4.17)
\]

\text{BelUpd}_i(\varphi) \text{ is read as “the beliefs of agent } i \text{ have just been updated with } \varphi.” \text{ A situation where } \text{BelUpd}_i(\varphi) \text{ holds can be illustrated as follows.}

\[
\begin{align*}
\text{BelUpd}_i(\varphi) & \quad \varphi \quad \neg \varphi \\
B_i & \quad j: \alpha \quad j: \alpha \\
\end{align*}
\]

In the state before the event \( j: \alpha \), agent \( i \) does not believe \( \varphi \); that is, it envisages worlds where \( \neg \varphi \) holds. In the state after the event \( j: \alpha \), agent \( i \) believes \( \varphi \); that is, in all worlds it holds as possible, \( \varphi \) is true.\(^2\) In that state, then, \text{BelUpd}_i(\varphi) \text{ is true. It should be noted that } \text{BelUpd}_i(\varphi) \text{ says nothing about the event (e.g., the action } j: \alpha \text{ in the illustration above) that actually brought about the belief update; all it expresses is that } \text{something} \text{ happened and as a consequence, agent } i \text{ believes } \varphi \text{ to }

\(^2\)With the danger of getting ahead of ourselves, we use possible world semantics to illustrate these definitions. Indeed, the belief and action modalities will be grounded using possible world semantics in Chapter 5. The definitions given in the present section do not really depend on such semantics; this illustration, then, only serves to get a feeling for what the defined constructs express.
be true. Now because $\text{BelUpd}_i(\varphi)$ represents an update of the beliefs of agent $i$, $\varphi$ can also be considered as a *percept* of agent $i$. From this point of view, $\text{BelUpd}$ and similar constructs can be used to define the perception of consequences of events, as the following will show.

With respect to emotions concerning consequences of events, the OCC model distinguishes between the types 'hope' and 'fear' on the one hand, and 'joy' and 'distress' on the other hand, based on whether the consequence in question is prospective or actual, respectively. As noted in Section 2.3.2, the notion of "prospect" is intentionally ambiguous; it is used to describe both future consequences and uncertain consequences. In Section 3.2.2 we used $\text{Prospective}_i(\varphi)$ and $\text{Actual}_i(\varphi)$ to express that agent $i$ considers $\varphi$ to be a prospective or actual consequence of an event, respectively. In particular, the trigger for 'joy' was defined as $\text{Joy}_i^c(\varphi) \overset{\text{def}}{=} \text{Pleased}_i^c(c) \land \text{Actual}_i(c)$, whereas the trigger for 'hope' was defined as $\text{Hope}_i^c(\varphi) \overset{\text{def}}{=} \text{Pleased}_i^c(c) \land \text{Prospective}_i(c)$. Using definitions similar to $\text{BelUpd}$ above, we define $\text{Prospective}$ and $\text{Actual}$ as follows.

**Definition 4.9 (Perceiving of different kinds of consequences (of events))**

Perceiving of a prospective consequence is defined as either a 'future update' or an 'uncertainty update'. Perceiving of an actual consequence is defined as a 'belief update'.

$$\text{Prospective}_i(\varphi) \overset{\text{def}}{=} \text{FutUpd}_i(\varphi) \lor \text{UncUpd}_i(\varphi) \quad (4.18)$$

$$\text{Actual}_i(\varphi) \overset{\text{def}}{=} \text{BelUpd}_i(\varphi) \quad (4.19)$$

where

$$\text{FutUpd}_i(\varphi) \overset{\text{def}}{=} \text{New B}_{i}^{\text{Fut}} \varphi \quad (4.20)$$

$$\text{UncUpd}_i(\varphi) \overset{\text{def}}{=} \text{New} (\neg \text{B}_{i}\varphi \land \neg \text{B}_{i}^{\neg} \varphi) \quad (4.21)$$

The definition of $\text{Prospective}$ is thus split into two cases in order to capture future as well as uncertain prospects. The definition of $\text{Actual}$ is the same as $\text{BelUpd}$ (see also the illustration above). As explained above, $\text{BelUpd}_i(\varphi)$ can be seen as expressing the perception of $\varphi$ by agent $i$, which is exactly what $\text{Actual}_i(\varphi)$ is also supposed to express, so the two constructs can be conflated.

The definition of $\text{FutUpd}$ is also like $\text{BelUpd}$ but then with $\varphi$ replaced by $\neg \varphi \land \text{Fut} \varphi$; that is, agent $i$ comes to believe that $\varphi$ is not true but that there exists a future in which $\varphi$ will be true. The definition of $\text{UncUpd}$ ("uncertainty update") also resembles $\text{BelUpd}$. If, in the illustration for $\text{BelUpd}$ above, the left 'cloud' would contain either only $\varphi$'s or only $\neg \varphi$'s, and the right 'cloud' would contain a mixture of $\varphi$'s and $\neg \varphi$'s, then $\text{UncUpd}_i(\varphi)$ would be true in the bottom right state. $\text{UncUpd}_i(\varphi)$ thus expresses that agent $i$ has just become uncertain about whether or not $\varphi$ holds.

$\text{Prospective}_i(\varphi)$ and $\text{Actual}_i(\varphi)$ are now defined such that they cover both cases for perceiving consequences of events that are distinguished in the OCC model. This means that $\text{PerceiveConseq}_i(\varphi)$, which we used to express that agent $i$ perceives consequence $\varphi$ of an event, can be (trivially) defined as the disjunction of the two.
Definition 4.10 (Perceiving consequences (of events))

Perceiving a consequence of an event is defined as either perceiving a prospective consequence or perceiving an actual consequence:

\[
\text{PerceiveConseq}_i(\phi) \overset{\text{def}}{=} \text{Prospective}_i(\phi) \lor \text{Actual}_i(\phi)
\] (4.22)

In effect, this definition specifies that perceiving a consequence of an event means either perceiving a prospective consequence or an actual consequence. Indeed, these are exactly the two cases distinguished in the OCC model. If one wishes to distinguish more kinds of consequences of events, more disjuncts covering those cases could be added. For example, if there would be a third kind of percept, formalized using construct \(X\), then \(\text{PerceiveConseq}_i(\phi) = \text{Prospective}_i(\phi) \lor \text{Actual}_i(\phi) \lor X\). In order to maintain proper division of kinds of percepts, however, care must be taken that \(\text{Prospective}_i(\phi) \land X\) and \(\text{Actual}_i(\phi) \land X\) are contradictions, just as \(\text{Prospective}_i(\phi) \land \text{Actual}_i(\phi)\) is currently a contradiction (see formula (4.26) below). So ideally, all disjuncts that make up \(\text{PerceiveConseq}_i(\phi)\) should be mutually exclusive.

To finish this section, let us study several properties of the presented definitions and their effects on the emotion triggers as defined in Chapter 3.

Proposition 4.11 (Properties of constructs regarding consequences of events)

Let \(\vdash_{\text{DD}}\) (where \(\text{DD}\) stands for dynamic doxastic logic) be a classical propositional entailment relation with the formulas in Definitions 3.2–3.13 and 4.1–4.10 as axioms and the assumption that belief distributes over conjunction (i.e., \(B_i\phi \land B_i\psi \leftrightarrow B_i(\phi \land \psi)\)). Then the following propositions are derivable:

\[
\begin{align*}
\vdash_{\text{DD}} & \neg(\text{FutUpd}_i(\phi) \land \text{UncUpd}_i(\phi)) \quad (4.23) \\
\vdash_{\text{DD}} & \neg(\text{BelUpd}_i(\phi) \land \text{UncUpd}_i(\phi)) \quad (4.24) \\
\Gamma \vdash_{\text{DD}} & \neg(\text{BelUpd}_i(\phi) \land \text{FutUpd}_i(\phi)) \quad (4.25) \\
\Gamma \vdash_{\text{DD}} & \neg(\text{Actual}_i(\phi) \land \text{Prospective}_i(\phi)) \quad (4.26) \\
\Gamma \vdash_{\text{DD}} & \neg(\text{Hope}_i^T(\phi) \land \text{Joy}_i^T(\phi)) \quad (4.27) \\
\Gamma \vdash_{\text{DD}} & \neg(\text{Fear}_i^T(\phi) \land \text{Distress}_i^T(\phi)) \quad (4.28) \\
\vdash_{\text{DD}} & \text{Pleased}_i^T(c) \leftrightarrow (\text{Hope}_i^T(c) \lor \text{Joy}_i^T(c)) \quad (4.29) \\
\vdash_{\text{DD}} & \text{Displeased}_i^T(c) \leftrightarrow (\text{Fear}_i^T(c) \lor \text{Distress}_i^T(c)) \quad (4.30)
\end{align*}
\]

where \(\Gamma = \neg(B_i\phi \land B_i\neg\phi)\).

The first three propositions state that \(\text{FutUpd}\), \(\text{UncUpd}\), and \(\text{BelUpd}\) are mutually exclusive. Because \(\text{Prospective}\) is defined in terms of \(\text{FutUpd}\) and \(\text{UncUpd}\), this immediately results in the fourth proposition. The fourth proposition then immediately leads to the fifth and sixth propositions, because (4.26) was exactly the assumption needed for propositions (3.42) and (3.43) (see page 52). The seventh and eighth propositions are the same as propositions (3.38) and (3.39) on page 52, except without needing assumptions due to the way \(\text{PerceiveConseq}\) has been defined. The assumption \(\Gamma\) expresses that beliefs must be consistent. When doxastic logic is used...
for modeling artificial agents, this assumption is commonly adopted as a constraint on the beliefs of agents. Indeed, upon introducing formal semantics in the next chapter, we will also constrain beliefs such that \( \neg(B_i \varphi \land B_i \neg \varphi) \) becomes a theorem itself.

The following propositions show that the triggers for ‘joy’ and ‘hope’ on the one hand, and ‘distress’ and ‘fear’ on the other hand, can be related, provided that (un)desirability of a future consequence of an event implies current (un)desirability.

**Proposition 4.12 (Relations between triggers for ‘joy’ and ‘hope’, and ‘distress’ and ‘fear’)**

The following propositions are derivable:

\[
\Gamma_1 \vdash \text{DD} \text{Joy}_T(\text{Fut}^+ \varphi) \rightarrow \text{Hope}_T(\varphi) \quad (4.31) \\
\Gamma_2 \vdash \text{DD} \text{Distress}_T(\text{Fut}^+ \varphi) \rightarrow \text{Fear}_T(\varphi) \quad (4.32)
\]

where \( \Gamma_1 = \text{Des}(\text{Fut}^+ \varphi) \rightarrow \text{Des}(\varphi) \) and \( \Gamma_2 = \text{Undes}(\text{Fut}^+ \varphi) \rightarrow \text{Undes}(\varphi) \).

Thus, informally speaking, ‘joy’ about a future consequence implies ‘hope’ and ‘distress’ about a future consequence implies ‘fear’. However, this is only true if desirability and undesirability were closed under the future operator. It should be noted that the way in which desirability and undesirability will be constrained in Section 5.4 will make neither \( \Gamma_1 \) nor \( \Gamma_2 \) derivable (hence they have to be explicitly mentioned as additional assumptions).

### 4.3 Agents and their Actions

In order to formalize agents and actions, and also for convenience of quantification in the remainder of this dissertation, the existence of sets enumerating all agents and actions is assumed.

**Definition 4.13 (Agents and actions)**

Let \( \text{agt} \) be the set of agent names with typical elements \( i \) and \( j \). Let \( \text{act} \) be the set of actions instances with typical element \( \alpha \).

As mentioned in the previous chapter, it is assumed each action is unique, i.e., it can be performed only once. This can be seen as each action in \( \text{act} \) being a unique instance of an action. Unique instances of actions can easily be constructed from some set \( \mathcal{A} \) of basic actions, namely by associating each action in \( \mathcal{A} \) with a unique identifier. \( \text{act} \) could then be defined as follows:

\[
\text{act} = \bigcup_{a \in \mathcal{A}} \{a_{id} \mid id \in \mathbb{N}\}
\]

where \( \mathbb{N} \) is the set of natural numbers. Having said that, we will not go into this any further here and simply write \( a \in \text{act} \). The constraint that the same action instance cannot be performed more than once will be formalized in the next chapter (see Constraint 5.7).
As usual in dynamic logic, basic actions can be composed to form plans. For example, a plan to move a cup from one table to another may be specified as \( \text{Cup} = \text{reach} ; \text{grab} ; \text{move} ; \text{release} \), where the semicolon is used to form sequential compositions of actions. Formally, we define the set plans of plans by induction on the set act of actions.

**Definition 4.14 (Plans)**
The set plans contains all actions, sequential compositions of actions, and converse actions. That is, plans is the smallest set such that:

- \( \text{act} \subseteq \text{plans} \);
- If \( \pi_1 \in \text{plans} \) and \( \pi_2 \in \text{plans} \), then \( (\pi_1 ; \pi_2) \in \text{plans} \);
- If \( \pi \in \text{plans} \), then \( \pi \sim \in \text{plans} \).

\( \pi \) is used as a typical element of plans.

The notation of the dynamic operator is extended to plans, such that we can write \([i: \pi] \varphi\) for \( \pi \in \text{plans} \). Furthermore, we write the dual of the action operator as usual, namely \( \langle i: \pi \rangle \varphi = \neg[i: \pi] \neg \varphi \). It should be noted that the current syntax is not able to express subsequent actions of different agents inside the action modality. For example, \([ (i: \alpha) ; (j: \beta) ] \varphi\) is not valid syntax. However, this is not a problem because such statements can always be written using nesting; in this example, one can write \([i: \alpha][j: \beta] \varphi\), which is syntactically valid.

Converse actions (e.g., \( \alpha \sim \)) are useful for expressing what was true before the execution of an action. For example, \([i: \alpha \sim] \varphi\) expresses that, if it is the case that agent \( i \) has just performed action \( \alpha \), then \( \varphi \) was true before that action. Because we will often need to express that some agent has just performed some action, we define a convenient shorthand for this.

**Definition 4.15 (Done)**
An agent \( i \) has just done action \( \alpha \) iff there exists a state after the converse of \( \alpha \):

\[
\text{Done}(i: \alpha) \overset{\text{def}}{=} \langle i: \alpha \sim \rangle \top \tag{4.33}
\]

\( \text{Done}(i: \alpha) \) can thus be read as “agent \( i \) has just performed action \( \alpha \).”

In the Chapter 3, we used \( \text{PerceiveAction} (j: \alpha) \) to express that agent \( i \) perceives that agent \( j \) has performed action \( \alpha \). In particular, this construct was then used to define the eliciting conditions of ‘approving’ and ‘disapproving’; for example, \( \text{Approving}_T (j: \alpha) \overset{\text{def}}{=} \text{PerceiveAction} (j: \alpha) \land \text{Praisew}_i (j: \alpha) \). Making use of the \( \text{BelUpd} \) construct introduced above, we define \( \text{PerceiveAction} \) as follows.

**Definition 4.16 (Perceiving actions of agents)**
Agent \( i \) perceives action \( \alpha \) of agent \( j \) iff the beliefs of \( i \) are updated with the fact that \( j \) has done \( \alpha \) some time in the past:

\[
\text{PerceiveAction} (j: \alpha) \overset{\text{def}}{=} \text{BelUpd}_i (\text{Past Done}(j: \alpha)) \tag{4.34}
\]
Note the use of the Past operator here. Because Done\((j:\alpha)\) only expresses that agent \(j\) has just performed action \(\alpha\), the Past operator is needed to express perceptions of actions that have been performed at some arbitrary time in the past. Thus PerceiveAction\((j:\alpha)\) does not specify when exactly agent \(j\) performed action \(\alpha\), just that agent \(i\) now believes it did and that \(i\) did not believe so before.

For the compound emotion types (gratification, remorse, gratitude, anger) it was necessary to express a (presumed) relation between an action of an agent and a consequence, for which we used PerceiveRelated. For example, the eliciting conditions of ‘gratification’ were defined as

\[
\text{Gratification}^T(i:a,c) \triangleq \text{Past Pride}^T(i:a) \land \text{Past Joy}^T_i(c) \land \text{PerceiveRelated}_i(j:a,c).
\]

In order to define PerceiveRelated we make good use of the BelUpd construct again.

**Definition 4.17 (Perceiving relations)**

Agent \(i\) perceives a relation between action \(\alpha\) of agent \(j\) and consequence \(\varphi\) iff the beliefs of \(i\) are updated with the fact that action \(\alpha\) of agent \(j\) co-occurred with \(\varphi\) becoming true:

\[
\text{PerceiveRelated}_i(j:\alpha,\varphi) \overset{\text{def}}{=} \text{BelUpd}_i(\text{Related}(j:\alpha,\varphi))
\]

(4.35)

where

\[
\text{Related}(i:\alpha,\varphi) \overset{\text{def}}{=} \text{Past}(\text{Done}(i:\alpha) \land \text{New } \varphi)
\]

(4.36)

For convenience we define a construct for relatedness separately (as it will be useful later on). Related\((i:\alpha,\varphi)\) expresses that some time in the past, \(\varphi\) became true just when agent \(i\) had performed action \(\alpha\). We do not suggest this establishes a causal relationship between the action and the formula; indeed, the relation merely exists in their co-occurrence. Note that this definition correctly expresses a relation because of the assumption of uniqueness of actions. It should also be noted that the figure on page 63 illustrates this construct; in particular, \(w_2 \models \text{Related}(i:\alpha,\varphi)\). With these definitions, then, PerceiveRelated\((j:\alpha,\varphi)\) expresses that agent \(i\) perceives action \(\alpha\) of agent \(j\) to be related to consequence \(\varphi\) if and only if agent \(i\) comes to believe that \(\varphi\) became true exactly when action \(\alpha\) was performed by agent \(j\).

To finish this section, we present several propositions illustrating how the action-based operators interact. These propositions cannot be called theorems yet, because we have not introduced formal semantics yet. However, they are shown here to get a feeling for the workings of the action-based operators. Proofs of these propositions are provided in Section 4.7 using the semantics presented in Chapter 5.

The following propositions show how sequential compositions and converse actions behave.

**Proposition 4.18 (Properties of sequential compositions and converse actions)**

Given the semantics to be introduced in Chapter 5, the following propositions are valid:

\[
[i:(\alpha_1;\alpha_2)]\varphi \leftrightarrow [i:\alpha_1][i:\alpha_2]\varphi
\]

(4.37)

\[
[i:(\alpha_1;\alpha_2^\leftarrow)]\varphi \leftrightarrow [i:(\alpha_2^\leftarrow;\alpha_1^\leftarrow)]\varphi
\]

(4.38)
The first proposition states that sequential compositions of actions can be reduced by nesting of action modalities. The second proposition states that the converse of a sequence of actions is equivalent to a reversed sequence of converse actions. The third, fourth, fifth, and sixth proposition state that ‘rolling back’ an action leads to the original state of affairs. Finally, the seventh proposition states that there is always at most one previous state; that is, the history is linear. Note that unlike the other propositions, this last proposition is not valid in dynamic logic in general. However, with the semantics and constraints that will be introduced in Chapter 5, the seventh proposition will be provable. Note also the similarity between \( \langle i : \alpha^- \rangle \varphi \leftrightarrow \neg \langle i : \alpha^- \rangle \neg \varphi \land \langle i : \alpha^- \rangle \top \) (which is just another way of writing formula (4.43)) and formula (4.9); indeed, this is no coincidence, because both propositions depend on the (for now assumed) linearity of the history.

The following propositions show how the temporal and action-based operators interact.

**Proposition 4.19 (Properties of temporal and action-based operators)**

*Given the semantics to be introduced in Chapter 5, the following propositions are valid:*

\[
[ i : \alpha ] \text{Done}( i : \alpha ) \tag{4.44}
\]

\[
\text{Done}( i : \alpha ) \rightarrow \text{Prev} \top \tag{4.45}
\]

\[
\text{Done}( i : \alpha ) \land \varphi \rightarrow \text{Prev} \langle i : \alpha \rangle \varphi \tag{4.46}
\]

\[
\text{Prev} [ i : \alpha ] \varphi \land \text{Done}( i : \alpha ) \rightarrow \varphi \tag{4.47}
\]

The first proposition reads rather tautologically: after the execution of action \( \alpha \) by agent \( i \), \( i \) has done \( \alpha \). The second proposition states that, if an action has just been done, there must exist a previous state. The third proposition states that everything that is true now must previously have been a possible result of the last performed action. Finally, the fourth proposition states that all necessary results of the last performed action must be true now.

The following propositions result from our assumption of unique actions; that is, each instance of an action can only be performed once. We emphasize that this does *not* imply that, e.g., Alice can open her front door only once in her lifetime. What uniqueness of actions does mean is that each time Alice opens her front door, this will be seen as a unique instance of the action of opening the front door. Because the \( \alpha \) written in the action modality represents such a unique action instance, the following propositions will be valid (given the semantics and constraints to be introduced in Chapter 5).
Proposition 4.20 (Properties resulting from uniqueness of actions)
Given the semantics to be introduced in Chapter 5, the following propositions are valid:

\[
\begin{align*}
\text{Past Done}(i:a) & \rightarrow [i:a] \perp \\
[i:a] \neg \text{Fut} & \neg [i:a] \perp
\end{align*}
\] (4.48)

The first proposition states that, if some action has been executed sometime in the past, then it cannot be executable now. The second proposition states that, necessarily after the execution of an action, this action cannot be executed again in all possible futures.

4.4 Objects and their Aspects

In order to formalize objects, and also for convenience of quantification in the remainder of this dissertation, the existence of a set enumerating all objects is assumed.

Definition 4.21 (Objects)
Let \( \text{obj} \) be the set of names of objects that can be perceived by agents. \( x \) is used as a typical element of \( \text{obj} \).

In the OCC model the object-based emotions can also be directed towards agents. For example, Bob may like his new car, but he may also like Alice. To account for this, agents must be able to be viewed as objects. Formally, this can be captured by requiring that the set of objects contains the set of agents. We therefore adopt the following constraint.

Constraint 4.22 (Agents are objects) \( \text{agt} \subseteq \text{obj} \).

In order to be able to form propositions about objects, there must exist an atomic proposition for each object that identifies that object. This is modeled using the following constraint.

Constraint 4.23 (Objects are representable as propositions) \( \{ \text{object}_x \mid x \in \text{obj} \} \subseteq \text{atm} \).

The notation \( \text{object}_x \) is used to refer to the proposition identifying \( x \) as an object. For example, if \( x = \text{mona lisa} \in \text{obj} \), then \( \text{object}_x \) may be the proposition \( \text{mona lisa is an object} \in \text{atm} \). Using this notation, the construct \( \text{PerceiveObject} \) used to capture the perception of objects can easily be defined using the \( \text{BelUpd} \) construct. Recall that \( \text{PerceiveObject} \) was used in Chapter 3 in the formalization of the eliciting conditions of ‘liking’ and ‘disliking’; for example, the trigger for ‘liking’ was defined as \( \text{Liking}_i^T(x) \equiv \text{PerceiveObject}_i(x) \land \text{Appeal}_i(x) \). Now \( \text{PerceiveObject} \) is defined as follows.

Definition 4.24 (Perceiving objects)
An object \( x \) is perceived by agent \( i \) iff the proposition identifying \( x \) is added to the beliefs of \( i \):

\[
\text{PerceiveObject}_i(x) \equiv \text{BelUpd}_i(\text{object}_x)
\] (4.50)
So perceiving an object is equated with a reference to the object being added to the agent’s beliefs.

With this last definition, we now have the property that every emotion trigger subsumes a belief update. In other words, every formula of the form $\text{Emotion}^T_i(X)$ (e.g., $\text{Admiration}^T_i(j:a)$) entails a formula of the form $\text{BelUpd}_i(Y)$ (e.g., $\text{BelUpd}_i(\text{Past Done}(j:a))$. But, of course, the same fact cannot count as an ‘update’ in two immediately successive states. An interesting consequence of this is that an emotion trigger, as defined in Chapter 3, cannot be satisfied with respect to the same argument in two successive states. For example, if joy with respect to winning the lottery is triggered in state $s$, then in state $s$ the agent in question must believe it has won the lottery, which means that in the next state joy with respect to winning the same lottery cannot be triggered again, because this would require that the agent did not believe having won the lottery in the previous state $s$, which contradicts the initial assumption. This idea of no immediate re-triggering is captured by the following proposition.

**Proposition 4.25 (No immediate re-triggering of emotions)**

Let $\vdash_{\text{DD}}$ (where DD stands for dynamic doxastic logic) be a classical propositional entailment relation with the formulas in Definitions 3.2–3.13 and 4.1–4.24 as axioms and the assumption that $\text{Prev}$ distributes over conjunction (i.e., $\text{Prev} \varphi \land \text{Prev} \psi \leftrightarrow \text{Prev} (\varphi \land \psi)$). Then the following proposition is derivable:

$$
\vdash_{\text{DD}} \neg(\text{Emotion}^T_i(X) \land \text{Prev Emotion}^T_i(X)) \tag{4.51}
$$

for $\text{Emotion} \in \{\text{Joy, Distress, Satisfaction, Fears-confirmed, Relief, Disappointment, Happy-for, Resentment, Gloat, Pity, Approving, Disapproving, Pride, Shame, Admiration, Reproach, Gratification, Remorse, Gratitude, Anger, Liking, Disliking}\}$.

The proposition above is not valid for $\text{Emotion} \in \{\text{Positive, Negative, Pleased, Displeased, Hope, Fear}\}$, because the formal definitions of the eliciting conditions of these emotion types contain disjunctions of $\text{BelUpd}$ constructs. Consequently, these emotion types can be triggered in successive states, but only for different reasons. For example, in one state hope may be triggered because of a new future prospect (e.g., $\text{FutUpd}_i(\varphi)$), while in the next state hope may be triggered with respect to the same consequence but this time because of an uncertainty update (e.g., $\text{UncUpd}_i(\varphi)$). For the emotion types for which formula (4.51) is valid, re-triggering with respect to the same consequence, action, or object is only possible every other state, because a belief has to be retracted and then re-added. So for example, $\text{Joy}^T_i(\varphi) \land \text{Prev Prev Joy}^T_i(\varphi)$ is a satisfiable proposition. Indeed, this formula implies $\text{Prev} \neg B_i \varphi$; that is, in between the two states in which ‘joy’ is triggered, the belief in $\varphi$ must have been retracted.

It should be emphasized that Proposition 4.25 does not mean that, for example, an agent cannot be joyous about having won the lottery for two successive states. Of course, the experience of joy can last for many successive states, but that is an aspect of emotion different from triggering and will be the subject of Chapter 6.
4.5 Related Work

This section briefly describes three interesting differences of the current approach with related formalizations of emotion that were also discussed in Section 3.4.

4.5.1 Introspection

The introduction of an operator for expressing beliefs raises the question whether agents should have introspection with respect to their emotions; that is, if an agent has an emotion, should it believe it has the emotion? So far in our formalization, we have only dealt with the conditions that can trigger emotions, not with the actual experience of emotions. Therefore, studying introspection at this point does not make much sense because there is no representation of actual emotions yet.

The formalization by Adam et al. [2009] also makes use of doxastic logic; that is, they make use of an operator Bel for representing beliefs of agents. Adam defined the eliciting conditions of OCC’s emotions such that they are able to derive complete introspection of emotions; that is, they can prove that \( \vdash \text{Emotion}_i \varphi \leftrightarrow \text{Bel}_i \text{Emotion}_i \varphi \) and \( \vdash \neg \text{Emotion}_i \varphi \leftrightarrow \text{Bel}_i \neg \text{Emotion}_i \varphi \) hold in their framework for every type of “Emotion.” However, if \( \text{Emotion}_i \varphi \) is supposed to represent the satisfaction of the eliciting conditions of \( \text{Emotion} \), as Adam intends, then we find this counterintuitive; one does not have to be aware of what triggered an emotion. It is only intuitive to suppose that one is aware of what one does and does not feel; that is, if \( \text{Emotion}_i \varphi \) were to represent the subjective experience of \( \text{Emotion} \) with respect to \( \varphi \). Unfortunately, because of confusion between elicitation and experience in Adam’s formalization (as discussed in Section 3.4), it is difficult to judge the status of these introspection properties.

4.5.2 Consequences of Events

The difference between events and their consequences has not been properly taken into account in related formalizations of emotions (e.g., [Adam et al., 2009; Gratch and Marsella, 2004]) and not even in some of our previous work (e.g., [Steunebrink et al., 2007, 2008a]). In the work of Gratch and Marsella [2004] the distinction is not discussed at all, whereas in the work of Adam et al. [2009] it is recognized that events are always appraised with respect to their consequences, but in the formalization itself this distinction is lost again. For example, Adam defines the eliciting conditions of ‘joy’ as \( \text{Joy}_i \varphi \overset{\text{def}}{=} \text{Bel}_i \varphi \land \text{Des}_i \varphi \); that is, ‘joy’ with respect to \( \varphi \) is triggered when the agent both believes and desires \( \varphi \) to be true. Although \( \varphi \) is used here as an argument of desirability, in the text \( \varphi \) is described as representing an event, whereas OCC stipulate that desirability is only applicable to consequences [Ortony and Clore, 2009]. In contrast, in our approach every emotion trigger subsumes a belief update; for example, we have formalized the eliciting conditions of ‘joy’ as \( \text{Joy}_i^T(\varphi) = \text{B}_i \varphi \land \neg \text{Prev B}_i \varphi \land \text{Des}_i(\varphi) \). Here \( \varphi \) represents a consequence of a change, and if that consequence is desirable, then ‘joy’ is triggered. This way we have made

---

3In the following, we simply use “Adam” to refer to Adam et al. [2009].
a clear distinction between events (which are taken to be state transitions) and their consequences (which are taken to be the changes in truth values, in particular in agents’ beliefs, resulting from state transitions).

4.5.3 Prospects

Although the OCC model uses the notion of prospect in two different ways (i.e., uncertainty about the current state and future possibility), no formalization of the OCC model that we are aware of recognizes this.

For example, in the formalization of the OCC model by Gratch and Marsella [2004] likelihood of a desirable event is given as a precondition for hope, in line with the OCC model. However, likelihood of an event is equated with the believed probability of the event, such that likelihood can also be used as a precondition for joy (in particular, if the likelihood of the event equals one). This results in the following specification of the conditions that trigger ‘hope’ and ‘joy’ with respect to some proposition $p$.

$$\text{Hope}(p) \text{ if } \text{Des}(\text{self}, p) > 0 \text{ and } \text{Likelihood}(\text{self}, p) < 1.0$$

$$\text{Joy}(p) \text{ if } \text{Des}(\text{self}, p) > 0 \text{ and } \text{Likelihood}(\text{self}, p) = 1.0$$

Now the question is, what does Likelihood($\text{self}, p$) measure: the probability of $p$ happening some time in the future, or the probability of $p$ being true now? By also using Likelihood($\text{self}, p$) in the specification for ‘joy’, the latter is suggested, but no clear answer is given in the text.

The formalization of the OCC model by Adam et al. [2009] is less ambiguous at this point. There the eliciting conditions of ‘hope’ are defined as follows:

$$\text{Hope}_i(\pi, \varphi) \overset{\text{def}}{=} \neg \text{Bel}_i \varphi \land \text{Prob}_i \varphi \land \text{Des}_i \varphi$$

Adam’s handling of the notion of prospects thus depends on the interpretation of the Prob operator. Specifically, Prob$_i \varphi$ is used to express that agent $i$ considers $\varphi$ to be probably true in the current state, where “probably” is used to mean “more than 50% certain.” So Adam’s formalization does not take into account prospects regarding future possibilities.

In contrast, in previous work [Steunebrink et al., 2007] we provided the following formalization of the eliciting conditions of ‘hope’:

$$\text{Hope}_i(\pi, \varphi) \overset{\text{def}}{=} \text{Bel}_i (\langle i; \pi \rangle \varphi \land \text{Goal}_i \varphi \land \text{Ab}_i \pi \land \text{Com}_i \pi)$$

where $\text{Ab}$ stands for “ability-to-perform” and Com stands for “committed-to”.$^4$ So it was specified that agent $i$ hopes to achieve $\varphi$ with plan $\pi$ iff agent $i$ believes $\varphi$ is a goal ($\text{Goal}_i \varphi$) and the execution of plan $\pi$ can lead to the achievement of $\varphi$ ($\langle i; \pi \rangle \varphi$) and it has the ability to perform $\pi$ ($\text{Ab}_i \pi$) and it is committed to performing $\pi$ ($\text{Com}_i \pi$). Here

$^4$The notation used in [Steunebrink et al., 2007] was a bit different from that used in the formula displayed here. We have change its notation for clarity of presentation and in order to avoid confusion with the current formalization.
the notion of prospect is captured by the term \( (i:\pi)\varphi \); that is, there exists a possibility of goal \( \varphi \) being achieved by performing plan \( \pi \). Obviously, then, this formalization only takes into account prospects regarding the future.

The formalization presented in this chapter takes into account both uncertainty-directed and future-directed prospect by defining a prospect as a disjunction of either uncertainty about the current state of affairs or believed future possibility (see Definition 4.9).

4.6 Concluding Remarks

We have so far ‘reduced’ the eliciting conditions of the emotion types of the OCC model to formulas involving propositional connectives and operators from dynamic doxastic logic. In particular, we have used the \( B \) operator to represent beliefs of agents, and the \( \text{Prev}, \text{Past}, \) and \( \text{Fut} \) operators for representing states of affairs in past and future states. The following nine constructs used in Chapter 3 are still undefined:

<table>
<thead>
<tr>
<th>Des</th>
<th>Praisew</th>
<th>Appeal</th>
<th>Confirms</th>
<th>CogUnit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undes</td>
<td>Blamew</td>
<td>Unappeal</td>
<td>Disconfirms</td>
<td></td>
</tr>
</tbody>
</table>

These constructs will be grounded in the KARO framework (which extends dynamic doxastic logic) in the next chapter. There are several reasons why they have not been defined in this chapter. First, the six appraisal constructs (\( \text{Des}, \text{Undes}, \text{Praisew}, \text{Blamew}, \text{Appeal}, \text{Unappeal} \)) require the notions of goals, standards, and attitudes, which are absent in pure dynamic doxastic logic. Second, for the confirmation constructs (\( \text{Confirms}, \text{Disconfirms} \)), two consequences of events must be compared in order to determine whether one confirms or disconfirms the other. But in order to be able to perform such comparisons, we want to be more precise about what is being compared first. Although it may be possible to define what it means for one arbitrary kind of consequence to confirm or disconfirm another arbitrary kind of consequence just in propositional logic, we can be more precise if we know what kinds of consequences will ever be compared at all. From formulas (3.24)–(3.27) it can be observed that the things being compared for (dis)confirmation are arguments of event-based emotion types, which means that they must be related to goals. Therefore, \( \text{Confirms} \) and \( \text{Disconfirms} \) can be given more specific definitions when we know what goals look like, which means postponing defining \( \text{Confirms} \) and \( \text{Disconfirms} \) until we have formalized goals. Third, a proper formalization of the notion of cognitive unit (as expressed by \( \text{CogUnit} \)) would require substantially more (psychological) research in order to formulate a precise specification of this notion. The next chapter provides a simple grounding of the \( \text{CogUnit} \) construct, leaving this aspect as open as possible for future extensions.

4.7 Proofs

The following proofs of propositions presented in this chapter make use of the formal semantics introduced in Chapter 5, in particular Definition 5.8.
Proposition (4.2). Take an arbitrary model–state pair \((M, s)\) and formula \(\varphi\) and assume \(M, s \models \varphi\). Then by definition of reflexive transitive closure, \(((M, s), (M, s)) \in (\bigcup R)^+\). According to Definition 5.8, the conditions for both \(M, s \models \text{Past} \varphi\) and \(M, s \models \text{Fut} \varphi\) are now satisfied. Because \((M, s)\) and \(\varphi\) were arbitrary, we have that \(\varphi \rightarrow \text{Past} \varphi \land \text{Fut} \varphi\) is valid. 

Proposition (4.3). Take an arbitrary model–state pair \((M, s)\) and formula \(\varphi\) and assume \(M, s \models \varphi\). Take an arbitrary agent \(i\), action \(\alpha\), and model–state pair \((M', s')\) such that \(((M, s), (M', s')) \in \mathcal{R}_{\text{act}}\). This implies that \(((M, s), (M', s')) \in (\bigcup R)\). By Definition 5.8 we now have that \(M', s' \models \text{Prev} \varphi\). Because \(i: \alpha\) was arbitrary, we have that \(M, s \models [i: \alpha] \text{Prev} \varphi\). Because \((M, s)\) and \(\varphi\) were arbitrary, we have that \(\varphi \rightarrow [i: \alpha] \text{Prev} \varphi\) is valid. 

Proposition (4.4). Take an arbitrary model–state pair \((M, s)\) and formula \(\varphi\) and assume \(M, s \models \varphi\). Take an arbitrary model–state pair \((M', s')\) such that \(((M, s), (M', s')) \in (\bigcup R)^+\). By Definition 5.8 we now have that \(M', s' \models \neg \text{Fut} \neg \text{Past} \varphi\). Because \((M', s')\) was arbitrary, we have that \(M, s \models \neg \text{Fut} \neg \text{Past} \varphi\). Because \((M, s)\) and \(\varphi\) were arbitrary, we have that \(\varphi \rightarrow \neg \text{Fut} \neg \text{Past} \varphi\) is valid. 

Proposition (4.5). The proof of this proposition is analogous to that of formula (4.4). 

Proposition (4.6). The fact that writing \(\text{Prev} \text{Past} \varphi\) is equivalent to \(\text{Past} \text{Prev} \varphi\) can be seen by realizing that both \(M, s \models \text{Prev} \text{Past} \varphi\) and \(M, s \models \text{Past} \text{Prev} \varphi\) are interpreted as \(\exists (M', s') \in S : M', s' \models \varphi\) and \(((M', s'), (M, s)) \in (\bigcup R)^+\); that is, they are interpreted over the transitive closure of all action-accessibility relations. 

Proposition (4.7). First we recall that from the definition of reflexive transitive closure it can be derived that \(R^* = \text{id}_U \cup R^+\) for every relation \(R\) on \(U\). This equation can be mapped onto the interpretation of the proposition \(\text{Past} \varphi \leftrightarrow \varphi \lor \text{Prev} \text{Past} \varphi\) symbol for symbol by realizing that \((\bigcup R)\) is a relation on \(S\). \(\text{Past} \varphi\) is interpreted using \((\bigcup R)^*\), \(\varphi\) can be seen to be interpreted using \(\text{id}_S\), and \(\text{Prev} \text{Past} \varphi\) is interpreted using \((\bigcup R)^+\) (see the proof of (4.6)). The symbols \(\equiv\) and \(\lor\) then correspond to \(\leftrightarrow\) and \(\lor\), respectively. 

Proposition (4.8). The expression \(M, s \models \text{Fut} \varphi\) is interpreted as \(\exists (M', s') \in S : ((M, s), (M', s')) \in (\bigcup R)^+\) and \(M', s' \models \varphi\). This is the same as \(M, s \models \varphi\) or \(\exists n \in \mathbb{N} : \exists i_0, \ldots, i_n \in \text{agt} : \exists a_0, \ldots, a_n \in \text{act} : \exists ((M, s), (M', s')) \in (\bigcup R)_{\text{act}_0} \circ \cdots \circ (\bigcup R)_{\text{act}_n} : M', s' \models \varphi\), i.e., \(M, s \models \varphi \lor \langle i_1: a_1 \rangle \cdots \langle i_n: a_n \rangle \varphi\). So \(\text{Fut} \varphi \leftrightarrow \exists i_0, \ldots, i_n \exists a_0, \ldots, a_n ((i_1: a_1) \cdots (i_n: a_n) \varphi)\) is valid (although strictly speaking we do not have quantification in our object language). 

Proposition (4.9). Assume \(M, s \models \text{Prev} \top\) for arbitrary \((M, s)\). This implies that \(\exists (M', s') : ((M', s'), (M, s)) \in (\bigcup R)\). But by Constraint 5.6 \(\cup R\) is injective, so \(\exists (M', s') : ((M', s'), (M, s)) \in (\bigcup R)\) and \(M', s' \models \varphi\) is true if and only if \(\forall (M', s') : ((M', s'), (M, s)) \in (\bigcup R)\) implies \(M', s' \models \varphi\) is true, i.e., \(M, s \models \text{Prev} \varphi \leftrightarrow \neg \text{Prev} \neg \varphi\). Because \((M, s)\) was arbitrary, \(\text{Prev} \top \rightarrow (\text{Prev} \varphi \leftrightarrow \neg \text{Prev} \neg \varphi)\) is valid. And because
Proofs

Proposition (4.12). If New \((\varphi \land \psi)\) then currently \(\varphi \land \psi\) and previously \(\neg(\varphi \land \psi)\), i.e., \(\neg \varphi \lor \neg \psi\). So either \(\varphi\) is ‘new’ or \(\psi\) is ‘new’. Therefore, New \((\varphi \land \psi)\) implies New \(\varphi \lor\) New \(\psi\).

Proposition (4.13). New \(\varphi \land\) New \(\psi\) equals \(\varphi \land \psi \land \neg \text{Prev} \varphi \land \neg \text{Prev} \psi\) equals \(\varphi \land \psi \land \neg \text{Prev}\neg (\neg \varphi \land \neg \psi)\) equals \(\varphi \land \psi \land \neg \text{Prev} \neg (\varphi \land \psi)\) equals New \((\varphi \land \psi)\). So New \(\varphi \land\) New \(\psi\) \(\rightarrow\) New \((\varphi \land \psi)\) is valid.

Proposition (4.14). \((\text{New} \varphi) \land \psi\) equals \(\varphi \land \psi \land \neg \text{Prev} \varphi\) equals \(\varphi \land \psi \land \neg \text{Prev} \neg (\neg \varphi)\) equals \((\varphi \land \psi)\) \(\land \neg \text{Prev} (\varphi \land \psi)\) equals New \((\varphi \land \psi)\). So \((\text{New} \varphi) \land \psi\) \(\rightarrow\) New \((\varphi \land \psi)\) is valid.

Proposition (4.15). New New \(\varphi\) equals \((\varphi \land \neg \text{Prev} \varphi) \land \neg \text{Prev} (\varphi \land \neg \text{Prev} \varphi)\). The first conjunct equals New \(\varphi\), so it remains to be shown that New \(\varphi\) implies \(\neg \text{Prev} (\varphi \land \neg \text{Prev} \varphi)\). If \(\neg \text{Prev} \varphi\) is true, then either previously \(\neg \varphi\) was true, or there exists no previous state. In the latter case, obviously \(\neg \text{Prev} (\varphi \land \neg \text{Prev} \varphi)\) is true and we are done. In the former case, also \(\neg \varphi \lor \text{Prev} \varphi\) was previously true, i.e., \(\text{Prev} \neg (\varphi \land \neg \text{Prev} \varphi)\) is true. But then by formula (4.9), \(\neg \text{Prev} (\varphi \land \neg \text{Prev} \varphi)\) is true, which was what remained to be shown. In conclusion, New \(\varphi \leftrightarrow\) New New \(\varphi\) is valid.

Proposition (4.16). If \(\neg \text{Prev} \top\) holds, then for any \(\varphi\), \(\neg \text{Prev} \varphi\) holds. New \(\varphi\) is defined as \(\varphi \land \neg \text{Prev} \varphi\), so \(\varphi \land \neg \text{Prev} \top \rightarrow\) New \(\varphi\) is valid.

Proposition (4.23). \(\text{FutUpd}_i(\varphi)\) implies \(B_i \neg \varphi\), whereas \(\text{UncUpd}_i(\varphi)\) implies \(\neg B_i \neg \varphi\). This contradiction makes \(\neg (\text{FutUpd}_i(\varphi) \land \text{UncUpd}_i(\varphi))\) valid.

Proposition (4.24). \(\text{BelUpd}_i(\varphi)\) implies \(B_i \varphi\), whereas \(\text{UncUpd}_i(\varphi)\) implies \(\neg B_i \varphi\). This contradiction makes \(\neg (\text{BelUpd}_i(\varphi) \land \text{UncUpd}_i(\varphi))\) valid.

Proposition (4.25). \(\text{BelUpd}_i(\varphi)\) implies \(B_i \varphi\), whereas \(\text{FutUpd}_i(\varphi)\) implies \(B_i \neg \varphi\), which gives rise to a contradiction given the assumption that \(\neg (B_i \varphi \land B_i \neg \varphi)\). So \(\neg (\text{BelUpd}_i(\varphi) \land \text{UncUpd}_i(\varphi))\) valid assuming \(\neg (B_i \varphi \land B_i \neg \varphi)\).

Proposition (4.26). This proposition follows directly from combining Definitions 4.9 with propositions (4.24) and (4.25).

Propositions (4.27) and (4.28). These propositions follow directly from combining Definitions 3.8 and 3.9 with proposition (4.26).

Propositions (4.29) and (4.30). These propositions follow directly from combining Definitions 3.5, 3.8, 3.9, and 4.10.

Proposition (4.31). \(\text{Joy}^T_i(\text{Fut}^+ \varphi)\) equals New \(B_i \text{Fut}^+ \varphi \land \text{Des}_i(\text{Fut}^+ \varphi)\), which, using the assumption that \(\text{Des}_i(\text{Fut}^+ \varphi) \rightarrow \text{Des}_i(\varphi)\), implies \(\text{FutUpd}_i(\varphi) \land \text{Des}_i(\varphi)\), which
implies $\text{Hope}_t^T(\varphi)$. So $\text{Joy}_t^T(\text{Fut}^+\varphi) \rightarrow \text{Hope}_t^T(\varphi)$ is valid assuming $\text{Des}_t(\text{Fut}^+\varphi) \rightarrow \text{Des}_t(\varphi)$. □

**Proposition (4.32).** The proof of this proposition is analogous to that of formula (4.31). □

**Propositions (4.37), (4.38), (4.39), (4.40), (4.41), and (4.42).** Proofs of these propositions can be found in any comprehensive textbook on dynamic logic, e.g., [Harel et al., 2000]. □

**Proposition (4.43).** Take an arbitrary model–state pair $(M, s)$, agent $i$, action $\alpha$, and formula $\varphi$ and assume $M,s \models \langle i: \alpha^\neg \rangle \varphi$. According to Definition 5.8, this means that there exists a model–state pair $(M', s')$ such that $((M', s'), (M, s)) \in R_{\text{ext}}$ and $M', s' \models \varphi$. But by Constraint 5.6, $\bigcup R$ is injective. Because $R_{\text{ext}} \subseteq \bigcup R$, $R_{\text{ext}}$ must be injective as well. But this means that the aforementioned existential quantification can be made universal, i.e., for all model–state pairs $(M', s')$ such that $((M', s'), (M, s)) \in R_{\text{ext}}$, $M', s' \models \varphi$. This is the same as $M, s \models [i: \alpha^\neg] \varphi$. Because $(M, s)$ was arbitrary, $\langle i: \alpha^\neg \rangle \varphi \rightarrow [i: \alpha^\neg] \varphi$ is valid. □

**Proposition (4.44).** Take an arbitrary agent $i$ and action $\alpha$. Now for all $((M, s), (M', s')) \in R_{\text{ext}}$ we obviously have that $((M', s'), (M, s)) \in R_{\text{ext}}$ and $M, s \models T$, i.e., $M', s' \models \langle i: \alpha^\neg \rangle T$, which is the same as $M', s' \models \text{Done}(i: \alpha)$. Because $(M', s')$ was an arbitrary $R_{\text{ext}}$-successor of $(M, s)$, we have that $M, s \models [i: \alpha] \text{Done}(i: \alpha)$. Because $(M, s)$ was arbitrary, $[i: \alpha] \text{Done}(i: \alpha)$ is valid. □

**Proposition (4.45).** Assume $M, s \models \text{Done}(i: \alpha)$ for arbitrary $(M, s)$, $i$, $\alpha$. Then there exists a model–state pair $(M', s')$ such that $((M', s'), (M, s)) \in R_{\text{ext}}$. Because $R_{\text{ext}} \subseteq \bigcup R$, $((M', s'), (M, s)) \in (\bigcup R)$. Because $M', s' \models T$, we have that $M, s \models \text{Prev} T$. Because $(M, s)$ was arbitrary, $\text{Done}(i: \alpha) \rightarrow \text{Prev} T$ is valid. □

**Proposition (4.46).** Assume $M, s \models \text{Done}(i: \alpha)$ and $M, s \models \varphi$ for arbitrary $(M, s)$, $i$, $\alpha$, $\varphi$. Let $(M', s')$ be the model–state pair such that $((M', s'), (M, s)) \in R_{\text{ext}}$ (there can only be one such $(M', s')$ because $\bigcup R$ is injective). Now $M', s' \models \langle i: \alpha \rangle \varphi$ and therefore $M, s \models \text{Prev} \langle i: \alpha \rangle \varphi$. Because $(M, s)$ was arbitrary, $\text{Done}(i: \alpha) \rightarrow \text{Prev} \langle i: \alpha \rangle \varphi$ is valid. □

**Proposition (4.47).** Assume $M, s \models \text{Prev} \langle i: \alpha \rangle \varphi$ and $M, s \models \text{Done}(i: \alpha)$ for arbitrary $(M, s)$, $i$, $\alpha$, $\varphi$. Let $(M', s')$ be the model–state pair such that $((M', s'), (M, s)) \in R_{\text{ext}}$ (there can only be one such $(M', s')$ because $\bigcup R$ is injective). Now $M', s' \models \langle i: \alpha \rangle \varphi$ and therefore $M, s \models \varphi$. Because $(M, s)$ was arbitrary, we have that $\text{Prev} \langle i: \alpha \rangle \varphi \land \text{Done}(i: \alpha) \rightarrow \varphi$ is valid. □

**Proposition (4.48).** Assume $M', s' \models \text{Past Done}(i: \alpha)$ for arbitrary $(M', s')$, $i$, $\alpha$, $\varphi$. This implies that there exist two model–state pairs $(M, s), (M', s') \in S$ such that $((M, s), (M', s')) \in R_{\text{ext}}$ and $((M', s'), (M', s''')) \in (\bigcup R)^*$. Then by Constraint 5.7, $R_{\text{ext}} \cap (S' \times S') = \emptyset$ where $S' = \{(M'', s''') \mid ((M', s'), (M'', s''')) \in (\bigcup R)^* \}$. But in particular, $(M', s'') \in S'$. This implies that there exists no $(M'', s''')$ such that $((M'', s'''), (M'', s''')) \in R_{\text{ext}}$ (otherwise $((M'', s'''), (M'', s''')) \in (S' \times S')$, vio-
lating Constraint 5.7); that is, $M'', s'' \models [i: \alpha] \bot$. Because $(M'', s'')$ was arbitrary, 
\[ \text{Past Done}(i: \alpha) \rightarrow [i: \alpha] \bot \] is valid. □

**Proposition (4.49).** Take an arbitrary tuple $((M, s), (M', s')) \in R_{ca}$. Now let
$S' = \{(M'', s'') \mid ((M', s'), (M'', s'')) \in (\cup R')^* \}$ be the set of all possible model–
state pairs after and including $(M', s')$. Then by Constraint 5.7 we have that for
all $((M'', s''), (M''', s''')) \in (S' \times S')$, it must be that $((M', s'), (M'', s'')) \in (\cup R')^*$ but
$((M'', s''), (M''', s''')) \not\in R_{ca}$. This means that $M'', s'' \models [i: \alpha] \bot$ and that $M', s' \models \neg \text{Fut} \neg [i: \alpha] \bot$. And because $((M, s), (M', s'))$ was arbitrary, we have that
$[i: \alpha] \neg \text{Fut} \neg [i: \alpha] \bot$ is valid. □

**Proposition (4.51).** It is easy to check that all definitions of emotion triggers are
of the form $\text{BelUpd}_i(Y) \land Z$, except for Positive, Negative, Pleased, Displeased, Hope, and Fear. For all other emotion types, $\text{Emotion}_i^T(X)$ implies $\text{BelUpd}_i(Y)$
for some $Y$, which implies $\neg \text{Prev } Y$. (Note that $\text{Prev}$ distributes over conjunction
given the semantics presented in Chapter 5.) In contrast, $\text{Prev Emotion}_i^T(X)$ implies
$\text{Prev BelUpd}_i(Y)$, which implies $\text{Prev } Y$. Because of this contradiction, the proposition
$\neg (\text{Emotion}_i^T(X) \land \text{Prev Emotion}_i^T(X))$ is valid. □
Emotion Elicitation III: Grounding in BDI

In this chapter we introduce a framework that grounds all definitions presented in Chapters 3 and 4, thereby finishing our formalization of the eliciting conditions of the emotions of the OCC model. The framework used for this is KARO [Meyer, 2006; Meyer et al., 1999].

The KARO framework is a mixture of dynamic logic, epistemic/doxastic logic, and several additional (modal) operators for dealing with the motivational aspects of artificial agents. KARO was originally proposed as a specification logic for rational agents. It was thus designed to serve a purpose similar to that of the logics of Cohen and Levesque [1990] and Rao and Georgeff [1998]. A crucial difference with these approaches is that KARO is primarily based on dynamic logic [Harel et al., 2000] rather than temporal logic [Emerson, 1990]. So one could view KARO as a kind of BDI (belief, desire, intention) logic based on dynamic logic. Although the specification of informational and motivational attitudes (such as knowledge and beliefs, and desires, goals, and commitments, respectively) had been the main aim for devising KARO ([Hoek et al., 1999; Meyer et al., 1999]), the logic has also proven to be applicable for the description of agent behavior, more in general. For example, in [Meyer, 2006] it has been used to specify four basic emotion types. In this chapter, we present a modest extension\(^1\) of KARO, such that the eliciting conditions of all emotion types of the OCC model can be specified in this framework.

This chapter is organized as follows. Section 5.1 introduces the motivational operators and constructs of KARO that are used in our formalization of emotions. Section 5.2 then presents the formal semantics of KARO. The interpretation of the

\(^1\)The extension consists of the addition of operators for expressing past states of affairs; that is, the \texttt{Prev} and \texttt{Past} operators from Chapter 4.
operators used in Chapters 3, 4, and 5 are then presented in Section 5.3. In Section 5.4 we show how the appraisal operators (desirable, praiseworthy, appealing) can be interpreted in terms of the motivational construct in KARO. Section 5.5 provides a simple formalization of the notion of cognitive unit. Then Section 5.6 present definitions of the confirmation and disconfirmation construct, thereby finishing our formalization of the eliciting conditions of emotions. Properties of the formalization are then discussed in Section 5.7. We return back to the bigger picture in Section 5.8 to discuss how everything from Chapters 3, 4, and 5 fits together. Finally, Section 5.9 concludes this chapter.

**Notational Preliminaries**

Before moving on to KARO, we first introduce some notation that will be used in the rest of this chapter. Specifically, we will be using some standard concepts regarding, and operations on, relations.

A relation is a set of tuples and can be visualized as the edges of a directed graph. A relation $R$ on $U$ is reflexive iff $R$ includes the identity relation on $U$, i.e., $\text{id}_U \subseteq R$ where $\text{id}_U = \{ (x, x) | x \in U \}$. A relation $R$ on $U$ is serial iff every node in $U$ has an $R$-successor, i.e., $\forall x \in U : \exists y \in U : (x, y) \in R$. A relation $R$ on $U$ is transitive iff every node that can be reached in two ‘steps’ can also be reached in a single ‘step’, i.e., $\forall x, y, z \in U : (x, y) \in R$ and $(y, z) \in R$ implies $(x, z) \in R$. A relation $R$ on $U$ is euclidean iff every two $R$-successors of a single node are themselves connected, i.e., $\forall x, y, z \in U : (x, y) \in R$ and $(x, z) \in R$ implies $(y, z) \in R$. A relation $R$ on $U$ is symmetric iff every edge in $R$ is bidirectional, i.e., $\forall x, y \in U : (x, y) \in R$ implies $(y, x) \in R$. A relation $R$ on $U$ is injective iff every node in $U$ is $R$-accessible from at most one other node in $U$, i.e., $\forall x, y, z \in U : (x, z) \in R$ and $(y, z) \in R$ implies $x = y$.

The composition of two binary relations $R_1$ and $R_2$, denoted $R_1 \circ R_2$, is defined as:

$$ R_1 \circ R_2 = \{ (x, z) | \exists y : (x, y) \in R_1, (y, z) \in R_2 \} $$

In the case where $R$ is binary and $Y$ in unary, the composition $R \circ Y$ is defined as:

$$ R \circ Y = \{ x | \exists y : (x, y) \in R, y \in Y \} $$

The $n$-fold composition of a binary relation $R$ on $U$ is denoted as $R^n$ and is defined inductively as follows:

$$ R^0 = \text{id}_U $$

$$ R^{n+1} = R \circ R^n $$

The $n$-fold composition is then used to define the transitive closure of a relation $R$, denoted as $R^+$, and the reflexive transitive closure, denoted as $R^*$, as follows:

$$ R^+ = \bigcup_{n \geq 1} R^n $$

$$ R^* = \bigcup_{n \geq 0} R^n $$
It can be proved that $R^+$ is the smallest transitive relation containing $R$ and that $R^*$ is the smallest relation containing $R$ that is both reflexive and transitive [Harel et al., 2000]. The converse of a binary relation $R$, denoted as $R^-$, is defined as:

$$R^- = \{ (y, x) | (x, y) \in R \}$$

It may be interesting to note that the converse operation $^-$ commutes with the reflexive transitive closure operation $^*$, i.e., $(R^-)^* = (R^*)^-$ [Harel et al., 2000].

### 5.1 Motivational Constructs

In KARO, there are several operators and constructs built from these operators for expressing motivational attitudes of agents (cf. [Hoek et al., 1999; Meyer et al., 1999]). Here we introduce three basic operators, namely for goals, abilities, and commitments.

**Definition 5.1 (Basic KARO operators)**

The following operators from KARO are used:

- $G_i \varphi$: Agent $i$ has the declarative goal $\varphi$. Here $\varphi$ represents a state of affairs which agent $i$ wants to achieve. A goal $\varphi$ is said to have been achieved when agent $i$ believes $\varphi$ holds, i.e., when $B_i \varphi$ is true.\(^2\)
- $A_i \pi$: Agent $i$ has the ability to do $\pi$.
- $\text{Com}_i(\pi)$: Agent $i$ is committed to doing $\pi$ (or: plan $\pi$ is on agent $i$’s agenda).

Using these operators, several constructs are defined expressing (possible) motivations of an agent, as follows.

**Definition 5.2 (Motivational constructs)**

The following constructs expressing (possible) motivations are used:

- $\text{PracPoss}_i(\pi, \varphi) \equiv A_i \pi \land \langle i: \pi \rangle \varphi$ (5.1)
- $\text{Can}_i(\pi, \varphi) \equiv B_i \text{PracPoss}_i(\pi, \varphi)$ (5.2)
- $\text{AchvGoal}_i(\varphi) \equiv B_i (\neg \varphi \land G_i \varphi)$ (5.3)
- $\text{PossIntend}_i(\pi, \varphi) \equiv \text{Can}_i(\pi, \varphi) \land \text{AchvGoal}_i(\varphi)$ (5.4)

where $\text{PracPoss}$ stands for “practical possibility,” $\text{AchvGoal}$ stands for “achievement goal,” and $\text{PossIntend}$ stands for “possible intention.”

The above constructs can be read as follows. An agent has the practical possibility to perform an action/plan $\pi$ to bring about $\varphi$ iff it has the ability to perform $\pi$ and doing so may bring about $\varphi$. An agent can perform $\pi$ to bring about $\varphi$ iff it believes it has the practical possibility to do so. An agent has the goal to achieve $\varphi$ iff it believes

\(^2\)It is not assumed that achieved goals automatically “disappear.” To remove a goal $\varphi$, it is assumed there exists a special action $\text{drop}(\varphi)$; see also the $\text{DropG}(i)$ assumption on page 98.
that $\varphi$ does not hold and that $\varphi$ is one of its goals. An agent has the possible intention to perform $\pi$ to accomplish $\varphi$ iff it can do so and it believes $\varphi$ is an achievement goal.

So far we have defined several operators and constructs based on the notion of a ‘goal’ without explicating what goals look like. In line with previous work [Steunebrink et al., 2007, 2008a], we define goal formulas as non-empty, consistent conjunctions of literals. This way goals can easily be broken up in subgoals; in particular, every non-empty ‘subconjunction’ of a goal formula is considered to be a subgoal. For example, if $\text{breakfasted} \land \text{dressed} \land \neg \text{after 8AM}$ is a goal, then $\text{breakfasted}$, $\text{dressed}$, $\neg \text{after 8AM}$, $\text{breakfasted} \land \text{dressed}$, $\text{breakfasted} \land \neg \text{after 8AM}$, $\text{dressed} \land \neg \text{after 8AM}$, and $\text{breakfasted} \land \text{dressed} \land \neg \text{after 8AM}$ are subgoals. This notion of consistent conjunctions of literals as goal formulas can be formalized. In particular, we require goal formulas to be drawn from the set $\text{ccl}$, which is defined as follows.

**Definition 5.3 (Consistent conjunctions of literals)**

The set of literals is defined as the set of atomic propositions together with the set of negated atomic propositions:

$$\text{LIT} = \text{atm} \cup \{ \neg p \mid p \in \text{atm} \}$$  \hspace{1cm} (5.5)

The set of consistent sets of literals is defined as all nonempty subsets of literals from which falsum cannot be derived:

$$\text{csl} = \{ \Phi \mid \emptyset \subset \Phi \subseteq \text{LIT}, \Phi \not\models_{\text{PC}} \bot \}$$  \hspace{1cm} (5.6)

where $\text{PC}$ stands for Propositional Calculus (so each set in $\text{csl}$ is consistent). The set of consistent conjunctions of literals is then defined as the set of conjunctions formed over the set of consistent sets of literals:

$$\text{ccl} = \{ \land \Phi \mid \Phi \in \text{csl} \}$$  \hspace{1cm} (5.7)

By requiring that goal formulas are drawn from the set $\text{ccl}$, it is ensured that goal formulas are satisfiable, and thus that goals are achievable. However, it is not assumed that goals are mutually consistent. So an agent may have multiple goals that contradict each other. This does not have to be problematic, because an agent could just pursue and achieve conflicting goals in sequence.

### 5.2 Semantics

In this section the formal semantics of all operators (i.e., $\lfloor \cdot \rfloor$, *Prev*, *Past*, *Fut*, *B*, *G*, *A*, *Com*) are presented. For the dynamic ($\lfloor \cdot \rfloor$) and doxastic (*B*) operators we use possible world semantics. The concept of possible world semantics is illustrated in Figure 5.1. The box, identified as $M$, represents the belief model of two agents $i$ and $j$. The model consists of possible worlds (s and t1 to t6) and accessibility relations ($R_i$ and $R_j$). The possible worlds are called *states* henceforth. Let us consider s to be the ‘actual’ state. In this state, agent $i$ considers states $t_1$, $t_2$, and $t_3$ to be possible, whereas agent $j$ considers states $t_4$, $t_5$, and $t_6$ to be possible, which is formally denoted as
Figure 5.1: An example of a belief model using possible world semantics.

(s, t₁), (s, t₂), (s, t₃) ∈ Rᵢ and (s, t₄), (s, t₅), (s, t₆) ∈ Rⱼ, respectively. Anything that is true in state t₁ is said to be held as possible by agent i, and similarly for the other states. Anything that is true in all Rᵢ-accessible states (in this example, t₁, t₂, and t₃) is said to be believed by agent i. Note that the actual state (in this example, s) does not have to be among the states held as possible by an agent.

Formally, belief models are defined in KARO using Kripke semantics, as follows.

**Definition 5.4 (Belief models)**

Belief models are of the form M = ⟨S, R, V⟩, where

- S is a non-empty set of states (or ‘possible worlds’).
- R = {Rᵢ | i ∈ AGT} is a set of accessibility relations on S, one for each agent name, hence the notation Rᵢ. So Rᵢ ⊆ S × S for each Rᵢ ∈ R.
- V : S → 2^{αₜ} is a valuation function, indicating which atomic propositions hold per state.

As is common in doxastic logic, each belief-accessibility relation Rᵢ is required to be serial, transitive, and euclidean, i.e., the modal logic KD45 is used for belief models.

This view of belief models is quite static; a model of the form M = ⟨S, R, V⟩ only represents the beliefs of agents at a certain moment in time. But in KARO, agents can “change their minds” by performing actions. Actions are modeled in the same way as beliefs; just as an agent can hold multiple states as possible, an action can have multiple possible results. We can thus model actions using possible world semantics, but with belief models as the possible worlds. This nesting is illustrated in Figure 5.2. For clarity, agent indices have been omitted in this illustration. Assume the agent is in state s of belief model M, where it holds states t₁, t₂, and t₃ as possible. If the agent performs action α₁, it can end up in one of two new states, namely either state s'₁ of belief model M'₁ or state s'₂ of belief model M'₂. This is formally denoted as (((M, s), (M'₁, s'₁)), ((M, s), (M'₂, s'₂)) ∈ Rₐ₁. Because the belief-accessibility relations starting from s'₁ and s'₂ are different from those from s, action α₁ can be said to change the agent’s mental state. For example, one possible effect of action α₁ is that it makes
the agent stop holding $t_3$ to be a possible world (see model $M'_2$). After performing action $\alpha_1$, the agent can perform a new action, as suggested by the arrows labeled $\alpha_2$. The complete action model is identified as $M$.

Formally, the semantics of actions are defined over the Kripke models of belief, as specified in the definition below. Furthermore, an action model contains functions for indicating the goals, abilities, and commitments of agents. They are included in the action models, because in principle actions must be able to change all aspects of the mental states of agents; and indeed, goals, abilities, and commitments are part of the mental state of an agent. The same holds for the appraisal operators (desirability, praiseworthiness, etc.): they are part of the mental state of an agent and can be changed though actions, so they are included in the action models.

**Definition 5.5 (Action models)**

Action models are of the form $M = \langle S, R, Aux, Emo \rangle$, where

- $S$ is a non-empty set of possible model–state pairs, where a model is of the form $M$ as in Definition 5.4 and a state is from $S$ therein. That is, if $(M, s) \in S$ and $M = \langle S, R, V \rangle$ then it must be that $s \in S$.

- $R = \{ R_{i,\alpha} | i \in \text{agt}, \alpha \in \text{act} \}$ is a set of accessibility relations on $S$. Each transition is labeled with an agent name and an action, hence the notation $R_{i,\alpha}$.

- $Aux = \langle \text{Goals}, \text{Caps}, \text{Agd} \rangle$ is a structure of auxiliary functions, indicating per agent and model–state pair which goals (Goals), capabilities (Caps), and commitments (Agd) the agent has.
The mappings of the three auxiliary functions are as follows. Goals: \( \text{agt} \times S \rightarrow 2^{\text{ccl}} \) is a function returning the set of goals an agent has per model–state pair; Caps: \( \text{agt} \times S \rightarrow 2^{\text{plans}} \) is a function that returns the set of actions that an agent is capable of performing per model–state pair; and Agd: \( \text{agt} \times S \rightarrow 2^{\text{plans}} \) is a function that returns the set of actions that an agent is committed to (are on its ‘agenda’) per model–state pair.

\[ \text{Emo} = \langle \text{Des}, \text{Undes}, \text{Praisew}, \text{Blamew}, \text{Appeal}, \text{Unappeal}, \text{CogUnit} \rangle \]

is a structure of appraisal and judgment functions, indicating per agent and model–state pair how that agent appraises consequences (Des, Undes), actions of agents (Praisew, Blamew), and objects (Appeal, Unappeal), and how it judges cognitive units (CogUnit).

The mappings of the appraisal and judgment functions are as follows. Des, Undes: \( \text{agt} \times S \rightarrow 2^L \) for desirability and undesirability (where \( L \) is the set of all well-formed formulas); Praisew, Blamew: \( \text{agt} \times S \rightarrow 2^{\text{agt}\times\text{act}} \) for praiseworthiness and blameworthiness; Appeal, Unappeal: \( \text{agt} \times S \rightarrow 2^{\text{obj}} \) for appealingness and unappealingness; and CogUnit: \( \text{agt} \times S \rightarrow 2^{\text{agt}} \) for cognitive unit.

With respect to the action-accessibility relations, it should be noted that many different actions may be executable in any given state. This may not have been clear from the illustration above; however, drawing out all possible actions and their results even for a single state would already result in a very messy diagram. If action \( \alpha \) can reach at least one state from state \( s \), \( \alpha \) is said to be executable in \( s \). If action \( \alpha \) can reach at most one state from state \( s \), \( \alpha \) is said to be deterministic in \( s \). So with respect to the illustration above, \( \alpha_1 \) is executable in \( (M, s) \) but \( \alpha_1 \) is not deterministic in \( (M, s) \).

All action-accessibility relations \( R_{\text{iat}} \in R \) are extended to plans as usual in dynamic logic [Harel et al., 2000]; namely, by induction on the structure of the plan in question, as follows:

\[
R_{\text{iat}} \subseteq S \times S \quad \text{for } \alpha \in \text{act}
\]

\[
R_{\text{i}}(\pi_1 ; \pi_2) = R_{\text{i}}(\pi_1) \circ R_{\text{i}}(\pi_2)
\]

\[
R_{\text{i}}(\pi)^{-1} = (R_{\text{i}}(\pi))^{-1}
\]

So sequential composition of actions is modeled using the relational composition of their action-accessibility relations, whereas converse actions are modeled using the inverse of their accessibility relations.

Again we emphasize that in dynamic logic and in KARO in particular, actions are used as an abstraction of time. The possibility of multiple actions being executable in any state and the possibility of actions being nondeterministic means that we have a branching future. In principle the same holds for the past. However, in Chapter 4 we assumed that the history would be linear. Having now formally introduced action models, we can constrain them such that histories will indeed be linear. This is done formally by placing the following constraint on the action-accessibility relations.

**Constraint 5.6 (Linear history)** \( \bigcup R \) is injective.

This constraint ensures that any model–state pair can be reached from at most one other model–state pair, and thus that the history is linear. Note however that
this does not exclude parallel actions. For example, if \(((M, s), (M', s')) \in R_{ia}\) and 
\(((M, s), (M', s')) \in R_{ip},\) then state \((M', s')\) is a result of the parallel execution of \(i:a\) and
\(j:b\) in state \((M, s)\).

Another assumption made in Chapters 3 and 4 was that action instances are unique. We are now in a position to formalize this assumption as a constraint on \(R\), as follows.

**Constraint 5.7 (Uniqueness of actions)**

\[ \forall R_{ia} \in R : \forall ((M, s), (M', s')) \in R_{ia} : R_{ia} \cap (S' \times S') = \emptyset \]

where \(S' = \{(M'', s'') \mid ((M', s'), (M'', s'')) \in (\bigcup R)^*\}\).

This constraint can be read as follows. If state \((M', s')\) is a result of action \(a\) by agent \(i\) 
\((\forall ((M, s), (M', s')) \in R_{ia} : \ldots)\), then no possible future state after \((M', s')\) (collected in \(S'\)) can be reachable by \(i\) doing \(a\) again \((R_{ia} \cap (S' \times S') = \emptyset)\). It may be interesting to note that this constraint implies that \(\bigcup R\) must be free of circles, i.e., \((\bigcup R)^* \cap id_S = \emptyset\).

### 5.3 Interpretation in KARO

We now have all ingredients necessary for the interpretation of formulas in KARO, presented below. Formulas are interpreted in state \(s\) of model \(M\), where \((M, s) \in S\). It should be noted that the pair \((M, s)\) is itself a state of model \(M\); that is, belief models \((M)\) are nested in action models \((M)\), as explained and illustrated in the previous section. Strictly speaking, we should write \((M, (M, s)) \models \ldots\), but we drop the \(M\) for notational convenience.

**Definition 5.8 (Interpretation of formulas)**

Let \(M = \langle S, R, Aux, Emo \rangle\), \((M, s) \in S\), and \(M = \langle S, R, V \rangle\); formulas are then interpreted as follows.

**Basic connectives are interpreted as usual in modal logic:**

\[
\begin{align*}
M, s & \models p & \text{iff } & p \in V(s) & \text{ for } p \in \text{atm} \\
M, s & \models \neg \varphi & \text{iff } & \text{not } M, s \models \varphi \\
M, s & \models \varphi \land \psi & \text{iff } & M, s \models \varphi & \text{ and } M, s \models \psi
\end{align*}
\]

Only interpretations for negation and conjunction are given; the other connectives are defined as the abbreviations \(\varphi \lor \psi \equiv \neg (\neg \varphi \land \neg \psi)\), \(\varphi \rightarrow \psi \equiv \neg \varphi \lor \psi\), and \(\varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi)\). Beliefs (B) are interpreted using the Kripke structure \(M\). Goals (G), abilities (A), and commitments (Com) are simply interpreted using the sets in \(Aux = \langle \text{Goals}, \text{Caps}, \text{Agd} \rangle\):

\[
\begin{align*}
M, s & \models B_i \varphi & \text{iff } & \forall s' \in S : (s, s') \in R_i & \text{ implies } & M, s' \models \varphi \\
M, s & \models G_i \varphi & \text{iff } & \varphi \in \text{Goals}(i)(M, s)
\end{align*}
\]
M, s \models A_i \pi \quad \text{iff} \quad \pi \in \text{Caps}(i)(M, s)

M, s \models \text{Com}_i(\pi) \quad \text{iff} \quad \pi \in \text{Agd}(i)(M, s)

Actions are interpreted using the Kripke structure $M$ in the standard way. For clarity of presentation the interpretation of $(i:\pi)\varphi$ is given; necessary results of actions are then simply defined as $[i:\pi]\varphi \overset{\text{def}}{=} \neg(i:\pi)\neg \varphi$. The future, past, and previous operators are interpreted over all action-accessibility relations in $R$, which is done by taking the union $\bigcup R$. $(\bigcup R)^\ast$ is then a relation connecting model–state pairs reachable in zero or more actions of agents. Notice that $(M, s)$ and $(M', s')$ are reversed for the future and past operators:

$M, s \models (i:\pi)\varphi \quad \text{iff} \quad \exists (M', s') \in S : M', s' \models \varphi \quad \text{and} \quad ((M, s), (M', s')) \in R_{i:}\pi$

$M, s \models \text{Fut}\varphi \quad \text{iff} \quad \exists (M', s') \in S : M', s' \models \varphi \quad \text{and} \quad ((M, s), (M', s')) \in (\bigcup R)^\ast$

$M, s \models \text{Past}\varphi \quad \text{iff} \quad \exists (M', s') \in S : M', s' \models \varphi \quad \text{and} \quad ((M', s'), (M, s)) \in (\bigcup R)^\ast$

$M, s \models \text{Prev}\varphi \quad \text{iff} \quad \exists (M', s') \in S : M', s' \models \varphi \quad \text{and} \quad ((M', s'), (M, s)) \in (\bigcup R)$

The appraisal operators and the cognitive unit operator are interpreted using the sets in $\text{Emo} = \langle \text{Des}, \text{Undes}, \text{Praisew}, \text{Blamew}, \text{Appeal}, \text{Unappeal}, \text{CogUnit} \rangle$:

$M, s \models \text{Des}_i(\varphi) \quad \text{iff} \quad \varphi \in \text{Des}(i)(M, s)$

$M, s \models \text{Undes}_i(\varphi) \quad \text{iff} \quad \varphi \in \text{Undes}(i)(M, s)$

$M, s \models \text{Praisew}_i(j:\alpha) \quad \text{iff} \quad (j, \alpha) \in \text{Praisew}(i)(M, s)$

$M, s \models \text{Blamew}_i(j:\alpha) \quad \text{iff} \quad (j, \alpha) \in \text{Blamew}(i)(M, s)$

$M, s \models \text{Appeal}_i(x) \quad \text{iff} \quad x \in \text{Appeal}(i)(M, s)$

$M, s \models \text{Unappeal}_i(x) \quad \text{iff} \quad x \in \text{Unappeal}(i)(M, s)$

$M, s \models \text{CogUnit}_i(j) \quad \text{iff} \quad j \in \text{CogUnit}(i)(M, s)$

As usual, $\models \varphi$ is used to denote that $\varphi$ is valid, i.e., $\varphi$ is satisfied in all possible model–state pairs.

We are not done yet. Having provided formal semantics for the basic operators of dynamic doxastic logic, we have now grounded all the work done in Chapter 4. However, it remains to be answered how the sets in $\text{Emo} (\text{Des}, \text{etc.})$ get their elements before we can declare the work done in Chapter 3 to be grounded. Determining this will constitute the final step of our formalization of the eliciting conditions of OCC’s emotions.

5.4 Appraisal Operators

Until now we have deferred the problem of formalizing appraisal to the functions $\text{Des}, \text{Undes}, \text{Praisew}, \text{Blamew}, \text{Appeal}, \text{and Unappeal}$. In this chapter we have introduced (achievement) goals, which will allow us to give meaning to these functions. The notion of achievement goals allows us to do this only to a limited degree, however, because there are many other kinds of concerns, such as norms, interests, preservation, etc. Therefore, we will not define these appraisal functions; instead, we will
constrain them such that they capture appraisal for agents with achievement goals only. The idea is that one can simply add more constraints to these appraisal function if the framework is enriched with more kinds of concerns.

Before introducing these constraints on the appraisal functions, we define two helper sets for matching subparts of goals and inverting goals, respectively. Recall that we require achievement goals to be consistent conjunctions of literals, which means that goal formulas are drawn from the set $ccl$. The relation $\text{sub}$ on $ccl$ will then be convenient for making subgoals, and the relation $\text{inv}$ will be convenient for inverting entire (sub)goals. For example, if $p_1 \land \neg p_2$ is a goal, then $p_1$ and $\neg p_2$ (and $p_1 \land \neg p_2$ itself) are subgoals, and $\neg p_1 \land p_2$ is the inverted goal. Formally: if $p_1 \land \neg p_2 \in ccl$, then $(p_1, p_1 \land \neg p_2)$ and $(\neg p_2, p_1 \land \neg p_2)$ are subgoals, and $(\neg p_1 \land p_2, p_1 \land \neg p_2) \in \text{inv}$. 

**Definition 5.9 (Subgoals and inverted goals)**

$\text{sub}$ is a relation on $ccl$ capturing the notion of subgoals. $\text{inv}$ is a relation on $ccl$ capturing the notion of inverted goals. $\text{sub}$ and $\text{inv}$ are defined as follows:

\[
\text{sub} = \{ (\bigwedge \Phi_1, \bigwedge \Phi_2) \mid \Phi_1, \Phi_2 \in \text{csl}, \Phi_1 \subseteq \Phi_2 \} \tag{5.8}
\]

\[
\text{inv} = \{ (\bigwedge \Phi_1, \bigwedge \Phi_2) \mid \Phi_1 \in \text{csl}, \Phi_2 = \{ \text{neg}(\varphi) \mid \varphi \in \Phi_1 \} \} \tag{5.9}
\]

where $\text{neg}(p) = \neg p$ and $\text{neg}(\neg p) = p$.

It will be obvious from this definition that $\text{sub}$ and $\text{inv}$ are binary relations on $ccl$, i.e., $\text{sub, inv} \subseteq ccl \times ccl$. It may be interesting to note that $\text{inv}$ is symmetric (i.e., $\text{inv} = \text{inv}^-$) and that inverting a subgoal is equivalent to taking a subpart of an inverted goal (i.e., $\text{inv} \circ \text{sub} = \text{sub} \circ \text{inv}$).

### 5.4.1 Desirability and Undesirability

We are now ready to constrain the appraisal operators. Let us start with the simplest case of desirability. According to the OCC model, desirability is related to goals, including achievement goals. Therefore, the achievement of a goal or subgoal should be desirable, and the undermining\(^3\) of a subgoal should be undesirable. Let $\text{Des}$ then be constrained such that every subgoal is desirable, and let $\text{Undes}$ be constrained such that every inverted subgoal is undesirable.

**Constraint 5.10 (Desirability of achieving own goals)**

For all agents $i \in \text{agt}$ and model–state pairs $(M, s) \in S$, $\text{Des}$ and $\text{Undes}$ are constrained as follows:

\[
\text{Des}(i)(M, s) \supseteq \text{sub} \circ \text{Goals}(i)(M, s) \tag{5.10}
\]

\[
\text{Undes}(i)(M, s) \supseteq \text{inv} \circ \text{sub} \circ \text{Goals}(i)(M, s) \tag{5.11}
\]

\(^3\)With respect to achievement goals, we use “undermining” to mean undoing previously achieved subgoal(s).
These two constraints read just as they are written: Des contains all subgoals, and Undes contains all inverted subgoals (recall the definition of relation composition from page 82). Let us illustrate the above constraint on Undes: if breakfasted ∧ dressed ∧ ¬after_8AM is a goal (i.e., breakfasted ∧ dressed ∧ ¬after_8AM ∈ Goals(i)(M, s)), then breakfasted ∧ dressed is a subgoal (because (breakfasted ∧ dressed, breakfasted ∧ dressed ∧ ¬after_8AM) ∈ sub) and ¬breakfasted ∧ dressed is an inverted subgoal (because (¬breakfasted ∧ dressed, breakfasted ∧ dressed) ∈ inv), so ¬breakfasted ∧ dressed is undesirable (i.e., ¬breakfasted ∧ dressed ∈ Undes(i)(M, s)).

With respect to the fortunes-of-others emotion types (‘happy-for’, ‘pity’, ‘gloating’, ‘resentment’), it may be interesting to investigate how the desires of agents can be related to the desires of other agents. In the ‘good-will’ case, it can be said that an agent finds it desirable that anything happens that is presumed to be desirable for appealing agents only. However, we have specified i and j as typical elements of agt, so we assume it is understood that we are quantifying over appealing agents only.

Although not problematic, it should be noted that these constraints may lead to contradicting desires. For example, if p ∈ Goals(i)(M, s) and ¬p ∈ Goals(j)(M, s′) for j ∈ Appeal(i)(M, s), then p, ¬p ∈ Des(i)(M, s). However, it was never assumed that an agent’s own achievement goals are consistent either. For example, it is possible that p, ¬p ∈ Goals(i)(M, s) to begin with. We do not consider inconsistencies in goals and desires to be problematic, because usually not all goals are pursued simultaneously and choices have to be made which goal(s) to give precedence. However, we will not go into the issue of preferences among goals in this chapter.

\[^4\text{Strictly speaking, we should write } \exists j \in \text{Appeal}(i)(M, s) \cap \text{agt} : \ldots \text{in order to ensure that } j \text{ is really an agent, because appealingness is defined with respect to all objects, not just agents. However, we have specified } i \text{ and } j \text{ as typical elements of agt, so we assume it is understood that we are quantifying over appealing agents only.}\]
These possible constraints can be changed in an interesting way by swapping the existential and the universal quantifications. By writing \( \text{Des}(i)(M, s) \supseteq \{ \varphi \mid \forall s' \in R_i(s) : \exists j \in \text{Appeal}(i)(M, s) : \varphi \in \text{Des}(j)(M, s') \} \) it would be specified that \( \varphi \) is desirable for agent \( i \) if in all states held as possible by agent \( i \), there is an appealing agent \( j \) to which \( \varphi \) is desirable. The difference with the shown constraints is that there may be different appealing agents to which \( \varphi \) is desirable, but none of them have to find \( \varphi \) desirable in all states held as possible by agent \( i \). However, in order to closely follow the OCC model, the appealing agent has to be fixed and then it has to be examined whether or not there are formulas that are believed to be desirable for that appealing agent. The emotion types that depend on this constraint on \( \text{Des} \) and \( \text{Undes} \) are the fortunes-of-others emotions (‘happy-for’, ‘pity’, ‘gloating’, ‘resentment’). According to the OCC model, these emotion types are elicited with respect to a single agent (other than the appraising agent), so fixing to a single agent (i.e., by putting \( \exists j \) … in front) makes sure this approach is followed.

However, the reason we do not formally adopt the above formulas as constraints is that they can lead to a ‘paradox’. Because the set of desirable consequences of events for one agent (say, Alice) would be constrained to include what another appealing agent (say, Bob is appealing to Alice) finds desirable, this can lead to a paradox if Bob would find anything desirable which Alice does not. That is, if we assume that \( \text{Appeal}(Alice)(M, s) = \{ Bob \} \) and \( \text{Des}(Bob)(M, s') = \{ \varphi \mid \varphi \notin \text{Des}(Alice)(M, s') \} \) for all \( (s, s') \in R_{Alice} \)—plus some additional assumptions, such as neither having any achievement goals, but we will not go into the minute details—then we would have that \( \varphi \in \text{Des}(Alice)(M, s) \) iff \( \varphi \notin \text{Des}(Alice)(M, s) \). Such a paradox is of course highly undesirable, therefore we have not formally adopted the four constraints above. Still, it is interesting to investigate how the desires of one agent interact with the presumed desires of other agents, but we leave the search for ‘paradox-free’ constraints for future work.\(^5\)

### 5.4.2 Praiseworthiness and Blameworthiness

The OCC model considers praiseworthiness (and its negative counterpart blameworthiness) to be determined with respect to one’s standards. However, OCC note that the praiseworthiness of an action may be evaluated with respect to the desirability of the events resulting from that action. Since we do not explicitly consider standards here, we will constrain \( \text{Praisew} \) and \( \text{Blamew} \) using \( \text{Des} \) and \( \text{Undes} \), respectively. Of course, different or additional constraints may be studied if an explicit representation of standards were added to the logical framework.

We now constrain \( \text{Praisew} \) and \( \text{Blamew} \) as follows. An action of an agent is appraised as praiseworthy or blameworthy when the appraising agent believes that the action is related to a desirable or undesirable consequence, respectively.

**Constraint 5.11 (Praiseworthiness and blameworthiness)**

For all agents \( i \in \text{agt} \) and model–state pairs \( (M, s) \in S \), \( \text{Praisew} \) and \( \text{Blamew} \) are constrained

\(^5\)Thanks to Wiebe van der Hoek for pointing out the paradox and the possibility to swap the quantifiers.
as follows:

\[
Praisew(i)(M, s) \supseteq \{ (j, \alpha) | \exists \varphi \in Des(i)(M, s) : M, s \models B, Related(j, \alpha, \varphi) \} \quad (5.12)
\]

\[
Blamew(i)(M, s) \supseteq \{ (j, \alpha) | \exists \varphi \in Undes(i)(M, s) : M, s \models B, Related(j, \alpha, \varphi) \} \quad (5.13)
\]

where \textit{Related} is as defined in Definition 4.17.

We did not spell out the condition \( M, s \models B, Related(j, \alpha, \varphi) \) using the semantics because that would have made these constraints considerably more difficult to read without becoming more enlightening.

### 5.4.3 Appealingness and Unappealingness

According to the OCC model, appealingness and unappealingness are determined with respect to one’s attitudes. Here we will not constrain the functions \textit{Appeal} and \textit{Unappeal} with respect to objects that are not agents. Instead, we will only consider the appealingness of agents, as follows. An agent is appealing to the appraising agent if it has performed at least one praiseworthy action, and unappealing if it has performed at least one blameworthy action.

**Constraint 5.12 (Appealingness and unappealingness)**

For all agents \( i \in \text{agt} \) and model–state pairs \( (M, s) \in S \), \textit{Appeal} and \textit{Unappeal} are constrained as follows:

\[
Appeal(i)(M, s) \supseteq \{ j | \exists \alpha : M, s \models \text{Past Approving}_{i}^{T}(j, \alpha) \} \quad (5.14)
\]

\[
Unappeal(i)(M, s) \supseteq \{ j | \exists \alpha : M, s \models \text{Past Disapproving}_{i}^{T}(j, \alpha) \} \quad (5.15)
\]

Because the current framework lacks a notion of ‘attitude’, we are forced to constrain the appealingness and unappealingness of agents in terms of the praiseworthiness and blameworthiness of their actions. Of course the \textit{amount} of appealingness of an agent would depend on the number of praiseworthy actions it has performed versus the number of blameworthy actions, plus probably some other considerations. For example, the time of these actions may be important; if an agent used to perform many blameworthy actions but has bettered itself by performing only praiseworthy action recently, then that agent may still be found appealing. However, what we want to achieve with this qualitative formalization is to specify the \textit{minimal} conditions for when to start calculating the ‘actual’ amount of appealingness. One praiseworthy or one blameworthy action is then this minimal condition. For further quantitative considerations, see Chapter 6.

Note that the definitions of ‘approving’ (3.8) and ‘disapproving’ (3.9) include the perception of the praiseworthy or blameworthy action. Again, we did not spell out these conditions using the semantics in order to keep the constraints concise and easy to read. It will be clear that other agents can be found to be both appealing and unappealing, allowing one agent to have “mixed feelings” about another agent. Also note that it is perfectly possible for an agent to find itself appealing or unappealing.

It should be emphasized that we have started out with just achievement goals as the only concerns of agents. We have then defined (or rather, constrained) desirability and undesirability in terms of achievement goals, then defined praiseworthiness
and blameworthiness in terms of desirability and undesirability, and then defined appealingness and unappealingness in terms of praiseworthiness and blameworthiness. Thus we have given meaning to the appraisal operators using just achievement goals. We re-emphasize that these appraisal operators can be refined by introducing more types of concerns and placing appropriate constraints. This effort is left for future work, however.

5.5 Cognitive Unit

In this section we briefly consider the function $\text{CogUnit}$ indicating which other agents are considered to be in a cognitive unit with the appraising agent. As noted in Section 4.6, a deep investigation into the notion of cognitive unit is left for future research. Nevertheless, one particular property of cognitive units should not be overlooked, namely regarding the self.

We do not constrain an agent’s judgment of when it considers itself to be in a cognitive unit with another agent, except that we require each agent to at least be in a cognitive unit with itself. Even this simple constraint may be too strong in general, because it may preclude a kind of ‘insanity’ where one does not consider the self as the (cognitive) author of one’s own actions. However, the current formalization is aimed at normal, healthy agents, so it should be safe to place the following constraint.

**Constraint 5.13 (Cognitive unit)**

For all agents $i \in \mathbf{agt}$ and model–state pairs $(M, s) \in S$, $\text{CogUnit}$ is constrained as follows:

$$\text{CogUnit}(i)(M, s) \supseteq \{i\}$$

This constraint ensures that $\text{Approving}_i^T(i:\alpha)$ is equivalent to $\text{Pride}_i^T(i:\alpha)$ and that $\text{Disapproving}_i^T(i:\alpha)$ is equivalent to $\text{Shame}_i^T(i:\alpha)$, as anticipated in Chapter 3 (cf. formulas (3.50) and (3.51) on page 53).

5.6 Confirmation and Disconfirmation

The only constructs yet undefined are $\text{Confirms}$ and $\text{Disconfirms}$. Recall from Chapter 3 that $\text{Confirms}(\varphi, \psi)$ expresses that agent $i$ considers consequence $\varphi$ to confirm consequence $\psi$, and likewise for disconfirmation. Now that we have restricted concerns of agents to achievement goals only and defined achievement goals as conjunctions of literals, representing (dis)confirmation has become quite straightforward.

For convenience, we first introduce some additional syntax. The operator $\sqsubseteq$ will be used as the syntactic counterpart of the relation $\sqsubseteq$, and the operator $\ominus$ as the syntactic counterpart of the relation $\sqsubseteq$. The interpretation of $\sqsubseteq$ and $\ominus$ is thus as follows:

$$\models \varphi \sqsubseteq \psi \quad \text{iff} \quad (\varphi, \psi) \in \sqsubseteq$$

$$\models \varphi \ominus \psi \quad \text{iff} \quad (\varphi, \psi) \in \ominus$$
φ ⊑ ψ is then read as “φ is a (logical) part of ψ,” and φ ⊕ ψ is read as “φ is the inverse conjunction of ψ.” Note that the interpretations of φ ⊑ ψ and φ ⊕ ψ do not depend on the Kripke models for belief and action.

To ease notation, we will not use the ⊕ operator but instead write ϕ, meaning the ‘inverse’ of ϕ in the sense of inv. In other words, for all (ϕ, ψ) ∈ inv, ψ = ϕ. This simplification is not problematic because any expression containing ϕ can always be rewritten to an equivalent one containing ⊕. Formally speaking, if χ is an expression containing ϕ, we can substitute all occurrences of ϕ by a fresh variable ψ and append . . . ∧ ϕ ⊕ ψ; that is, χ(ϕ) = χ(ψ) ∧ ϕ ⊕ ψ where ψ does not occur in χ. Note that inv is symmetric, so that ϕ = ϕ, as expected.

Using these new constructs, Confirms and Disconfirms are defined as follows.

**Definition 5.14 (Confirmation and disconfirmation)**

A goal formula φ confirms a goal formula ψ for agent i iff agent i believes φ is a subgoal of ψ.

A goal formula φ disconfirms a goal formula ψ for agent i iff agent i believes φ is an inverted subgoal of ψ:

\[
\text{Confirms}_i(ϕ, ψ) \overset{\text{def}}{=} B_i(ϕ ⊑ ψ) \quad (5.17)
\]

\[
\text{Disconfirms}_i(ϕ, ψ) \overset{\text{def}}{=} B_i(ϕ ⊑ ψ) \quad (5.18)
\]

This definition expresses that a consequence φ confirms another consequence ψ when φ is a (logical) part of ψ. It is incorrect to require that φ be (logically) stronger than ψ, because the idea of ‘confirms’ is that it must also account for partial confirmations. For example, assume that Alice learns that a plane carrying four of her relatives has crashed; she will then fear they have all perished but hope for survivors. When later she learns that two of her relatives have survived the crash, this will both partially confirm her fear and partially confirm her hope. To account for partial confirmations, then, we use the construct φ ⊑ ψ.

We emphasize that the presented definitions of Confirms and Disconfirms depend on our focus on achievement goals and our representation of goal formulas as conjunctions of literals. Consequently, if different kinds of goals (e.g., interests, preservation) would be added to the formalization, the definitions of Confirms and Disconfirms will likely have to be adapted. For example, the usage of Confirms_i(ϕ, ψ) and Disconfirms_i(ϕ, ψ) (in Definition 3.12) ensures that ϕ is the subject of a belief update. That is, ϕ comes from either Joy^T_i(ϕ) or Distress^T_i(ϕ), both of which entail BelUpd_i(ϕ). So from the perspective of achievement goals, ϕ is already something new; all that the specification of Disconfirms then has to do is to check for partial disconfirmation, which is indeed what the expression ϕ ⊑ ψ does. However, in the case of, e.g., interest goals the notion of disconfirmation is subtly different. In that case, consequence ϕ can also be said to disconfirm interest goal ψ if ϕ represents the fact that ψ can never be achieved in the future. So when one wishes to extend the current formalization with interest goals, the Confirms and Disconfirms constructs will likely have to be extended to take into account presumptions about the future.

Finally, we should note that it is not impossible to define confirmation more generally, instead of just with respect to conjunctions of literals. For example, “φ (partially) confirms ψ” can also be expressed as “ψ ⊢_{PC} φ,” i.e., ψ logically entails
\( \varphi \). But then we would need a construct for representing entailment in the object language. In effect, \( \sqsubseteq \) is a kind of entailment relation (\( \varphi \sqsubseteq \psi \) implies \( \psi \vdash_{PC} \varphi \)), but restricted to goal formulas (i.e., conjunctions of literals).

### 5.7 Properties

In this section we discuss some properties of the presented formalization in KARO of the eliciting conditions of emotions of the OCC model. As usual, \( \models \varphi \) expresses that the formula \( \varphi \) is a validity; that is, every state of every model satisfies \( \varphi \). \( \Gamma \models \varphi \) is then used to denote that \( \varphi \) is valid assuming \( \Gamma \). All definitions and constraints presented in Chapters 3, 4, and 5 are assumed to be in effect. Proofs of the propositions below can be found in Section 5.10.

First of all, the following propositions show how the appraisal operators stem from just (achievement) goals and beliefs. Of course, concerns other than achievement goals can influence desirability, praiseworthiness, and appealingness, but we have restricted our study of appraisal to achievement goals in this dissertation.

**Proposition 5.15 (Properties of appraisal)**

The following propositions are valid:

\[
\begin{align*}
\models G_i \varphi \land \psi \sqsubseteq \varphi & \rightarrow \text{Des}_i(\psi) \land \text{Undes}_i(\overline{\psi}) \\
\models \text{Des}_i(\varphi) \land B_i\text{Related}(j;\alpha, \varphi) & \rightarrow \text{Praisew}_i(j;\alpha) \\
\models \text{Undes}_i(\varphi) \land B_i\text{Related}(j;\alpha, \varphi) & \rightarrow \text{Blamew}_i(j;\alpha) \\
\models \text{Past Approving}^T_i(j;\alpha) & \rightarrow \text{Appeal}_i(j) \\
\models \text{Past Disapproving}^T_i(j;\alpha) & \rightarrow \text{Unappeal}_i(j)
\end{align*}
\]  

Note that these propositions correspond directly to the constraints specified in Section 5.4. The notation \( \overline{\psi} \) was explained in Section 5.6. We emphasize again that desirability and undesirability, praiseworthiness and blameworthiness, and appealingness and unappealingness are not mutually exclusive; see formulas (3.52), (3.53), and (3.54) on page 54. Furthermore, desirability and undesirability are not assumed to be individually consistent either. For example, \( \text{Des}_i(\varphi) \land \text{Des}_i(\overline{\varphi}) \) and \( \text{Undes}_i(\varphi) \land \text{Undes}_i(\overline{\varphi}) \) are not required to be contradictions.

The following propositions are restatements of (3.42), (3.43), (3.50), and (3.51), respectively.

**Proposition 5.16 (Properties of concrete emotion types)**

The following propositions are valid:

\[
\begin{align*}
\models \neg(\text{Hope}^T_i(\varphi) \land \text{Joy}^T_i(\varphi)) \\
\models \neg(\text{Fear}^T_i(\varphi) \land \text{Distress}^T_i(\varphi)) \\
\models \text{Pride}^T_i(j;\alpha) & \leftrightarrow \text{Approving}^T_i(j;\alpha) \\
\models \text{Shame}^T_i(j;\alpha) & \leftrightarrow \text{Disapproving}^T_i(j;\alpha)
\end{align*}
\]
In previous chapters, additional assumptions were required to make these propositions derivable. In our formalization in KARO, we have turned these assumptions into constraints, making these propositions truly theorems.

In the following, we drop the agent subscripts (e.g., $i$ and $j$) to ease notation; all terms requiring one are assumed to have the same agent subscript. The four propositions below express the triggering conditions for the event-based (and self-based) emotion types in BDI-like terms, i.e., in terms of beliefs and goals.

**Proposition 5.17 (Properties of event-based emotion triggers in terms of beliefs and goals)**

The following propositions are valid:

\[
| \models G \varphi \land \psi \subseteq \varphi \land \text{BelUpd}(\psi) \rightarrow \text{Joy}^T(\psi) \quad (5.28)
\]

\[
| \models G \varphi \land \psi \subseteq \varphi \land \text{BelUpd}(\overline{\psi}) \rightarrow \text{Distress}^T(\overline{\psi}) \quad (5.29)
\]

\[
| \models G \varphi \land \psi \subseteq \varphi \land \text{BelUpd}(\text{Fut}^+ \psi) \rightarrow \text{Hope}^T(\psi) \quad (5.30)
\]

\[
| \models G \varphi \land \psi \subseteq \varphi \land \text{BelUpd}(\text{Fut}^- \overline{\psi}) \rightarrow \text{Fear}^T(\overline{\psi}) \quad (5.31)
\]

The first proposition states that ‘joy’ is triggered with respect to $\psi$ if $\psi$ is a subgoal of the agent and it has just updated its beliefs with $\psi$ (i.e., subgoal $\psi$ has just been achieved). Analogously, the second proposition states that ‘distress’ is triggered with respect to an inverted subgoal $\overline{\psi}$ if the subgoal $\psi$ has just been undermined (i.e., part of $\psi$ had previously been achieved but the agent now believes the inverse $\overline{\psi}$ to be true). The third and fourth propositions have similar readings, save for being future-directed.

The following propositions show that in the extreme case where the (sub)goal hoped for or feared is achieved or undermined in its entirety, ‘satisfaction’, ‘fears-confirmed’, ‘relief’, or ‘disappointment’ is triggered with respect to the complete (sub)goal.

**Proposition 5.18 (Properties of prospect-based emotion triggers)**

The following propositions are valid:

\[
| \models \text{Joy}^T(\varphi) \land \text{Past Hope}^T(\varphi) \rightarrow \text{Satisfaction}^T(\varphi, \varphi) \quad (5.32)
\]

\[
| \models \text{Distress}^T(\overline{\varphi}) \land \text{Past Fear}^T(\overline{\varphi}) \rightarrow \text{Fears-confirmed}^T(\overline{\varphi}, \overline{\varphi}) \quad (5.33)
\]

\[
| \models \text{Joy}^T(\varphi) \land \text{Past Fear}^T(\overline{\varphi}) \rightarrow \text{Relief}^T(\varphi, \overline{\varphi}) \quad (5.34)
\]

\[
| \models \text{Distress}^T(\overline{\varphi}) \land \text{Past Hope}^T(\varphi) \rightarrow \text{Disappointment}^T(\overline{\varphi}, \varphi) \quad (5.35)
\]

Comparing these propositions to Definition 3.12 reveals that the propositions above have lost the confirmation and disconfirmation terms. In this case they can be left out because $| \models \text{Confirms}(\varphi, \varphi)$ and $| \models \text{Disconfirms}(\overline{\varphi}, \varphi)$ are valid, which are exactly the ‘missing’ terms.

Of course it must be recognized that the current formalization also has its limitations. The OCC model describes emotions that are triggered with respect to events, actions, and objects, and therefore the elicitation of emotions is naturally described in terms of the perception of events, actions, and objects. However, this emphasis
on perception fails to incorporate changes in the appraisal of goals, standards, and attitudes as triggers for emotions. For example, the perception of a desirable consequence of an event can trigger joy, but a new desire for a known consequence does not trigger joy in the current formalization. Formally, we now have:

\[
\begin{align*}
\models & \text{New } B\varphi \land \text{Des}(\varphi) \rightarrow \text{Joy}^T(\varphi) \\
\not\models & B\varphi \land \text{New } \text{Des}(\varphi) \rightarrow \text{Joy}^T(\varphi)
\end{align*}
\]

But the latter should intuitively be valid as well. Of course it is possible to define the emotion triggers such that changes in appraisal are taken into account, but this has not been the focus of this work and will be left as a topic of future work.

The propositions above have shown various ways in which many of the emotion types are triggered in response to beliefs and desires. With respect to the BDI perspective, this has left the notion of intention out of the picture. The final propositions that we discuss relate intention with tracking of goal achievements and undermining. Given several (reasonable) assumptions, the notion of intention as used in KARO (see formula (5.4)) is related with a simultaneous elicitation of ‘pride’, ‘joy’, and ‘gratification’. Before formally presenting the propositions, let us first name a number of possible constraints that will be useful here and later on.

- \(Acd(i,j;\alpha) = \forall \psi : B_i[j;\alpha]\psi \rightarrow [j;\alpha]B_i\psi\): action \(\alpha\) of agent \(j\) is accordant for agent \(i\), i.e., agent \(i\) does not forget the results of \(j;\alpha\).
- \(Det(i;\alpha) = \forall \psi : (i;\alpha)\psi \rightarrow [i;\alpha]\psi\): action \(\alpha\) by agent \(i\) is deterministic.
- \(NoPar(i;\alpha) = \forall j,\beta : \neg (\text{Done}(i;\alpha) \land \text{Done}(j;\beta))\) for \(\alpha \neq \beta\): no action can be performed in parallel to \(i;\alpha\).
- \(BG(i) = \forall \psi : B_i G_i \psi \rightarrow G_i \psi\): believed goals are true goals for agent \(i\).
- \(DropG(i) = \forall \psi : \text{Prev } G_i \psi \land \neg G_i \psi \rightarrow \text{Done}(i;\text{drop}(\psi))\): only ‘drop’ actions can remove goals of agent \(i\).
- \(NoDrop(\alpha) = \forall \psi : \alpha \neq \text{drop}(\psi)\): action \(\alpha\) will not drop a goal.
- \(NoCU(i,j) = \neg \text{CogUnit}_i(j)\): agent \(i\) never considers itself to be in a cognitive unit with agent \(j\). Note that because of Constraint 5.13, \(NoCU(i,j)\) implies that \(i \neq j\).

Taking a subset of these constraints as assumptions yields the following proposition. If an agent (possibly) intends to perform action \(\alpha\) to achieve goal \(\varphi\), then after actually performing \(\alpha\), ‘pride’ about having done so will be triggered, as well as ‘joy’ about the resulting achievement, and ‘gratification’ about the action leading to the achievement.

**Proposition 5.19 (Executing an intention triggers emotions)**

The following proposition is valid:

\[
\Gamma_i^\tau \models \text{PossIntend}_i(\alpha, \varphi) \rightarrow [i;\alpha] (\text{Pride}^T_i(i;\alpha) \land \text{Joy}^T_i(\varphi) \land \text{Gratification}^T_i(i;\alpha, \varphi))
\]

where \(\Gamma_i^\tau = \{Acd(i,j;\alpha), Det(i;\alpha), NoPar(i;\alpha), BG(i), DropG(i), NoDrop(\alpha)\}\) is a set of assumptions.
Interestingly, in contrast to ‘pure’ KARO, the current framework allows us to reason about subgoals. This makes it possible to define a less strict version of PossIntend; namely, one expressing that an agent can achieve a subgoal with some action or plan (in contrast to a complete goal as required by PossIntend). Analogously, we can define a construct expressing that an agent can undermine a subgoal with some action or plan. These two constructs are defined below as PossAch and PossUnd, respectively.

\[
\text{PossAch}_i(j; \pi, \psi, \varphi) \overset{\text{def}}{=} B_i(\text{PracPoss}_j(\pi, \psi) \land G_i \varphi \land \psi \subseteq \varphi \land \psi)
\]

(5.37)

\[
\text{PossUnd}_i(j; \pi, \psi, \varphi) \overset{\text{def}}{=} B_i(\text{PracPoss}_j(\pi, \psi) \land G_i \varphi \land \neg \psi \subseteq \varphi \land \psi)
\]

(5.38)

PossAch\(_i(\pi, \psi, \varphi)\) is read as “agent \(i\) can possibly achieve subgoal \(\psi\) of goal \(\varphi\) with plan \(\pi\),” and PossUnd\(_i(\pi, \psi, \varphi)\) is read as “agent \(i\) can possibly undermine subgoal \(\psi\) of goal \(\varphi\) with plan \(\pi\)” (here \(\psi\) is thus an inverted subgoal of \(\varphi\)). It may be instructive to compare formula (5.37) to formula (5.4) on page 83. PossIntend\(_i(\pi, \varphi)\) can be rewritten as \(B_i(\text{PracPoss}_j(\pi, \varphi) \land G_i \varphi \land \neg \varphi)\), which will make it obvious that PossAch\(_i(i; \pi, \varphi)\) implies PossIntend\(_i(\pi, \varphi)\). Using PossAch we can strengthen Proposition 5.19, and using PossUnd we can add an analogous case for the simultaneous elicitation of ‘shame’, ‘distress’, and ‘remorse’. Moreover, analogous cases for agents \(i \neq j\) can be obtained, as follows.

**Proposition 5.20 (Properties of PossAch and PossUnd)**

The following propositions are valid:

\[
\models \text{PossAch}_i(i; \pi, \varphi, \varphi) \rightarrow \text{PossIntend}_i(\pi, \varphi)
\]

(5.39)

\[
\Gamma^a_i \models \text{PossAch}_i(i; \alpha, \psi, \varphi) \rightarrow [i; \alpha](\text{Pride}^T_i(i; \alpha) \land \text{Joy}^T_i(\psi) \land \text{Gratification}^T_i(i; \alpha, \psi))
\]

(5.40)

\[
\Gamma^a_i \models \text{PossUnd}_i(i; \alpha, \psi, \varphi) \rightarrow [i; \alpha](\text{Shame}^T_i(i; \alpha) \land \text{Distress}^T_i(\psi) \land \text{Remorse}^T_i(i; \alpha, \psi))
\]

(5.41)

\[
\Gamma^a_{ij} \models \text{PossAch}_j(j; \alpha, \psi, \varphi) \rightarrow [j; \alpha](\text{Admiration}^T_j(j; \alpha) \land \text{Joy}^T_j(\psi) \land \text{Gratitude}^T_j(j; \alpha, \psi))
\]

(5.42)

\[
\Gamma^a_{ij} \models \text{PossUnd}_j(j; \alpha, \psi, \varphi) \rightarrow [j; \alpha](\text{Reproach}^T_j(j; \alpha) \land \text{Distress}^T_j(\psi) \land \text{Anger}^T_j(j; \alpha, \psi))
\]

(5.43)

where \(\Gamma^a_i = \{Acд(i, i; \alpha), Det(i; \alpha), \text{NoPar}(i; \alpha), BG(i), \text{DropG}(i), \text{NoDrop}(\alpha)\}\) and \(\Gamma^a_{ij} = \{Acд(i, j; \alpha), Det(j; \alpha), \text{NoPar}(j; \alpha), BG(i), \text{DropG}(i), \text{NoDrop}(\alpha), \text{NoCU}(i, j)\}\) are sets of assumptions.

For example, the last proposition can be read as follows: if agent \(i\) believes that agent \(j\) can possibly undermine subgoal \(\psi\) of agent \(i\) by performing action \(\alpha\), then if agent \(j\) indeed performs action \(\alpha\), a number of emotions will be triggered for agent \(i\), namely reproach for agent \(j\) having performed action \(\alpha\), distress about the undermining of subgoal \(\psi\), and anger about that action leading to this result.

Notice that \(\Gamma^a_i\) and \(\Gamma^a_{ij}\) contain different assumptions. The assumption \(Acд(i, i; \alpha)\), expressing that an agent does not forget the effects of its actions, is not uncommon.
in dynamic logic. Interestingly, adopting this $\text{Acd}(i, i; \alpha)$ results in agent $i$ always being ‘aware’ of the fact that it has performed action $\alpha$. This is because we would then in particular have $B_i[i; \alpha] \text{Done}(i; \alpha) \rightarrow [i; \alpha]B_i \text{Done}(i; \alpha)$. But $B_i[i; \alpha] \text{Done}(i; \alpha)$ is always true (by necessitation on formula (4.44)), so $[i; \alpha]B_i \text{Done}(i; \alpha)$ is also valid. The assumption $\text{Acd}(i, j; \alpha)$ results in agent $i$ always being ‘aware’ of the fact that it has performed action $\alpha$. This is because we would then in particular have $B_i[i; \alpha] \text{Done}(i; \alpha)$.

The assumption $\text{Acd}(i, j; \alpha)$ is slightly stronger and entails that, when agent $j$ has performed action $\alpha$, agent $i$ will be ‘aware’ of this fact. That is, $[j; \alpha]B_i \text{Done}(j; \alpha)$ will be valid. The assumption $\text{Det}(i; \alpha)$ of determinism is needed to ensure that $\alpha$ will indeed succeed in achieving or undermining subgoal $\psi$. The assumptions $\text{NoPar}(i; \alpha)$, $\text{BG}(i)$, $\text{DropG}(i)$, and $\text{NoDrop}(\alpha)$ are needed to ensure that $\alpha$ will not cause goal $\varphi$ to be dropped; indeed, no emotions would be triggered if $\alpha$ causes $\varphi$ to be dropped, because then agent $i$ would not care about subgoal $\psi$ of $\varphi$ anymore. Finally, the assumption $\text{NoCU}(i, j)$ forces ‘admiration’ or ‘reproach’ to be triggered instead of ‘pride’ or ‘shame’.

### 5.8 Putting Everything Together

Let us reconsider Figure 3.1 on page 38. In this chapter and the previous two, we have been concerned with the Appraisal part of this procedural perspective of emotion. The presented formalization of emotion triggers constitutes the process of emotion elicitation which combines percepts and concerns. The output of the process of emotion elicitation is labeled as triggered emotions in Figure 3.1; a definition of what constitutes triggered emotions would finish our formalization of the Appraisal part of emotion. Indeed, this will be the subject of this section.

In order to define triggered emotions in general, it must be possible to quantify over all emotion types distinguished in the OCC model. For this purpose the set EMO-TYPES is defined containing all 28 emotion types used in this dissertation.

**Definition 5.21 (Emotion types)**

The set EMO-TYPES contains all emotion types of the OCC model:

$$\text{EMO-TYPES} = \text{UNDIFFERENTIATED} \cup \text{CONSEQ-BASED} \cup \text{ACTION-BASED} \cup \text{OBJECT-BASED} \cup \text{COMPOUND} \cup \text{PROSPECT-BASED} \cup \text{FORTUNES-OF-OTHERS}$$

where

- \text{UNDIFFERENTIATED} = \{\text{Positive, Negative}\}
- \text{CONSEQ-BASED} = \{\text{Pleased, Displeased, Hope, Fear, Joy, Distress}\}
- \text{ACTION-BASED} = \{\text{Approving, Disapproving, Pride, Shame, Admiration, Reproach}\}
- \text{OBJECT-BASED} = \{\text{Liking, Disliking}\}
- \text{COMPOUND} = \{\text{Gratification, Remorse, Gratitude, Anger}\}
- \text{PROSPECT-BASED} = \{\text{Satisfaction, Fears-confirmed, Relief, Disappointment}\}
- \text{FORTUNES-OF-OTHERS} = \{\text{Happy-for, Resentment, Gloating, Pity}\}
**Emotion** is used as a typical element of emo-types.

In general, formulas representing the triggering of an emotion have the form $\text{Emotion}_i^T(\bar{o})$, where **Emotion** $\in$ emo-types is the type of the emotion in question, $i \in \text{agt}$ is the appraising agent, and $\bar{o}$ is a vector of arguments of the emotion (i.e., the thing towards which the emotion is triggered). For example, $\text{Joy}_{Alice}^T(\text{won lottery})$ can be used to represent the fact that joy is triggered for Alice with respect to winning the lottery. However, each formula of the form $\text{Emotion}_i^T(\bar{o})$ has been defined in Chapter 3 as a macro for the eliciting conditions of an emotion of type **Emotion**, so we cannot quantify over them. For example, the formula $\text{Joy}_{Alice}^T(\text{won lottery})$ does not really “exist”; indeed, it is merely an abbreviation of the formula $B_{Alice,\text{won lottery}} \wedge \neg \text{Prev } B_{Alice,\text{won lottery}} \wedge \text{Des}_{Alice}(\text{won lottery})$. Nevertheless, we can define a set of all possible emotions independent of the macros for emotion triggering, namely by considering all possible type–agent–arguments triples. An example of such a triple would then be $(\text{Joy}, Alice, \text{won lottery})$.

**Definition 5.22 (Possible emotions)**
The set of all possible emotions is defined as the following set of triples:

\[
\text{EMOTION} = (\text{UNDIFFERENTIATED } \times \text{agt} \times (\text{cq } \cup \text{act-agt } \cup \text{obj})) \\
\cup (\text{CONSEQ-BASED } \times \text{agt} \times \text{cq}) \\
\cup (\text{ACTION-BASED } \times \text{agt} \times \text{act-agt}) \\
\cup (\text{OBJECT-BASED } \times \text{agt} \times \text{obj}) \\
\cup (\text{COMPOUND } \times \text{agt} \times (\text{act-agt } \times \text{cq})) \\
\cup (\text{PROSPECT-BASED } \times \text{agt} \times (\text{cq } \times \text{cq})) \\
\cup (\text{FORTUNES-OF-OTHERS } \times \text{agt} \times (\text{cq } \times \text{agt}))
\]  

where $\text{cq } = \mathcal{L}$ is the set of formulas that can represent consequences of events (i.e., all well-formed formulas) and $\text{act-agt} = \{i: \alpha \mid i \in \text{agt}, \alpha \in \text{act}\}$ is the set of actions of agents. $(\text{Emotion}, i, \bar{o})$ is used as a typical element of emotion.

This definition explicates the types of arguments that can be used for different emotion types. For example, the inclusion of “OBJECT-BASED $\times$ AGT $\times$ OBJ” specifies that the object-based emotion types (‘liking’ and ‘disliking’) only take objects as arguments. Recall from Constraint 4.22 that agents are also considered to be objects (i.e., AGT $\subseteq$ OBJ). So $\text{Liking}_{Alice}^T(\text{Bob})$—expressing that for Alice liking towards Bob is triggered—is a well-formed formula iff $(\text{Liking}, Alice, Bob) \in \text{EMOTION}$.

Using the set $\text{EMOTION}$ we can very straightforwardly define the set of all emotions that are triggered in a certain state, as follows.

**Definition 5.23 (Triggered emotions)**
$\text{AllTriggered} : S \rightarrow 2^{\text{EMOTION}}$ is a function indicating per model–state pair which emotions are triggered for all agents. AllTriggered is defined as follows:

\[
\text{AllTriggered}(M, s) = \{ (\text{Emotion}, i, \bar{o}) \in \text{EMOTION} \mid M, s \models \text{Emotion}_i^T(\bar{o}) \}
\]  

(5.46)
This then completes the Appraisal part of Figure 3.1, because AllTriggered(M,s) represents the contents of the data block labeled triggered emotions for each state (M,s).

5.9 Concluding Remarks

With this chapter our formalization of the eliciting conditions of the emotions of the OCC model is finished. The work presented in this chapter constitutes the third and last step of this formalization. We have used the KARO framework to ground all operators and constructs used in the previous two steps (presented in Chapters 3 and 4, respectively). This formalization constitutes a formal specification of the conditions that can trigger certain types of emotion. Nothing is said about how strongly triggered emotions are experienced, if at all. Emotional experience will be the subject of the next chapter.

We have tried to make it as easy as possible to extend the presented formalization, and to indicate clearly how this may be done. In particular, we have introduced one type of concern for appraisal, namely achievement goals, and shown that the specifications of desirability, praiseworthiness, and appealingness can be extended to account for more types of concerns. But just considering achievement goals already yields many interesting properties, as shown in Section 5.7. Moreover, by considering achievement goals and formally specifying the notion of subgoals, we have built a convenient bridge to actual BDI-based agent implementations, which often make use of achievement goals and subgoals. This bridge to agent implementations will be studied in more depth in Chapter 7.

5.10 Proofs

Below are proofs of the propositions presented in Section 5.7.

**Proposition** (5.19)–(5.23). These propositions follow immediately from Constraints 5.10–5.12.

**Proposition** (5.24) and (5.25). These propositions follow immediately from propositions (4.27) and (4.28) and the fact that $\models \neg (B_i \psi \land B_i \neg \psi)$ follows from the seriality of $R_i$.

**Proposition** (5.26) and (5.27). These propositions follow immediately from propositions (3.50) and (3.51) and the fact that $\models \text{CogUnit}_i(i)$ follows from Constraint 5.13.

**Proposition** (5.28)–(5.31). Joy$_i^T(\psi)$ is equivalent to Des$_i(\psi) \land \text{Actual}_i(\psi)$ and Distress$_i^T(\overline{\psi})$ is equivalent to Undes$_i(\overline{\psi}) \land \text{Actual}_i(\overline{\psi})$. By proposition (5.19), $G_i \psi \land \psi \sqsubseteq \varphi$ implies Des$_i(\psi)$ and Undes$_i(\overline{\psi})$. Because Actual$_i(\psi)$ is equivalent to BelUpd$_i(\psi)$, $G_i \psi \land \psi \sqsubseteq \varphi \land \text{BelUpd}_i(\psi)$ implies Joy$_i^T(\psi)$ and $G_i \psi \land \psi \sqsubseteq \varphi \land \text{BelUpd}_i(\overline{\psi})$ implies Distress$_i^T(\overline{\psi})$. Propositions (5.30) and (5.31) follow analogously, by noting that BelUpd$_i(\text{Fut}^+ \psi)$ is equivalent to FutUpd$_i(\psi)$, which implies Prospective$_i(\psi)$.
Proposition (5.32)–(5.35). By noting that the relation \(\text{sub}\) is reflexive it will be clear that \(\vdash \varphi \equiv \varphi\) holds. Then by necessitation \(\vdash B_i(\varphi \equiv \varphi)\) holds for all \(i\), i.e., \(\vdash \text{Confirms}_i(\varphi, \varphi)\). By taken the definition of Satisfaction \(\text{Satisfaction}_i^T(\varphi, \varphi)\) and crossing out the Confirms term we obtain proposition (5.32). Propositions (5.33)–(5.35) follow analogously. Recall that \(\overline{\varphi} = \varphi\), so \(\text{Disconfirms}_i(\overline{\varphi}, \varphi) = B_i(\overline{\varphi} \equiv \varphi)\). \(\square\)

Proposition (5.36). Assume \(M, s \models \text{PossIntend}_i(\alpha, \varphi)\) for arbitrary \((M, s), i, \alpha, \varphi\). Take an arbitrary model–state pair \((M', s')\) such that \(((M, s), (M', s')) \in \mathcal{R}_{i\alpha}\). To prove: \(M', s' \models \text{Pride}_i^T(i, \alpha) \land \text{Joy}_i^T(i, \alpha) \land \text{Gratification}_i^T(i, \alpha, \varphi)\), i.e., \(M', s' \models \text{PerceiveAction}_i(i, \alpha) \land \text{Praisew}_i(i, \alpha) \land \text{CogUnit}_i(i) \land \text{PerceiveConseq}_i(\varphi) \land \text{Des}_i(\varphi) \land \text{Actual}_i(\varphi) \land \text{Past Pride}_i^T(i, \alpha) \land \text{Past Joy}_i^T(i, \alpha) \land \text{PerceiveRelated}_i(i, \alpha, \varphi)\). We can immediately cross out several of the conjuncts: \(\text{CogUnit}_i(i)\) follows directly from Constraint 5.13; by Definition 4.10, \(\text{PerceiveConseq}_i(\varphi)\) follows from \(\text{Actual}_i(\varphi)\); and \(\text{Past Pride}_i^T(i, \alpha)\) and \(\text{Past Joy}_i^T(\varphi)\) will follow automatically because \(\psi \rightarrow \text{Past } \psi\) is a validity. Writing out definitions, we have to prove \(M', s' \models \text{New } B_i, \text{Past Done}_i(i, \alpha) \land \text{New } B_i, \varphi \land \text{New } B_i, \text{Related}_i(i, \alpha, \varphi) \land \text{Des}_i(\varphi) \land \text{Praisew}_i(i, \alpha)\). If \(M, s \models \neg \psi\) and \(M', s' \models \psi\), then \(M', s' \models \text{New } \psi\). So for state \((M, s)\) we have to show:

(i) \(M, s \models \neg B_i, \text{Past Done}_i(i, \alpha)\)
(ii) \(M, s \models \neg B_i, \varphi\)
(iii) \(M, s \models \neg B_i, \text{Related}_i(i, \alpha, \varphi)\)

and for state \((M', s')\) we have to show:

(a) \(M', s' \models B_i, \text{Past Done}_i(i, \alpha)\)
(b) \(M', s' \models B_i, \varphi\)
(c) \(M', s' \models B_i, \text{Related}_i(i, \alpha, \varphi)\)
(d) \(M', s' \models \text{Des}_i(\varphi)\)
(e) \(M', s' \models \text{Praisew}_i(i, \alpha)\)

By proposition (5.20), (e) is implied by the (d) and (c). Because \(\psi \rightarrow \text{Past } \psi\) is valid, so is \(B_i \psi \rightarrow B_i \text{Past } \psi\), which means that it suffices to show for \((M', s')\) that:

(A) \(M', s' \models B_i \text{Done}_i(i, \alpha)\)
(B) \(M', s' \models B_i, \varphi\)
(C) \(M', s' \models B_i, \neg \text{Prev } \varphi\)
(D) \(M', s' \models \text{Des}_i(\varphi)\)

From proposition (4.44), we have that \([i, \alpha] \text{Done}_i(i, \alpha)\) is valid. Then by necessitation \(B_i[i, \alpha] \text{Done}_i(i, \alpha)\) is also valid, and by the assumption that \(B_i[i, \alpha] \psi \rightarrow [i, \alpha] B_i \psi, [i, \alpha] B_i \text{Done}_i(i, \alpha)\) is also valid. But if \(M, s \models [i, \alpha] B_i \text{Done}_i(i, \alpha)\), then \(M', s' \models B_i \text{Done}_i(i, \alpha)\), which proves (A). \(M, s \models \text{PossIntend}_i(\alpha, \varphi)\) implies \(M, s \models B_i(i, \alpha) \varphi\), which implies \(M, s \models B_i[i, \alpha] \varphi\) (because \(\alpha\) was assumed to be deterministic), which implies \(M, s \models [i, \alpha] B_i \varphi\) (because \(\alpha\) was assumed to be accordant). But then \(M', s' \models B_i \varphi\), which proves (B). It is easy to verify that \(\neg \varphi \rightarrow [i, \alpha] \neg \text{Prev } \varphi\) is a validity; then \(B_i \neg \varphi \rightarrow B_i[i, \alpha] \neg \text{Prev } \varphi\) is also valid. Because \(\text{PossIntend}_i(\alpha, \varphi)\) implies \(B_i \neg \varphi, M, s \models B_i[i, \alpha] \neg \text{Prev } \varphi\). But then \(M, s \models [i, \alpha] B_i, \neg \text{Prev } \varphi\) and \(M', s' \models B_i, \neg \text{Prev } \varphi\), which proves (C). \(\text{PossIntend}_i(\alpha, \varphi)\) implies \(B_i G_i \varphi\), which implies \(G_i \varphi\) (because it was assumed
that $B,G,\varphi \rightarrow G,\varphi)$. So because $M,s \models G,\varphi$, $M',s' \models \text{Prev } G,\varphi$. Furthermore, it was assumed that $\alpha \neq \text{drop}(\varphi)$ and that $\neg(\text{Done}(i:a) \land \text{Done}(i:\text{drop}(\varphi)))$, so the fact that $M',s' \models \text{Done}(i:a)$ implies $M',s' \models \neg \text{Done}(i:\text{drop}(\varphi))$. But it was also assumed that $\text{Prev } G,\varphi \land \neg G,\varphi \rightarrow \text{Done}(i:\text{drop}(\varphi))$, so it must be that $M',s' \models G,\varphi$, which implies $M',s' \models \text{Des},(\varphi)$, which proves (D). From proposition (4.48) we have that $\text{Past } \text{Done}(i:a) \rightarrow [i:a]_\bot$, i.e., $\langle i:a \rangle \top \rightarrow \neg \text{Past } \text{Done}(i:a)$. Then $B,(i:a)\top \rightarrow B,\neg \text{Past } \text{Done}(i:a)$ is also valid. The antecedent is implied by $\text{PossIntend} ,(a,\varphi)$, so $M,s \models B,\neg \text{Past } \text{Done}(i:a)$. But by seriality of $R$, $M,s \models \neg B,\text{Past } \text{Done}(i:a)$, which proves (i). $\text{PossIntend} ,(a,\varphi)$ implies $B,\neg \varphi$, so by seriality of $R$, $M,s \models \neg B,\varphi$, which proves (ii). It is easy to verify that $\text{Past } \text{(Done}(i:a) \land \text{New } \varphi) \rightarrow \text{Past } \text{Done}(i:a)$ is a validity. But then by necessitation and contraposition $\neg B,\text{Past } \text{Done}(i:a) \rightarrow \neg B,\text{Past } \text{Done}(i:a) \land \text{New } \varphi$ is also a validity. So (i) implies (iii), which proves (iii). We can now conclude that indeed $M',s' \models \text{Pride}^T ,(i:a)\land \text{Joy}^T ,(\varphi)\land \text{Gratification}^T ,(i:a,\varphi)$. Because $(M',s')$ and $(M,s)$ were arbitrary, $\text{PossIntend} ,((\pi,\varphi) \rightarrow [i:a]((\text{Pride}^T ,(i:a) \land \text{Joy}^T ,(\varphi)\land \text{Gratification}^T ,(i:a,\varphi))))$ is valid.

Proposition (5.39). $\text{PossAch} ,((\pi,\varphi) \rightarrow [i:a]((\text{Pride}^T ,(i:a) \land \text{Joy}^T ,(\varphi)\land \text{Gratification}^T ,(i:a,\varphi))))$ is an abbreviation of $B,(\text{PracPoss} ,(\pi,\varphi) \land \text{Gratification}^T ,(i:a),\varphi)$. But sub is reflexive so $\varphi \sqsubseteq \varphi$ is valid. Moreover, $\overline{\varphi}$ is stronger than $\neg \varphi$. So $\text{PossAch} ,((\pi,\varphi) \rightarrow [i:a]((i:a) \land \neg \varphi)) \rightarrow \text{PossIntend} ,((\pi,\varphi) \rightarrow \text{PossIntend} ,(\pi,\varphi))$ is valid.

Proposition (5.40)–(5.43). The proofs of these propositions are largely the same as the proof of proposition (5.36) above. For example, assuming $\text{PossAch} ,((\pi,\varphi) \rightarrow [i:a]((\text{Pride}^T ,(i:a) \land \text{Joy}^T ,(\varphi)\land \text{Gratification}^T ,(i:a,\varphi))))$ or $\text{PossUnd} ,((\pi,\varphi) \rightarrow [i:a]((\text{Pride}^T ,(i:a) \land \text{Joy}^T ,(\varphi)\land \text{Gratification}^T ,(i:a,\varphi))))$ still implies point (ii), because we then have that $M,s \models B,\overline{\psi}$, that $\overline{\psi} \rightarrow \neg \psi$ is valid, that $R_i$ is serial, and thus that $M,s \models B,\psi$. To prove proposition (5.41), points (d) and (e) become $M',s' \models \text{Undes} ,(\psi) \land \text{Undes} ,(\varphi)$ and $M',s' \models \text{Blame} ,((i:a))$, respectively. $\text{PossUnd} ,((\pi,\varphi) \rightarrow [i:a]((\text{Pride}^T ,(i:a) \land \text{Joy}^T ,(\varphi)\land \text{Gratification}^T ,(i:a,\varphi))))$ implies $\text{Undes} ,(\overline{\psi}) = \text{Undes} ,(\psi)$ and then by proposition (5.21) implies $\text{Blame} ,((i:a))$. Finally, the proofs of propositions (5.42) and (5.43) only differ by using the assumption that $\neg \text{CogUnit} ,(j)$ for $i \neq j$. □
In Chapters 3, 4, and 5 we have presented a formalization of the eliciting conditions of emotions. This amounts to a specification of when certain types of emotion are triggered. However, it does not say anything about which emotions are actually experienced by an agent. An emotion which is triggered may have such a low ‘potential’ that it is not experienced. Also, an emotion may in fact be experienced even though currently its eliciting conditions do not hold. This may be because the emotion has recently been triggered with such a high ‘potential’ that it is still being felt. For example, suppose Alice spills her tea over her mother-in-law. The moment Alice realizes what she has done, the eliciting conditions for shame will hold, because she perceives her own blameworthy action. However, suppose also that an hour ago Alice learned that she is pregnant, something which she had enormously desired. Now Alice may be so elated that she does not feel ashamed, but instead excitedly continues to bring the good news to her mother-in-law. So in this case, the joy persists over some time, while the shame may never surface.

In this chapter we will study how emotional experience can be defined in the logical framework introduced in the previous chapters. This chapter is outline as follows. First, in Section 6.1 we will revisit our procedural view of emotion (see Figure 3.1) in order to explain in more detail what aspect of emotion is being formalized here. It will be shown that one important concept needed to define emotional experience has not yet been formalized. This concept is that of overall felt intensity, which is then studied in Section 6.2. The concept of emotional experience is formally defined in our logical framework in Section 6.3. We will move beyond the concept of emotion types in Section 6.4 and look at how specific instances of emotions can be represented in our formalism. For example, we have previously seen that an emotion of the type ‘fear’ can be triggered, but here we will study if and when such an emotion can be called ‘worried’, ‘anxious’, or ‘terrified’, to name a few tokens of the ‘fear’ type (see the box on page 21). All the work of this chapter will then be formally integrated into the KARO framework in Section 6.5. Finally, this chapter is concluded in Section 6.6 with some thoughts on related work and on the practical use of distinguishing between emotion elicitation and emotional experience.
6.1 The Procedural View of Emotion Revisited

Before going into formal details, let us first revisit the procedural view of emotion in order to get a feeling for how the work of this chapter fits in the bigger picture. For ease of reference the illustration of this procedural view is repeated in Figure 6.1. In Chapters 3, 4, and 5 we have been concerned with formalizing the Appraisal part of Figure 6.1. In this chapter, then, we move on to the Experience part.

According to our procedural view of emotion, emotional experience is modeled by the process of intensity assignment and its resulting output of experienced emotions. The process of intensity assignment is informed by triggered emotions, individual parameters (the emotional ‘characters’ of agents), and the mood of an agent. Recall that the contents of triggered emotions were formally defined in Definition 5.23. Individual parameters and mood will not be discussed in as much detail; that is, they will be treated as informational and not as central to the logical structure of emotions.

Let us recall Definition 3.1 on page 39. There emotional experience was (informally) defined in terms of emotion triggering and overall felt intensity. Now emotion triggering has been formalized in Chapters 3, 4, and 5. In order to be able to formally define emotional experience, we need a formalization of the concept of ‘overall felt intensity’, which is exactly what we will do next in Section 6.2. Indeed, this formal-
ization will also constitute a specification of the process of intensity assignment. The output of this process (i.e., experienced emotions) is then formalized in Section 6.3.

6.2 Overall Felt Intensity

In this section we investigate the structure of the concept of ‘overall felt intensity’, which is the magnitude at which an emotion is felt. ‘Overall felt intensity’ will be assumed to be unidimensional and as such expressible with a non-negative real number. Note that emotional experience itself is not assumed to be unidimensional; that is, there may be many parameters (aspects) through which the experience of an emotion manifests itself. For example, the tendency to flee out of fear or the inability to concentrate out of infatuation can also be experienced with a certain strength. Nevertheless, the parameter called ‘overall felt intensity’ is taken to be an indicator for all parameters of emotional experience, whatever they may be. So the ‘overall felt intensity’ of an emotion is assumed to be positive if and only if any of the parameters of emotional experience is felt.

For the moment we will leave any other parameters of emotional experience aside and only treat the ‘overall felt intensity’. In Section 6.3.2 we will have more to say about other parameters and how ‘overall felt intensity’ fits among them.

6.2.1 Emotion Episode

One of the most important differences between emotional experience and emotion elicitation is that emotion elicitation is a property of a single instance in time, whereas emotional experience is something that manifests itself over time. When the eliciting conditions of an emotion hold, we say that the emotion in question is triggered. But what happens next? As Figure 1.1 on page 6 had already illustrated, emotions have a duration; they endure for some time. The time during which an emotion is experienced is usually called an “emotion episode” [Frijda, 1987].

Intuitively, the course of an emotion over time can be illustrated as in Figure 6.2. When the triggering conditions of the emotion in question are satisfied, the intensity of this emotion may rise to some peak value, after which it starts to subside. The decrease in emotional intensity is not necessarily smooth though. Refocusing on the event, action, or object that gave rise to the emotion may cause it to recur. Finally, after some time the emotion may die off completely, although it may still resurface yet later. For example, let us recall the example from the introduction of this chapter. The pregnant Alice may later in the day watch some television, which completely distracts her mind. After switching off the television, however, she may remember the good news and become excited once again.

It should be noted that the initial rise of intensity does not have to coincide with the moment at which the triggering conditions of the emotion hold. For example, Alice may feel ashamed for spilling the tea when later (e.g., an hour after Alice perceived her own blameworthy action) her mother-in-law tells her about how it ruined her white scarf.
In Figure 6.2 we can already see several factors related to emotional intensity: the time at which the emotion was triggered, the peak intensity, the rate at which the intensity decreases over time, the total duration of the emotion episode, and the possibility of recurrences. A proper formalization of ‘overall felt intensity’ must take all these factors into account.

Ultimately, the intensity of an emotion (the y-axis in Figure 6.2) is a function of time (the x-axis in Figure 6.2). Therefore, in order to be able to be more formal about all the mentioned aspects of emotional intensity, a representation of time is needed. Let us then introduce a notion of time into our formal framework, as follows. Let a function \( \text{Time} : S \rightarrow \mathbb{R}_{\geq 0} \) associate a time value with each model–state pair, where time is represented as a non-negative real number. \( \text{Time} \) is used (only) to calculate actual emotion intensity values; therefore, it is assumed the same time is assigned to each state \( s \) of a belief model \( M \).

**Constraint 6.1 (All states in a belief model have the same time stamp)**

For all model–state pairs \((M, s) \in S\), the function \( \text{Time} \) is constrained as follows:

\[
\forall s' \in S : \text{Time}(M, s) = \text{Time}(M, s')
\]  
(6.1)

where \( M = \langle S, R, V \rangle \).

In the following, we express the time at a model–state pair \((M, s)\) as \( \text{Time}_M \) for clarity of presentation, because the belief state \( s \) is not relevant to the function \( \text{Time} \). It is also assumed that all actions take time, which ensures that time can only ‘increase’ when going ‘forward’ (in the sense of the action-accessibility relations \( R \)) through an action model. This is expressed by the following constraint.

**Constraint 6.2 (Actions take time)**

\[
\forall((M, s), (M', s')) \in (\bigcup R) : \text{Time}_M < \text{Time}_{M'}.
\]

Note that Constraint 5.7 implies that all action-accessibility relations are irreflexive, so it is guaranteed that \( M \) and \( M' \) in Constraint 6.2 are distinct belief models. The strict inequality in Constraint 6.2 is therefore not a problem.
Now we need a way to produce intensity functions. For this we introduce the function \( \text{IntFun} : \text{emotion} \times S \rightarrow I \), where \( I \) is the set of monotonically decreasing functions of type \( \mathbb{R} \rightarrow \mathbb{R} \). What \( \text{IntFun} \) does is associate with each triggered emotion a function of time. With the danger of getting ahead of ourselves, let us briefly explain the idea of assigning intensity functions to triggered emotions (as opposed to directly assigning intensity values), and why we demand monotonically decreasing intensity functions. By assigning intensity functions to emotions, we can say that ‘usually’, performing an action does not change the way emotion intensities behave. That is, for ‘most’ \( \alpha \), \(((M, s), (M', s')) \in R_{\text{Int}} \) implies \( \text{IntFun}(\text{Emotion}, i, \bar{o})(M', s') = \text{IntFun}(\text{Emotion}, i, \bar{o})(M, s) \). For certain actions we can then put useful constraints on \( \text{IntFun} \) such that these actions can be said to influence emotions by changing the intensity functions that are assigned to these emotions. Indeed, this is exactly what we will do in Section 7.2 on action tendency.

The idea that actions ‘usually’ do not change the intensity functions assigned to emotions also elucidates the reason why we demand monotonically decreasing intensity functions: all things being equal, emotions should subside by themselves. This is modeled by having monotonically decreasing functions of time. To have an emotion recur (i.e., have its intensity rise again), a special action must be performed which replaces the intensity function of the emotion in question with one producing higher intensity values. For example, such an action may be Alice refocusing on the good news of being pregnant after having switched off her television. But of course, this new intensity function must also be monotonically decreasing, allowing the emotion in question to subside again, all things being equal.

Again, the benefits of this construction will only really show in Section 7.2. But given that \( \text{IntFun} \) produces intensity functions of time, we can define the function \( \text{OFI} \), returning the ‘overall felt intensity’ of an emotion, as follows.

**Definition 6.3 (Overall felt intensity)**

Let \( \text{IntFun} : \text{emotion} \times S \rightarrow I \) be a function assigning monotonically decreasing intensity functions to emotions per state. Then the ‘overall felt intensity’ (\( \text{OFI} \)) of an emotion \( (\text{Emotion}, i, \bar{o}) \in \text{emotion in state } (M, s) \in S \) is calculated as follows:

\[
\text{OFI}(\text{Emotion}, i, \bar{o})(M, s) = \max(0, I(\text{Time}_M)) \quad (6.2)
\]

where \( I = \text{IntFun}(\text{Emotion}, i, \bar{o})(M, s) \).

Note that it is allowed for an intensity function to produce negative values; the ‘overall felt intensity’ as defined above ensures that the final intensity value will be at least zero. The form of the intensity function \( I \) produced by \( \text{IntFun} \) will be the subject of the remainder of this subsection.

**Intensity Functions of Time**

Let us first examine the simplest kind of intensity function. An intensity function is a function of time and depends on (at least) three parameters, here denoted as \( \tau, i, \) and \( \bar{o} \):
• $\tau \geq 0$ is the *triggering time*, i.e., the time point at which the emotion in question was triggered. If emotion $\text{Emotion}_i(\delta)$ was triggered in state $(M, s)$ (i.e., $M, s \models \text{Emotion}_i(\delta)$) then $\tau = \text{Time}_{M,s}$;

• $\iota \geq 0$ is the *initial intensity* associated with the emotion in question. It is assumed that any other affective processes such as mood have already been taken into account in determining $\iota$;

• $\delta > 0$ is the *duration* time of the emotion in question.

All other things being equal, the intensity of an emotion should drop from $\iota$ to zero within time $\delta$. An intensity function $I$ should thus satisfy at least the following two criteria:

1. $I(\tau) = \iota$: at the time $\tau$ of triggering, the intensity should be equal to the initial intensity $\iota$;

2. $I(\tau + \delta) = 0$: the intensity should be zero when the duration $\delta$ of the emotion episode is up.\(^1\)

It is quite trivial to find a function satisfying these two criteria. The simplest linear function satisfying these two constraints is:

$$I_{\iota,\tau,\delta}(x) = -\frac{\iota}{\delta}(x - \tau - \delta)$$

(6.3)

However, if one wants to consider a kind of function that has a gentle curve, it would be reasonable to consider an inverse sigmoid function. Inverse sigmoid functions have the following form:

$$\hat{\sigma}(x) = \frac{1}{1 + \exp(x)}$$

(6.4)

Of course, there are several parameters that have to be set to give the inverse sigmoid function the shape desired for modeling a particular emotion episode. Specifically, it must depend on the initial intensity $\iota$, the time at which the emotion was triggered $\tau$, and the duration $\delta$, as specified above. Moreover, a fall-off speed can be set, here denoted as $\lambda$:

• $\lambda > 0$ is the rate determining how steeply the intensity drops around halfway the sigmoid function. ($\lambda$ can be used to adjust this slope by multiplication, as in $\hat{\sigma}(\lambda x)$; without $\lambda$, $\hat{\sigma}(x)$ has a slope of $-\frac{1}{4}$ at $x = 0$.)

These parameters are then used in the inverse sigmoid function in the following manner:

$$\iota \cdot \hat{\sigma}(\lambda \cdot (x - \tau - \frac{1}{2}\delta))$$

where $x$ is the variable of the function (the time axis).

\(^1\)Even stronger, it may be demanded that $I(x) \leq 0$ for all $x \geq \tau + \delta$; that is, $\delta$ correctly specifies how long an emotion’s intensity can be positive. All intensity function discussed next will satisfy this stronger criterium as well.
The formula above does not satisfy the two aforementioned criteria, however. In particular, it should be scaled such that the intensity is equal to the initial intensity at the time of triggering (i.e., such that \( I(\tau) = \iota \)). Moreover, some value must be subtracted such that the intensity is zero when the duration of the emotion episode is up (i.e., such that \( I(\tau + \delta) = 0 \)). This means that we have to find \( a \) and \( b \) such that our function is \( I_{\iota,\tau,\delta,\lambda}(x) = \iota \cdot \hat{\sigma}(\lambda \cdot (x - \tau - \frac{1}{2} \delta)) \cdot a - b \) and the two mentioned criteria are satisfied. The desired values for \( a \) and \( b \) are 
\[
\begin{align*}
  a &= \frac{1}{\hat{\sigma}(-\lambda \cdot \frac{1}{2} \delta) - \hat{\sigma}(\lambda \cdot \frac{1}{2} \delta)} \\
  b &= \iota \cdot \hat{\sigma}(\lambda \cdot \frac{1}{2} \delta) \cdot a
\end{align*}
\]
Filling in \( a \) and \( b \) then results in the following sigmoid intensity function:
\[
I_{\iota,\tau,\delta,\lambda}(x) = \iota \cdot \frac{\hat{\sigma}(\lambda \cdot (x - \tau - \frac{1}{2} \delta)) - \hat{\sigma}(\lambda \cdot \frac{1}{2} \delta)}{\hat{\sigma}(-\lambda \cdot \frac{1}{2} \delta) - \hat{\sigma}(\lambda \cdot \frac{1}{2} \delta)}
\]
To obtain a normal sigmoid curve for \( I \), the fall-off rate \( \lambda \) should be set at around 1. Choosing larger values for \( \lambda \) will make the drop around \( x = \tau + \frac{1}{2} \delta \) steeper. Smaller values for \( \lambda \) (i.e., closer to zero) will give a more linear shape. It is interesting to note that this inverse sigmoid reduces to the linear function above (i.e., formula (6.3)) as \( \lambda \) goes to zero:
\[
\lim_{\lambda \to 0} I_{\iota,\tau,\delta,\lambda}(x) = -\frac{\iota}{\delta} (x - \tau - \delta)
\]
Also note that \( I_{\iota,\tau,\delta,\lambda}(\tau + \frac{1}{2} \delta) = \frac{1}{2} \iota \); that is, after half the duration, the intensity has halved regardless of the value for \( \lambda \).

To conclude, a reasonable default choice for intensity functions would be the inverse sigmoid specified above; that is, \( \text{IntFun}([\text{Emotion}, \iota, \delta], (M, s)) = I_{\iota,\tau,\delta,\lambda} \). But of course this raises the question of what the values of \( \iota, \tau, \delta, \lambda \) should be. One is trivial: the triggering time is \( \tau = \text{Time}_M \). The values for the duration \( \delta \) and fall-off rate \( \lambda \) will greatly depend on the experiencing agent’s emotional ‘character’, but we will not go into these any further here. Arguably the most interesting parameter here is the initial intensity \( \iota \), the calculation of which will be the subject of the remainder of this section.

### 6.2.2 Potential, Threshold, and Initial Intensity

In the OCC model, quantitative aspects of emotions are described in terms of **potentials**, **thresholds**, and **intensities**. For each of the 22 emotion types identified in the OCC model, a list of variables is provided (see also Appendix A under “variables affecting intensity” in each box). For example, for the emotions of type ‘gratitude’, the variables affecting intensity are given as (1) the degree of judged praiseworthiness, (2) deviations of the agent’s action from person/role-based expectations (i.e., unexpectedness), and (3) the degree to which the event is desirable. However, despite what the phrase “variables affecting intensity” suggests, the idea is that these variables determine the **potential** of an emotion. But the potential is again used to determine the intensity of an emotion, so these variables do indeed **affect** intensity, albeit not as directly as might be expected at first glance.

With respect to the variables affecting the intensity of an emotion, the idea is that the weighted sum of these variables equals the emotion’s potential. The intensity of an emotion is then defined as its potential minus its threshold, or zero if the threshold
is greater than the potential. The values of thresholds of emotions are not specified by OCC, but they are hinted to depend on a more diffuse affective state, also called ‘mood’. For example, if an agent is in a ‘good mood’, the thresholds of the negative emotions are increased, causing a lower (or zero) intensity to be associated with a negative emotion, if one is triggered. Emotions that are assigned a nonzero intensity may in turn influence the mood of an agent, entangling the dynamics of short-term emotions and a long-term mood. This entangling is also apparent in Figure 6.1, which depicts experienced emotions, mood, and intensity assignment in a loop.

The main idea behind separating emotion potentials and intensities is to be able to reason about why some emotions have no effect on an agent (i.e., the agent does not experience the emotion and it does not influence its behavior) even though their eliciting conditions hold. This happens when the eliciting conditions of an emotion hold, but its potential is calculated to be below its threshold. In this case we can still recognize that an emotion has been triggered for some agent, even though the emotion in question does not affect the agent. This may happen when, for example, the mood of the agent was ‘too good’ for him to be affected by shame, even though the agent is aware it performed a blameworthy action.

In the previous subsection we have used $i$ to represent the initial intensity assigned to a triggered emotion. Given OCC’s specification that intensity is calculated as potential minus threshold, we arrive at the following definition for the calculation of $i$.

**Definition 6.4 (Calculation of initial intensity)**

Let there be a state $(M, s) \in S$ and a triggered emotion $(\text{Emotion}, i, \bar{o}) \in \text{AllTriggered}(M, s)$, i.e., $M, s \models \text{Emotion}^T(\bar{o})$ holds (see Definition 5.23). Then the initial intensity $i$ of the triggered emotion is calculated as:

$$i = \max(0, \text{Potential}(\text{Emotion}, i, \bar{o})(M, s) - \text{Threshold}(\text{Emotion}, i, \bar{o})(M, s))$$  (6.7)

The initial intensity $i$ is thus calculated in terms of a function $\text{Potential}$ indicating the emotion’s potential and a function $\text{Threshold}$ indicating the emotion’s threshold. This $i$ is then used in the intensity function assigned to emotion $(\text{Emotion}, i, \bar{o})$, as described in Section 6.2.1.

In order to determine the potential of a triggered emotion, the OCC model specifies a number of variables for each type of emotion. Without specifying yet which emotions type has which variables, we can already capture the idea that potential is defined as their weighted sum. Assume for the moment that there exists a function $\text{Vars}$ which indicates for each type of emotion the variables belonging to that emotion type. For example, the variables for ‘gratitude’ are praiseworthiness, expectation deviation, and desirability, so $\text{Vars}(\text{Gratitude}) = \{\text{praiseworthiness}, \text{expectation deviation}, \text{desirability}\}$, where praiseworthiness, expectation deviation, and desirability are functions of emotions $(\text{emotion})$ and states $(S)$. Then the function $\text{Potential}$ is defined as follows.

**Definition 6.5 (Calculation of potential)**

For every state $(M, s) \in S$ and triggered emotion $(\text{Emotion}, i, \bar{o}) \in \text{AllTriggered}(M, s)$, the
emotion’s potential is calculated as:

\[
\text{Potential}(\text{Emotion}, i, \bar{o})(M, s) = \sum_{\text{var} \in \text{Vars}(\text{Emotion})} w_{\text{var}} \cdot \text{var}(\text{Emotion}, i, \bar{o})(M, s)
\]  

(6.8)

Each variable \(\text{var}\) is weighted by the variable-dependent weight \(w_{\text{var}} \geq 0\). These weights are dependent on the emotional ‘character’ of the experiencing agent, so we will not go into these any further in this dissertation. The variables in \(\text{Vars}\) are the subject of the entirety of Section 6.2.3.

Before moving on to our investigation of the variables affecting intensity, however, let us consider the Threshold function used in Definition 6.4. The reason for using thresholds is that the variables that determine the potential of a triggered emotion are quite ‘unemotional’; that is, they do not take into account the current affective state of the experiencing agent. Although the OCC model is not very specific about how emotion thresholds are calculated, a very reasonable choice would be to make them depend on mood. Recall from Chapter 1, in particular Figure 1.1 on page 6, that the affective phenomenon one level beyond “emotion” in terms of duration was identified as “mood.” Mood is a long-term affective state that is more diffuse than emotion in the sense that mood is not directed at anything in particular, whereas an emotion is always relative to an event, action, or object. What makes mood suitable for use as a threshold for emotional experience is that mood represents a sort of ‘average’ of all emotions recently experienced by an agent. Thus by using mood for thresholds, a sort of temporal consistency of emotional experience can be attained. For example, a ‘good mood’ then ensures that negative emotions are less likely to be experienced, which means that on average more positive emotions are experienced, thereby maintaining the ‘good mood’. It would then take several strong negative emotions or many moderate negative emotions in order to tip the balance to a ‘bad mood’.

The function Threshold is thus defined in terms of mood as follows.

**Definition 6.6 (Calculation of threshold)**
For every state \((M, s) \in S\) and triggered emotion \((\text{Emotion}, i, \bar{o}) \in \text{AllTriggered}(M, s)\), the emotion’s threshold is calculated as:

\[
\text{Threshold}(\text{Emotion}, i, \bar{o})(M, s) = \begin{cases} 
\text{Mood}(i)(M, s) & \text{if } \text{Emotion} \in \text{emo-types}^- \\
-Mood(i)(M, s) & \text{if } \text{Emotion} \in \text{emo-types}^+
\end{cases}
\]  

(6.9)

where \(\text{Mood}\) is a function indicating per agent and model–state pair the overall mood of the agent in that state, as a positive (‘good mood’) or negative (‘bad mood’) real number, or zero (‘neutral mood’); and where

\[
\text{emo-types}^+ = \{\text{Positive, Pleased, Approving, Liking, Hope, Joy, Pride, Admiration, Gratification, Gratitude, Satisfaction, Relief, Happy-for, Gloating}\}
\]  

(6.10)
EMOTION EXPERIENCE

\[ \text{emo-types}^- = \{\text{Negative, Displeased, Disapproving, Disliking,} \]
\[ \text{Fear, Distress, Shame, Reproach,} \]
\[ \text{Remorse, Anger,} \]
\[ \text{Fears-confirmed, Disappointment, Resentment, Pity} \} \]

Here \text{Mood} is thus assumed to return a real number. Like emotional intensity, mood may be multidimensional in humans; nevertheless, like ‘overall felt intensity’, it is assumed that a unidimensional ‘overall mood’ can always be determined. This ‘overall mood’ is then what the function \text{Mood} expresses.

Recall from Definition 6.4 that the threshold is subtracted from the potential. Consequently, the mood must be negated for positive emotion types in order to make the threshold enforce the potential in the case where the type of the emotion in question is congruent with the sign of the mood, and diminish the potential otherwise. To this end the emotion types are split up into \text{emo-types}^+ and \text{emo-types}^- such that \text{emo-types} = \text{emo-types}^+ \cup \text{emo-types}^-, as expected from Definition 5.21.

The usage of mood for determining initial intensities explains the role of the box labeled as \text{mood} in Figure 6.1, where it is shown to influence the process of \text{intensity assignment}. We speculate that mood is calculated based on the intensities of recently experienced emotions, and that mood is self-stabilizing (i.e., returns to zero over time if no emotions are triggered). Because our focus is on emotions in this dissertation, we will not go into the workings of mood any further. Nevertheless, the structure, dynamics, and mutual influence of mood with emotions are fascinating topics for future research.

6.2.3 Variables Affecting Intensity

In this subsection we investigate in more detail the structure of the variables affecting intensity, as used in Definition 6.5. It should be noted beforehand that in personal communication, Ortony [2009] has admitted that their classification of variables affecting intensity was rather speculative and has never been properly validated. Still, we believe that the presented classification has considerable face value and would be useful as a guide to computer scientists wishing to implement emotions and being able to calculate their intensities. Having said that, we will proceed with investigating how the variables affecting intensity, as identified and described in the OCC model, can be formalized and incorporated into our framework.

In total, the OCC model identifies 25 variables affecting intensity. First we will deal with the \text{structure} of the organization of these variables. After that we will (briefly) describe each variable and speculate on how they may be calculated.

The Structure of Variables Affecting Intensity

In the OCC model, for each emotion type between one and four variables affecting intensity are listed. There is considerable overlap though, as most of these variables affect the intensity of several emotion types. For example, the variable \text{desirability} affects all positive event-based emotion types. In total, 21 distinct variables are used in the emotion specifications of the OCC model. It should be noted that all
emotion specifications of the OCC model are summarized in Appendix A, including the variables affecting intensity for each emotion type.

One distinction regarding the variables affecting intensity that is made in the OCC model is between local and global variables. The aforementioned number of 21 variables includes only the local variables. They are called local because these variables do not affect the intensities of all emotion types. Only four variables are identified as global variables, namely sense of reality, psychological proximity, unexpectedness, and physiological arousal. Indeed, these variables are said to affect the intensities of all emotion types. Although the four global variables are not mentioned in Appendix A, we include them in our descriptions of the variables affecting intensity below. So in total the OCC model identifies 25 variables affecting intensity.

There are important regularities regarding which variable affects the intensity of which type of emotion. These regularities are illustrated in Figure 6.3. This figure should be read as follows. For each emotion type (e.g., hope, shame, anger, etc.), in order to determine which variables affect the intensity of emotions of that type, simply follow the hierarchy up from that emotion type and collect all variables (written on the lines) encountered on the way. So for example, the variables affecting ‘happy-for’ are liking, desirability for other, deservingness, desirability, physiological arousal, unexpectedness, psychological proximity, and sense of reality. Note that this is a superset of the variables affecting the intensity of ‘joy’. This is in line with the previous chapters where ‘happy-for’ was defined as a specialization of ‘joy’. Also note that the intensity of every emotion type is affected by sense of reality, psychological proximity, unexpectedness, and physiological arousal. Indeed, these are the four global variables.

We should note that on page 69 of OCC’s book [Ortony et al., 1988], there is also a figure illustrating the variables affecting intensity. For completeness, we have included this figure in Appendix A; see Figure A.2 on page 189. The reason that we present a new figure is because a strict reading of Figure A.2 does not lead to the same assignment of variables to emotion types as the assignment of variables according to the specifications given in the text of [Ortony et al., 1988]. Therefore we have redrawn the figure in order to obtain an unambiguous inheritance-based hierarchy, resulting in Figure 6.3. The main difference is in the repositioning of the emotion types ‘satisfaction’, ‘fears-confirmed’, ‘relief’, and ‘disappointment’.

It may also be interesting to compare Figure 6.3 to Figure 2.1, i.e., the inheritance-based view of the eliciting conditions of emotions. Again we see that ‘satisfaction’, ‘fears-confirmed’, ‘relief’, and ‘disappointment’ have been moved from under ‘joy’ and ‘distress’ to a branch of their own. The precise reason for this move will be clarified below in the description of the variables affecting intensity.

Even before going into what these variables actually express, we can already formally capture their structure as illustrated in Figure 6.3. First, we specify that all these variables affecting intensity will be treated as functions and are collected in the set \( \text{vars} \).
Figure 6.3: A view of the emotions of the OCC model based on variables affecting intensity. The upper figure features the positive emotion types, whereas the lower figure features the negative emotion types. How this figure should be read is explained in the text.
\[
\begin{align*}
\text{Vars(Positive)} &= \{\text{sense_of_reality}, \text{proximity}, \text{unexpectedness}, \text{arousal}\} \\
\text{Vars(Pleased)} &= \text{Vars(Positive)} \cup \{\text{desirability}\} \\
\text{Vars(Approving)} &= \text{Vars(Positive)} \cup \{\text{praiseworthiness}, \text{expectation\_deviation}\} \\
\text{Vars(Hope)} &= \text{Vars(Pleased)} \cup \{\text{likelihood}\} \\
\text{Vars(Joy)} &= \text{Vars(Pleased)} \\
\text{Vars(Pride)} &= \text{Vars(Approving)} \cup \{\text{strength\_of\_unit}\} \\
\text{Vars(Admiration)} &= \text{Vars(Approving)} \\
\text{Vars(Liking)} &= \text{Vars(Positive)} \cup \{\text{appealingness}, \text{familiarity}\} \\
\text{Vars(Satisfaction)} &= \text{Vars(Positive)} \cup \{\text{realization}, \text{attendant\_hope, effort\_to\_attain}\} \\
\text{Vars(Relief)} &= \text{Vars(Positive)} \cup \{\text{realization}, \text{attendant\_fear, effort\_to\_prevent}\} \\
\text{Vars(Gratification)} &= \text{Vars(Joy)} \cup \text{Vars(Pride)} \\
\text{Vars(Gratitude)} &= \text{Vars(Joy)} \cup \text{Vars(Admiration)} \\
\text{Vars(Happy-for)} &= \text{Vars(Joy)} \cup \{\text{deservingness, des\_for\_other, liking}\} \\
\text{Vars(Gloating)} &= \text{Vars(Joy)} \cup \{\text{deservingness, undes\_for\_other, disliking}\} \\
\text{Vars(Negative)} &= \{\text{sense_of_reality}, \text{proximity}, \text{unexpectedness}, \text{arousal}\} \\
\text{Vars(Displeased)} &= \text{Vars(Negative)} \cup \{\text{undesirability}\} \\
\text{Vars(Disapproving)} &= \text{Vars(Negative)} \cup \{\text{blameworthiness}, \text{expectation\_deviation}\} \\
\text{Vars(Fear)} &= \text{Vars(Displeased)} \cup \{\text{likelihood}\} \\
\text{Vars(Distress)} &= \text{Vars(Disapproving)} \\
\text{Vars(Shame)} &= \text{Vars(Disapproving)} \cup \{\text{strength\_of\_unit}\} \\
\text{Vars(Reproach)} &= \text{Vars(Disapproving)} \\
\text{Vars(Disliking)} &= \text{Vars(Negative)} \cup \{\text{unappealingness, familiarity}\} \\
\text{Vars(Disappointment)} &= \text{Vars(Negative)} \cup \{\text{realization, attendant\_hope, effort\_to\_attain}\} \\
\text{Vars(Fears-confirmed)} &= \text{Vars(Negative)} \cup \{\text{realization, attendant\_fear, effort\_to\_prevent}\} \\
\text{Vars(Remorse)} &= \text{Vars(Distress)} \cup \text{Vars(Shame)} \\
\text{Vars(Anger)} &= \text{Vars(Distress)} \cup \text{Vars(Reproach)} \\
\text{Vars(Pity)} &= \text{Vars(Distress)} \cup \{\text{undeservingness, undes\_for\_other, liking}\} \\
\text{Vars(Resentment)} &= \text{Vars(Distress)} \cup \{\text{undeservingness, des\_for\_other, disliking}\}
\end{align*}
\]

**Figure 6.4:** The assignment of variables affecting intensity to emotion types, corresponding directly to Figure 6.3.
**Definition 6.7 (Variables affecting intensity)**

The 25 variables affecting intensity are collected in the set \( \text{vars} \), as follows.

\[
\text{vars} = \{ \text{sense of reality}, \text{proximity}, \text{unexpectedness}, \text{arousal}, \\
\text{desirability}, \text{undesirability}, \text{praiseworthiness}, \text{blameworthiness}, \\
\text{appealingness}, \text{unappealingness}, \text{liking}, \text{disliking}, \\
\text{likelihood}, \text{familiarity}, \text{strength of unit}, \text{expectation deviation}, \\
\text{realization}, \text{attendant hope}, \text{attendant fear}, \text{effort to attain}, \text{effort to prevent}, \\
\text{deservingness}, \text{undeservingness}, \text{des for other}, \text{undes for other} \}
\]

(6.12)

Each \( \text{var} \in \text{vars} \) is a function with the following mapping:

\[
\text{var} : \text{emotion} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}
\]

(6.13)

So each variable affecting intensity is a function taking an emotion and a state, and returning a non-negative value for that variable.

Next it is assumed there exists a function \( \text{Vars} : \text{emo-types} \rightarrow 2^{\text{vars}} \) returning for each type of emotion the variables affecting its intensity (if an instance of that emotion type were triggered). The definition of \( \text{Vars} \) is presented in Figure 6.4. It will be clear that this definition of \( \text{Vars} \) accurately follows the inheritance-based structure illustrated in Figure 6.3. It may be interesting to note that the multiple inheritance of variables by the emotion types ‘gratification’, ‘remorse’, ‘gratitude’, and ‘anger’ is easily captured using set union.

The specification of the signature of each variable affecting intensity as a function of type \( \text{emotion} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0} \) may seem too liberal, because except for the four global variables, the variables are not applicable to each type of emotion. To solve this, the following constraint prevents possible ‘type errors’.

**Constraint 6.8 (Local variables are only applicable to particular emotion types)**

For all \((\text{Emotion}, i, \bar{o}) \in \text{emotion}\), for all \((M, s) \in \mathcal{S}\), and for all \(\text{var} \in \text{vars}\):

\[
\text{var}(\text{Emotion}, i, \bar{o})(M, s) \text{ is defined iff } \text{var} \in \text{Vars}(\text{Emotion})
\]

Let us now briefly describe each variable affecting intensity.

**Variables Affecting Intensity as Functions**

Below a description is provided of each of the 25 variables affecting intensity as identified in the OCC model. The first four variables are the global variables; the other 21 are the local variables. Which variables affect which emotion types is illustrated in Figure 6.3. After the (brief) descriptions we will show for some of the variables how they may be calculated, if the calculation can be expressed in our formal framework.

*sense of reality*: This global variable captures the *sense of reality* of the experiencing agent. A low sense of reality will result in low emotion potentials for triggered emotions and thus in weak or no emotional experience. A low sense of reality can
6.2 OVERALL FELT INTENSITY

result from imagining (as opposed to perceiving) events, actions, and objects. On the other hand, sufficiently believable portrayals of events, actions, and objects in media such as novels and movies can sometimes lead to emotional experiences, because these media can induce a sense of reality. In this way the sense of reality variable is used in the OCC model to account for vicarious emotions. This variable is also used to account for initial “numbness” in response to very grave situations (whether positive or negative), because such situations may need some time to become accepted by the agent in question, which can be captured by a slowly increasing sense of reality.

proximity: This global variable captures the psychological proximity of the situation to the experiencing agent. With psychological proximity the OCC model means proximity in all possible senses of the word, including the temporal, physical, relational, and geographical sense. For example, learning that a particular political murder happened ten years ago will most likely cause a far weaker emotional reaction (if any) than hearing about a political murder as breaking news, because the latter event has greater temporal proximity. Moreover, a political murder being committed in a far country is likely to cause a far weaker emotional reaction (if any) than one being committed in front of one’s eyes, because the latter event is happening physically much closer. Similarly, actions performed by individuals with which one has some personal relation are likely to elicit stronger emotional reactions (positive or negative). Therefore, psychological proximity is postulated to be positively correlated with overall felt intensity.

unexpectedness: This global variable captures the unexpectedness of the situation to the experiencing agent. An agent finding a situation (event, action, or object) unexpected may be said to be surprised by it. Note that the OCC model does not classify ‘surprise’ as an emotion itself, because it lacks valence. Surprise can be either positive or negative; that is, both pleasant surprise and unpleasant surprise exist. In the OCC model, surprise is considered as merely an aspect of a proper emotion. So for example, ‘pleasant surprise’ can be seen as an instance of ‘joy’ with relatively high unexpectedness. Because surprise makes an emotion more intense, this aspect has been incorporated in the OCC model as a variable affecting the intensity of emotions named unexpectedness. It should further be noted that unexpectedness is considered to be a backward-looking expectation. This means that, in contrast to likelihood discusses below, the unexpectedness of an event, action, or object is only evaluated after having been perceived. Thus unexpectedness of an event, action, or object does not depend on considerations prior to the perception of the event, action, or object.

arousal: This global variable captures the physiological arousal of the experiencing agent. Here the OCC model means arousal induced by the autonomic nervous system in humans. For example, a high level of adrenaline increases physiological arousal in humans, and thus intensifies any triggered emotion (whether positive or negative). Here we will not consider this variable further, because arousal in this sense goes beyond the cognitive perspective of this dissertation.

desirability, undesirability, praiseworthiness, blameworthiness, appealingness, unappealingness: These are the central appraisal variables that have been used since Chapter 3. It should be noted that the operators Des, Undes, Praisew, Blamew, Appeal, and Unappeal, as used in the previous chapters, are purely qualitative,
whereas the variables affecting intensity are quantitative. For example, \textbf{Praisew} is used to indicate \textit{whether or not} an action is praiseworthy, whereas the variable \textit{praiseworthiness} is used to indicate the \textit{degree of judged praiseworthiness}. As mentioned before, the OCC model considers values for \textit{desirability} and \textit{undesirability} to be determined based on an agent’s goals, \textit{praiseworthiness} and \textit{blameworthiness} to be determined based on an agent’s standards, and \textit{appealingness} and \textit{unappealingness} to be determined based on an agent’s attitudes, where tastes are considered to be a subset of attitudes. It should further be noted that the variable \textit{appealingness} is distinguished from the emotion \textbf{Liking}, in that \textit{appealingness} captures dispositional liking, whereas \textbf{Liking} captures momentary liking.

\textit{likelihood}: This is the \textit{likelihood} of an event going to occur or having occurred. In general, it is not assumed that likelihood estimates are made on the cognitive level, nor that they are absolute. That is, likelihood estimates may be comparative in nature (e.g., one event may simply be considered as more likely than another event), so calculating explicit probabilities may not be necessary. The \textit{likelihood} variable is only used for determining the potentials of ‘hope’ and ‘fear’ emotions. It should be noted that likelihood estimates can vary over time, probably causing congruent changes in the intensities at which ‘hope’ and ‘fear’ emotions are experienced. For example, seeing one’s chances at winning a game increase also raises one’s hope for winning and decreases one’s fear for losing.

\textit{expectation} \textit{deviation}: This is the \textit{deviation from person/role-based expectations} of an action by an agent. This variable is only used for the action-based emotion types and indicates the unexpectedness of an action from a particular agent in a particular role. For example, a person in the role of a licensed hunter can be expected to shoot a wild bear, while a person in the role of a nature preservation activist is not. It should be noted that, if one upholds the standard that shooting wild animals is bad, there is little reason to assign more blameworthiness to the activist’s action of shooting a bear than to the hunter’s action of shooting a bear, for it is the same kind of action with the same outcome. If to account for the natural reaction of being more angry at the activist’s action than at the hunter’s action, the \textit{expectation deviation} variable is introduced in the OCC model. According to this variable, the activist’s action deviates more from his or her role-based expectations than the hunter’s action, thereby increasing the value returned by \textit{expectation deviation} and thus giving a higher potential to the ‘anger’ directed at the activist.

\textit{strength} \textit{of} \textit{unit}: This is the \textit{strength of the cognitive unit} with the actual agent of an action. This variable is introduced in the OCC model to account for ‘pride’ and ‘shame’ emotions where the experiencing agent and the acting agent are not the same. For example, a mother may be proud of the achievements of her son because she considers herself to being a cognitive unit with her son, not because she considers herself to be directly responsible for her son’s actions. In general, the stronger the cognitive unit with the actual agent, the stronger the potential of a ‘pride’ or ‘shame’ emotion in the experiencing agent. It should be noted that the strength of the cognitive unit with one particular agent may be very dependent on the situation. For example, if Alice, who is from Amsterdam, meets Bob, who is from Berlin, at a party in Berlin with only other Germans present, Alice may consider herself to be an outsider and not as being in a cognitive unit with Bob. On the other hand, if Alice
and Bob meet at a party in Tokyo with only Japanese people, Alice and Bob may share a strong cognitive unit because they are the only Western Europeans present.

**familiarity**: This is the *degree of familiarity* an agent has with an object. In the OCC model, the *familiarity* variable is the only variable affecting the intensity of the object-based emotions besides appealingness/unappealingness and the global variables. The contribution of this variable to an emotion’s potential is a bit special though. For all other variables, the idea is that the greater the value of the variable, the greater the potential of the emotion in question. However, as the saying goes, “familiarity breeds contempt.” Specifically, the contribution of the *familiarity* variable to an emotion’s potential is not assumed to be linear, but to have a bell shape [Ortony et al., 1988]. So familiarity with an object greater than some optimal value will lessen the variable’s contribution to emotion potential. A way of capturing this idea will be discussed below.

**realization**: This is the *degree of realization* of an event that was hoped for or feared. Thus the *realization* variable is used in determining the potentials of ‘satisfaction’, ‘fears-confirmed’, ‘relief’, and ‘disappointment’ emotions. For partially attainable goals, the degree of realization can be seen as the ratio between the number of achieved subgoals and the total number of subgoals. On the other hand, for all-or-nothing goals, a degree of realization could also be determined by measuring how close the agent in question considered itself to be to achieving the goal.

**attendant_hope, attendant_fear**: These variables express the *intensity of the attendant hope emotion* and the *intensity of the attendant fear emotion*. These variables complete the temporal link between ‘hope’ and ‘fear’ on the one hand, and ‘satisfaction’, ‘fears-confirmed’, ‘relief’, and ‘disappointment’ on the other hand. ‘Satisfaction’ and ‘disappointment’ require a previously triggered ‘hope’ emotion (see Definition 3.12); the intensity of this ‘hope’ emotion then affects the potential of a ‘satisfaction’ or ‘disappointment’ emotion through the *attendant_hope* variable. Likewise, ‘relief’ and ‘fears-confirmed’ require a previously triggered ‘fear’ emotion, whose intensity affects the potential of the ‘relief’ or ‘fears-confirmed’ emotion in question through the *attendant_fear* variable.

**effort_to_attain, effort_to_prevent**: These are the *effort expanded in trying to attain the event* and the *effort expanded in trying to prevent the event*. The *effort_to_attain* variable is used in determining the potentials of ‘satisfaction’ and ‘disappointment’ emotions, whereas the *effort_to_prevent* variable is used for ‘relief’ and ‘fears-confirmed’ emotions. So *effort_to_attain* presupposes a ‘hope’ emotion and *effort_to_prevent* presupposes a ‘fear’ emotion. It should be noted that the term “effort” is used to include mental, physical, and material (e.g., invested money) effort. The OCC model actually mentions only one *effort* variable which is used for all four emotion types (‘satisfaction’, ‘fears-confirmed’, ‘relief’, and ‘disappointment’). However, a careful reading of the OCC model reveals that two kinds of effort are distinguished, namely effort expanded in trying to *attain* something and effort expanded in trying to *prevent* something. Therefore we have introduced two separate variables.

**deservingness, undeservingness**: These are the *deservingness for the other* and the *undeservingness for the other* of a consequence of an event. These variables are used in determining the potentials of ‘happy-for’, ‘resentment’, ‘gloating’, and ‘pity’ emotions. In the OCC model, the degree to which another agent deserves or does not
deserve a certain consequence of an event is described as depending on judgments of what is considered just and what is considered unjust. However, a proper treatment of such judgments is outside the scope of both the OCC model and this dissertation.

des_for_other, undes_for_other: These are the presumed desirability for the other and the presumed undesirability for the other of a consequence of an event. These variables are used in determining the potentials of ‘happy-for’, ‘resentment’, ‘gloating’, and ‘pity’ emotions. The des_for_other and undes_for_other variables require one agent to have a (mental) model of another agent and its goals. In this mental model methods similar to those used in determining values for the desirability and undesirability variables can be used in order to attain values for the des_for_other and undes_for_other variables. We have already seen this in Definition 3.13, where the Des and Undes constructs were “recycled.” For example, Des_{i}(\varphi) was used to denote that agent i desires \varphi, which was then nested in the expression Presume; Des_{j}(\varphi) to denote that agent i presumes that agent j desires \varphi.

liking, disliking: These are the degree of liking another person and the degree of disliking another person. These variables are used in determining the potentials of ‘happy-for’, ‘resentment’, ‘gloating’, and ‘pity’ emotions. They are similar to the appealingness and unappealingness variables, but liking and disliking are only applicable to agents. We re-emphasize that the liking and disliking variables should not be confused with the Liking and Disliking emotion types; the former expresses dispositions for liking or disliking another agent, whereas the latter is used in expressing momentary reactions of liking and disliking.

Calculating Variables in the Current Framework

Having described each of the variables affecting intensity, we now turn to investigating in which way many of them can be calculated in the current framework. Note that the formulas below are more suggestions than definite calculation methods. Moreover, many of the formulas below depend on parameters that can be set depending on the application and the desired emotional ‘character’ of the agent in question. Consequently, the formulas presented here stop short of providing actual values for the variables affecting intensity. This is in line with our focus on the logical structure of emotions. The aim of this subsection is to show how different variables affecting intensity are or can be related to each other. For desirability, praiseworthiness, and appealingness, their relations as presented here correspond to the ways in which they were related to each other in Section 5.4, which means that they are in line with the OCC model given that achievement goals are the only kinds of concerns for agents. For the other variables affecting intensity, the presented relations have been taken from the OCC model as well.

Let us start with desirability and undesirability. In Chapter 5 we constrained the desirability operator such that all subgoal achievements were desirable. The degree to which the achievement of subgoals is desirable can be calculated as the degree to which the subgoals contribute to the overall goal. Having defined goals as conjunctions of literals, such a degree of desirability can simply be calculated as the ratio between the number of subgoals (literals) achieved and the number of subgoals
(literals) in the overall goal. Thus the desirability variables can be defined as follows:

\[
\text{desirability}(\text{Emotion}, i, \varphi)(M, s) = \sigma(\sum \{ f(#\varphi, #\psi) \mid (\varphi, \psi) \in \text{sub}, \psi \in \text{Goals}(i)(M, s) \})
\]  

(6.14)  

for \text{Emotion} \in \{\text{Pleased, Hope, Joy}\}. Here \sigma is a squashing function (e.g., \(\sigma(x) = \tanh(x)\)), which is needed because there may be multiple goals (\(\psi\)) to which the subgoal (\(\varphi\)) contributes; a squashing function may thus be needed in order to keep the magnitude of desirability within some range. The notation \#\varphi is used to express the number of conjuncts in \(\varphi\); that is, \#\varphi = |\Phi| where \(\varphi = \bigwedge \Phi\) and \(\Phi \in \text{csl}\). The function \(f\) then weighs the number of newly achieved subgoals against the total number of subgoals; for example, \(f(x, y) = \frac{x}{y}\).

In the OCC model it is suggested\(^2\) that if the achievement of a subgoal is desirable to some degree, then the absence or undermining of that subgoal is likely to be undesirable to the same degree. By restraining goals to conjunctions of literals, it is easy to define the notion of absence or undermining; indeed, the relation \text{inv} (see Definition 5.9) can be used to invert goal formulas and thus to convert between expressing achievement and undermining. Using \text{inv}, the undesirability variable can simply be calculated in terms of the desirability variable, as follows:

\[
\text{undesirability}(\text{Emotion}, i, \varphi)(M, s) = \text{desirability}(\text{Pleased}, i, \psi)(M, s)
\]  

(6.15)  

for \text{Emotion} \in \{\text{Displeased, Fear, Distress}\} and \((\varphi, \psi) \in \text{inv}\). Note that a positive event-based emotion type must be supplied to \text{desirability}, which is done substituting \text{Displeased, Fear, or Distress} with \text{Pleased}.

The above calculations for desirability and undesirability do not work for the fortunes-of-others emotion types (‘happy-for’, ‘resentment’, ‘gloating’, ‘pity’), however. The fortunes-of-others emotion types take an additional argument; namely, another agent to which the emotion in question is directed. But even if another agent is involved, the (un)desirability variable is still supposed to indicate the (un)desirability for the self; that is, the consequence in question should be tested against one’s own goals. So for the fortunes-of-others emotion types, desirability and undesirability are simply calculated as follows:

\[
\text{desirability}(\text{Emotion}, i, (\varphi, j))(M, s) = \text{desirability}(\text{Pleased}, i, \varphi)(M, s)
\]  

(6.16)  

\[
\text{undesirability}(\text{Emotion’}, i, (\varphi, j))(M, s) = \text{undesirability}(\text{Displeased}, i, \varphi)(M, s)
\]  

(6.17)  

for \text{Emotion} \in \{\text{Happy-for, Gloating}\} and \text{Emotion’} \in \{\text{Resentment, Pity}\}.

The fortunes-of-others emotion types also use another desirability-based variable, namely the \text{des for other} variable expressing presumed desirability for another agent. Since presumptions were modeled using the notion of belief, presumptions of an agent \(i\) about desirability for other agents can be defined over the belief-accessibility relation \(R_i\), as follows:

\[
\text{des for other}(\text{Emotion}, i, (\varphi, j))(M, s) = \text{g}(\{\text{desirability}(\text{Pleased}, j, \varphi)(M, s’) \mid (s, s’) \in R_i\})
\]  

(6.18)  

\(^2\) [Ortony et al., 1988, page 113–114].
for $\text{Emotion} \in \{\text{Happy-for, Resentment}\}$. If all possible worlds are presumed to be equiprobable, one could choose $g$ to be the function $g(X) = \frac{1}{|X|} \sum X$. Similarly, the $\text{undes}_\text{for \ other}$ variable can be defined in terms of the $\text{undesirability}$ variable, as follows:

$$\text{undes}_\text{for \ other}(\text{Emotion}, i, \langle \varphi, j \rangle)(M, s) = g(\text{undesirability}(\text{Displeased}, j, \varphi)(M, s') \mid (s, s') \in R_i)$$  (6.19)

for $\text{Emotion} \in \{\text{Gloating, Pity}\}$.

Recall from Section 5.4 that we constrained the praiseworthiness of an action by considering the desirability of the consequences of the action. This same idea can be applied to the calculation of the $\text{praiseworthiness}$ variable, as follows:

$$\text{praiseworthiness}(\text{Emotion}, i, j; \alpha)(M, s) = \sigma(\sum \{ \text{desirability}(\text{Pleased}, i, \varphi)(M, s) \mid M, s \models \text{B}_i\text{Related}(j; \alpha, \varphi) \})$$  (6.20)

for $\text{Emotion} \in \{\text{Approving, Pride, Admiration, Gratification, Gratitude}\}$. So the praiseworthiness of action $\alpha$ of agents $j$ is determined by collecting all desirable consequences $\varphi$ of this action. These desirability values are then summed and squashed using a squashing function $\sigma$. An analogous method can be use to calculate the $\text{blameworthiness}$ variable:

$$\text{blameworthiness}(\text{Emotion}, i, j; \alpha)(M, s) = \sigma(\sum \{ \text{undesirability}(\text{Displeased}, i, \varphi)(M, s) \mid M, s \models \text{B}_i\text{Related}(j; \alpha, \varphi) \})$$  (6.21)

for $\text{Emotion} \in \{\text{Disapproving, Shame, Reproach, Remorse, Anger}\}$.

In Section 5.4 praiseworthiness was again used to constrain the appealingness of agents (though not the appealingness of objects). Effectively, what this accomplishes is that the appealingness of another agent is proportional to the praiseworthiness of the actions performed by that agent in the past. Thus the $\text{appealingness}$ variable can be calculated as follows:

$$\text{appealingness}(\text{Liking}, i, j)(M, s) = \sigma(\sum \{ \text{praiseworthiness}(\text{Approving}, i, j; \alpha)(M, s) \mid M, s \models \text{Past Approving}_j^T(\alpha) \})$$  (6.22)

for $j \in \text{AGT}$. Note that this demand that $j \in \text{AGT}$ means that the appealingness of objects that are not agents is left unspecified. The $\text{unappealingness}$ variable can be calculated analogously:

$$\text{unappealingness}(\text{Disliking}, i, j)(M, s) = \sigma(\sum \{ \text{blameworthiness}(\text{Disapproving}, i, j; \alpha)(M, s) \mid M, s \models \text{Past Disapproving}_j^T(\alpha) \})$$  (6.23)
OVERALL FELT INTENSITY

for $j \in \text{agt}$.

As noted before, the liking and disliking variables can be defined directly in terms of the appealingness and unappealingness variables, which already capture dispositional liking and disliking. This relation is captured as follows:

$$\text{liking}(\text{Emotion}, i, \langle \varphi, j \rangle)(M, s) = \text{appealingness}(\text{Liking}, i, j)(M, s)$$ (6.24)
$$\text{disliking}(\text{Emotion}', i, \langle \varphi, j \rangle)(M, s) = \text{unappealingness}(\text{Disliking}, i, j)(M, s)$$ (6.25)

for $\text{Emotion} \in \{\text{Happy-for, Pity}\}$ and $\text{Emotion}' \in \{\text{Gloating, Resentment}\}$.

Several times we have mentioned that the contribution of familiarity to the potential of a ‘liking’ or ‘disliking’ emotion is not linear. That is, both low and high familiarity do not contribute as much to potential as moderate familiarity. It should be noted, though, that this idea greatly depends on the word “contribution.” Familiarity is familiarity; it should be possible to calculate familiarity with an object regardless of how the resulting value is used, i.e., what it contributes. One simple way of doing this would be to let the familiarity of agent $i$ with object $x$ be proportional to the amount of exposure $i$ has had to $x$ (denoted as $\text{exposure}_{i,x}$) divided by the age of $i$ (denoted as $\text{age}_i$):

$$\text{familiarity}(\text{Emotion}, i, x)(M, s) = a \cdot \frac{\text{exposure}_{i,x}}{\text{age}_i}$$ (6.26)

for $\text{Emotion} \in \{\text{Liking, Disliking}\}$ and some scalar $a$. The contribution of a familiarity value is then regulated through the weight $w_{\text{familiarity}}$ assigned to the familiarity variable (see Definition 6.5). To be in line with the OCC model, this weight should be a bell-shaped function of familiarity. For example, a simple bell-shaped function is obtained by setting $w_{\text{familiarity}} = b \cdot \exp(- (\text{fml} - \text{peak})^2)$, where $\text{fml}$ is the familiarity as calculated above, and $\text{peak}$ is the “optimal” familiarity value; that is, the value at which the contribution of familiarity to emotion potential will be greatest. Some scalar $b$ can be used to bring the weight in the appropriate range.

Having defined goal formulas as conjunctions of literals, a simple calculation for the realization variable can be constructed, namely by counting the number of subgoals achieved and the number of subgoals hoped for or feared (to be undermined). Dividing these two then results in a realization value between 0 and 1, as follows:

$$\text{realization}(\text{Emotion}, i, \langle \varphi, \psi \rangle)(M, s) = \frac{\#\varphi}{\#\psi}$$ (6.27)

for $\text{Emotion} \in \{\text{Satisfaction, Disappointment, Relief, Fears-confirmed}\}$ and $\varphi, \psi \in \text{ccl}$.

Finally, let us consider the attendant hope and attendant fear variables. These variables are actually rather trivial to calculate, because the value asked for is simply the intensity of an attendant ‘hope’ or ‘fear’ emotion. For this purpose the overall felt intensity function $\text{OFI}$ can be used, as follows:

$$\text{attendant hope}(\text{Emotion}, i, \langle \varphi, \psi \rangle)(M, s) = \text{OFI}(\text{Hope}, i, \psi)(M, s)$$ (6.28)
$$\text{attendant fear}(\text{Emotion}', i, \langle \varphi, \psi \rangle)(M, s) = \text{OFI}(\text{Fear}, i, \psi)(M, s)$$ (6.29)
EMOTION EXPERIENCE

for \textit{Emotion} ∈ \{\textit{Satisfaction, Disappointment}\} and \textit{Emotion’} ∈ \{\textit{Relief, Fears-confirmed}\}. Note that it is the second argument of the ‘satisfaction’, ‘fears-confirmed’, ‘relief’, or ‘disappointment’ emotion (i.e., \(\psi\)) that must be supplied to the attendant ‘hope’ or ‘fear’ emotion (compare Definition 3.12 on page 49).

We should note that we have not considered possible calculations of the four global variables \textit{sense of reality}, \textit{proximity}, \textit{unexpectedness}, and \textit{arousal}, and of the local variables \textit{likelihood}, \textit{strength of unit}, \textit{expectation deviation}, \textit{effort to attain}, \textit{effort to prevent}, \textit{deservingness}, and \textit{undeservingness}. The reason for this is that most of these variables need notions (e.g., roles, justness, an autonomic nervous system) that have not been incorporated into the current framework. Therefore we cannot consider their possible calculations within the scope of this research.

6.2.4 Explicating Assumptions

Having formalized emotion intensity and the structure of its calculation, we are now in a position to uncover the assumptions that have to be made in order to bring our framework in line with observations and speculations made in the OCC model. For example, there is a discussion on page 113–114 of [Ortony et al., 1988] on the relation between the intensities of corresponding ‘hope’ and ‘fear’ emotions; that is, between the intensities of the hope to achieve a certain goal and the fear of failure (i.e., absence of the goal achievement). It is noted that in such a case, the intensities of these emotions should sum to a constant. In this subsection, we investigate whether this is indeed the case, or which additional assumption are needed to attain this property. For simplicity, let us restrict our analysis to the local variables affecting intensity and to initial intensities (in the sense of Definition 6.4) so that we can leave the time aspect out of the picture.

First of all, formulas of the form \(\text{Hope}_i^T(\varphi) \land \text{Fear}_i^T(\varphi)\) must be contingent. That is, hope with respect to the possible achievement of (sub)goal \(\varphi\) and fear with respect to failing to achieve \(\varphi\) (which is the case when \(\varphi\) holds) must be elicitable at the same time. This is indeed possible, as was also suggested by Proposition 5.17 on page 97.

Now assume \(M, s \models \text{Hope}_i^T(\varphi) \land \text{Fear}_i^T(\varphi)\) and let \(\iota_h\) be the initial intensity associated with \(\text{Hope}_i^T(\varphi)\) in state \((M, s)\) and let \(\iota_f\) be the initial intensity associated with \(\text{Fear}_i^T(\varphi)\) in state \((M, s)\), in the sense of Definition 6.4. The question is then whether the sum \(\iota_h + \iota_f\) is indeed constant. Recall that initial intensity is calculated in terms of potential threshold, so let us examine what the potentials and thresholds of \(\text{Hope}_i^T(\varphi)\) and \(\text{Fear}_i^T(\varphi)\) amount to.

According to Definition 6.6, the thresholds that are associated with the triggered emotions \(\text{Hope}_i^T(\varphi)\) and \(\text{Fear}_i^T(\varphi)\) are each other’s negative. This means that, if for both the triggered ‘hope’ and ‘fear’ emotion the potential is greater than the threshold, then \(\iota_h + \iota_f = \text{Potential}(\text{Hope}, i, \varphi)(M, s) + \text{Potential}(\text{Fear}, i, \varphi)(M, s)\). The question that thus remains is whether this sum of potentials is constant.

According to Definition 6.5, the potentials of ‘hope’ and ‘fear’ emotions are determined by the (un)desirability of the prospective consequence and its likelihood, which is expressed as follows:
Potential(\text{Hope}, i, \varphi)(M, s) = w_1 \cdot \text{desirability}(\text{Hope}, i, \varphi)(M, s) \\
+ w_2 \cdot \text{likelihood}(\text{Hope}, i, \varphi)(M, s) \\
Potential(\text{Fear}, i, \overline{\varphi})(M, s) = w_3 \cdot \text{undesirability}(\text{Fear}, i, \overline{\varphi})(M, s) \\
+ w_4 \cdot \text{likelihood}(\text{Fear}, i, \overline{\varphi})(M, s)

where \( w_1 = w_{\text{desirability}} \), \( w_3 = w_{\text{undesirability}} \), and \( w_2 = w_4 = w_{\text{likelihood}} \) are variable-dependent weights. Note that the global variables are left out for simplicity.

According to formula (6.15), and in line with the OCC model, the absence of a desirable event is assumed to be undesirable to the same degree as the event in question is desirable. Formally, this means that we can assume that \( \text{desirability}(\text{Hope}, i, \varphi)(M, s) = \text{undesirability}(\text{Fear}, i, \overline{\varphi})(M, s) \). As for the likelihood variable, it is reasonable to assume that it behaves such that \( \text{likelihood}(\text{Emotion}, i, \varphi)(M, s) = 1 - \text{likelihood}(\text{Emotion}, i, \overline{\varphi})(M, s) \) if likelihood estimates are expressed as (subjective) probabilities between 0 and 1. Note that this equation uses \( \varphi \) versus \( \overline{\varphi} \), whereas for our hope and fear example we have used \( \varphi \) versus \( \overline{\varphi} \). In order to bring these two in line, we have to restrict our analysis to single subgoals. This is because \( \overline{\varphi} \equiv \varphi \text{ iff } \varphi \in \text{lit} \); that is, \( \varphi \) is a single literal.

With these properties of the variables affecting intensity, we have that the sum of initial intensities amounts to \( \iota_h + \iota_f = (w_1 + w_3)\text{desirability}(\text{Hope}, i, \varphi)(M, s) + w_2 \). This sum is ‘constant’ insofar as the desirability of the (sub)goal and the variable-dependent weights are constant (which is not an unreasonable assumption over short to medium time spans). Thus, the (estimated) likelihood of the goal achievement can vary freely without affecting the sum above.

The reader may have noticed the frequent usage of the word “assumption” in this subsection, which shows that this formalization is capable of explicating many constraints (although reasonable) that are needed to capture the intuitions of the OCC model. Adopting these constraints will render our model completely in line with the OCC model.

### 6.3 Experience of a Type of Emotion

Having formalized the concept of ‘overall felt intensity’ (Definition 6.3) and investigated its structure and possible calculation in the preceding section, we are now in a position to formally define emotional experience.

### 6.3.1 Formalizing Emotional Experience

Definition 3.1 on page 39 informally explained how emotional experience can be defined in terms of triggering and intensity. Specifically, experience was defined as follows:

\textbf{Emotion} is experienced if and only if

1. \textbf{Emotion} has been triggered sometime in past and
2. overall felt intensity of \textbf{Emotion} is positive
where \textbf{Emotion} stands for any of the emotion types distinguished in the OCC model. Thus emotional experience is equated with a conjunction of two terms. The first of these two terms (i.e., emotion triggering) has been treated extensively in Chapters 3, 4, and 5. The notion of ‘overall felt intensity’, as used in the second point, was the subject of Section 6.2. Taken together, emotional experience is then formally defined as follows.

\textbf{Definition 6.9 (Emotional experience, formal)}

For all \((\text{Emotion}, i, \bar{o}) \in \text{emotion}\), the fact that agent \(i\) experiences emotion \(\text{Emotion}\) with respect to \(\bar{o}\) is denoted as \(\text{Emotion},(\bar{o})\), which is defined as follows:

\[
\text{Emotion},(\bar{o}) \overset{\text{def}}{=} \text{Past Emotion}^T_i(\bar{o}) \land \text{OFI(Emotion, }i,\bar{o})
\]  

(6.30)

where \(\text{OFI}\) stands for “overall felt intensity.”

Before going into the details of this definition, it should be noted that this definition is actually a shorthand for 28 definitions, one for each \(\text{Emotion} \in \text{emo-types}\). For example:

\[
\text{Fear}(\varphi) \overset{\text{def}}{=} \text{Past Fear}^T_i(\varphi) \land \text{OFI(Fear, }i,\varphi)
\]

\(\text{Fear}(\varphi)\) is then read as “agent \(i\) fears consequence \(\varphi\) (of an event).” It is crucial to note that this is different from the reading of \(\text{Fear}^T_i(\varphi)\). \(\text{Fear}^T_i(\varphi)\) expresses that ‘fear’ with respect to \(\varphi\) is triggered for agent \(i\), whereas \(\text{Fear}(\varphi)\) expresses that fear with respect to \(\varphi\) is currently experienced by agent \(i\).

The construction \(\text{Past Emotion}^T_i(\bar{o})\) in Definition 6.9 corresponds to point (1) above and expresses that the emotion type in question has been triggered some time in the past, where emotion triggering is defined as in Chapter 3. The construction \(\text{OFI(Emotion, }i,\bar{o})\) captures the ‘overall felt intensity’ part in point (2) above, as follows. We specify that an \(\text{OFI}\) construct is true if and only if the overall felt intensity of the provided emotion type is greater than zero in the current state. For example:

\[
M, s \models \text{OFI(Fear, }i,\varphi) \quad \text{iff} \quad \text{OFI(Fear, }i,\varphi)(M, s) > 0
\]

where \(\text{OFI}\) is a function taking an emotion type, the experiencing agent, the argument(s) of the emotion, and a model–state pair. It then returns the overall felt intensity of that emotion in that state, as a non-negative real number. The function \(\text{OFI}\) was formally defined in Definition 6.3.

It should be noted that the \(\text{OFI}\) construct has not been formally defined yet. In keeping with the top-down approach taken in this dissertation, the \(\text{OFI}\) construct is used before it is grounded, because we prefer to put constructs in their proper context before delving into the details of how they may be defined. In line with this approach, the next subsection formally defines the \(\text{OFI}\) construct in terms the \(\text{OFI}\) function. However, while we are still at the level of emotional experience, let us consider several properties of the concept of emotional experience as formalized in Definition 6.9. The properties presented below may seem trivial given Definition 6.9, but it is important to emphasize these properties because they display the essence of our distinction between emotion elicitation and experience, and the relations between the two.
Proposition 6.10 (The relation between emotion elicitation and experience)

The following propositions show which implications between emotion triggers and experienced emotions are valid or not. For all \((\text{Emotion}, i, \bar{o}) \in \text{emotion}\), we have that:

\[
\begin{align*}
\not\implies \text{Emotion}_i^T(\bar{o}) &\implies \text{Emotion}_i(\bar{o}) \\
\not\implies \text{Emotion}_i(\bar{o}) &\implies \text{Emotion}_i^T(\bar{o}) \\
\implies \text{Emotion}_i(\bar{o}) &\implies \text{Past Emotion}_i^T(\bar{o})
\end{align*}
\]  

(6.31)  
(6.32)  
(6.33)

The first proposition (which abbreviates, e.g., \(\not\implies \text{Joy}_i^T(\varphi) \implies \text{Joy}_i(\varphi)\)) states that a newly triggered emotion is not necessarily experienced, whereas the second proposition (e.g., \(\not\implies \text{Joy}_i(\varphi) \implies \text{Joy}_i^T(\varphi)\)) states that an emotion that is currently experienced is not necessarily a newly triggered one. Finally, the third proposition states that for an emotion to be experienced, it must have been triggered some time in the past.

We are now in a position to define what is in the data block labeled \textit{experienced emotions} in Figure 6.1, in a way similar to that of the \textit{triggered emotions}.

Definition 6.11 (Experienced emotions)

AllExperienced : \(S \rightarrow 2^{\text{emotion}}\) is a function indicating per model–state pair which emotions are experienced for all agents. AllExperienced is defined as follows:

\[
\text{AllExperienced}(M, s) = \{ \text{Emotion}, i, \bar{o} \in \text{emotion} \mid M, s \models \text{Emotion}_i(\bar{o}) \}
\]  

(6.34)

Thus \text{AllExperienced}(M, s) represents the contents of the data block labeled \textit{experienced emotions} for each state \((M, s)\). Note that this definition is very similar to Definition 5.23. This is of course no coincidence, but the product of our consistent notation in formalization.

6.3.2 Parameters of Emotional Experience

Let us put the concept of ‘overall felt intensity’ better in context. Although we use ‘overall felt intensity’ to define general emotional experience, it is not the only aspect of emotional experience that is valued. Emotional experience is most likely multidimensional. For example, in [Frijda et al., 1992] sixteen parameters of subjective emotional experience were identified. The arguably most important among these are “drasticness of emotion-induced urges to act” and, again, “overall felt intensity.” The former is also called \textit{action tendency}. It is interesting to note that Frijda practically identifies the notion of emotion with “felt action tendency” [Frijda, 1987], although this view is not uncontroversial among psychologists [Ortony, 2009]. We will have more to say about action tendency in Section 7.2.

Given that there are multiple parameters that together constitute emotional experience and that it is not (yet) known exactly which are these parameters, we will abstract from the specific parameters and assume a set \text{exp-param} of parameters as given. Being general, we can then specify that a parameter \(P \in \text{exp-param}\) holds for a certain emotion type if and only if the value it returns for the current state is greater than zero. This is expressed in the following definition.
Definition 6.12 (Parameters of emotional experience)

Let $\text{exp-param}$ be a given set of parameters of emotional experience. For all $P \in \text{exp-param}$ let there exist a function $P$ assigning for each state a non-negative value to an emotion instance. Each function $P$ has the following mapping:

$$P : \text{emotion} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0} \quad (6.35)$$

$P$ is then used to define a construct $P$ in the object language, indicating when this parameter is experienced by an agent. $P$ is interpreted as follows:

$$M, s \models P(\text{Emotion}, i, \bar{o}) \text{ iff } P(\text{Emotion}, i, \bar{o})(M, s) > 0 \quad (6.36)$$

In line with [Frijda et al., 1992], we now assume that ‘overall felt intensity’ is among the given parameters, where it is named OFI. Demanding that OFI $\in \text{exp-param}$ then completes Definition 6.9, because we can write $\text{OFI}/\text{OFI}$ instead of $P/P$ in formula (6.36). Indeed, the example of “$M, s \models \text{OFI}(\text{Fear}, i, \varphi)$ iff $\text{OFI}(\text{Fear}, i, \varphi)(M, s) > 0$” given on page 128 is just an instance of (6.36).

In anticipation of Section 7.2 on action tendencies and in line with [Frijda et al., 1992], it is also assumed that action tendencies are among the parameters of emotional experience. An action tendency parameter is denoted as $T_\alpha$, which is read as “the tendency to perform action $\alpha$.” We thus place the following constraint on the parameters included in $\text{exp-param}$.

Constraint 6.13 (Parameters of emotional experience)

It is assumed that ‘overall felt intensity’ (OFI) and ‘action tendency’ ($T_\alpha$ for each action $\alpha$) are parameters of emotional experience:

$$\text{OFI} \in \text{exp-param}$$

$$T_\alpha \in \text{exp-param} \quad \text{for all } \alpha \in \text{act}$$

We reiterate that with this constraint Definition 6.9 of emotional experience is finally grounded, because Constraint 6.13 allows Definition 6.12 to tie the operator OFI to the function OFI.

6.3.3 Comparing Emotions

A natural extension of having values for parameters of emotional experience is to compare these values. This way we can reason about, say, the intensity of one emotion being greater than the intensity of another emotion. The interpretation of a comparison of two parameters is then interpreted as follows.

Definition 6.14 (Comparison of parameters of emotional experience)

Let $P_1, P_2 \in \text{exp-param}$ be two parameters of emotional experience, let $(\text{Emotion}, i, \bar{o})$, $(\text{Emotion}', j, \bar{o}') \in \text{emotion}$ be two emotions, and let $(M, s) \in \mathcal{S}$ be a model–state pair; then the comparison of parameter $P_1$ of $(\text{Emotion}, i, \bar{o})$ with parameter $P_2$ of $(\text{Emotion}', j, \bar{o}')$ in state $(M, s)$ is interpreted as follows:
6.4 Emotion Types versus Tokens

So far in this dissertation we have only been concerned with emotion types. However, for each emotion type treated in the OCC model, a list of tokens is provided, indicating

\[ M, s \models P_1(\text{Emotion}, i, \varnothing) < P_2(\text{Emotion}', j, \varnothing') \quad \text{iff} \quad P_1(\text{Emotion}, i, \varnothing)(M, s) < P_2(\text{Emotion}', j, \varnothing')(M, s) \]  

(6.37)

and similarly for the other usual relational operators \( >, \leq, \geq, =, \) and \( \neq \).

To illustrate this definition, assume that we would want to compare the overall felt intensity of the emotion \( \text{Hope}(\varphi) \) with that of \( \text{Fear}(\psi) \). The assertion that the overall felt intensity of this hope emotion is smaller the overall felt intensity of this fear emotion is interpreted as follows:

\[ M, s \models \text{OFI(Hope, } i, \varnothing) < \text{OFI(Fear, } i, \psi) \quad \text{iff} \quad \text{OFI(Hope, } i, \varnothing)(M, s) < \text{OFI(Fear, } i, \psi)(M, s) \]

This formula is well-grounded because \( \text{OFI} \) was defined in Section 6.2.

It must be noted that it may not always make sense to compare any two parameters. Comparing the overall felt intensities of hope and fear may not sound problematic, but it may not make sense to compare, say, the magnitude of the tendency to flee out of fear with the magnitude of the tendency to touch an object of interest. In a numerical sense, values can always be compared, but that does not mean it makes sense to do so. In principle, the framework allows any comparison, but the result may not always be useful or sensible. Indeed, in the following we will only make comparisons between equal parameters or equal emotion instances. With this restriction in mind, we define the following abbreviations for making comparisons using one parameter or one emotion instance:

\[ \text{Emotion},(\varnothing) <_P \text{Emotion}',(\varnothing') \equiv P(\text{Emotion}, i, \varnothing) < P(\text{Emotion}', j, \varnothing') \]  

(6.38)

\[ P_1 <_{\text{Emotion},(\varnothing)} P_2 \equiv P_1(\text{Emotion}, i, \varnothing) < P_2(\text{Emotion}, i, \varnothing) \]  

(6.39)

So now we can write \( \text{Hope},(\varphi) <_{\text{OFI}} \text{Fear},(\bar{\varphi}) \) meaning that the overall felt intensity of hope for (achieving goal) \( \varphi \) is smaller than the fear of (failing to achieve goal \( \varphi \) and thus attain) \( \bar{\varphi} \). We can also write \( T_{\text{flight}} <_{T_{\text{flight}},(\varphi)} T_{\text{flight}}' \) meaning that the fear of \( \varphi \) gives agent \( i \) a greater tendency to flight and to fight. These example are interpreted as follows:

\[ M, s \models \text{Hope},(\varphi) <_{\text{OFI}} \text{Fear},(\bar{\varphi}) \quad \text{iff} \quad \text{OFI(Hope, } i, \varnothing)(M, s) < \text{OFI(Fear, } i, \bar{\varphi})(M, s) \]

\[ M, s \models T_{\text{flight}} <_{T_{\text{flight}},(\varphi)} T_{\text{flight}}' \quad \text{iff} \quad T_{\text{flight}}(\text{Fear, } i, \varnothing)(M, s) < T_{\text{flight}}'(\text{Fear, } i, \varnothing)(M, s) \]

Note that the concept of action tendency will be formalized in Section 7.2, thereby grounding the last example.

In the next section we will investigate how the notion of parameters of emotional experience can be used to define formulas with arbitrary emotion words in the object language.
which emotion words can be classified as belonging to the emotion type in question (see also the box on page 21). For example, ‘worried’, ‘scared’, and ‘petrified’ are emotions of the fear type, whereas ‘irritation’, ‘exasperation’, and ‘rage’ are emotions of the anger type. Although we now have a definition of \textit{Fear}(\varphi), we do not yet have a way to express that agent \textit{i} is \textit{scared} of consequence \varphi, which would naturally be denoted as \textit{scared}(i, \varphi) in our formal language. In this section, we present a way to formalize arbitrary emotion words, inspired by the psychological work of Frijda, Ortony, Sonnemans, and Clore [1992].

6.4.1 The Logical Structure of Tokens

The OCC model is not (and does not attempt to be) exhaustive with respect to which tokens belong to which types. As can be checked in Appendix A, for some emotion types (e.g., joy, distress, fear, anger) a large list of tokens is provided followed by “etc.”. For other emotion types (e.g., relief, gloating, pride), no other tokens fitting the emotion type in question appear to exist; at least not in the English language. For yet other emotion types (e.g., fears-confirmed, happy-for), no single-word (English) tokens appear to exist at all. It may be interesting to note that the negative emotion types usually have the most tokens.

To capture the fact that certain tokens belong to a certain type of emotion, a function \textit{Tokens} can be defined for each emotion type, indicating which words can be used to describe an emotion of the type in question. For example, the following tokens for ‘fear’ and ‘anger’ are mentioned in both [Ortony et al., 1988] and [Frijda et al., 1992]:

\begin{align}
\text{Tokens}(\text{Fear}) & \supseteq \{ \text{anxious}, \text{apprehensive}, \text{cowering}, \text{dread}, \text{fear}, \\
& \qquad \text{nervous}, \text{petrified}, \text{scared}, \text{terrified}, \text{timid}, \text{worried} \} \\
\text{Tokens}(\text{Anger}) & \supseteq \{ \text{anger}, \text{annoyed}, \text{exasperated}, \text{furious}, \\
& \qquad \text{incensed}, \text{irritated}, \text{livid}, \text{outraged}, \text{rage} \}
\end{align}

Note that, by convention, we write all types with a first upper case character whereas tokens begin with a lower case character. Also note that, as expected, the type name is also among its tokens (i.e., \textit{fear} \in \text{Tokens}(\text{Fear})). We write \supseteq in the examples above because these lists of tokens are not exhaustive. Moreover, the function \textit{Tokens} is supposed to be language-independent; for example, one could include \text{Tokens}(\text{Fear}) \supseteq \{ \ldots, \text{furcht}, \text{paura}, \text{miedo} \}. At this point it does not matter that different tokens may have additional or differing connotations; this is just a summarization of tokens. However, the idea of different tokens having different connotations will become crucial later on.

For now, let us focus on the logical structure of emotion words. According to our inheritance-based perspective of the emotion types (see Figure 2.1 on page 31), each emotion type is a generalization of the emotion type(s) below it. Consequently, the tokens of each emotion type must at least include all those of its specializations. For example, all tokens for ‘joy’ and ‘hope’ can be used to describe an emotion of type ‘pleased’, because they are specializations of ‘pleased’. In order to enforce this inheritance of tokens, the function \textit{Tokens} must be constrained as follows.
Constraint 6.15 (Inheritance hierarchy of emotion types)
The function Tokens assigning emotion words to emotion types is constrained in accordance with the emotion type hierarchy depicted in Figure 2.1, as follows:

\[
\begin{align*}
\text{Tokens(Positive)} & \supseteq \text{Tokens(Pleased)} \cup \text{Tokens(Approving)} \cup \text{Tokens(Liking)} \\
\text{Tokens(Negative)} & \supseteq \text{Tokens(Displeased)} \cup \text{Tokens(Disapproving)} \cup \text{Tokens(Disliking)} \\
\text{Tokens(Pleased)} & \supseteq \text{Tokens(Hope)} \cup \text{Tokens(Joy)} \\
\text{Tokens(Displeased)} & \supseteq \text{Tokens(Fear)} \cup \text{Tokens(Distress)} \\
\text{Tokens(Approving)} & \supseteq \text{Tokens(Pride)} \cup \text{Tokens(Admiration)} \\
\text{Tokens(Disapproving)} & \supseteq \text{Tokens(Shame)} \cup \text{Tokens(Reproach)} \\
\text{Tokens(Joy)} & \supseteq \text{Tokens(Gratification)} \cup \text{Tokens(Gratitude)} \cup \text{Tokens(Satisfaction)} \cup \text{Tokens(Relief)} \cup \text{Tokens(Happy-for)} \cup \text{Tokens(Gloating)} \\
\text{Tokens(Distress)} & \supseteq \text{Tokens(Remorse)} \cup \text{Tokens(Anger)} \cup \text{Tokens(Fears-confirmed)} \cup \text{Tokens(Disappointment)} \cup \text{Tokens(Resentment)} \cup \text{Tokens(Pity)} \\
\text{Tokens(Pride)} & \supseteq \text{Tokens(Gratification)} \\
\text{Tokens(Shame)} & \supseteq \text{Tokens(Remorse)} \\
\text{Tokens(Admiration)} & \supseteq \text{Tokens(Gratitude)} \\
\text{Tokens(Reproach)} & \supseteq \text{Tokens(Anger)}
\end{align*}
\]

Note that the multiple inheritance of the compound emotion types (‘gratification’, ‘remorse’, ‘gratitude’, ‘anger’) is captured by having their tokens “inherited” by ‘joy’ and ‘distress’ on the one hand, and ‘pride’, ‘shame’, ‘admiration’, and ‘reproach’ on the other hand.

6.4.2 Formalizing Emotional Experience for any Token

Now let us return to the observation that different emotion words classified as belonging to the same emotion type can have (and most often, will have) different connotations. For example, one connotation that the token ‘cowering’ adds to the type ‘fear’ is that the experiencing agent has a high tendency to flee. In contrast, ‘petrified’ entails a low fleeing tendency but high overall felt intensity. Here it is assumed that every possible “connotation” can be expressed as a combination of a parameter of emotional experience (in the sense of exp-param) and a threshold value.

To formalize such connotations, the function \( \Theta \) is introduced. When applied to a token, it returns a set of parameter–threshold tuples. For example:

\[
\begin{align*}
\Theta(\text{cowering}) & \supseteq \{ \langle \text{OFI}, 0.4 \rangle, \langle \text{T_{flight}}, 0.9 \rangle, \langle \text{T_{fight}}, 0.01 \rangle \} \\
\Theta(\text{petrified}) & \supseteq \{ \langle \text{OFI}, 0.8 \rangle, \langle \text{T_{flight}}, 0.01 \rangle, \langle \text{T_{fight}}, 0.1 \rangle \} \\
\Theta(\text{scared}) & \supseteq \{ \langle \text{OFI}, 0.6 \rangle, \langle \text{T_{flight}}, 0.4 \rangle, \langle \text{T_{fight}}, 0.4 \rangle \}
\end{align*}
\]
Here $\Theta$ specifies that for a fear emotion to be called “cowering,” its overall felt intensity must be greater than 0.4, the tendency to flee must be greater than 0.9, the tendency to fight must be greater than 0.01, etc.\(^3\) Note that OFI, $T_{flight}$, and $T_{fight}$ are parameters of emotional experience; that is, OFI, $T_{flight}$, $T_{fight} \in \text{EXP-PARAM}$. Each threshold value is a positive real number.

It should be noted that $\Theta$ does not have to associate a threshold with each $P \in \text{EXP-PARAM}$; if no threshold is specified for a certain emotion word and parameter of emotional experience, this just means that the emotion word in question has no connotations regarding the parameter in question. For example, if one does not consider the token ‘worried’ to say anything about a tendency to flee out of fear, then this is modeled by having $\Theta(\text{worried})$ not contain a tuple $(T_{flight}, \theta)$.

The function $\Theta$ and the threshold values it returns can be set according to psychological research into relations among emotion words, such as [Frijda et al., 1992]. In the following we will not be concerned with investigating possible connotations of emotion words and specifying $\Theta$ accordingly. Instead, an appropriate specification of $\Theta$ is assumed as given.

Given these parameter–threshold tuples as specified by $\Theta$, we can now apply parameters to types as well as tokens, in a way very similar to Definition 6.12.

**Definition 6.16 (Parameters of emotional experience and thresholds)**

For all emotion types $\text{Emotion} \in \text{emo-types}$, for all tokens $\text{emotion} \in \text{Tokens(Emotion)}$, and for all $(P, \theta) \in \Theta(\text{emotion})$, the parameter $P$ holds for token $\text{emotion}$ if and only if the value the parameter returns when applied to $\text{Emotion}$ is greater than the threshold $\theta$:

$$M, s \models P(\text{emotion}, i, \bar{o}) \iff P(\text{Emotion}, i, \bar{o})(M, s) > \theta \quad (6.42)$$

For example, if $(T_{flight}, 0.4) \in \Theta(\text{scared})$, this means that for any agent to be called ‘scared’ it must have a tendency to flee\(^4\) out of fear greater than 0.4. This connotation of ‘scared’ is then interpreted as follows:

$$M, s \models T_{flight}(\text{scared}, i, \varphi) \iff T_{flight}(\text{Fear}, i, \varphi)(M, s) > 0.4$$

It is now easy to define emotional experience for any emotion word. We have already defined what the experience of an emotion of a certain type (e.g., $\text{Fear}(\varphi)$) means, namely by way of Definition 6.9. We are now in a position to define the interpretation of, e.g., $\text{scared}(\varphi)$ (which expresses that agent $i$ is scared of $\varphi$).

**Definition 6.17 (Emotional experience with respect to specific tokens)**

A specific emotion labeled as $\text{emotion}$ is experienced if and only if its type is experienced and all parameters of emotional experience associated with $\text{emotion}$ are satisfied. That is, for all types $\text{Emotion} \in \text{emo-types}$ and for all tokens $\text{emotion} \in \text{Tokens(Emotion)}$:

$$\text{emotion}_i(\bar{o}) \overset{\text{def}}{=} \text{Emotion}_i(\bar{o}) \land \bigwedge_{(P, \theta) \in \Theta(\text{emotion})} P(\text{emotion}, i, \bar{o}) \quad (6.43)$$

\(^3\)For the purpose of illustration, it is assumed values are scaled between 0 and 1, but this is not necessary. The figures for overall felt intensity (OFI) have been chosen after [Frijda et al., 1992]; other figures have been made up for the sake of illustration.

\(^4\)The action $\text{flight}$ can be seen as a special action that initiates a fleeing behavior.
To illustrate this definition, let us write out the interpretation of $\text{scared}_i(\varphi)$ using the semantics, given that $\Theta(\text{scared}) = \{\langle \text{OFI}, 0.6 \rangle, \langle T_{\text{flight}}, 0.4 \rangle, \langle T_{\text{fight}}, 0.4 \rangle\}$:

\[
M, s \models \text{scared}_i(\varphi) \ \text{iff} \ \ M, s \models \text{Fear}_i(\varphi) \ \\
\text{and} \ OFI(\text{Fear}, i, \varphi)(M, s) > 0.6 \ \\
\text{and} \ T_{\text{flight}}(\text{Fear}, i, \varphi)(M, s) > 0.4 \ \\
\text{and} \ T_{\text{fight}}(\text{Fear}, i, \varphi)(M, s) > 0.4
\]

So an agent is ‘scared’ when all corresponding threshold, as specified by $\Theta$, are exceeded. This exceeding of thresholds then models the satisfaction of all connotations associated with the word ‘scared’. On a side note, the term “$M, s \models \text{Fear}_i(\varphi)$” in this example may be replaced by “$M, s \models \text{Past Fear}_i(\varphi)$,” because the next term “$\text{OFI}(\text{Fear}, i, \varphi)(M, s) > 0.6$” already entails the second of the two conjuncts constituting $\text{Fear}_i(\varphi)$ (cf. Definitions 6.9 and 6.12).

### 6.4.3 Orderings of Tokens

If we look back at the tokens in formulas (6.40) and (6.41), we see that they have been ordered alphabetically. However, there are obviously more interesting orderings possible. For example, ‘petrified’ should not come before ‘scared’ if the fear tokens were ordered by increasing typical intensity [Frijda et al., 1992]. On the other hand, ‘petrified’ does probably come before ‘scared’ if they were ordered by increasing typical tendency to flee. Throughout this chapter we have refrained from assigning specific values to emotional experience and its parameters, instead focusing on the logical structure underlying emotional experience. By studying orderings of emotion words we can again capture some of the logical structure of emotion without being concerned with (individual-dependent) numerical values.

In particular, tokens of the same emotion type can be ordered with respect to some parameter of emotional experience by comparing the threshold values associated with these parameters for these tokens. For example, ‘petrified’ and ‘scared’, both of type ‘fear’, can be ordered by overall felt intensity because $\langle \text{OFI}, 0.8 \rangle \in \Theta(\text{petrified})$ and $\langle \text{OFI}, 0.6 \rangle \in \Theta(\text{scared})$; then ‘scared’ should come before ‘petrified’, denoted as $\text{scared} \prec_{\text{OFI}} \text{petrified}$, because $0.6 < 0.8$. This is formalized for arbitrary tokens as follows.

**Definition 6.18 (Orderings of tokens)**

For all emotion types $\text{Emotion} \in \text{emo-types}$ and for all parameters of emotional experience $P \in \text{exp-param}$, an ordering $\prec_{\text{Emotion}}^P$ is defined as follows. For all $\text{emotion}_1, \text{emotion}_2 \in \text{Tokens(Emotion)}$, for all $\langle P_1, \theta_1 \rangle \in \Theta(\text{emotion}_1)$, and for all $\langle P_2, \theta_2 \rangle \in \Theta(\text{emotion}_2)$, where $P_1 = P_2 = P$:

\[
\text{emotion}_1 \prec_{\text{Emotion}}^P \text{emotion}_2 \ \text{iff} \ \theta_1 < \theta_2
\]  (6.44)

To illustrate how this definition can be used, let us consider the tokens given for the emotion types ‘fear’ and ‘anger’. In a survey presented by Frijda et al. [1992], participants were asked to rate the typical intensity they associate with a given set
of emotion words associated with fear and anger. The results of this survey can be formalized as constraints on orderings by ‘overall felt intensity’; that is, as constraints on \( \prec_{\text{Fear}} \) and \( \prec_{\text{Anger}} \).

**Constraint 6.19 (Orderings of ‘fear’ and ‘anger’ by overall felt intensity)**

The orderings \( \prec_{\text{Fear}} \) and \( \prec_{\text{Anger}} \) are constrained in accordance with the findings presented by Frijda et al. [1992] as

\[
\begin{align*}
\text{timid} & \prec_{\text{Fear}} \text{cowering} & \text{cowering} & \prec_{\text{Fear}} \text{apprehensive} & \text{apprehensive} & \prec_{\text{Fear}} \text{worried} & \text{worried} & \prec_{\text{Fear}} \text{fear} & \text{fear} & \prec_{\text{Fear}} \\
\text{nervous} & \prec_{\text{Fear}} \text{scared} & \text{scared} & \prec_{\text{Fear}} \text{anxious} & \text{anxious} & \prec_{\text{Fear}} \text{dread} & \text{dread} & \prec_{\text{Fear}} \text{terrified} & \text{terrified} & \prec_{\text{Fear}} \\
\text{annoyed} & \prec_{\text{Anger}} \text{irritated} & \text{irritated} & \prec_{\text{Anger}} \text{incensed} & \text{incensed} & \prec_{\text{Anger}} \text{exasperated} & \text{exasperated} & \prec_{\text{Anger}} \text{livid} & \text{livid} & \prec_{\text{Anger}} \\
\text{angry} & \prec_{\text{Anger}} \text{furious} & \text{furious} & \prec_{\text{Anger}} \text{outraged} & \text{outraged} & \prec_{\text{Anger}} \text{rage} & \text{rage} & \text{(6.45)}
\end{align*}
\]

Of course, choosing a different parameter of emotional experience may result in a very different ordering. For example, the example for \( \Theta \) on page 133 results in the following ordering of ‘cowering’, ‘petrified’, and ‘scared’ by typical tendency to flee:

\[
\text{petrified} \prec_{\text{Fear}_T \text{flight}} \text{scared} \prec_{\text{Fear}_T \text{flight}} \text{cowering}
\]

So from this ordering we see that an emotion of type ‘fear’ cannot be called ‘scared’ if the tendency to flee out of fear is at most that of ‘petrified’.

### 6.5 Integration in KARO

In order to finalize the formalization of emotional experience, the state-dependent functions introduced in this chapter must be integrated into the KARO framework used in the preceding chapters. This integration is the subject of this section.

Recall from Chapter 5 that KARO formulas are interpreted using nested Kripke models. Specifically, Kripke models of belief are nested inside Kripke models of action, which is done to allow belief changes through actions to be modeled. Actions also provide an abstraction of time, because actions are assumed to take time. Thus action models are the most top-level structures in KARO.

According to Definition 5.5, an action model is denoted as \( M \) and consists of a set \( S \) of belief model–state pairs, a set \( R \) of action-accessibility relations on \( S \), a structure \( \text{Aux} \) of auxiliary functions (for goals, capabilities, and commitments), and a structure \( \text{Emo} \) of appraisal functions (for desirability, praiseworthiness, appealingness, and cognitive unit). To accommodate the functions necessary for interpreting emotional experience, action model \( M \) is extended with a structure \( \text{Exp} \) containing these functions. So now an action model is a structure \( M = \langle S, R, \text{Aux}, \text{Emo}, \text{Exp} \rangle \). This additional structure \( \text{Exp} \) then contains the following functions and sets:

\[
\text{Exp} = \langle \text{Time}, \text{IntFun}, \text{Potential}, \text{Threshold}, \text{vars}, \text{exp-param}, \text{AllTriggered}, \text{AllExperienced} \rangle
\]
where \( \text{Time} \) is the time-stamping function, \( \text{IntFun} \) is the function assigning intensity functions to emotions, \( \text{Potential} \) and \( \text{Threshold} \) are the emotion potential and threshold functions, \( \text{vars} \) is the set of variables affecting intensity, \( \text{exp-param} \) is the set of parameters of emotional experience, and \( \text{AllTriggered} \) and \( \text{AllExperienced} \) are the functions returning all emotions triggered and experienced per state. Other sets and functions introduced in this chapter are not dependent on states and actions, so they do not have to be included in the action models. The inclusion of \( \text{Exp} \) in \( M \) then formally finalizes the formalization of emotional experience.

As a final remark, it may be interesting to note that throughout this chapter a number of constructs have been used that together could be said to define the emotional “characters” of agents. For example, there have been the weights for the variables affecting intensity: \( \{w_{\text{var}} | \text{var} \in \text{vars}\} \). Also differing per individual agent are the settings of the durations (\( \delta \)) and fall-off rates (\( \lambda \)) of the emotion intensity functions. A detailed analysis of the contents of the data block labeled as \textit{individual parameters} in Figure 6.1 would have to consider these elements as possible contents, because the real magnitudes of all valenced reactions are determined by how these are set.

### 6.6 Concluding Remarks

To the best of our knowledge, formally distinguishing between emotion elicitation and emotional experience is quite unique in computer science literature on emotions. Our first own work introducing a formalization of this distinction was presented in [Steunebrink et al., 2008a]. In that paper a method was introduced by which emotion elicitation and emotional experience could both be represented in the same logical language. It should be noted that the work of this chapter follows a similar line (for example, properties like in Proposition 6.10 were already present in [Steunebrink et al., 2008a]), but the current approach is much more refined and comprehensive.

Although some related works do attempt to calculate intensities for certain types of emotion, most often these calculations are quite ad hoc, because only limited means are available in the target application. Although our approach also suffers from limited means (by using only BDI concepts), we have at least rigourously followed a comprehensive psychological model of human emotion, using the familiar BDI constructs where possible, and leaving the rest open for future extensions.

Finally, let us spend a few words on emphasizing the usefulness of distinguishing between emotion elicitation and emotional experience. According to our procedural view of emotion, as illustrated in Figure 6.1, \textit{experienced emotions} come about through the process of \textit{intensity assignment}, which uses \textit{triggered emotions} as input. It is important to note that the process of \textit{intensity assignment} does not attempt to assign a (possibly zero) emotional intensity to just any percept. The process of \textit{emotion elicitation} serves the intermediate function of “filtering” the percepts for salience (by holding them against the concerns) before passing them on to \textit{intensity assignment}. So the emotions triggers, as formalized in Chapters 3, 4, and 5, indicate which percepts to consider at all before starting the (probably costly) process of calculating emotion intensities. Note that this idea becomes apparent in Definition 6.4, where
only triggered emotions are assigned initial intensity values.
Chapter 7

Emotion Regulation

When emotion is entirely left out of the reasoning picture, as happens in certain neurological conditions, reason turns out to be even more flawed than when emotion plays bad tricks on our decisions.
– Antonio Damasio [1994]

Having formalized when emotions are triggered (Chapters 3, 4, and 5) and how intensely they are experienced (Chapter 6), we turn to the possible effects of emotions on behavior in this chapter. In psychological literature, the link between emotion and action selection is usually called emotion regulation [Gross, 2007]. Note that this term is used to mean both the regulation of action selection by emotions, and the regulation of emotions by performing certain actions. There is a circularity in here though: emotions can be seen as regulating action selection in order to regulate the emotions themselves. Of course the “purpose” of emotions is not to keep themselves in balance, but to keep the experiencing agent’s environment in balance. Emotions thus not only indicate what is right and wrong in the agent’s environment, but they also suggest ways to keep the good aspects and fix the bad aspects.

The investigation into how exactly emotions influence action selection is performed from two different perspectives. The first is a more theoretical perspective, treated in Section 7.2. In that section the notion of emotion-induced action tendency is discussed and formalized, inspired by the psychological works of Frijda [1987] and Gross and Thompson [2007]. (Note that we cannot use the OCC model here, because emotion regulation is not treated in the OCC model.) The second is a more applied perspective, treated in Section 7.3. In that section we look at how the action selection of BDI-based agents is often implemented, and then investigate how our formal model of emotions can be used to influence this action selection. However, before going into formal details, let us first revisit the procedural view of emotion in order to get a feeling for how the work of this chapter fits in the bigger picture.
7.1 The Procedural View of Emotion Revisited

For ease of reference the illustration of our procedural view of emotion is repeated in Figure 7.1. In Chapters 3, 4, and 5 we have been concerned with formalizing the Appraisal part of Figure 7.1. In Chapter 6 we have been concerned with the Experience part. In this chapter, then, we move on to the Effects on behavior part. This part consists of a process called emotion regulation which influences the experienced emotions of an agent but also influences the selection of actions. Indeed, performing an action can influence the experience of emotions, so these two aspects of emotion regulation are strongly tied. For example, a fire can cause fear to be elicited, which in turn induces a tendency to flee away from the danger. But upon reaching safety, the fear will subside, because the danger to the self has been averted (i.e., the situation has been “fixed”). Thus the action selected under pressure of the fear is also the action that is supposed to lead to a state where the fear is mitigated.

It should be noted that besides emotion, also mood can be regulated [Clore et al., 2001; Gross, 2007]. However, this fact has not been incorporated in Figure 7.1 (e.g., by drawing an additional rounded box saying “mood regulation” and pointing at “mood”) because only emotion regulation is discussed in this dissertation. On the subject of affect regulation, one may further wonder whether other aforementioned
affective phenomena (impulses, stress, expressions, emotional disorders, temperament) can be regulated. In the classification by Gross and Thompson [2007] described in Chapter 1, impulses are characterized as being inflexible, although some limited learning through conditioning may be possible. The regulation of stress, usually called coping, is a great source of income for psychiatrists and therapists; however, since coping with stress is not directed at the regulation of individual emotions, this falls outside the scope of this dissertation on emotion. With respect to Figure 1.1, expressions and autonomic changes are reflex-like and not subject to cognitive regulation (although they may change over time through conditioning). The regulation of emotional disorders and personality traits falls outside the scope of this dissertation for the same reasons as stress. So indeed many types of affect regulation can be distinguished and studied, but in this chapter we will focus on the regulation of emotions proper.

7.2 Action Tendency

In this section we investigate the effects of emotions on behavior from the perspective of action tendency.

Following Frijda, action tendencies are “states of readiness to execute a given kind of action, [which] is defined by its end result aimed at or achieved” ([Frijda, 1987, page 70]). In the case of negative emotions, reaching the associated end state should mitigate its experience (e.g., fear subsides once one believes the object of one’s fear cannot reach oneself anymore), whereas positive emotions generally put an individual in a “mode of relational action readiness” (e.g., joy can put one in a mode of readiness for new interactions). Action tendencies in response to negative emotions are particularly powerful, because these tendencies signal that some things are not as they should be, making it possible to identify actions that can fix the situation. Indeed, in our formalization, the experience of negative emotions will be constrained such that for some actions, reaching their end state removes the emotional experience. Action tendencies will then follow naturally in response to emotional experience.

It should be noted that action tendencies differ from intentions in that they are not goal-directed, but rather stimulus-driven [Frijda, 1987]. For example, fear does not (necessarily) spawn a goal to flee towards safety, but rather gives the urge to flee away from the perceived danger. The “end state” of safety does matter, however, in determining when the fear is supposed to subside. An example of several emotion types and their associated action tendencies and end states is provided in Table 7.1. Note that not all emotion types have a clearly identifiable end state; these are usually positive emotions.

The notion of action tendency that we formalize is based on the following idea. Consider again the example from the introduction where Alice is watching a horror movie, which happens to be very distressing for her. Psychologists such as Gross and Thompson [2007] have made classifications of the kinds of strategies that Al-
ice could employ to lessen her distress. For example, she could mute the sound of the movie (“situation selection”); she could close her eyes during a gory scene (“attentional deployment”); she could tell herself the blood and the storyline are all fake (“cognitive change”); or she could take refuge in drinking or smoking to vent the distress (“response modulation”). There is one common thread in all of these emotion regulation strategies: the regulatory actions that Alice can perform all serve to decrease the overall felt intensity of her distress. This common thread is captured by the notion of action tendency. If Alice believes that muting the sound of the movie will lessen her distress, then she will have the tendency to mute the sound. So action tendencies point to ways to “fix” a situation. Just as emotions arise from concerns being compromised or achieved, action tendencies indicate how the situation can be improved by mitigating the grievance or extending the celebration. The notion of action tendency that we formalize is described generally as follows:

If an agent expects that performing a particular action will decrease the overall felt intensity of a negative emotion, or increase the overall felt intensity of a positive emotion, then that agent has the tendency to perform that action.

Note that this notion of action tendency is always relative to a particular emotion. For example, Alice’s tendency to mute the sound of the horror movie is relative to her distress from watching the movie.

**Notational Preliminaries**

Before moving on to the formal details, we briefly introduce some notation that will be used in the rest of this section.

In the following we use the notion of expected value from Probability Theory. Let us therefore recall that the expected value of a function $f$, given a probability

---

2In humans, the “belief” that some action can change the intensity of an emotion does not have to be explicit. Probably many emotion regulation strategies do not have to be cognitively selected; they are learned, automated responses. In future work the current framework may be refined with a distinction between explicit and implicit beliefs such that action tendencies can also follow from implicit beliefs.
Distribution $p$ and a (discrete) sample space $\Omega$, is defined as:

$$E_p[f] = \sum_{x \in \Omega} f(x) \cdot p(x)$$

Note that for $p$ to be a probability distribution on $\Omega$, it has to satisfy the following two constraints:

- $\forall x \in \Omega : p(x) \geq 0$ (probabilities are non-negative);
- $\sum_{x \in \Omega} p(x) = 1$ (probabilities sum to one).

Finally, recall that for all $(M, s) \in S$, $M = (S, R, V)$ is a belief model, where $R$ is a set of belief-accessibility relations on $S$. That is, for each $R_i \in R$, we have that $R_i \subseteq S \times S$. However, in the following we would like each $R_i$ to be a relation on $S$, i.e., on the possible worlds of the action model. For this we will write $R^M_i$, which is defined as:

$$R^M_i = \{ ((M, s_1), (M, s_2)) | (s_1, s_2) \in R_i \} \quad (7.1)$$

Now we have that $R^M_i$ is a relation on $S$, i.e., $R^M_i \subseteq S \times S$.

### 7.2.1 Formalizing Action Tendency

Let us now move on to the formal details. Recall from Section 6.3.2 that, like ‘overall felt intensity’, ‘action tendency’ is considered to be a parameter of emotional experience [Frijda et al., 1992]. This means that we already have an interpretation of action tendencies in our object language. Specifically, according to Definition 6.12 we have the following interpretation:

$$M, s \models T_\alpha(\text{Emotion}, i, \bar{o}) \quad \text{iff} \quad T_\alpha(\text{Emotion}, i, \bar{o})(M, s) > 0 \quad (7.2)$$

Here $T_\alpha(\text{Emotion}, i, \bar{o})$ can be filled in with any action and emotion instance; for example, the proposition $T_{\text{flight}}(\text{Fear}, Alice, bitten \text{ by spider})$ is read as “Alice’s fear of being bitten by the spider gives her the tendency to flee.” As an example of a ‘positive’ action tendency, the proposition $T_{\text{touch}}(\text{Liking}, Bob, fluffy kitten)$ can liberally be read as “Bob liked the fluffy kitten so much that he wanted to touch it.”

The question posed by formula (7.2) is how the function $T$ is defined. From our (informal) specification of action tendency above, we can see that action tendency has to do with beliefs about the possible results of actions. In particular, an agent experiencing a certain emotion must be able to form expectations about the overall felt intensity of that emotion in the possible states resulting from performing an action. If this expected resulting overall felt intensity is better than currently (i.e., lower for negative emotions, higher for positive emotions), then the agent has the tendency to perform the action for the emotion. In order to be able to capture expectations about the results of actions, we introduce the function $AcPb$, which will then be used in the definition of the action tendency function $T$. 


Definition 7.1 (AcPb as discrete probability distribution on presumed results of actions)
Let $AcPb : \mathcal{A} \times \mathcal{A} \rightarrow S \rightarrow S \rightarrow \mathbb{R}$ be a function assigning subjective probabilities to possible results of actions. $AcPb(i, \alpha)(M, s)(M', s') = x$ means that agent $i$ considers the probability that performing action $\alpha$ in state $(M, s)$ will result in state $(M', s')$ to be $x$. The function $AcPb(i, \alpha)(M, s)$ of type $S \rightarrow \mathbb{R}$ is constrained as usual in Probability Theory such that it is a discrete probability distribution with $S$ as sample space. That is, for all agents $i \in \mathcal{A}$, actions $\alpha \in \mathcal{A}$, and model–state pairs $(M, s) \in S$, $AcPb$ satisfies the following constraints:

- $\forall (M', s') \in S : AcPb(i, \alpha)(M, s)(M', s') \geq 0$ (i.e., probabilities are non-negative);

- $\sum_{(M', s') \in S} AcPb(i, \alpha)(M, s)(M', s') = 1$ if $\exists (M', s') \in S : ((M, s), (M', s')) \in R^M_i \circ R^s_{ia}$ (i.e., probabilities sum to one, provided that there exists a presumed possible result of action $i: \alpha$ at all);

- $AcPb(i, \alpha)(M, s)(M', s') = 0$ if $((M, s), (M', s')) \notin R^M_i \circ R^s_{ia}$ (i.e., states that are not held as possible results of action $i: \alpha$ are assigned zero probability).

So $AcPb(i, \alpha)(M, s)$ is a discrete probability distribution on $S$, indicating for each possible state $(M', s') \in S$ the subjective probability for agent $i$ of the execution of action $\alpha$ in state $(M, s)$ resulting in state $(M', s')$. Note the usage of $R^M_i \circ R^s_{ia}$, which is a relation between model–state pairs that can be reached by taking one belief ‘step’ and then one action ‘step’. Thus $R^M_i \circ R^s_{ia}$ indicates which states are held by agent $i$ as possible results of performing action $\alpha$. In other words, if $((M, s), (M', s')) \in R^M_i \circ R^s_{ia}$ and $M', s' \models \varphi$, then $M, s \models \neg B_i \neg (i: \alpha) \varphi$.

The introduction of $AcPb$ furthers our quest to define the action tendency function $T$ as follows. The (informal) specification of action tendency above requires us to be able to calculate the expected difference in overall felt intensity resulting from performing an action. Using $AcPb$ the expected overall felt intensity after performing action $i: \alpha$ in a state $(M, s)$ can be calculated using the following function.

Definition 7.2 (Expected overall felt intensity)
For each action $\alpha \in \mathcal{A}$, $E_{OFI_\alpha}$ is a function of type $\text{Emotion} \times S \rightarrow \mathbb{R}_{\geq 0}$, indicating per emotion and model–state pair the expected overall felt intensity of that emotion after performing action $\alpha$ in that state. $E_{OFI_\alpha}$ is defined as follows:

$$E_{OFI_\alpha}(\text{Emotion}, i, \partial)(M, s) = E_p[OFI(\text{Emotion}, i, \partial)]$$

(7.3)

where $p = AcPb(i, \alpha)(M, s)$ is the probability distribution and $S$ is the sample space of the expectation operator.

Note that, given the constraints on $AcPb$ above, the term $E_p[OFI(\text{Emotion}, i, \partial)]$ is equivalent to:

$$\sum \{ OFI(\text{Emotion}, i, \partial)(M', s') \cdot AcPb(i, \alpha)(M, s)(M', s') \mid ((M, s), (M', s')) \in R^M_i \circ R^s_{ia} \}$$
A possible concern here is that the summation may be taken over the empty set, in which case $E_{OFI_{\alpha}}$ simply returns zero. This is not of concern in the following, however, because this calculation will only be used in a setting where it is guaranteed that there exists an $((M, s), (M', s')) \in R^M \circ R_{\alpha}$.

We now have all ingredients at hand to introduce the definition of the $T$ function, thereby formalizing the notion of action tendency. The action tendency function $T$ is defined as follows.

**Definition 7.3 (Action tendency)**

For each action $\alpha \in \text{act}$, $T_{\alpha}$ is a function of type $\emotion \times S \rightarrow \mathbb{R}_{\geq 0}$ indicating the magnitude of the tendency to perform $\alpha$. $T_{\alpha}$ is defined as follows:

$$T_{\alpha}(\emotion, i, \bar{o})(M, s) = \begin{cases} w \cdot \text{gain} & \text{if } M, s \models \emotion_i(\bar{o}) \land \text{Can}_i(\alpha, \top) \\ 0 & \text{otherwise} \end{cases}$$

(7.4)

where $w > 0$ is a weight and gain is determined as follows:

$$\text{gain} = \begin{cases} \max(0, \text{OFI}_{\text{diff}}) & \text{if } \emotion \in \text{emo-types}^+ \\ \max(0, -\text{OFI}_{\text{diff}}) & \text{if } \emotion \in \text{emo-types}^- \end{cases}$$

where

$$\text{OFI}_{\text{diff}} = E_{OFI_{\alpha}}(\emotion, i, \bar{o})(M, s) - OFI(\emotion, i, \bar{o})(M, s)$$

Let us explain this definition in more detail. In the first place, this definition ensures that an action tendency can only arise with respect to an emotion that is actually experienced and an action that can presumably be performed. This is ensured by the condition $M, s \models \emotion_i(\bar{o}) \land \text{Can}_i(\alpha, \top)$. The magnitude of the tendency to perform an action is specified to be proportional to the gain achieved by performing the action. The gain is then defined as the difference (OFI_{diff}) between the expected overall felt intensity after performing the action and the current overall felt intensity of the emotion. For the negative emotion types, this difference is negated so that we have a positive gain if the expected overall felt intensity is less than the current overall felt intensity of the emotion.

The weight $w$ (with $w > 0$) serves two purposes. One is to scale the magnitude of the action tendency such that it falls in the appropriate range. The range of the magnitudes of action tendencies is application-dependent and also dependent on the range of overall felt intensities. Since we are not concerned with specific applications in this dissertation, and because we have said nothing about the range of overall felt intensities, we cannot go any further into how a range for action tendencies might be set. The other purpose is to function as one of the parameters of the emotional ‘characters’ of agents. Recall from Figure 7.1 that emotions are affected by Individual parameters; these parameters govern the individual response propensities of agents. This weight is then supposed to account for part of the response propensities of the agent in question.\(^3\)

---

\(^3\)This means that, strictly speaking, there should be an arrow in Figure 7.1 from Individual parameters to Emotion regulation as well. We have omitted this in order not to clutter the diagram.
7.2.2 Properties

In this subsection we study some of the properties of our formalization of the notion of action tendency. Because this formalization depends on expectations about future intensities of emotions, it will be natural to bring this notion to the object language such that the strongest and most insightful propositions can be attained (i.e., equivalence of action tendency with something else). At this point it is crucial to note that the function $E_{OFI}$ for expected intensity has the same type as the functions $T_\alpha$ and $OFI$. This means that, just like ‘action tendency’ and ‘overall felt intensity’, the notion of ‘expected overall felt intensity’ can be represented in our object language by adding it to $\text{exp-param}$, which is the set of parameters of emotional experience. Definition 6.12 (see page 130) then takes care of the rest.\footnote{It may be argued here that we “abuse” Definition 6.12 because it may be dubious to consider ‘expected overall felt intensity’ as a parameter of emotional experience in the sense of [Frijda et al., 1992]. However, the mechanism provided by Definition 6.12 by which functions of type $\text{emotion} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ can be reified in the object language is seen here simply as a useful tool for constructing insightful properties of our formalization of ‘action tendency’.

Proposition 7.4 (Equivalences regarding action tendencies)

For each action $\alpha \in \text{act}$, let $E_{OFI_\alpha}$ be a parameter of emotional experience, i.e., \{ $E_{OFI_\alpha}$ $|$ $\alpha \in \text{act}$ \} $\subseteq$ $\text{exp-param}$. Then the following equivalences are valid:

\[
\begin{align*}
\models T_\alpha(\text{Emotion}^+, i, \bar{o}) & \iff \text{Emotion}^+_\bar{o}(\alpha) \land \text{Can}_i(\alpha, T) \land E_{OFI_\alpha} >_{\text{Emotion}^+_\bar{o}} \text{OFI} \quad (7.5) \\
\models T_\alpha(\text{Emotion}^-, i, \bar{o}) & \iff \text{Emotion}^-_{\bar{o}}(\alpha) \land \text{Can}_i(\alpha, \top) \land E_{OFI_\alpha} <_{\text{Emotion}^-_{\bar{o}}} \text{OFI} \quad (7.6)
\end{align*}
\]

where $\text{Emotion}^+ \in \text{emo-types}^+$ and $\text{Emotion}^- \in \text{emo-types}^-$.}

These two propositions express that an agent has the tendency to perform a certain action for a positive / negative emotion iff (1) the emotion is currently experienced by the agent, (2) the agent ‘can’ perform the action, and (3) the agent expects that performing the action will increase / decrease the overall felt intensity of the emotion. Recall from formula (6.39) on page 131 that, for example, $M, s \models E_{OFI_{\text{smoke}}} <_{\text{Distress}_{\text{Alice}}(\text{horror})} \text{OFI}$ abbreviates $E_{OFI_{\text{smoke}}}(\text{Distress}, \text{Alice}, \text{horror})(M, s) < \text{OFI}(\text{Distress}, \text{Alice}, \text{horror})(M, s)$, which can liberally be read as “Alice expects that after having smoked, the intensity of her distress from having watched the horror movie to be less than the current intensity.”

Even without using the $E_{OFI}$ function interesting properties of action tendency can be obtained. In particular, instead of looking forward using expectation, we can also look backward at what a certain action has accomplished for the overall felt intensity of an emotion. Because action tendency invariably concerns comparisons between overall felt intensities across different action-accessible states, it will be convenient to have a function at hand that returns the previous overall felt intensity of an emotion. This previous value can then be used in propositions that compare this value to the current overall felt intensity (as returned by the function $OFI$). The function $\text{PrevOFI} : \text{emotion} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ is defined very straightforwardly to return
the overall felt intensity of the provided emotion in the previous state:

\[
\text{PrevOFI}(\text{Emotion}, i, \delta)(M, s) = \begin{cases} 
\text{OFI}(\text{Emotion}, i, \delta)(M', s') & \text{if } ((M', s'), (M, s)) \in (\cup \mathcal{R}) \\
0 & \text{otherwise}
\end{cases}
\] (7.7)

Recall that the history was constrained to be linear, so if \(((M', s'), (M, s)) \in (\cup \mathcal{R})\) then \((M', s')\) must be the state before \((M, s)\). If no previous state exists, this means there was previously no intensity either, so then zero is returned by \(\text{PrevOFI}\). Just as in the properties above, we make use of the fact that any function of type \(\text{emotion} \times \mathcal{S} \rightarrow \mathbb{R} \geq 0\) can be reified in the object language by specifying \(\text{PrevOFI}\) to be a parameter of emotional experience. Indeed, this is exactly what is done in the following propositions.

**Proposition 7.5 (Properties of action tendency)**

Let \(\text{PrevOFI}\) be a parameter of emotional experience, i.e., \(\text{PrevOFI} \in \text{exp-param}\). Let \(\text{IntBel}\) be the constraint that emotion intensity functions, as returned by the function \(\text{IntFun}\), are invariant across belief-accessible states; that is, for all emotions \((\text{Emotion}, i, \delta) \in \text{emotion}\) and states \((M, s) \in \mathcal{S}\), \(\text{IntFun}(\text{Emotion}, i, \delta)(M, s) = \text{IntFun}(\text{Emotion}, i, \delta)(M, s')\) if \((s, s') \in R_i\) (where \(M = (\mathcal{S}, R, V)\)). Then the following propositions are valid:

\[
\text{IntBel} \models \text{OFI}(\text{Emotion}, i, \delta) \leftrightarrow \text{B}_i\text{OFI}(\text{Emotion}, i, \delta)
\] (7.8)

\[
\text{BD}^a_i \models \text{Emotion}^+_i(\overline{\delta}) \wedge \text{Can}_i(\alpha, \text{OFI} \triangleright \text{Emotion}^+_i(\overline{\delta}) \text{PrevOFI}) \rightarrow \text{T}_a(\text{Emotion}^+_i, i, \delta)
\] (7.9)

\[
\text{BD}^a_i \models \text{Emotion}^-_i(\overline{\delta}) \wedge \text{Can}_i(\alpha, \text{OFI} \triangleleft \text{Emotion}^-_i(\overline{\delta}) \text{PrevOFI}) \rightarrow \text{T}_a(\text{Emotion}^-_i, i, \delta)
\] (7.10)

\[
\text{BD}^a_i \models \text{Emotion}^-_i(\overline{\delta}) \wedge \text{Can}_i(\alpha, \neg \text{OFI}(\text{Emotion}^+_i, i, \delta)) \rightarrow \text{T}_a(\text{Emotion}^+_i, i, \delta)
\] (7.11)

\[
\text{IntBel} \models \text{Emotion}^-_i(\overline{\delta}) \wedge \text{B}_i[i: \alpha] \neg \text{OFI}(\text{Emotion}^+_i, i, \delta) \wedge \text{Can}_i(\alpha, \top) \rightarrow \text{T}_a(\text{Emotion}^+_i, i, \delta)
\] (7.12)

\[
\text{IntBel} \models \text{B}_i[i: \alpha][\text{OFI} \preceq \text{Emotion}^+_i(\overline{\delta}) \text{PrevOFI}] \rightarrow \neg \text{T}_a(\text{Emotion}^+_i, i, \delta)
\] (7.13)

\[
\text{IntBel} \models \text{B}_i[i: \alpha][\text{OFI} \succeq \text{Emotion}^+_i(\overline{\delta}) \text{PrevOFI}] \rightarrow \neg \text{T}_a(\text{Emotion}^+_i, i, \delta)
\] (7.14)

where \(\text{Emotion}^+ \in \text{emo-types}^+\) and \(\text{Emotion}^- \in \text{emo-types}^-\). Furthermore, the set of assumptions \(\text{BD}^a_i\) stands for \(\{\text{IntBel}, \text{Det}(i: \alpha)\}\), where \(\text{Det}(i: \alpha)\) is the assumption that action \(i: \alpha\) is deterministic (see page 98).

The assumption \(\text{IntBel}\) entails that actual emotion intensities are equal to presumed emotion intensities. This then leads immediately to the first proposition, which states that an emotion is felt if and only if it is believed to be felt (an agent is thus always ‘conscious’ of the overall felt intensity of an emotion). This first proposition does not talk about action tendencies, but it has been included to give a feeling for the \(\text{IntBel}\) assumption used in the other propositions.

The second proposition then states that action tendency with respect to a positive emotion arises when the emotion is currently experienced and the action ‘can’ be performed to yield a situation where the overall felt intensity of the emotion has increased. Analogously, the third proposition states that action tendency with respect to a negative emotion arises when the emotion is currently experienced and the action ‘can’ be performed to yield a situation where the overall felt intensity of
the emotion has decreased. The fourth proposition is a special case of the third proposition; it states that an agent has the tendency to perform an action for a negative emotion if the agent currently experiences the emotion and it 'can' perform the action to mitigate the experience of the negative emotion completely. (Note that $\neg \text{OFI}(\text{Emotion}, i, \delta)$ implies $\neg \text{Emotion}(\delta)$.) It should be noted that the second, third, and fourth proposition are only valid when the action in question is assumed to be deterministic. If this assumption is dropped, a property very much like the fourth proposition can still be formulated. Indeed, the fifth proposition reads exactly like the fourth one. Finally, the sixth and seventh propositions show when action tendencies do not arise. Specifically, if an agent believes that performing a certain action cannot result in an increase / decrease of the overall felt intensity of a positive / negative emotion, then the agent does not tend to perform this action for this emotion. We emphasize that in such a situation the agent could still perform the action in question anyway; indeed, a lack of tendency does not imply impossibility of execution.

Proofs of these propositions can be found in Section 7.5. Next we show how, for certain types of actions, constraints can be put on the framework, such that specific emotions can be shown to lead to action tendencies.

### 7.2.3 Investigating Specific Cases

Consider the most basic emotion regulation strategy: letting feelings subside by themselves. Since time is supposed to “heal all wounds” (and negative emotions in particular), the presented formalization of action tendency should straightforwardly capture tendency towards idling in response to negative emotions. An example of idling as an emotion regulation strategy is to count till ten before acting when feeling angry.

Formally, let idle denote the action that has no effects other than the passage of time. (Here it does not matter how long an agent will actually be idling; by Constraint 6.1 performing an idle action always causing at least some time to pass.) It is not unreasonable to assume that an agent always has the practical possibility of idling. As anticipated in Chapter 6, we also constrain the function IntFun such that performing idle does not cause any changes in intensity functions. Formally, these requirements translate to the following constraint.

**Constraint 7.6 (Properties of idling)**

For all states $(M, s) \in S$ and agents $i \in \text{agt}$, idle $\in \text{Caps}(i)(M, s)$ and $R_{\text{idle}}$ is serial.\(^5\)

Furthermore, the function IntFun is constrained such that performing idle does not cause any changes in intensity functions assigned to emotions. That is, for all $(\text{Emotion}, i, \delta) \in \text{emotion}$:

$$\forall((M, s), (M', s')) \in R_{\text{idle}} : \text{IntFun}(\text{Emotion}, i, \delta)(M, s) = \text{IntFun}(\text{Emotion}, i, \delta)(M', s')$$

\(^5\)Strictly speaking the constraint “$R_{\text{idle}}$ is serial” is in conflict with the earlier Constraint 5.7 of uniqueness of actions. What is actually meant is that for each state there exists a (unique) instance of the idle action that can be performed by the agent. For convenience of notation, however, we will not be so strict in explicitly distinguishing between different instances of the idle action.
Now let $IntSMD$ stand for the assumption that for all emotions and states, the intensity functions returned by $IntFun(\text{Emotion}, i, \overline{\delta})(M, s)$ are strictly monotonically decreasing.\textsuperscript{6} With these constraints we attain the following properties of action tendencies towards idling.

**Proposition 7.7 (Properties of tendencies towards idling)**
The following propositions are valid:

\begin{align*}
IntBel \models \neg T_{\text{idle}}(\text{Emotion}^+, i, \overline{\delta}) & \quad (7.15) \\
IntSMD, IntBel \models \text{Emotion}^-(\overline{\delta}) \leftrightarrow T_{\text{idle}}(\text{Emotion}^-, i, \overline{\delta}) & \quad (7.16)
\end{align*}

where $\text{Emotion}^+ \in \text{emo-types}^+$ and $\text{Emotion}^- \in \text{emo-types}^-$. The assumption $IntBel$ is the same as in Proposition 7.5.

The first proposition states that an agent *never* tends to idle with respect to a positive emotion. This is because our formalization of action tendency requires emotional intensity to *improve*, which for positive emotion types means an increase in overall felt intensity. But in Chapter 6 it was explicitly demanded that intensity functions be monotonically decreasing. Because intensity functions are not changed when performing an $\text{idle}$ action, no increase in overall felt intensity can occur, and thus no positive action tendency will arise with respect to $\text{idle}$ actions. The second proposition states that an agent *always* tends to idle in response to a negative emotion. Note that this does not mean that an agent experiencing a negative emotion will actually choose to idle. An agent experiencing a negative emotion may have many actions tendencies; all that formula (7.16) states is that idling will always be among them. Obviously, idling is only a valid strategy for negative emotions whose intensities actually decrease over time, hence the added requirement of strict monotonicity for the second proposition ($IntSMD$). We re-emphasize that the intensity function discussed in Chapter 6 satisfy this $IntSMD$ assumption.

The beautiful thing about tendencies with respect to idling is that idling is both conceptually and formally a very simple kind of action. Still, this very simple formalization of idling gives rise to two interesting cases of action tendency; namely, the two extremes of never tending and always tending. Because of its simplicity, the $\text{idle}$ action serves as a sort of “default” test for the formalization of action tendency.

The properties above with respect to tendencies to idle result largely from the constraint placed on $IntFun$ (Constraint 7.6). By placing appropriate constraints on the function $IntFun$, it is also possible to model action tendencies in response to emotions concerning other agents. In particular, recall that the so-called fortunes-of-others emotion types (‘happy-for’, ‘pity’, ‘gloating’, and ‘resentment’) involve presumed consequences of events for agents other than the self. Other emotion types that involve agents other than the experiencing agent are the action-based emotion types ‘admiration’, ‘reproach’, ‘gratitude’, and ‘anger’, as well as the object-based emotion types ‘liking’ and ‘disliking’. Below we establish relations between these emotion types through action tendencies.

\textsuperscript{6}All intensity functions discussed in the previous chapter were strictly monotonically decreasing, but this strictness was not a formal constraint.
In the following, it should be kept in mind that according to Chapters 3–5, pity or resentment can be triggered when an agent believes that another agent’s (sub)goal has been undermined (undone) or accomplished, respectively, and the agent views this as undesirable for itself. The (sub)goal that has been accomplished for the other agent \( j \) is denoted as \( \varphi \), or as \( \overline{\varphi} \) if it has been undermined. For example, the proposition \( \text{Pity}_i(\varphi, j) \) states that agent \( i \) pities agent \( j \) for the undermining of goal \( \varphi \). Furthermore, ‘gratitude’ or ‘anger’ can be triggered when a (sub)goal has been accomplished or undermined, respectively, by an action of another agent. For example, the proposition \( \text{Gratitude}_j(i, \alpha, \varphi) \) states that agent \( j \) is grateful towards agent \( i \) having performed action \( \alpha \) resulting in the achievement of (sub)goal \( \varphi \) of agent \( j \).

A reasonable constraint on \( \text{IntFun} \) would now be to require that the intensity of a pity emotion is decreased if the agent believes that the action it performs re-accomplishes the (sub)goal of the other agent that it pitied, which is exactly the case when gratitude is triggered in the other agent. Similarly, the intensity of a resentment emotion can be decreased if the agent believes that the action it performs undermines the (sub)goal of the other agent that it resented, which is exactly the case when anger is triggered in the other agent. Formally, we constrain \( \text{IntFun} \) as follows.

**Constraint 7.8 (Intensity changes in fortunes-of-others emotions)**

For all agents \( i \in \text{agt} \), actions \( \alpha \in \text{act} \), and all \( ((M, s), (M', s')) \in \mathcal{R}_{\text{act}} \):

\[
\text{IntFun}(\text{Pity}, i, (\varphi, j))(M', s') =
\begin{cases}
I_0 & \text{if } M', s' \models \mathbf{B} \text{Gratitude}_j(i, \alpha, \varphi) \\
\text{IntFun}(\text{Pity}, i, (\varphi, j))(M, s) & \text{otherwise}
\end{cases}
\]

and

\[
\text{IntFun}(\text{Resentment}, i, (\varphi, j))(M', s') =
\begin{cases}
I_0 & \text{if } M', s' \models \mathbf{B} \text{Anger}_j(i, \alpha, \overline{\varphi}) \\
\text{IntFun}(\text{Resentment}, i, (\varphi, j))(M, s) & \text{otherwise}
\end{cases}
\]

where \( I_0(x) = 0 \) for all \( x \).

This constraint thus expresses that the intensity function assigned to a ‘pity’ or ‘resentment’ emotion is left unchanged when the experiencing agent performs an action, except when that action results in the agent believing that ‘gratitude’ or ‘anger’ is triggered in the other agent towards which the ‘pity’ or ‘resentment’ emotion was directed. In this case, the overall felt intensity of the ‘pity’ or ‘resentment’ emotion is specified to be set to zero.\(^7\) With these constraints the following properties are attained.

\(^7\)It may be argued that this is a bit drastic, but the zero function \( I_0 \) is the simplest function guaranteed to return values less than the original intensity function. Of course, there is nothing against using a different, more complicated function instead of \( I_0 \), as long as it returns values less than the previous intensity function assigned to the emotion in question.
Proposition 7.9 (Properties of action tendency relative to fortunes-of-others emotions)
The following propositions are valid:

\[ BDA_i^\alpha \models \text{Pity}((\varphi, j) \land \text{Can}_i(\alpha, \text{Gratitude}^T(i: \alpha, \varphi)) \rightarrow T_a(\text{Pity}, i, (\varphi, j)) \]  
\[ BDA_i^\alpha \models \text{Resentment}((\varphi, j) \land \text{Can}_i(\alpha, \text{Anger}^T(i: \alpha, \varphi)) \rightarrow T_a(\text{Resentment}, i, (\varphi, j)) \]

(7.17) (7.18)

where \( BDA_i^\alpha = (\text{IntBel}, \text{Det}(i: \alpha), \text{Acd}(i, i: \alpha)) \). The assumption \( \text{IntBel} \) is the same as in Proposition 7.5. The assumptions \( \text{Det}(i: \alpha) \) and \( \text{Acd}(i, i: \alpha) \) mean that action \( i: \alpha \) is deterministic and accordant, respectively (see page 98).

The first proposition states that if an agent pities another agent because its (sub)goal \( \varphi \) has been undermined, then it has the tendency to perform any action with which it can trigger, in the other agent, gratitude towards itself with respect to \( \varphi \). Agent \( i \) will thus tend to “help” agent \( j \). The second proposition reads similarly, except that in this case agent \( i \) tends to “take revenge” on agent \( j \).

Continuing in this vein, we can relate the object-based emotion types ‘liking’ and ‘disliking’ to the action-based emotion types ‘admiration’ and ‘reproach’. This is done by constraining \( \text{IntFun} \) such that the intensity function assigned to a ‘liking’ or ‘disliking’ emotion is replaced by a function returning higher or lower values, respectively, as a result of an action that triggers ‘admiration’ or ‘reproach’ in the other agent which is the object of the ‘liking’ or ‘disliking’ emotion. Formally, we constrain \( \text{IntFun} \) as follows.

Constraint 7.10 (Intensity changes in object-based emotions)
For all agents \( i \in \text{agt} \), actions \( \alpha \in \text{act} \), and all \((M, s), (M', s')\) \( \in \mathcal{R}_{Ea} \):

\[ \text{IntFun}(\text{Liking}, i, j)(M', s') = \]
\[
\begin{cases} 
1^+ & \text{if } M', s' \models B_i\text{Admiration}^T(i: \alpha) \\
\text{IntFun}(\text{Liking}, i, j)(M, s) & \text{otherwise}
\end{cases}
\]

for some function \( 1^+ \) such that \( 1^+(\text{Time}_{M'}) > \text{IntFun}(\text{Liking}, i, j)(M, s)(\text{Time}_M) \), and

\[ \text{IntFun}(\text{Disliking}, i, j)(M', s') = \]
\[
\begin{cases} 
1^- & \text{if } M', s' \models B_i\text{Reproach}^T(i: \alpha) \\
\text{IntFun}(\text{Disliking}, i, j)(M, s) & \text{otherwise}
\end{cases}
\]

for some function \( 1^- \) such that \( 1^-(\text{Time}_{M'}) < \text{IntFun}(\text{Disliking}, i, j)(M, s)(\text{Time}_M) \).

This constraint is largely analogous to Constraint 7.8 above. Note that here we cannot use the zero function \( l_0 \) for the ‘liking’ case because an intensity function is needed that will return a higher value at the new time \( \text{Time}_{M'} \) than the old intensity value at time \( \text{Time}_M \) (i.e., \( \text{IntFun}(\text{Liking}, i, j)(M, s)(\text{Time}_M) \)). This requirement placed on the new intensity function \( 1^+ \) ensures that the overall felt intensity of the ‘liking’ emotion in question will be greater after performing the admiration-triggering action than before. The second constraint is completely analogous.

With these constraints the following properties are attained.
Proposition 7.11 (Properties of action tendency relative to object-based emotions)
The following propositions are valid:

\[ BDA^\alpha_i \models \text{Liking}(j) \land \text{Can}(\alpha, \text{Admiration}^T(j;i)) \rightarrow T^{\alpha}(\text{Liking}, i, j) \]  
(7.19)

\[ BDA^\alpha_i \models \text{Disliking}(j) \land \text{Can}(\alpha, \text{Reproach}^T(j;i)) \rightarrow T^{\alpha}(\text{Disliking}, i, j) \]  
(7.20)

where \( BDA^\alpha_i = \{\text{IntBel}, \text{Det}(i;\alpha), \text{Acd}(i, i;\alpha)\} \) is the same set of assumptions as used in Proposition 7.9.

The first proposition states that if an agent \( i \) likes another agent \( j \), then it will tend to perform actions with which it ‘can’ trigger admiration in agent \( j \) with respect to itself. Analogously, the second proposition states that if an agent \( i \) dislikes another agent \( j \), then it will tend to perform actions with which it ‘can’ trigger reproach in agent \( j \) with respect to itself.

It should be noted that the properties presented in this subsection may or may not be desirable for certain applications. For example, if one wants to prevent “vengeful” tendencies in an affective companion robot, one must make sure that no negative emotions are mitigated or positive emotions strengthened when harm is done to another agent. In particular, constraints like the second formula of Constraint 7.8 must then be prevented.

7.2.4 A Concluding Remark

Although a formalization of the notion of ‘action tendency’ can be used to study the influence of emotions on action selection, we should emphasize that the presence of action tendencies still does not specify which action will actually be chosen by an agent. Nevertheless, it is easy to have a single action designated as the “best” action to perform with respect to the emotions of an agent. Specifically, an ordering can easily be defined on the subset of actions that an agent tends to perform by comparing their ‘gain’, i.e., the difference in intensity that the agent believes to obtain by performing an action. Then one can determine which action an agent tends to perform most and is thus most likely to be selected if the agent is very emotion-driven. For example, a construct \( T^\alpha_i \) can be introduced, representing that agent \( i \) tends to perform action \( \alpha \) most. \( T^\alpha_i \) is then interpreted as follows:

\[ M, s \models T^\alpha_i \iff \alpha = \arg \max_{\alpha} \max_{(\text{Emotion}, i, \bar{\delta})} T^{\alpha}(\text{Emotion}, i, \bar{\delta})(M, s) \]  
(7.21)

Of course, there may be ties; in the most extreme case (i.e., when agent \( i \) experiences no emotions at all) \( \text{Can}(\alpha, \top) \rightarrow T^\alpha_i \) will hold for all actions \( \alpha \in \text{act} \). Knowing for which \( \alpha \) \( T^\alpha_i \) holds may be useful in implementations of affective agents.

7.2.5 Related Work

In previous work on emotion formalization at Utrecht University, Meyer [2006] formalizes four basic emotion types (i.e., happiness, sadness, anger, and fear) inspired by the psychological work of Oatley and Jenkins [1996]. In addition to formalizing
their triggering conditions, a heuristic is associated with each emotion type, indicating how an agent should act on it. However, lacking a formalization of quantitative aspects, it is left unspecified how executing such a heuristic influences the experience of the associated emotion. Moreover, in our approach, any number of ‘heuristics’ can be defined; the action tendency operator will pick up on any action that can improve the situation. For example, in Section 7.2.3 we have shown how actions can be specified to influence emotional intensity by constraining the function $\text{IntFun}$; it is then through these changes in emotional intensity (particularly their expected changes) that action tendencies arise.

Adam [2007] proposes a purely qualitative formalization of the OCC model also incorporating emotion regulation. However, only the regulation of negative event-based emotions (i.e., distress, disappointment, fear, fears-confirmed, pity, and resentment) is investigated. To this end, seven coping strategies are defined. Some coping strategies (e.g., denial, resign) change the beliefs or desires of an agent such that the triggering conditions for the negative emotion cease to hold. Other coping strategies (e.g., mental disengagement, venting) lead to the adoption of intentions to bring about new positive emotions that “divert the individual from the current negative one” [Adam, 2007]. However, in contrast to our approach, quantitative aspects of emotions are not taken into account, so it is left unspecified in Adam’s work how these coping strategies actually mitigate the experience of negative emotions. Moreover, it is not clear how Adam measures whether the situation after coping is better than before.

Gratch and Marsella [2004] have been working on a computational framework for modeling emotions inspired by the OCC model, among others. An implementation, named EMA, is used for social training applications. Like Adam, their framework incorporates a number of coping strategies. However, in EMA, the link from appraisal to coping is rather direct. In contrast, we put a notion of action tendency in between emotions and action selection; an agent can then still decide to act on its tendencies or not. Moreover, few formal details about the logic underlying EMA are provided.

### 7.3 Controlling Deliberation

In this section we take a different approach at investigating the effects of emotions on behavior.\(^8\) The previous section on action tendency attempted to capture the general and main idea behind emotion regulation, namely the idea that agents tend to select actions that maximize emotional reward. Here we take a more applied approach by considering a specific, popular class of agent implementations and investigating how emotions can be incorporated into these implementations, in a principled way.

Artificial agents, and BDI-based agents in particular, are often implemented by means of a sense–reason–act cycle. The sense–reason–act cycle is based on the idea that, in order to decide which action(s) to execute, an agent must reason about its knowledge of its environment and its goals; and in order to keep its knowledge of its environment actual, it must use its sensing capabilities to update its knowledge.

\(^8\)This section is largely based on a paper titled “Emotions to Control Agent Deliberation” [Steunebrink et al., 2010].
Although this general idea may seem quite sensible, actually implementing such a sense–reason–act cycle forces one to make a lot of design choices, many of which are not trivial. For example, at any particular time, an agent may have the option to generate new plans for its goals, to revise existing plans, and to execute a previously generated plan. But based on what principles does one decide when and which plan to generate, revise, or execute?

In the most general sense–reason–act cycle, all such decisions are left unspecified, and so all choices are made nondeterministically. For example, if a plan can be revised according to three different procedures, the procedure to apply is chosen nondeterministically. But there exist many agent programming languages that implement the deliberation of agents by means of some form of sense–reason–act cycle, for example 2APL [Dastani, 2008], GOAL [Hindriks, 2008], Jason [Bordini et al., 2007], Jadex [Pokahr et al., 2005], and Jack [Winikoff, 2005]. These agent implementations do not actually choose nondeterministically, but often make use of ad hoc rules for breaking ties. For example, in 2APL reasoning rules are always tried in the order in which they appear in the agent program, even though its formal semantics leave this to nondeterminism.

The approach taken here is to look at psychological models of human emotions for ways in which agent deliberation can be controlled. According to psychological literature, emotions function as a feedback mechanism with respect to one’s performance. Particularly in a task-oriented situation, positive and negative emotions function as signals for continuing or halting current inclinations, respectively. In this section we investigate how emotions can be used to specify constraints on an agent’s sense–reason–act cycle; these constraints then reduce nondeterminism in a principled way.

Our approach is to take a slightly simplified version of 2APL [Dastani, 2008] and strip its sense–reason–act cycle down to the point where it is completely general and nondeterministic. We then take four emotion types (i.e., hope, fear, joy, distress) from the OCC model and for each of these four emotion types we formally specify a constraint corresponding to that emotion type. The hope-based constraints turns out to closely match one of the ad hoc choices used in 2APL, whereas the fear-based constraint differs from the corresponding ad hoc choice used in 2APL. The joy-based and distress-based constraints, on the other hand, introduce entirely new dynamics. Although in the end there is still nondeterminism, this can be further reduced by introducing more principled, possibly emotion-inspired, constraints.

### 7.3.1 A Practical Agent Programming Language

We will begin by illustrating the choices that have to be made when designing the sense–reason–act cycle of a BDI agent. To this end, we define the semantics of a simple agent. These semantics are based on 2APL [Dastani, 2008] (“A Practical Agent Programming Language”), but the message of this section extends to any BDI-based agent programming language that makes use of beliefs, declarative goals, and reasoning rules for implementing agents. Such programming languages include GOAL [Hindriks, 2008], Jason [Bordini et al., 2007], Jadex [Pokahr et al., 2005], and Jack [Winikoff, 2005]. Although 2APL is a platform for programming multi-agent
system, here we will only consider the single-agent case. Also, we will not be concerned with the external environment in which the agent might ‘live’ and therefore leave this out of our formalization for simplicity.

In 2APL, an agent is programmed in terms of beliefs, declarative goals, plans, and two kinds of reasoning rules: plan generation (PG) rules and plan revision (PR) rules. The analysis of 2APL presented here is purely semantic; the exact syntax of 2APL programs is not of concern here and can be found elsewhere [Dastani, 2008]. On the semantic level, then, an agent is represented as a configuration containing data structures for storing beliefs, goals, plan, and reasoning rules. This configuration is changed by the execution of actions; the precise effects of actions on a configuration is specified using transition rules.

**Definition 7.12 (Agent configuration)**

An agent configuration is a tuple \( \langle i, B, G, P, PG, PR \rangle \), where \( i \) is the agent name, \( B \) is the belief base, \( G \) is the goal base, \( P \) is the plan base, \( PG \) is the set of plan generation rules, and \( PR \) is the set of plan revision rules.

We will typically use the symbol \( C \) to refer to an agent configuration. It should be noted that during the execution of an agent, only its beliefs, goals, and plans can change. So an agent’s name and reasoning rules are static.

Let us now introduce what is contained in the beliefs, goals, plans, and reasoning rules of an agent configuration. First, belief bases are defined as follows.

**Definition 7.13 (Beliefs)**

Let \( L_{PC} \) be a set of well-formed formulas, built as usual by induction from a set of atomic propositions \( \text{atm} \) and the usual propositional connectives. The set of all possible belief bases is \( \Sigma = \{ \sigma \subseteq L_{PC} | \sigma \nvdash_{PC} \bot \} \). The following typical element is used: \( \beta \in L_{PC} \) and \( B \in \Sigma \).

So an agent’s belief base is assumed to be consistent; note that “PC” stands for Propositional Calculus. Belief bases can be queried as follows.

**Definition 7.14 (Belief querying)**

A belief base is queried using the relation \( \vdash_b \subseteq \Sigma \times L_{PC} \). \( \vdash_b \) is specified as follows:

\[
B \vdash_b \beta \quad \text{iff} \quad B \vdash_{PC} \beta
\]  

(7.22)

Note that a closed world assumption is used with respect to beliefs; for each formula, either the formula or its negation follows from the belief base. This is formalized by the following constraint.

**Constraint 7.15 (Closed world assumption)**

For all \( \beta \in L_{PC} \), either \( B \vdash_b \beta \) or \( B \vdash_b \neg \beta \).

Second, goals are defined as conjunctions of atomic propositions. Note that this deviates from previous chapters, where goals were represented as conjunctions of literals. The restriction to conjunctions of atomic propositions for representing goals will simplify the current treatment a bit without weakening its message.
Definition 7.16 (Achievement goals)
The set of all possible goal bases is $\Gamma = 2^K$. Goal formulas are thus drawn from the set $\mathcal{K}$, which is the set of conjunctions of atomic propositions, defined as follows:

$$\mathcal{K} = \{ \bigwedge \Phi \mid \emptyset \subseteq \Phi \subseteq \text{atm} \}$$

(7.23)

$$\overline{\mathcal{K}} = \{ \bigwedge \Phi \mid \emptyset \subseteq \Phi \subseteq \overline{\text{atm}} \}$$

(7.24)

where $\overline{\text{atm}} = \{ \neg p \mid p \in \text{atm} \}$. The set $\overline{\mathcal{K}}$ of conjunctions of negated propositions will be useful for representing unachieved goals. The following typical elements are used: $\kappa \in \mathcal{K}$, $\overline{\kappa} \in \overline{\mathcal{K}}$, and $G \in \Gamma$.

It is thus assumed for each configuration $\langle i, B, G, P, PG, PR \rangle$ that $B \in \Sigma$ and $G \in \Gamma$. So the goals of an agent are assumed to be specified as conjunctions of atomic propositions. Note that $\mathcal{K} \cup \overline{\mathcal{K}} = \text{ccl}$; see Definition 5.3 on page 84. With slight abuse of notation, we use the ‘overline’ to convert from $\mathcal{K}$ to $\overline{\mathcal{K}}$ and vice versa. So we have that $\overline{\mathcal{K}} = \{ \overline{\kappa} \mid \kappa \in \mathcal{K} \}$ and that $\overline{\kappa} = \kappa$.

In the following it will be useful to be able to talk about subgoals of goals. This notion is formalized using the following relation.

Definition 7.17 (Subgoal relation)
The subset relation $\sqsubseteq$ on $\mathcal{K}$ is defined as:

$$\sqsubseteq = \{ (\bigwedge \Phi_1, \bigwedge \Phi_2) \mid \emptyset \subseteq \Phi_1 \subseteq \Phi_2 \subseteq \text{atm} \}$$

(7.25)

So $\kappa' \sqsubseteq \kappa$ iff the conjuncts comprising $\kappa'$ are a non-empty subset of those comprising $\kappa$. Normally, we will use this relation when $\kappa$ is a goal; in that case $\kappa' \sqsubseteq \kappa$ expresses that $\kappa'$ is a subgoal of $\kappa$.

With the use of $\sqsubseteq$, querying of goal bases can now be easily defined as follows.

Definition 7.18 (Goal querying)
A goal base is queried using the relation $\models_\delta \subseteq \Gamma \times \mathcal{K}$. $\models_\delta$ is specified as follows:

$$G \models_\delta \kappa \iff \exists \gamma \in G : \kappa \subseteq \gamma$$

(7.26)

This definition expresses that $\kappa$ follows from the goal base iff $\kappa$ is a goal or subgoal of the agent. A (sub)goal $\kappa$ is said to have been achieved if it follows from the belief base, i.e., $B \models_\delta \kappa$. The goal base is assumed not to contain irrelevant goals; that is, it is assumed that goals are removed from the goal base as soon as they have been achieved. This is expressed using the following constraint.

Constraint 7.19 (Achievement goals are relevant)
$\gamma \in G$ implies $B \models_\delta \neg \gamma$.

Third, a plan base $P$ consists of plan–(sub)goal–rule triples, i.e., $P \subseteq \Pi \times \mathcal{K} \times (PG \cup PR)$ where $\Pi$ is the set of all possible plans. For example, $(\pi, \kappa, r) \in P$ means that the agent is committed to performing plan $\pi$ in order to achieve (sub)goal $\kappa$, and
that $\pi$ was obtained by applying rule $r$. Plans may consist of belief queries, belief updates, external actions (such as sending a message and sensing and manipulating the environment, but we will not go into this any further here), if–then–else and while–do constructs, and sequential compositions of actions. It is assumed that plans are removed as soon as the associated goal has been achieved or has become irrelevant. This is expressed using the following constraint.

**Constraint 7.20 (Plans are relevant)**

$$(\pi, \kappa, r) \in P \implies B \models \neg \kappa \text{ and } G \models \kappa.$$ 

Fourth and last, the reasoning rules are represented as follows. PG-rules are of the form $\kappa \mid \beta \rightarrow \pi$, which specifies that if the agent believes $\beta$ to be the case, then it can perform plan $\pi$ to achieve (sub)goal $\kappa$. PR-rules are of the form $\pi' \mid \beta \rightarrow \pi$, which specifies that if the agent believes $\beta$ to be the case, then it can rewrite (replace) $\pi'$ by plan $\pi$. In the following, we will call $\kappa$ respectively $\pi'$ the head of the rule, $\beta$ the guard (belief condition) of the rule, and $\pi$ the plan of the rule. Three accessors for rules are then used: $H(r)$ denotes the head of rule $r$ (i.e., $\kappa$ or $\pi'$), $G(r)$ denotes the guard of rule $r$ (i.e., $\beta$), and $P(r)$ denotes the plan of rule $r$ (i.e., $\pi$).

Reasoning rules can be **applied** in order to generate new plans (for PG-rules) or revise existing plans (for PR-rules). Applicability of a rule in a certain agent configuration is formally defined as follows.

**Definition 7.21 (Applicability of reasoning rules)**

A PG-rule $r = (\kappa \mid \beta \rightarrow \pi)$ is applicable with respect to a configuration $C = \langle i, B, G, P, PG, PR \rangle$ iff the head of the rule ($\kappa$) follows from the goal base ($G$) but not from the belief base ($B$) and the guard of the rule ($\beta$) follows from the belief base. This is abbreviated as follows:

$$\text{Applicable}_{C}(r) \overset{\text{def}}{=} r \in PG \land G \models \kappa \land B \models \beta \land \neg \kappa \quad (7.27)$$

Similarly, a PR-rule $r = (\pi' \mid \beta \rightarrow \pi)$ is applicable to a plan $\pi''$ iff $\pi''$ is in the plan base, the head of the rule ($\pi'$) is unifiable with $\pi''$, and the guard of the rule follows from the belief base. This is abbreviated as follows:

$$\text{Applicable}_{C}(r, \pi'') \overset{\text{def}}{=} r \in PR \land (\exists \kappa', r' : (\pi'', \kappa', r') \in P) \land \text{Unifiable}(\pi', \pi'') \land B \models \beta \quad (7.28)$$

where Unifiable is a predicate indicating whether two plans are unifiable.

In 2APL, the head $\pi'$ and plan $\pi$ of a PR-rule $\pi' \mid \beta \rightarrow \pi$ can contain variables to allow PR-rules to be applied to many different instances of plans. However, we cannot go into the definition of Unifiable without considering the syntax of plans in 2APL. Therefore we simply assume a suitable definition of Unifiable such that it indicates whether or not two plans match. The interested reader can find a proper specification of plan unification in [Dastani, 2008], but knowledge of these details is not necessary for understanding the message of this section.
7.3.2 Nondeterminism in Operational Semantics

We will now specify the effects of applying a PG-rule or PR-rule. For the application of PG-rule \( r \) we specify the meta-action \( \text{Gen}(r) \) ("generate"), and for the application of PR-rule \( r' \) we specify the meta-action \( \text{Rev}(r') \) ("revise"). These are meta-actions because they are part of the sense–reason–act cycle of an agent. \( \text{Gen} \) and \( \text{Rev} \) cannot appear in the plans of an agent; they are only used in specifying the execution of the agent’s deliberation cycle.

If a PG-rule \( r \) is applied, the triple \((P(r), H(r), r)\) is added to the plan base \( P \). If a PR-rule \( r' \) is applied to \((\pi, \kappa, r') \in P\), the triple \((\pi, \kappa, r')\) is replaced by \((P(r), \kappa, r)\) in the plan base \( P \). But of course a rule can only be applied if it is applicable (in the sense of Definition 7.21). The meta-actions \( \text{Gen}(r) \) and \( \text{Rev}(r') \) are specified using transition semantics, as follows:

\[
\begin{align*}
\text{Applicable}_C(r) & \implies \exists \pi : \text{Applicable}_C(r, \pi) \\
\text{Applicable}_C(r) & \implies \mathcal{C}' \\
\text{Applicable}_C(r) & \implies \mathcal{C}'' \\
\end{align*}
\]

The plan base of configuration \( \mathcal{C}' \) is \( P' = P \cup \{(P(r), H(r), r)\} \). The plan base of configuration \( \mathcal{C}'' \) is \( P'' = (P \setminus \{((\pi, \kappa, r')\}) \cup \{(P(r), \kappa, r)\} \).

We will use a third meta-action which specifies the execution of a plan of an agent. This meta-action is denoted as \( \text{Do}(\pi) \) and is specified with the following transition rule:

\[
\begin{align*}
\mathcal{C} & \xrightarrow{\alpha} \mathcal{C}'''' \\
\mathcal{C} & \xrightarrow{\text{Do}(\pi)} \mathcal{C}'''' \\
\end{align*}
\]

where \( \alpha \) is the first action of plan \( \pi \), i.e., \( \pi = \alpha; \pi' \) for some (possibly empty) \( \pi' \). So the meta-action \( \text{Do} \) only executes the first action of the provided plan. This allows plans to be interleaved if the plan base contains more than one plan. \( \mathcal{C} \xrightarrow{\alpha} \mathcal{C}'''' \) means that configuration \( \mathcal{C} \) can make a ‘normal’ transition to \( \mathcal{C}'''' \) by performing action \( \alpha \). Here we will not go into the transition rules for actions that can be performed by an agent; for 2APL all such transition rules can be found in [Dastani, 2008]. These transition rules typically include a condition to ensure that the plan being executed (i.e., \( \pi \)) is in the plan base and that the goal associated with the plan is still relevant (e.g., \( \exists \kappa, r : (\pi, \kappa, r) \in P \land G \models_{\mathcal{G}} \kappa \land B \models_{\mathcal{B}} \neg \kappa \)). Note that the meta-action for executing a plan only executes one action at the time; that is, plans are not assumed to be atomic, so this meta-action will have to be performed repeatedly to finish a plan.

We now have sufficient ingredients to illustrate that a sense–reason–act cycle typically contains many choice points. In the following, we use notation as is common in Dynamic Logic, i.e., ‘;’ denotes sequential composition, ‘∗’ denotes repetition (execute zero or more times), and ‘+’ denotes nondeterministic choice. A complete sense–reason–act cycle is then denoted as \((\text{Sense}; \text{Reason}; \text{Act})\), where \text{Sense} is a meta-action that senses the environment and updates the belief base and goal base accordingly. Although sensing is an essential part of any sense–reason–act cycle, we will focus on the reasoning and acting part in the rest of this paper. The reasoning and acting of an agent can be described using the meta-actions introduced above, as follows.
Definition 7.22 (The general form of the reason–act cycle)
The ‘reason’ and ‘act’ part of a sense–reason–act cycle is defined in its most general form as follows:

\[
\text{Reason}_\text{Act} \overset{\text{def}}{=} (\text{ApplyPGrule} + \text{ApplyPRrule} + \text{ExecutePlan})^*
\]

- \(\text{ApplyPGrule} \overset{\text{def}}{=} \text{Gen}(pg_1) + \cdots + \text{Gen}(pg_n)\)
- \(\text{ApplyPRrule} \overset{\text{def}}{=} \text{Rev}(pr_1) + \cdots + \text{Rev}(pr_m)\)
- \(\text{ExecutePlan} \overset{\text{def}}{=} \text{Do}(\pi_1) + \cdots + \text{Do}(\pi_k)\)

where \(PG = \{pg_1, \ldots, pg_n\}\), \(PR = \{pr_1, \ldots, pr_m\}\), and \(P = \{(\pi_1, \kappa_1, r_1), \ldots, (\pi_k, \kappa_k, r_k)\}\).

So with respect to reasoning and acting, the agent can do three kinds of things: apply PG-rules, apply PR-rules, and execute plans. The above specification is completely general: all these meta-actions can be done in any particular order; all choices are nondeterministic; there is no commitment to any particular order of execution of the presented meta-actions.

Of course choices have to be made when implementing an agent programming language. For example, in 2APL, PG-rules are tried and applied in the order in which they appear in the source code of the agent program; PR-rules are only applied when execution of a plan fails; and plans are interleaved if there is more than one in the plan base. Although seemingly reasonable choices, they remain ad hoc choices. It will be clear that, among the many different ways of designing a sense–reason–act cycle, some will be better than others. For example, constantly checking each PG-rule and each PR-rule for applicability after performing a single atomic action will obviously yield a very inefficient agent. But how does one decide, in a principled way, how to choose between generating, revising, and executing plans? In the next subsection we show how emotions can be used as such design principles.

7.3.3 A Formalization of Emotion Triggers in 2APL

Let us recall from Chapters 3 and 4 that emotions of the types ‘hope’, ‘fear’, ‘joy’, and ‘distress’ are triggered in response to the perception of a consequence of an event that is either prospective or actual, and that is appraised as either desirable or undesirable. Specifically, the triggering conditions of ‘hope’, ‘fear’, ‘joy’, and ‘distress’ were formalized in accordance with the OCC model as the following macros:

\[
\begin{align*}
\text{Hope}_T(\phi) & \overset{\text{def}}{=} \text{Pleased}_T(\phi) \land \text{Prospective}_i(\phi) \quad (7.31) \\
\text{Fear}_T(\phi) & \overset{\text{def}}{=} \text{Displeased}_T(\phi) \land \text{Prospective}_i(\phi) \quad (7.32) \\
\text{Joy}_T(\phi) & \overset{\text{def}}{=} \text{Pleased}_T(\phi) \land \text{Actual}_i(\phi) \quad (7.33) \\
\text{Distress}_T(\phi) & \overset{\text{def}}{=} \text{Displeased}_T(\phi) \land \text{Actual}_i(\phi) \quad (7.34) \\
\text{Pleased}_T(\phi) & \overset{\text{def}}{=} \text{PerceiveConseq}_i(\phi) \land \text{Des}_i(\phi) \quad (7.35) \\
\text{Displeased}_T(\phi) & \overset{\text{def}}{=} \text{PerceiveConseq}_i(\phi) \land \text{Undes}_i(\phi) \quad (7.36)
\end{align*}
\]
PerceiveConseq_i(\varphi) \stackrel{\text{def}}{=} \text{Prospective}_i(\varphi) \lor \text{Actual}_i(\varphi) \quad (7.37)

\text{Prospective}_i(\varphi) \stackrel{\text{def}}{=} \text{FutUpdate}_i(\varphi) \lor \text{UncUpdate}_i(\varphi) \quad (7.38)

\text{Actual}_i(\varphi) \stackrel{\text{def}}{=} \text{New } B_i\varphi \quad (7.39)

\text{FutUpdate}_i(\varphi) \stackrel{\text{def}}{=} \text{New } B_i\text{Fut}^+\varphi \quad (7.40)

\text{UncUpdate}_i(\varphi) \stackrel{\text{def}}{=} \text{New } (\neg B_i\varphi \land \neg B_i\neg \varphi) \quad (7.41)

\text{New } \varphi \stackrel{\text{def}}{=} \varphi \land \text{Prev } \neg \varphi \quad (7.42)

The basic operators that are thus used are B for belief, Prev for previous state, Fut^+ for possible future state, Des for desirable, and Undes for undesirable.

We are now in a position to define a satisfaction relation \models for agent configurations, such that the eliciting conditions of ‘hope’, ‘fear’, ‘joy’, and ‘distress’ can be said to hold in a particular configuration. In order to be able to interpret the Prev construct, it is assumed that agent configurations contain an additional structure \(C_{-1}\), storing the belief base, goal base, and plan base of the previous configuration, as follows.

**Constraint 7.23 (Previous configuration)**

*If a transition \(C \rightarrow C'\) is made, where \(C = \langle i, B, G, P, PG, PR, C_{-1} \rangle\) and \(C' = \langle i, B', G', P', PG, PR, C'_{-1} \rangle\), then \(C'_{-1} = \langle B, G, P \rangle\).*

So only the previous configuration is stored, not a complete history. Formulas are then interpreted on agent configurations as follows.

**Definition 7.24 (Interpretation of formulas on agent configurations)**

Let \(C = \langle i, B, G, P, PG, PR, C_{-1} \rangle\) be an agent configuration; then \(C \models \varphi\) is specified as follows:

- \(C \models \neg \varphi\) iff not \(C \models \varphi\)
- \(C \models \varphi_1 \land \varphi_2\) iff \(C \models \varphi_1\) & \(C \models \varphi_2\)
- \(C \models B_i\beta\) iff \(B \models_b \beta\)
- \(C \models \text{Des}_i(\kappa)\) iff \(G \models_g \kappa\)
- \(C \models \text{Prev } \varphi\) iff \(C_{-1} \models \varphi\)
- \(C \models B_i\text{Fut}^+\xi\) iff \(\text{Fut}_C^+ (\xi)\)

where \(\text{Fut}_C^+ (\xi)\) abbreviates, for \(\xi \in \mathcal{K} \cup \overline{\mathcal{K}}:\)

\[
\text{Fut}_C^+ (\xi) \stackrel{\text{def}}{=} \text{Fut}_C (\xi) \land B \models_b \neg \xi \quad (7.43)
\]

\[
\text{Fut}_C (\kappa) \stackrel{\text{def}}{=} \exists r \in PG : \kappa \subseteq \mathcal{H}(r) \land B \models_b \mathcal{G}(r) \quad (7.44)
\]

\[
\text{Fut}_C (\overline{\kappa}) \stackrel{\text{def}}{=} \exists (\pi, \kappa', r) \in P : \overline{\kappa} \in \text{PostCond}_C (r) \quad (7.45)
\]

Below we will explain Definition 7.24 in detail.

In the interpretation of \text{Prev } \varphi, writing \(C_{-1} \models \varphi\) is strictly speaking not allowed, because \(C_{-1}\) is not a proper configuration. Indeed, \(C_{-1} \models \varphi\) is short for
the reasoning rules of a configuration, which are static, and that it is not assumed ("undermined") after performing the plan of the rule. Its mapping is thus each PG-rule and PR-rule a subset of \( C \subseteq \text{PostCond}(\cdot) \) of plan the agent is committed to performing the plan. Thus the agent should believe each \( \text{Fut} \) fails if \( \text{Prev} \); \( \text{Fut} \) definition of \( C \) is nested (as in \( \text{Prev} \text{Prev} \cdot \)), this is not problematic because a situation where nesting of \( \text{Prev} \) is required will not occur. Note that such nesting also fails for \( \text{B} \), \( \text{Des} \), and \( \text{Fut}^+ \). So the kinds of formulas that can be interpreted on agent configurations is limited with respect to the language used in previous chapters, but sufficient for the purposes of this section. For example, according to Definition 7.24 the \( \text{Fut}^+ \) operator can only be interpreted inside a \( \text{B} \) operator, but \( \text{Fut}^+ \) is never used outside \( \text{B} \) (cf. Definition 4.9 on page 65), so this restriction does not pose a problem.

The reasoning behind the \( \text{Fut}_C(\kappa) \) construct is as follows. Assume \( \kappa \models \beta \to \pi \) is a PG-rule of agent \( i \) in configuration \( C \); this means that the agent programmer promises that \( \kappa \) can be achieved by executing \( \pi \) in a state where \( \beta \) holds. Thus \( (\kappa \models \beta \to \pi) \in \text{PG} \) expresses that agent \( i \) believes that \( \beta \to \langle \pi \rangle \kappa \) is (always) true.\(^9\) So we would have that \( C \models B_i(\beta \to \langle \pi \rangle \kappa) \) iff \( (\kappa \models \beta \to \pi) \in \text{PG} \). Now \( \text{Fut} \varphi \) expresses that \( \varphi \) will hold in some possible future, i.e., it can be seen as an existential quantification over all possible agents and plans (see also formula (4.8) on page 62; note that \( \text{Fut} \) does not include intention). So if there exists an agent \( i \) and a plan \( \pi \) such that \( (\langle \pi \rangle \kappa) \), then we also have \( \text{Fut} \kappa \). Replacing \( (\pi) \kappa \) by \( \text{Fut} \), we would then have that \( C \models B_i(\beta \to \text{Fut} \kappa) \) iff \( \exists r \in \text{PG} : \kappa = H(r) \& \beta = G(r) \). But the guard of the rule is satisfied if \( B \models_b G(r) \), so \( C \models B_i \text{Fut} \kappa \) iff \( \exists r \in \text{PG} : \kappa = H(r) \& B \models_b G(r) \). But if \( \kappa \) holds after executing \( \pi \), then so do all its subgoals, so \( C \models B_i \text{Fut} \kappa \) iff \( \exists r \in \text{PG} : \kappa \subseteq H(r) \& B \models_b G(r) \). Hence the definition of \( \text{Fut}_C(\kappa) \) above.

As for the \( \text{Fut}_C(\overline{\kappa}) \) construct, the fact that a plan \( \pi \) is in the plan base means that the agent is committed to performing the plan. Thus the agent should believe each postcondition of plan \( \pi \) to be possibly true after the execution of \( \pi \). So we would have that \( C \models B_i(\langle \pi \rangle \overline{\kappa}) \) iff \( \exists r \in \text{PG} : \kappa = H(r) \& \beta = G(r) \). But the guard of the rule is satisfied if \( B \models_b G(r) \), so \( C \models B_i \text{Fut} \kappa \) iff \( \exists r \in \text{PG} : \kappa = H(r) \& B \models_b G(r) \). But if \( \kappa \) holds after executing \( \pi \), then so do all its subgoals, so \( C \models B_i \text{Fut} \kappa \) iff \( \exists r \in \text{PG} : \kappa \subseteq H(r) \& B \models_b G(r) \). Hence the definition of \( \text{Fut}_C(\kappa) \) above.

The \( \text{Fut}_C^+(\xi) \) construct is then a strict version of the \( \text{Fut}_C(\kappa) \) and \( \text{Fut}_C(\overline{\kappa}) \) constructs, in the same sense as \( \text{Fut}^+ \varphi \) being a strict version of \( \text{Fut} \varphi \) (see formula (4.1) on page 61). Namely, \( \text{Fut}_C^+(\xi) \) requires that \( \xi \) is not derivable from the current belief base.

In order to determine which formulas can possibly hold after the execution of a plan in the plan base, we thus make use of the \( \text{PostCond}_C \) function. In the following we are actually only interested in a special kind of formula as postcondition, namely inverted goal formulas. The reason for this is that we want to find out whether one plan threatens the goal of another plan. A plan threatens a goal if an inverted subgoal of the goal is among the postconditions of the plan. The set of all (sub)goals that can possibly be threatened is \( \mathcal{K}_C = \{ \kappa \mid \kappa \in \text{PostCond}(\cdot) \} \). The set of inverted subgoals is then \( \overline{\mathcal{K}}_C = \{ \overline{\kappa} \mid \kappa \in \mathcal{K}_C \} \). The function \( \text{PostCond}_C \) then associates with each PG-rule and PR-rule a subset of \( \overline{\mathcal{K}}_C \) indicating which subgoal may be false ("undermined") after performing the plan of the rule. Its mapping is thus \( \text{PostCond}_C : (\text{PG} \cup \text{PR}) \to \overline{\mathcal{K}}_C \). It should be noted that the function \( \text{PostCond}_C \) only depends on the reasoning rules of a configuration, which are static, and that it is not assumed

\(^9\)Here we use \( (\langle \pi \rangle \kappa) \) to express that \( \kappa \) is a possible result of agent \( i \) performing \( \pi \), as usual in Dynamic Logic. But note that \( \beta \to (\langle \pi \rangle \kappa) \) will not appear in our actual object language; it is only used to clarify the interpretation of \( B_i \text{Fut}^+ \xi \).
to take into account preconditions at ‘run-time’. Therefore we may assume that
the postconditions of all rules are determined at ‘compile-time’, thereby making
PostCondC a cheap lookup function. The postconditions of a rule can be determined
by performing some analysis of the rule’s plan, or by letting the agent programmer
annotate each reasoning rule with the (relevant) postconditions.

Of all constructs used in the formalization of eliciting conditions, only the con-
struct Undes for expressing undesirability is still undefined. Here we define unde-
sirability simply as an ‘inverse’ of desirability:

\[ \text{Undes}_i(\varphi) \overset{\text{def}}{=} \text{Des}_i(\overline{\varphi}) \quad (7.46) \]

So the satisfaction of undesirability in a configuration becomes \( C \mid= \text{Undes}_i(\overline{\kappa}) \) iff \( G \mid= g_\kappa \) (recall that \( \overline{\kappa} = \kappa \)). Note that this definition of undesirability is not unreasonable
in light of a restriction to achievement goals.\(^{10}\) In fact, it is noted by OCC that if an
event is desirable to some degree, the absence of that event may be undesirable to
the same degree [Ortony et al., 1988]. Because we have formalized a desirable event
as the satisfaction of a goal formula, the absence of a desirable event can thus be
formalized as the satisfaction of an inverted goal formula.

Using formulas (7.31)–(7.42) and (7.46) as macros, it will now follow that these
equivalences hold for all possible agent configurations:

**Proposition 7.25 (Eliciting conditions of ‘hope’, ‘fear’, ‘joy’, and ‘distress’)**

The following equivalences are valid:

\[
\begin{align*}
\text{Hope}^T_i(\kappa) & \iff \text{Des}_i(\kappa) \land \text{B}_i\text{Fut}^+\kappa \land \neg \text{Prev B}_i\text{Fut}^+\kappa \quad (7.47) \\
\text{Fear}^T_i(\overline{\kappa}) & \iff \text{Des}_i(\kappa) \land \text{B}_i\text{Fut}^+\overline{\kappa} \land \neg \text{Prev B}_i\overline{\kappa} \quad (7.48) \\
\text{Joy}^T_i(\kappa) & \iff \text{Des}_i(\kappa) \land \text{B}_i\kappa \land \neg \text{Prev B}_i\kappa \quad (7.49) \\
\text{Distress}^T_i(\overline{\kappa}) & \iff \text{Des}_i(\kappa) \land \text{B}_i\overline{\kappa} \land \neg \text{Prev B}_i\overline{\kappa} \quad (7.50)
\end{align*}
\]

When comparing (7.47) and (7.48) to formula (7.38), it may appear that they miss the
‘uncertainty’ aspect (recall that ‘prospect’ was used to refer to both current uncer-
tainty and future possibility). However, because of our closed world assumption on
belief bases, uncertainty in that sense cannot exist; a proposition either follows from
the belief base or it does not. Given (7.46), it is easy to verify that (7.49) and (7.50) do
correspond directly to formulas (7.33) and (7.34), respectively.

The propositions above are translated to the level of agent configurations as
follows:

**Proposition 7.26 (Interpretation of emotion triggers in 2APL)**

Let \( C = \langle i, B, G, P, PG, PR, C_{-1} \rangle \) be an agent configuration, where \( C_{-1} = \langle B', G', P' \rangle \). Then
the following equivalences are valid:

\[
\begin{align*}
C \mid= \text{Hope}^T_i(\kappa) \quad & \text{iff} \quad G \mid= g_\kappa \land \text{Fut}^+_C(\kappa) \land \neg \text{Fut}^+_{C_{-1}}(\kappa) \quad (7.51) \\
C \mid= \text{Fear}^T_i(\overline{\kappa}) \quad & \text{iff} \quad G \mid= g_\kappa \land \text{Fut}^+_C(\overline{\kappa}) \land \neg \text{Fut}^+_{C_{-1}}(\overline{\kappa}) \quad (7.52)
\end{align*}
\]

\(^{10}\)When types of goals other than achievement goals are incorporated, the framework may have to be
extended such that desirability and undesirability are defined independently, as was done in the previous
chapters.
CONTROLLING DELIBERATION

7.3

7.3.4 Constraining the Deliberation Cycle

We now have all ingredients necessary to formally specify principled constraints on the deliberation cycle based on the emotion types ‘hope’, ‘fear’, ‘joy’, and ‘distress’.

In preparation of the next section, we will show two more interesting theorems with respect to ‘hope’ and ‘fear’. Let \( r \) be a PG-rule of configuration \( C \). The following macro then expresses that hope is triggered with respect to \( r \) as soon as the PG-rule becomes applicable:

\[
\text{Hope}_C(r) \overset{\text{def}}{=} \text{Applicable}_C(r) & \neg \text{Fut}^+_C(H(r))
\]  

(7.55)

The term \( \neg \text{Fut}^+_C(H(r)) \) expresses that rule \( r \) cannot have been applicable in the previous configuration. Including this term ensures that \( \text{Hope}_C(r) \) accurately expresses that hope is triggered, as is shown by the following property.

**Proposition 7.27 (Hope triggered by an applicable PG-rule)**

The following implication is valid:

\[
\text{Hope}_C(\kappa | \beta \rightarrow \pi) \text{ implies } C \models \text{Hope}_T(\kappa)
\]  

(7.56)

It should be noted that, although we put a complete PG-rule in \( \text{Hope}_C \), the object of the triggered hope is actually the (sub)goal that can be achieved by applying the PG-rule (and executing its plan).

Finally, we have that ‘fear’ is triggered as soon the plan base contains two conflicting plans, in the sense that a postcondition of one plan contradicts the goal of the other plan. To express this, we define the following macro:

\[
\text{Fear}_C(\pi, \kappa) \overset{\text{def}}{=} \neg \text{Fut}^+_C(\kappa) & \exists (\pi_1, \kappa_1, r_1), (\pi_2, \kappa_2, r_2) \in P : \kappa \in \text{PostCond}_C(r_2)
\]

\[
& \kappa \sqsubseteq \kappa_1 & \pi = \pi_2
\]  

(7.57)

Note again the use of the term \( \neg \text{Fut}^+_C(\kappa) \) to ensure that what is expressed is a triggering condition for fear. Thus \( \text{Fear}_C(\pi, \kappa) \) expresses that fear is triggered because plan \( \pi \) threatens (sub)goal \( \kappa \) (by promising \( \kappa \)) of another plan. It should be noted that it is possible that \( (\pi_1, \kappa_1, r_1) = (\pi_2, \kappa_2, r_2) \). This does not necessarily mean that \( \pi \) (= \( \pi_1 = \pi_2 \)) is a ‘bad’ plan; it might simply be the case that \( \pi \) is not guaranteed to succeed in achieving its goal. The fact that \( \text{Fear}_C(\pi, \kappa) \) accurately expresses that fear is triggered is shown by the following property.

**Proposition 7.28 (Fear triggered by a plan threatening a goal)**

The following implication is valid:

\[
\text{Fear}_C(\pi, \kappa) \text{ implies } C \models \text{Fear}_T(\kappa)
\]  

(7.58)
We will specify emotion-inspired constraints in order to limit the choices in applying reasoning rules and executing plans. In psychological literature, affective feelings (including emotions) are often described as informing an individual about his or her performance. In particular, when one is task-oriented, positive emotions function as a “go” signal for pursuing currently accessible inclinations, whereas negative emotions function as a “stop” signal to allow for seeking alternatives [Clore et al., 2001]. This view of emotions as “go” and “stop” signals can be used to decide when to generate a new plan (“go ahead and do it”), when to revise a plan (“stop and reconsider”), which plan to choose for execution (“do what is making you feel good”), and when to stop a plan’s execution (“stop when you don’t feel good about what you’re doing”). Indeed, we will now show how ‘hope’, ‘fear’, ‘joy’, and ‘distress’ can be used as the described signals, respectively, to constrain the deliberation cycle.

**Hope**

The generation of new plans is constrained by only allowing a PG-rule to be applied when hope is triggered with respect to the head of the rule. This is done by replacing $Gen$, as used in Definition 7.22, by $Gen'$. The transition rule for $Gen'$ is specified as follows:

$$
\frac{\text{Hope}_C(r)}{\text{C} \xrightarrow{\text{Gen}(r)} \text{C}'}
$$

where $C'$ is updated as in rule (7.29). Recall from formula (7.55) that $\text{Hope}_C(r)$ includes $\text{Applicable}_C(r)$, which was the condition for $Gen$ used in rule (7.29). What the condition $\text{Hope}_C(r)$ adds to the original (ad hoc) condition is that previously applicable PG-rules are not reconsidered for application.

**Fear**

The revision of existing plans is restricted by only allowing a PR-rule to be applied when fear is triggered with respect to a possible conflict between two plans, in the sense of formula (7.57). This is done by replacing $Rev$, as used in Definition 7.22, by $Rev'$. The transition rule for $Rev'$ is specified as follows:

$$
\exists \pi : \text{Applicable}_C(r, \pi) \& \exists \kappa : \text{Fear}_C(\pi, \kappa)
\frac{\text{Rev}(r)}{\text{C} \xrightarrow{\text{Rev}(r)} \text{C}''}
$$

where $C''$ is updated as in rule (7.29). The condition that fear must have been triggered for a PR-rule to be applied is different from the condition used in 2APL [Dastani, 2008]. In 2APL, the deliberation cycle only tries to find applicable PR-rules when an action of a plan has failed. In 2APL’s precursor 3APL, PR-rules were applied whenever applicable, and applicable PR-rules were sought each time a single action had been executed. The constraint presented above takes a middle road by allowing threatening plans to be revised as soon as a possible conflict is perceived.

It may be interesting to note that it is not guaranteed that a revised plan is any less threatening. So after plan revision, a new fear may immediately be triggered, thus allowing for multiple successive revisions, until a non-threatening plan has been found.
Joy

The nondeterministic choice between which of the current plans to execute can be constrained by specifying a preference on the plans in the plan base. Specifically, the “go” signal given by joy can make an agent prefer to execute plans that are going well, i.e., for which subgoals are being achieved. This is done by replacing $\text{ExecutePlan}$, as used in Definition 7.22, by $\text{ExecutePlan}'$. $\text{ExecutePlan}'$ is defined as follows:

$$\text{ExecutePlan}' \overset{\text{def}}{=} \text{Do}(\pi) \quad (7.61)$$

where $(\pi, \kappa, r) = \min_{<_C} P$ (ties broken arbitrarily). The preference relation $<_C$ is a strict partial order on plan base $P$ of configuration $C$, defined as:

$$<_C = \{ (\pi, \kappa, r) \in P \mid \text{Joy}_C(\kappa) \} \times \{ (\pi, \kappa, r) \in P \mid \neg \text{Joy}_C(\kappa) \} \quad (7.62)$$

where $\text{Joy}_C(\kappa) \overset{\text{def}}{=} \exists \kappa' : \kappa' \sqsubseteq \kappa \& C \models \text{Joy}_T(\kappa')$. So $<_C$ divides the plan base in two; those plans that have made progress and those that have not. $\text{ExecutePlan}'$ then chooses the plan that has made the most progress. Obviously, this constraint still allows for nondeterminism, because $<_C$ is only a partial order. But there are ways to extend this preference order on plans, for example by taking into account the number of subgoals having been achieved. Furthermore, $<_C$ could be made to take into account goal achievements over a longer history than just the previous state.

Distress

The interleaving of the execution of plans is restricted by specifying that if a plan is executed, it is not interleaved unless it triggers distress. Thus distress will interrupt the agent’s ‘attention’ from the current plan and allow it to consider, e.g., plan revision or executing another plan. This is accomplished by interleaving each plan with tests for distress. So $\text{Do}$, as used in Definition 7.22, is replaced by $\text{Do}'$, which is defined inductively as follows:

$$\text{Do}'(\alpha) \overset{\text{def}}{=} \text{Do}(\alpha) \quad (7.63)$$

$$\text{Do}'(\alpha; \pi) \overset{\text{def}}{=} \text{Do}(\alpha); (\text{Distress}_C? + (\neg \text{Distress}_C?; \text{Do}'(\pi))) \quad (7.64)$$

where $\text{Distress}_C \overset{\text{def}}{=} \exists \kappa : C \models \text{Distress}_T(\kappa)$. Observe that “if $\varphi$ then $\pi_1$ else $\pi_2$” is a common abbreviation of $(\varphi?; \pi_1) + (\neg \varphi?; \pi_2)$. So $\text{Do}'(\pi)$ expands to $\pi$ interleaved with $\neg \text{Distress}_C?$ tests. The triggering of distress then effectively functions as a “stop” signal with respect to the execution of a plan.

7.3.5 Discussion

To summarize, we have constrained the deliberation cycle such that (1) PG-rules are only applied to goals that have triggered hope, (2) PR-rules are only applied to plans that have triggered fear, (3) plans that have triggered joy are preferred for execution, and (4) plan execution is interrupted as soon as distress is triggered. It will
be clear that these constraints do not resolve all nondeterminism in the specification of \textit{Reason\_Act} in Definition 7.22. Specifically, the following nondeterminism remains:

- When different PG-rules can be applied (in the sense of rule (7.59)), it is not specified how many of them are to be applied and in which order.

- When different PR-rules can be applied (in the sense of rule (7.60)), it is not specified how many of them are to be applied and in which order.

- The order $\prec_C$ imposed on the plan base by joy may not result in a unique most preferred plan. As noted before $\prec_C$ can be refined to further reduce nondeterministic choices.

- The triggering of distress interrupts the execution of a plan, but it is not specified what should happen after that. The current construction makes it possible for the agent to switch to another plan, but it is also allowed to continue with the same plan.

- Although $(\text{ApplyPGrule} + \text{ApplyPRrule} + \text{ExecutePlan})^*$ in Definition 7.22 appears to suggest the order “first apply PG-rules, then apply PR-rules, then execute plans,” the use of nondeterministic choice (+) inside an iteration (•) effectively allows any order for performing \text{ApplyPGrule}, \text{ApplyPRrule}, and \text{ExecutePlan}.

- It is not specified how sensing of the environment is interleaved with reasoning and acting.

Investigating principled ways to resolve this remaining nondeterminism is left for future work. In particular, it will be interesting to study other emotion types for their suitability to control agent deliberation. It should be noted though that probably not all emotion types will be useful for this purpose. As described before, psychological, sociological, and neurological studies have shown that an absence of emotions is very detrimental to proper functioning in society and thus bad for the survival of individuals. This implies that at least some emotion types serve important purposes. However, for other emotion types there appears to be no good reason other than that they are vestigial mechanisms, i.e., left-overs from human evolution. For example, there would seem to be no advantages for an individual to ruminate, be depressed, or have phobias. Therefore, when implementing intelligent agents it may not be sensible to try and incorporate every type of emotion (from the OCC model).

Still, it is important to keep in mind why one wants to incorporate emotions in an artificial system. For example, if one wants to build a companion robot, it makes little sense to make it capable of becoming depressed. On the other hand, in (serious) games it may be interesting to have virtual characters that are capable of hurting themselves through emotions. Interacting with such characters may also be used to test theories of human emotion in a safe way.

\subsection*{7.3.6 Related Work}

Let us close this subsection by comparing the presented work with related approaches at using emotions in reasoning.
Dastani and Meyer [2006] have proposed to constrain the deliberation cycle of 2APL using heuristics inspired by the emotions happiness, sadness, fear, and anger, as described in the psychological model of emotions of Oatley and Jenkins [1996]. These heuristics were then added on top of the deliberation cycle of 2APL. However, the existing 2APL deliberation cycle itself, which is based on ad hoc choices, was not changed in a principled way (i.e., the emotion-inspired heuristics were only used to extend the deliberation cycle). In contrast, our approach starts with a clean slate by assuming complete nondeterminism. Then we have added several constraints in accordance with the common psychological view of emotions as “go” and “stop” signals in task-oriented situations. Note that it is our intention that any remaining nondeterminism be resolved by investigating and adding more principled (possibly emotion-based) constraints.

Gratch and Marsella [2004] have been working on a computational framework for modeling emotions inspired by the OCC model, among others. An implementation, named EMA, is used for social training applications. Like Adam [2007], their framework incorporates a number of coping strategies. However, few formal details of how emotions affect the reasoning of an agent are provided, so it is hard to assess how much of the behavior of an EMA agent results from ad hoc choices or emotion-inspired principles. Concerning the three main topics of modeling emotions discussed in this dissertation (i.e., appraisal, experience, and regulation), Gratch and Marsella distinguish only between appraisal and regulation. Probably owing to their computational, quantitative approach, emotional experience appears to be merged with appraisal in EMA. Their main focus is on modeling how emotions influence behavior, emphasizing modeling of coping strategies for artificial agents.

### 7.4 Concluding Remarks

In this chapter we have investigated the third of the three main phases of emotion, namely emotion regulation. The main idea behind emotion regulation is that agents want to select actions such that the experience of positive emotions is maximized and the experience of negative emotions is minimized. When an agent has an action at its disposal with which it can increase the intensity a positive emotion or decrease the intensity of a negative emotion, the agent will experience a tendency to perform this action. This notion of ‘action tendency’ has been formalized in this chapter in a very general form, and its properties investigated by studying theorems involving action tendencies (or the lack thereof). Although there is no formal, consensual specification of ‘action tendency’ in the literature, the formalized notion of ‘action tendency’ captures the basic idea that tendencies arise towards those actions that are expected to result in a gain of emotional reward (i.e., more positive and less negative emotions). Of course it is possible to come up with different ways of formalizing ‘action tendency’; even to have them together in a single framework. For example, in previous work [Steunebrink et al., 2009a] we have discussed and formally compared several alternative “flavors” of action tendency, namely a ‘strict’ version (requiring “active” improvement), a ‘long-term’ version (avoiding short-sighted improvements), and an ‘overall’ version (not being relative to any particular
emotion.

A second, more applied approach to studying the effects of emotions on behavior has also been presented. For this we have taken an implementation of a simple but generic BDI-based agent and investigated how its deliberation can be controlled using four emotion types (hope, fear, joy, and distress). This was done in a principled manner, by first doing away with all ad hoc design choices present in a typical implementation of an agent’s sense–reason–act cycle. Emotion-inspired constraint were then added to reduce some of the nondeterminism in the stripped-down sense–reason–act cycle. Interestingly, some of the emotion-inspired constraints closely matched previous (ad hoc) design choices, while others deviated or introduced entirely new dynamics. This applied approach of using emotions to constrain agent deliberation can thus be used to build support for certain existing design choices, while also offering novel but principled options for implementing intelligent agents.

7.5 Proofs

Action Tendency

Below are proofs of the propositions presented throughout Section 7.2.

Proposition (7.5). \((\rightarrow)\) Take an arbitrary model–state pair \((M, s) \in S\), positive emotion \((\text{Emotion}^+, i, \bar{o}) \in \text{Emotion}\), and action \(\alpha \in \text{act}\) and assume \(M, s \models T_a(\text{Emotion}^+, i, \bar{o})\). Then by Definition 6.12, \(T_a(\text{Emotion}^+, i, \bar{o})(M, s) > 0\), which by Definition 7.3 implies \(M, s \models \text{Emotion}^+_\alpha(\bar{o}) \land \text{Can}_\alpha(\alpha, \tau) \land w \cdot \text{gain} > 0\). Because \(w > 0\) by definition, it must be that \(\text{gain} > 0\), which is exactly the case when \(\text{E}_\text{OFI}_\alpha(\text{Emotion}^+, i, \bar{o})(M, s) > \text{OFI}(\text{Emotion}^+, i, \bar{o})(M, s)\). But by formula (6.39) this is the same as \(\text{E}_\text{OFI}_\alpha > \text{Emotion}^+_\alpha(\bar{o})\) \text{OFI}\. Because \((M, s)\) was arbitrary, we have that \(T_a(\text{Emotion}^+, i, \bar{o}) \rightarrow \text{Emotion}^+_\alpha(\bar{o}) \land \text{Can}_\alpha(\alpha, \tau) \land \text{E}_\text{OFI}_\alpha > \text{Emotion}^+_\alpha(\bar{o})\) \text{OFI}\ is valid.

\((\leftarrow)\) Take an arbitrary model–state pair \((M, s) \in S\), positive emotion \((\text{Emotion}^+, i, \bar{o}) \in \text{Emotion}\), and action \(\alpha \in \text{act}\) and assume \(M, s \models \text{Emotion}^+_\alpha(\bar{o}) \land \text{Can}_\alpha(\alpha, \tau) \land \text{E}_\text{OFI}_\alpha > \text{Emotion}^+_\alpha(\bar{o})\) \text{OFI}\. To prove: \(M, s \models T_a(\text{Emotion}^+, i, \bar{o})\), i.e., \(T_a(\text{Emotion}^+, i, \bar{o})(M, s) > 0\), i.e., \(w \cdot \text{gain} > 0\), i.e., \(\text{E}_\text{OFI}_\alpha(\text{Emotion}^+, i, \bar{o})(M, s) > \text{OFI}(\text{Emotion}^+, i, \bar{o})(M, s)\). But by formula (6.39) this is the same as \(\text{E}_\text{OFI}_\alpha > \text{Emotion}^+_\alpha(\bar{o})\) \text{OFI}\, which holds by assumption. Because \((M, s)\) was arbitrary, we have that \(\text{Emotion}^+_\alpha(\bar{o}) \land \text{Can}_\alpha(\alpha, \tau) \land \text{E}_\text{OFI}_\alpha > \text{Emotion}^+_\alpha(\bar{o})\) \text{OFI}\ \rightarrow T_a(\text{Emotion}^+, i, \bar{o})\) is valid.

In conclusion, \(T_a(\text{Emotion}^+, i, \bar{o}) \leftrightarrow \text{Emotion}^+_\alpha(\bar{o}) \land \text{Can}_\alpha(\alpha, \tau) \land \text{E}_\text{OFI}_\alpha > \text{Emotion}^+_\alpha(\bar{o})\) \text{OFI}\ is valid. \(\square\)

Proposition (7.6). The proof of this propositions is analogous to that of proposition (7.5) above. \(\square\)

Proposition (7.8). \((\rightarrow)\) Take an arbitrary model–state pair \((M, s) \in S\) (where \(M = \langle S, R, V \rangle\)) and emotion \((\text{Emotion}, i, \bar{o}) \in \text{Emotion}\) and assume \(M, s \models \text{OFI}(\text{Emotion}, i, \bar{o})\). Then by Definition 6.12, \(\text{OFI}(\text{Emotion}, i, \bar{o})(M, s) > 0\). Now take an arbitrary state \(s' \in S\) such that \((s, s') \in R_i\). Then by the IntBel assumption, \(\text{OFI}(\text{Emotion}, i, \bar{o})(M, s) = \text{OFI}(\text{Emotion}, i, \bar{o})(M, s') > 0\). So \(M, s' \models \text{OFI}(\text{Emotion}, i, \bar{o})\). Because \(s'\) was an arbi-
trary belief-accessible state, we have that $M, s \models B, OFI(\text{Emotion}, i, \bar{o})$. Because $(M, s)$ was arbitrary, we have that $OFI(\text{Emotion}, i, \bar{o}) \rightarrow B, OFI(\text{Emotion}, i, \bar{o})$ is valid.

$(\leftarrow)$ Take an arbitrary model–state pair $(M, s) \in S$ (where $M = (S, R, V)$) and emotion $(\text{Emotion}, i, \bar{o}) \in \text{emotion}$ and assume $M, s \models B, OFI(\text{Emotion}, i, \bar{o})$. Then for all $s' \in S$ such that $(s, s') \in R$, we have that $M, s' \models OFI(\text{Emotion}, i, \bar{o})$, i.e., $OFI(\text{Emotion}, i, \bar{o})(M, s') > 0$. But for state $s$ we have that $OFI(\text{Emotion}, i, \bar{o})(M, s) = OFI(\text{Emotion}, i, \bar{o})(M, s') > 0$. So $M, s \models OFI(\text{Emotion}, i, \bar{o})$. Because $(M, s)$ was arbitrary, we have that $B, OFI(\text{Emotion}, i, \bar{o}) \rightarrow OFI(\text{Emotion}, i, \bar{o})$ is valid.

In conclusion, $OFI(\text{Emotion}, i, \bar{o}) \leftrightarrow B, OFI(\text{Emotion}, i, \bar{o})$ is valid. □

Proposition (7.9). Take an arbitrary model–state pair $(M, s) \in S$, positive emotion $(\text{Emotion}^+, i, \bar{o}) \in \text{emotion}$, and action $\alpha \in \text{act}$ and assume $M, s \models \text{Emotion}^+_\bar{o}(\bar{o}) \land \text{Can}_i(\alpha, OFI) >^{\text{Emotion}^+_\bar{o}} \text{PrevoFI}$. To prove: $M, s \models T_\alpha(\text{Emotion}^+, i, \bar{o})$, i.e., $T_\alpha(\text{Emotion}^+, i, \bar{o})(M, s) > 0$, i.e., $w \cdot \text{gain} > 0$, i.e., $E_{OFI_\alpha}(\text{Emotion}^+, i, \bar{o})(M, s) > OFI(\text{Emotion}^+, i, \bar{o})(M, s)$. Let us note that $\text{Can}_i(\alpha, \phi)$ implies $B_\alpha(\alpha, \phi)$, which by the assumption that action $\alpha$ is deterministic implies $B_\alpha[\alpha, \phi]$. So $M, s \models \text{Can}_i(\alpha, OFI) >^{\text{Emotion}^+_\bar{o}} \text{PrevoFI}$ implies that for all model–state pairs $(M', s') \in S$ such that $((M, s), (M', s')) \in R^M \circ R_{\text{act}}$ it holds that $M', s' \models OFI >^{\text{Emotion}^+_\bar{o}} \text{PrevoFI}$, i.e., $OFI(\text{Emotion}^+, i, \bar{o})(M', s') > PrevoFI(\text{Emotion}^+, i, \bar{o})(M', s')$. By the definition of $\text{PrevoFI}$, this is the same as $OFI(\text{Emotion}^+, i, \bar{o})(M', s') > OFI(\text{Emotion}^+, i, \bar{o})(M'', s'')$ where $((M'', s''), (M', s')) \in R_{\text{act}}$. Now this model–state pair $(M'', s'')$ is reachable from $(M, s)$ by $R^M$, i.e., $((M, s), (M'', s'')) \in R^M$. But that means that $M = M''$ and $s, s'' \in R$. Then by the IntBel assumption we have that $OFI(\text{Emotion}^+, i, \bar{o})(M', s') > OFI(\text{Emotion}^+, i, \bar{o})(M'', s'') = OFI(\text{Emotion}^+, i, \bar{o})(M, s)$. But if in every possible presumed state resulting from performing $\alpha$ the overall felt intensity is greater than currently, this means that the expected overall felt intensity is greater than the current overall felt intensity. That is, $E_{OFI_\alpha}(\text{Emotion}^+, i, \bar{o})(M, s) > OFI(\text{Emotion}^+, i, \bar{o})(M, s)$, i.e., $E_{OFI_\alpha} >^{\text{Emotion}^+_\bar{o}} \text{OFI}$. Then by formula (7.5), $M, s \models T_\alpha(\text{Emotion}^+, i, \bar{o})$. Because $(M, s)$ was arbitrary, we have that $\text{Emotion}^+_\bar{o}(\bar{o}) \land \text{Can}_i(\alpha, OFI) >^{\text{Emotion}^+_\bar{o}} \text{PrevoFI} \rightarrow T_\alpha(\text{Emotion}^+, i, \bar{o})$ is valid. □

Proposition (7.10). The proof of this propositions is analogous to that of proposition (7.9) above. □

Proposition (7.11). Take an arbitrary model–state pair $(M, s) \in S$, negative emotion $(\text{Emotion}^-, i, \bar{o}) \in \text{emotion}$, and action $\alpha \in \text{act}$ and assume $M, s \models \text{Emotion}^-_\bar{o}(\bar{o}) \land \text{Can}_i(\alpha, \neg OFI(\text{Emotion}^-, i, \bar{o}))$. To prove: $M, s \models T_\alpha(\text{Emotion}^-, i, \bar{o})$, i.e., $E_{OFI_\alpha}(\text{Emotion}^-, i, \bar{o})(M, s) < OFI(\text{Emotion}^-, i, \bar{o})(M, s)$. Let us note from the proof of formula (7.9) above that $M, s \models \text{Can}_i(\alpha, \neg OFI(\text{Emotion}^-, i, \bar{o}))$ implies that for all model–state pairs $(M', s') \in S$ such that $((M, s), (M', s')) \in R^M \circ R_{\text{act}}$ it holds that $OFI(\text{Emotion}^-, i, \bar{o})(M', s') = 0$. Moreover, let us note that $OFI(\text{Emotion}^-, i, \bar{o})(M', s') = 0 < OFI(\text{Emotion}^-, i, \bar{o})(M'', s'') = OFI(\text{Emotion}^-, i, \bar{o})(M, s)$ for all $((M, s), (M'', s'')) \in R^M$. This means that in every possible presumed state resulting from performing $\alpha$ the overall felt intensity is zero, which is less than the current intensity. That is, $E_{OFI_\alpha}(\text{Emotion}^-, i, \bar{o})(M, s) < OFI(\text{Emotion}^-, i, \bar{o})(M, s)$, i.e., $E_{OFI_\alpha} <^{\text{Emotion}^-_\bar{o}} \text{OFI}$. Then by formula (7.6) we have that $M, s \models T_\alpha(\text{Emotion}^-, i, \bar{o})$. Because $(M, s)$ was arbi-
trary, we have that $\text{Emotion}_i(\overline{\partial}) \land \text{Can}_i(\alpha, \neg \text{OFI}(\text{Emotion}^-, i, \overline{\partial})) \rightarrow T_{\overline{\alpha}}(\text{Emotion}^-, i, \overline{\partial})$ is valid.

**Proposition (7.12).** The proof of this propositions is analogous to that of proposition (7.11) above.

**Proposition (7.13).** Take an arbitrary model–state pair $(M, s) \in S$, positive emotion $(\text{Emotion}^+, i, \overline{\partial}) \in \text{EMOTION}$, and action $\alpha \in \text{ACT}$. Now assume that $M, s \models B_i[i;\alpha](\text{OFI} \leq \text{Emotion}^+(\overline{i}) \ \text{PrevOFI})$. Then for all $((M, s), (M', s')) \in R_i^M \circ R_{\text{st}}$ it holds that $M', s' \models \text{OFI} \leq \text{Emotion}^+(\overline{i}) \ \text{PrevOFI}$, i.e., $\text{OFI}(\text{Emotion}^+, i, \overline{\partial})(M', s') \leq \text{PrevOFI}(\text{Emotion}^+, i, \overline{\partial})(M', s')$. By the definition of $\text{PrevOFI}$, this is the same as $\text{OFI}(\text{Emotion}^+, i, \overline{\partial})(M', s') \leq \text{OFI}(\text{Emotion}^+, i, \overline{\partial})(M'', s'')$ where $((M'', s''), (M', s')) \in R_{\text{st}}$. Now this model–state pair $(M'', s'')$ is reachable from $(M, s)$ by $R_i^M$, i.e., $((M, s), (M'', s'')) \in R_i^M$. But that means that $M = M''$ and $(s, s'') \in R_i$. Then by the IntBel assumption we have that $\text{OFI}(\text{Emotion}^+, i, \overline{\partial})(M', s') \leq \text{OFI}(\text{Emotion}^+, i, \overline{\partial})(M'', s'') = \text{OFI}(\text{Emotion}^+, i, \overline{\partial})(M, s)$. But this means that $E_{\text{OFI}_i}(\text{Emotion}^+, i, \overline{\partial})(M, s) \leq \text{OFI}(\text{Emotion}^+, i, \overline{\partial})(M, s)$, because the above holds for all $(M', s')$ reachable by $R_i^M \circ R_{\text{st}}$ from $(M, s)$. Then by Definition 7.3, the gain associated with action $i;\alpha$ cannot be positive. Therefore, $T_{\overline{\alpha}}(\text{Emotion}^+, i, \overline{\partial})(M, s) = 0$, i.e., $M, s \models \neg T_{\overline{\alpha}}(\text{Emotion}^+, i, \overline{\partial})$. Because $(M, s)$ was arbitrary, we have that the formula $B_i[i;\alpha](\text{OFI} \leq \text{Emotion}^+(\overline{i}) \ \text{PrevOFI}) \rightarrow \neg T_{\overline{\alpha}}(\text{Emotion}^+, i, \overline{\partial})$ is valid.

**Proposition (7.14).** The proof of this propositions is analogous to that of proposition (7.13) above.

**Proposition (7.15).** By Constraint 7.6 the monotonically decreasing intensity functions assigned to emotions are not changed, so the proposition $[i;\text{idle}](\text{OFI} \leq \text{Emotion}^+(\overline{i}) \ \text{PrevOFI})$ must be valid. That is, for each emotion the overall felt intensity in a state resulting from performing an idle action can be at most that of the previous state. By necessitation, the proposition $B_i[i;\text{idle}](\text{OFI} \leq \text{Emotion}^+(\overline{i}) \ \text{PrevOFI})$ must also be valid. Combining with formula (7.13) then yields that $\neg T_{\text{idle}}(\text{Emotion}^+, i, \overline{\partial})$ is valid.

**Proposition (7.16).** By Constraint 7.6 the monotonically decreasing intensity functions assigned to emotions are not changed, and they are all strict by the IntSMD assumption, so the proposition $[i;\text{idle}](\text{OFI} \prec \text{Emotion}^+(\overline{i}) \ \text{PrevOFI})$ must be valid. That is, for each emotion the overall felt intensity in a state resulting from performing an idle action must be less than that of the previous state. By necessitation, the proposition $B_i[i;\text{idle}](\text{OFI} \prec \text{Emotion}^+(\overline{i}) \ \text{PrevOFI})$ must also be valid. By Constraint 7.6 we also have that $\text{Can}_i(\text{idle}, \top)$ is valid. Combining with formula (7.10) then yields that $\text{Emotion}^+(\overline{\partial}) \rightarrow T_{\text{idle}}(\text{Emotion}^-, i, \overline{\partial})$ is valid. The reverse implication also holds by formula (7.6), so $\text{Emotion}^+(\overline{\partial}) \leftrightarrow T_{\text{idle}}(\text{Emotion}^-, i, \overline{\partial})$ is valid.

**Proposition (7.17).** The proposition to prove is $\text{Pity}_i(\overline{\varphi}, j) \land \text{Can}_i(\alpha, \text{Gratitude}^T_i(i;\alpha, \varphi)) \rightarrow T_{\overline{\alpha}}(\text{Pity}_i, i, (\overline{\varphi}, j))$. If we assume $M, s \models \text{Pity}_i(\overline{\varphi}, j) \land \text{Can}_i(\alpha, \text{Gratitude}^T_i(i;\alpha, \varphi))$ for an arbitrary model–state pair $(M, s)$ and if we can show that
M, s \models \text{Can}_i(\alpha, \neg\text{OFI}(\text{Pity}, i, \langle \overline{q}, j \rangle)), \text{ then we can see from formula (7.11) that we obtain } T_\alpha(\text{Pity}, i, \langle \overline{q}, j \rangle) \text{ and we are done. Now the assumption } M, s \models \text{Can}_i(\alpha, \text{Gratitude}^T(i\alpha, \varphi)) \text{ implies } M, s \models B_i(i\alpha)\text{Gratitude}^T(i\alpha, \varphi), \text{ which by the assumption that action } i\alpha \text{ is deterministic implies } M, s \models B_i[i\alpha]\text{Gratitude}^T(i\alpha, \varphi). \text{ But by the assumption that action } i\alpha \text{ is accordant implies } M, s \models B_i[i\alpha]B_i\text{Gratitude}^T(i\alpha, \varphi). \text{ (To see why this is, recall that action } i\alpha \text{ being accordant means that } B_i[i\alpha]\varphi \rightarrow [i\alpha]B_i\varphi \text{ is valid; then by necessitation } B_iB_i[i\alpha]\varphi \rightarrow B_i[i\alpha]B_i\varphi, \text{ and by transitivity of the belief-accessibility relations } B_i[i\alpha]\varphi \rightarrow B_i[i\alpha]B_i\varphi \text{ is valid.) Now take an arbitrary state } (M', s') \in S \text{ such that } ((M, s), (M', s')) \in R_i^M \circ R_{i\alpha}. \text{ Because } M', s' \models B_i\text{Gratitude}^T(i\alpha, \varphi), \text{ we can apply Constraint 7.8 and obtain that } \text{IntFun}(\text{Pity}, i, \langle \overline{q}, j \rangle)(M', s') = I_0. \text{ So the overall felt intensity assigned to the } \text{Pity},(\overline{q}, j) \text{ emotion in state } (M', s') \text{ will be zero, i.e., } M', s' \models \neg\text{OFI}(\text{Pity}, i, \langle \overline{q}, j \rangle). \text{ Because } (M', s') \text{ was arbitrary, we have that } M, s \models B_i[i\alpha]\neg\text{OFI}(\text{Pity}, i, \langle \overline{q}, j \rangle). \text{ Because action } i\alpha \text{ 'can' be performed (i.e., } M, s \models \text{Can}_i(\alpha, \top)), \text{ we obtain } M, s \models \text{Can}_i(\alpha, \neg\text{OFI}(\text{Pity}, i, \langle \overline{q}, j \rangle)), \text{ which is what we had to prove.} \\

**Proposition (7.18).** The proof of this propositions is analogous to that of proposition (7.17) above.

**Proposition (7.19).** The proposition to prove is \( \text{Liking}_i(j) \land \text{Can}_i(\alpha, \text{Admiration}^T(i\alpha)) \rightarrow T_\alpha(\text{Liking}, i, j) \). If we assume \( M, s \models \text{Liking}_i(j) \land \text{Can}_i(\alpha, \text{Admiration}^T(i\alpha)) \) for an arbitrary model–state pair \((M, s)\) and if we can show that \( M, s \models \text{Can}_i(\alpha, \text{OFI} > \text{Liking}_i(j) \text{-prevOFI}) \), then we can see from formula (7.9) that we obtain \( T_i(\text{Liking}, i, j) \) and we are done. Now the assumption \( M, s \models \text{Can}_i(\alpha, \text{Admiration}^T(i\alpha)) \) implies \( M, s \models B_i[i\alpha]B_i\text{Admiration}^T(i\alpha) \) by the same reasoning as in proposition (7.17). Now take an arbitrary state \((M', s') \in S \) such that \(((M, s), (M', s')) \in R_i^M \circ R_{i\alpha}. \text{ Because } M', s' \models B_i\text{Admiration}^T(i\alpha), \text{ we can apply Constraint 7.10 and obtain that } \text{IntFun}(\text{Liking}, i, j)(M', s') = I^+, \text{ where } I^+ \text{ is an intensity function such that we have that } \text{OFI}(\text{Liking}, i, j)(M', s') > \text{OFI}(\text{Liking}, i, j)(M'', s'') \text{ for } ((M'', s''), (M', s')) \in R_{i\alpha}. \text{ Then by formula (7.7) we have that } \text{OFI}(\text{Liking}, i, j)(M', s') > \text{PrevOFI}(\text{Liking}, i, j)(M', s'), \text{ which by formula (6.39) equals } M', s' \models \text{OFI} > \text{Liking}_i(j) \text{-prevOFI}. \text{ Because } (M', s') \text{ was arbitrary, we have that } M, s \models B_i[i\alpha]\text{OFI} > \text{Liking}_i(j) \text{-prevOFI}. \text{ Because action } i\alpha \text{'can' be performed (i.e., } M, s \models \text{Can}_i(\alpha, \top)), \text{ we have that } M, s \models \text{Can}_i(\alpha, \text{OFI} > \text{Liking}_i(j) \text{-prevOFI}), \text{ which is what we had to prove.} \\

**Proposition (7.20).** The proof of this propositions is analogous to that of proposition (7.19) above.

**Controlling Deliberation**

Below are proofs of the propositions presented throughout Section 7.3.

**Propositions (7.47)–(7.50).** According to Definition 3.8, \( \text{Hope}_i^T(\kappa) \) is equivalent to \( \text{Des}_i(\kappa) \land \text{FutUpd}_i(\kappa) \lor \text{UncUpd}_i(\kappa) \). But \( \text{UncUpd}_i(\kappa) \) implies \( \neg B_i\kappa \land \neg B_i \neg \kappa \), which is a contradiction because of our closed world assumption. So \( \text{Hope}_i^T(\kappa) \) is equivalent to
Desₜ(κ) ∧ New (B; ∼κ ∧ B; Fut κ). The same reasoning holds for Fearₜ(κ). The theorems for Joyₜ(κ) and Distressₜ(κ) are just the macros of Definition 3.9 expanded.

Propositions (7.51)–(7.54). These theorems are direct rewrites of (7.47)–(7.50) (see above). It should be noted that the term New (B; ∼ξ ∧ B; Fut ξ) in theorem (7.47) and (7.48) expands to B; ∼ξ ∧ B; Fut ξ ∧ Prev (B; ∼ξ ∧ B; Fut ξ). Interpreting this formula in a configuration C yields C ⊨ B; ∼ξ ∧ B; Fut ξ and C ⊨ Prev (B; ∼ξ ∧ B; Fut ξ), i.e., C⁻¹ ⊭ B; ∼ξ ∧ B; Fut ξ. By Definition 7.24 this is equivalent to Futₜ⁺(κ) & ¬Futₜ⁻(κ).

Proposition (7.56). To obtain this theorem it suffices to show that Applicableₜ(κ | β → π) implies Futₜ⁺(κ) and G ⊨ g κ. Applicableₜ(κ | β → π) expands to (κ | β → π) ∈ PG & G ⊨ g κ & B ⊨ b (β ∧ ¬κ), while Futₜ⁺(κ) expands to B ⊨ b ¬κ & ∃r ∈ PG : κ ⊑ τ(r) & B ⊨ b G(r). Assume Applicableₜ(κ | β → π). Then immediately we have G ⊨ g κ and B ⊨ b ¬κ. But because (κ | β → π) ∈ PG and κ ⊑ κ, we have that ∃r ∈ PG : κ ⊑ τ(r). Moreover, B ⊨ b G(r) is now the same as B ⊨ b β, which we have by assumption. So indeed Applicableₜ(κ | β → π) implies Futₜ⁺(κ) and G ⊨ g κ.

Proposition (7.58). To obtain this theorem it suffices to show that ∃(π₁, κ₁, r₁), (π₂, κ₂, r₂) ∈ P : κ ∈ PostCondₜ(r₂) & κ ⊑ κ₁ implies Futₜ⁺(κ) and G ⊨ g κ. Assume the antecedent above. Then ∃(π, κ', r) ∈ P : κ ∈ PostCondₜ(r), which is equivalent to Futₜ⁺(κ). But by Constraint 7.20 (π₁, κ₁, r₁) ∈ P implies G ⊨ g κ₁. Because κ ⊑ κ₁, also G ⊨ g κ. So indeed ∃(π₁, κ₁, r₁), (π₂, κ₂, r₂) ∈ P : κ ∈ PostCondₜ(r₂) & κ ⊑ κ₁ implies Futₜ⁺(κ) and G ⊨ g κ.
In a nutshell, this dissertation has built a bridge between psychological models of human emotions on the one hand, and implementation of emotions in robots and virtual characters on the other hand. The work presented in this dissertation concerns very much the logical structure of emotions, as the title suggests. For example, we have not been concerned with determining (numerical) values for, say, the intensity of a fear emotion; rather, we have been more concerned with relating the intensity of the fear emotion—whatever value it may have—to the intensity of a corresponding hope emotion. For this reason the work has been verified using formal analysis instead of implementation. Still, we have provided as many details on calculation as can be found in the psychological OCC model, and have studied possible calculations in light of our formal framework.

In order to put the presentation of the main results of this dissertation in context, let us quickly recall the basic “three-phase model” of emotion (see Figure 8.1) that has been used as the backbone of our formalization. First, an emotion is elicited by testing the agent’s percepts against its concerns and adding some individual parameters. So three kinds of data affect the elicitation of emotions. Second, the result of emotion elicitation is that some emotions will be experienced. Note that not all newly elicited emotions are necessarily experienced (they may lie below some threshold), and that emotions may be experienced that have been elicited some time in the past. Third, the emotions experienced by an agent can be actively regulated by the agent by selecting actions that influence its experienced emotions. Typically, the agent will be inclined to select actions that lead to or strengthen positive emotions and that weaken, mitigate, or prevent negative emotions.

Let us now present the main results and other highlights of this dissertation.
8.1 Main Results

In short, the main results of this dissertation consist of the formalization of each of the three phases of emotion (elicitation, experience, regulation), where the formalization of each successive step builds on the results of the previous step. The resulting formalization constitutes our answer to the main research question, described in the introduction as “How can emotions be formalized?”. On the way many other interesting aspects of emotions were identified and incorporated into the formalization. Moreover, right at the start, the preparations for the formalization have resulted in a novel analysis and structure of the existing types of emotion. Below we give a “chronological” overview of the results; that is, they are shown in the order in which they appear in this dissertation.

- The backbone of our formalization of emotions has been formed by a *procedural view* of emotion. This view distinguishes certain types of data (e.g., percept, concerns, experienced emotions) from processes (e.g., emotion elicitation, emotion regulation). Although similar perspectives of emotion can be found throughout the existing psychological literature on emotion (indeed, our

---

**Figure 8.1:** A view of emotion in terms of data (square boxes) and processes (rounded boxes). This figure is a repetition of Figure 1.2.
procedural view is based on these), the precise articulation of emotion in terms of data and processes is novel to the best of our knowledge. At the start of each chapter initiating a new phase in the formalization (i.e., Chapters 3, 6, and 7), we have reiterated our procedural view in order to put the work into context.

- With respect to emotion elicitation, we have constructed a novel *compositional structure* of the eliciting conditions of emotions as specified in the psychological “OCC model” (see Figure 2.1). This has been done in order to prepare the psychological model for translation to formal logic, which is compositional. As a result, our mapping of the psychological notions used in specifying the eliciting conditions to formal constructs has been very direct and transparent. Given the popularity of the OCC model in the affective computing community, our novel compositional structure of OCC’s emotion types can be a useful tool for others who want to construct a new formalization or implementation of emotions in robots or virtual characters. It should be noted that the presentation of this structure constitutes our answer to Research Question 1.

- The process of formalizing the OCC model has brought several *ambiguities* in its specifications to light. (This is another good reason to do formalization.) As shown in the discussion of related work in Section 3.4, different computer scientists attempting to formalize the OCC model have chosen different interpretations regarding these ambiguities. However, we have contacted the original authors [Ortony and Clore, 2009; Ortony, 2009] to obtain the intended readings of ambiguous aspects of OCC’s specifications. This brings our formalization more in line with psychological work on emotion.

- We have formalized the eliciting conditions of emotions in *three stages*. This has been done to provide different levels of commitment to formalisms, such that extensions and modifications can easily be made to the framework without having to start from scratch. The three stages were presented in Chapters 3, 4, and 5, respectively. In the first stage, only propositional logic was used and all notions that could not be properly captured therein were represented using predicates with suggestive names. In the second stage a commitment was made to dynamic doxastic logic. This allowed us to ground many of the predicates used in the first stage. In the third stage, a commitment was made to KARO, which is an extension of dynamic doxastic logic with operators for representing motivational attitudes of agents. This allowed us to ground the remaining appraisal operators (e.g., desirability, praiseworthiness, appealingness) in terms of an agent’s motivations. The advantages of this staged approach are transparency thanks to the gradual complexification of the formal framework, and convenient extension and modification points for other researchers. It should be noted that the formalization of emotion elicitation resulting from this staged approach constitutes our answer to Research Question 2.

- The presented formalization of emotions has been built on the idea that emotion elicitation, experience, and regulation can and should be distinguished. In the affective computing literature the first two are often together referred to as ‘appraisal’. In the other hand, the psychological OCC model of human emotions
that we have followed makes a clear distinction between the conditions that trigger emotions, and the variables that affect the intensities of emotions. The former constitutes a qualitative description of emotion elicitation, whereas the latter constitutes a quantitative description of emotional experience. In line with this distinction, our formalization has the property that emotion elicitation does not imply emotional experience, nor vice versa. Rather, they are tied together using the notion of overall felt intensity, which is a property indicating for any (currently or previously) elicited emotion whether or not the emotion in question is currently experienced. Thus we have treated emotional experience (in Chapter 6) as a concept separate from emotion elicitation, in line with our procedural view of emotion. Emotional experience was defined in terms of (past) satisfaction of the eliciting conditions of an emotion, plus a positive overall felt intensity of the emotion. With this construction, the elicitation and the experience of an emotion do not have to co-occur (although experience does require elicitation to have happened some time in the past).

- Quantitative aspects of emotions have been described in the OCC model in terms of variables affecting intensity. The OCC model identifies 25 variables that affect the intensities of emotions (e.g., arousal, likelihood, familiarity), but not all variables affect all types of emotion. For example, all emotions of an agent are affected by the arousal of the agent, but not all of its emotions require a likelihood estimate to be made. In order to capture the structure of the quantitative aspects of emotions, we have constructed another compositional view of emotions, but this time based on the assignment of the variables affecting intensity to the emotion types (see Figure 6.3). This structure was then incorporated in its entirety into the formal framework. Even though we have not discussed possible calculations for all variables affecting intensity, they are all present in the framework, together with a formalization of any relations between them. It should be noted that the treatment of the variables affecting intensity, and their incorporation into an overall felt intensity, constitutes our answer to Research Question 3.

- Throughout this dissertation we have been more concerned with the logical structure underlying emotions than with calculating their magnitudes. Thus we have been more concerned with relations than numbers. Note that this logical structure is supposed to be the same for every ('normal') individual. An interesting result of focusing on relations and deferring the actual calculation of intensities is that it explicates what are the individual-dependent and environment-dependent weights and parameters that eventually determine the strengths of the emotional responses. The collection of these weights and parameters can be said to represent the emotional 'character' of an agent, because they are what is different between agents.

- One thing that makes formalizing emotional experience hard is that it is not known what is the dimensionality of emotional experience. Even though usually people can make an estimate of the overall felt intensity of an emotion on a unidimensional scale, this does not mean that there are no other dimen-
8.2 Future Work

Although our formalization of emotions has been comprehensive with respect to the psychological OCC model, there remain several areas where the work can and should be extended. Most of the limitations of the presented formalization pertain to our choice to focus on the popular BDI paradigm (Belief, Desire, Intention), and to our focus on structure instead of numerical values. Below we give an overview of the topics that will be interesting for future work.

- The types of concerns of agents have been restricted to achievement goals. This has been a convenient choice given the popularity of the BDI paradigm and the availability of BDI-based agent implementations that rely (mostly) on achievement goals. In future extensions of this work, the integration of other types of goals, such as interest goals and maintenance goals, has to be studied because these may give rise to new and interesting properties. However, not all emotion
types are related to goals. According to the OCC model, events are appraised with respect to goals, but actions are appraised with respect to standards, and objects are appraised with respect to attitudes (including tastes). In this dissertation we have grounded the notions of standards and attitudes in achievement goals, but in future work it will have to be considered how standards and attitudes can be represented in their own right, and what are the properties that this extension will give rise to.

- The representation of objects in our framework is not particularly strong. For example, it is not possible to describe aspects of objects except as other objects. This may limit the power of the object-based emotion types of 'liking' and 'disliking'. For example, Alice may express a liking for the architecture of Bob's new house and a disliking for the color of the house, while adding that on the whole she likes Bob's house. In the current framework it is not straightforward to represent affective reactions to different aspects of objects and relating them to affective reactions to the overall object.

- Although the relations between the variables affecting the intensity of emotions have been explored, the actual values these are supposed to output have not. Moreover, values for weights and parameters (such as durations and fall-off rates) used in modeling these variables have not been specified. We have speculated before that these weights and parameters together may account (in large part) for the emotional 'character' of an agent. Still, finding appropriate settings for them does not have to involve too much work, as is noted by Ortony, Clore, and Collins themselves:

"( . . ) it is not necessary to assume great precision in the computational mechanisms. The assignment of values to individual variables can be, and doubtless often is, quite imprecise, having a strong qualitative flavor. It seems unlikely that the internal psychophysics of value assignment is, or need be, very precise, and, in fact, even the notion of "assigning" values may be too strong. It may be preferable, and sufficient, to think of values being available rather than being assigned (. . ). What is important is what happens to the variables and how they have their effects."

(Italics in original. Quoted from [Ortony, Clore, and Collins, 1988, page 83].)

- The influence of mood and other affective phenomena on emotions and vice versa remains to be explicated. In this dissertation we have used mood for determining thresholds for the experience of emotions, and we have speculated that experienced emotions in turn influence the mood. This entanglement of the dynamics of emotions and mood, and probably other affective phenomena too, is something that remains to be properly investigated.

- Finally, as was also highlighted in the quote above, the most important aspect of emotions is how they affect behavior and action selection in particular. In Chapter 7 we have shown how emotions can be used to specify principled ways
of reducing nondeterminism in the deliberation of agents including action selection. It will be interesting to pursue this direction of research further because not all nondeterminism has been eliminated yet in a principled way, forcing ad hoc design choices in implementations of the deliberation mechanisms of artificial agents.

8.3 Final Remarks

The formalization of emotions presented in this dissertation has always been developed with the idea that it should be modifiable and extensible, and moreover, that it will be clear at which points and how modifications and extensions can be made. We have done this because we recognize that emotion is a huge subject about which the final word has not been written yet in psychology and neurology and other fields. Therefore we have endeavored to make our work flexible enough to allow for new insights to be incorporated. For example, the formalization of eliciting conditions has been undertaken in three stages to allow different formalisms with different properties to be plugged in, if desired, without having to start from scratch. The appraisal operators have been constrained (as opposed to defined) with respect to achievement goals, so that other types of concerns can easily be incorporated by introducing more constraints on these appraisal operators. With respect to emotional experience, we have allowed any dimension of experience to be specified; any emotion word from any language with all its connotations can be defined; and emotional ‘characters’ can be defined by appropriately setting the identified weights and parameters.

So the presented formalization of emotions is not the end, the final word. Quite the opposite is true: our formalization is intended to be the starting point of further investigations (and appropriate modifications) and implementation in affective robots and virtual characters.
## Summary of the OCC Model

This appendix summarizes the specifications of emotions from the book *The Cognitive Structure of Emotions* by Ortony, Clore, and Collins [1988], also known as the OCC model. The intended readings of these specifications is explained on page 21. They are included here for easy reference and to make this dissertation more self-contained.

### Joy Emotions

**Type Specification:** (pleased about) a desirable event  
**Tokens:** contented, cheerful, delighted, ecstatic, elated, euphoric, feeling good, glad, happy, joyful, jubilant, pleasantly surprised, pleased, etc.  
**Variables Affecting Intensity:**  
1. The degree to which the event is desirable  
**Example:** The man was pleased when he realized he was to get a small inheritance from an unknown distant relative.

### Distress Emotions

**Type Specification:** (displeased about) an undesirable event  
**Tokens:** depressed, distressed, displeased, dissatisfied, distraught, feeling bad, feeling uncomfortable, grief, homesick, lonely, lovesick, miserable, regret, sad, shock, uneasy, unhappy, upset, etc.  
**Variables Affecting Intensity:**  
1. The degree to which the event is undesirable  
**Example:** The driver was upset about running out of gas on the freeway.
HAPPY-FOR EMOTIONS
TYPE SPECIFICATION: (pleased about) an event presumed to be desirable for someone else
TOKENS: delighted-for, happy-for, pleased-for, etc.
VARIABLES AFFECTING INTENSITY:
(1) the degree to which the desirable event for the other is desirable for oneself
(2) the degree to which the event is presumed to be desirable for the other person
(3) the degree to which the other person deserved the event
(3) the degree to which the other person is liked
EXAMPLE: Fred was happy for his friend Mary because she won a thousand dollars.

PITY\textsuperscript{1} EMOTIONS
TYPE SPECIFICATION: (displeased about) an event presumed to be undesirable for someone else
TOKENS: compassion, pity, sad-for, sorry-for, sympathy, etc.
VARIABLES AFFECTING INTENSITY:
(1) the degree to which the undesirable event for the other is undesirable for oneself
(2) the degree to which the event is presumed to be undesirable for the other person
(3) the degree to which the other person did not deserved the event
(3) the degree to which the other person is liked
EXAMPLE: Fred was sorry for his friend Mary because her husband was killed in a car crash.

RESENTMENT EMOTIONS
TYPE SPECIFICATION: (displeased about) an event presumed to be desirable for someone else
TOKENS: envy, jealousy, resentment, etc.
VARIABLES AFFECTING INTENSITY:
(1) the degree to which the desirable event for the other is undesirable for oneself
(2) the degree to which the event is presumed to be desirable for the other person
(3) the degree to which the other person did not deserved the event
(3) the degree to which the other person is not liked
EXAMPLE: The executive resented the large pay raise awarded to a colleague whom he considered incompetent.

\textsuperscript{1}In [Ortony et al., 1988, page 93] it says SORRY-FOR EMOTIONS here, but the label ‘pity’ is used in OCC’s figures and in our formalization.
<table>
<thead>
<tr>
<th>EMOTION TYPE</th>
<th>TYPE SPECIFICATION</th>
<th>TOKENS</th>
<th>VARIABLES AFFECTING INTENSITY</th>
<th>EXAMPLE</th>
</tr>
</thead>
</table>
| Gloating Emotions | (pleased about) an event presumed to be undesirable for someone else | gloating, Schadenfreude, etc. | (1) the degree to which the undesirable event for the other is desirable for oneself  
(2) the degree to which the event is presumed to be undesirable for the other person  
(3) the degree to which the other person deserved the event  
(4) the degree to which the other person is not liked | Political opponents of Richard Nixon gloated over his ignominious departure from office. |
| Hope Emotions | (pleased about) the prospect of a desirable event | anticipation, anticipatory excitement, excitement, expectancy, hope, hopeful, looking forward to, etc. | (1) the degree to which the event is desirable  
(2) the likelihood of the event | As she thought about the possibility of being asked to the dance, the girl was filled with hope. |
| Fear Emotions | (displeased about) the prospect of an undesirable event | apprehensive, anxious, cowering, dread, fear, fright, nervous, petrified, scared, terrified, timid, worried, etc. | (1) the degree to which the event is undesirable  
(2) the likelihood of the event | The employee, suspecting he was no longer needed, feared that he would be fired. |
| Satisfaction Emotions | (pleased about) the confirmation of the prospect of a desirable event | gratification, hopes-realized, satisfaction, etc. | (1) the intensity of the attendant hope emotion  
(2) the effort expended in trying to attain the event  
(3) the degree to which the event is realized | When she realized that she was indeed being asked to go to the dance by the boy of her dreams, the girl was gratified. |
FEARS-CONFIRMED EMOTIONS
TYPE SPECIFICATION: (displeased about) the confirmation of the prospect of an undesirable event
TOKENS: fears-confirmed, worst fears realized
VARIABLES AFFECTING INTENSITY:
(1) the intensity of the attendant fear emotion
(2) the effort expended in trying to prevent the event
(3) the degree to which the event is realized
EXAMPLE: The employee’s fears were confirmed when he learned that he was indeed going to be fired.

RELIEF EMOTIONS
TYPE SPECIFICATION: (pleased about) the disconfirmation of the prospect of an undesirable event
TOKENS: relief
VARIABLES AFFECTING INTENSITY:
(1) the intensity of the attendant fear emotion
(2) the effort expended in trying to prevent the event
(3) the degree to which the event is realized
EXAMPLE: The employee was relieved to learn that he was not going to be fired.

DISAPPOINTMENT EMOTIONS
TYPE SPECIFICATION: (displeased about) the disconfirmation of the prospect of a desirable event
TOKENS: dashed-hopes, despair, disappointment, frustration, heartbroken, etc.
VARIABLES AFFECTING INTENSITY:
(1) the intensity of the attendant hope emotion
(2) the effort expended in trying to attain the event
(3) the degree to which the event is realized
EXAMPLE: The girl was disappointed when she realized that she would not be asked to the dance after all.

PRIDE EMOTIONS
TYPE SPECIFICATION: (approving of) one’s own praiseworthy action
TOKENS: pride
VARIABLES AFFECTING INTENSITY:
(1) the degree of judged praiseworthiness
(2) the strength of the cognitive unit with the actual agent
(3) deviations of the agent’s action from person/role-based expectations (i.e., unexpectedness)
EXAMPLE: The woman was proud of saving the life of a drowning child.
**SHAME\(^2\) EMOTIONS**

**TYPE SPECIFICATION:** (disapproving of) one’s own blameworthy action  
**TOKENS:** embarrassment, feeling guilty, mortified, self-blame, self-condemnation, self-reproach, shame, (psychologically) uncomfortable, uneasy, etc.  
**VARIABLES AFFECTING INTENSITY:**  
(1) the degree of judged blameworthiness  
(2) the strength of the cognitive unit with the actual agent  
(3) deviations of the agent’s action from person/role-based expectations (i.e., unexpectedness)  
**EXAMPLE:** The spy was ashamed of having betrayed his country.

---

**ADMIRATION\(^3\) EMOTIONS**

**TYPE SPECIFICATION:** (approving of) someone else’s praiseworthy action  
**TOKENS:** admiration, appreciation, awe, esteem, respect, etc.  
**VARIABLES AFFECTING INTENSITY:**  
(1) the degree of judged praiseworthiness  
(2) deviations of the agent’s action from person/role-based expectations (i.e., unexpectedness)  
**EXAMPLE:** The physicist’s colleagues admired him for his Nobel-prize-winning work.

---

**REPROACH EMOTIONS**

**TYPE SPECIFICATION:** (disapproving of) someone else’s blameworthy action  
**TOKENS:** appalled, contempt, despise, disdain, indignation, reproach, etc.  
**VARIABLES AFFECTING INTENSITY:**  
(1) the degree of judged blameworthiness  
(2) deviations of the agent’s action from person/role-based expectations (i.e., unexpectedness)  
**EXAMPLE:** Many people despised the spy for having betrayed his country.

---

**GRATITUDE EMOTIONS**

**TYPE SPECIFICATION:** (approving of) someone else’s praiseworthy action and (being pleased about) the related desirable event  
**TOKENS:** appreciation, gratitude, feeling indebted, thankful, etc.  
**VARIABLES AFFECTING INTENSITY:**  
(1) the degree of judged praiseworthiness  
(2) deviations of the agent’s action from person/role-based expectations (i.e., unexpectedness)  
(3) the degree to which the event is desirable  
**EXAMPLE:** The woman was grateful to the stranger for saving the life of her child.

---

\(^2\)In [Ortony et al., 1988, page 137] it says **SELF-REPROACH EMOTIONS** here, but the label ‘shame’ is used in OCC’s figures and in our formalization.  
\(^3\)In [Ortony et al., 1988, page 145] it says **APPRECIATION EMOTIONS** here, but the label ‘admiration’ is used in OCC’s figures and in our formalization.
ANGRy EMOTIONS
TYPE SPECIFICATION: (disapproving of) someone else’s blameworthy action and (being displeased about) the related undesirable event
TOKENS: anger, annoyance, exasperation, fury, incensed, indignation, irritation, livid, offended, outrage, rage, etc.
VARIABLES AFFECTING INTENSITY:
(1) the degree of judged blameworthiness
(2) deviations of the agent’s action from person/role-based expectations (i.e., unexpectedness)
(3) the degree to which the event is undesirable
EXAMPLE: The woman was angry with her husband for forgetting to buy the groceries.

GRATIFICATION EMOTIONS
TYPE SPECIFICATION: (approving of) one’s own praiseworthy action and (being pleased about) the related desirable event
TOKENS: gratification, pleased-with-oneself, self-satisfaction, smug, etc.
VARIABLES AFFECTING INTENSITY:
(1) the degree of judged praiseworthiness
(2) the strength of the cognitive unit with the agent
(3) deviations of the agent’s action from person/role-based expectations (i.e., unexpectedness)
(4) the degree to which the event is desirable
EXAMPLE: The man was gratified by his daughter’s achievements.

REMorSE EMOTIONS
TYPE SPECIFICATION: (disapproving of) one’s own blameworthy action and (being displeased about) the related undesirable event
TOKENS: penitent, remorse, self-anger, etc.
VARIABLES AFFECTING INTENSITY:
(1) the degree of judged blameworthiness
(2) the strength of the cognitive unit with the agent
(3) deviations of the agent’s action from person/role-based expectations (i.e., unexpectedness)
(4) the degree to which the event is undesirable
EXAMPLE: The spy felt remorse at the damage he had done in betraying his country.

\(^4\)In [Ortony et al., 1988, page 148] it says “undesirable” here, but this is obviously a typographical error.
LIKING EMOTIONS
TYPE SPECIFICATION: (liking) an appealing object
TOKENS: adore, affection, attracted-to, like, love, etc.
VARIABLES AFFECTING INTENSITY:
(1) the degree to which the object is appealing
(2) the degree of familiarity with the object
EXAMPLE: Mary was filled with affection as she gazed at her newborn infant.

DISLIKING EMOTIONS
TYPE SPECIFICATION: (disliking) an unappealing object
TOKENS: aversion, detest, disgust, dislike, hate, loathe, repelled-by, revulsion, etc.
VARIABLES AFFECTING INTENSITY:
(1) the degree to which the object is unappealing
(2) the degree of familiarity with the object
EXAMPLE: John disliked the concert so much that he left in the middle.

Finally, we include two diagrams, printed on pages 19 and 69 of [Ortony, Clore, and Collins, 1988], respectively. They are included here so that they can be compared with our novel structures presented in Figure 2.1 and Figure 6.3, respectively.
Figure A.1: A view of the emotions of the OCC model based on focus of attention. Taken from [Ortony, Clore, and Collins, 1988, page 19].
Figure A.2: A view of the emotions of the OCC model based on variables affecting intensity. Taken from [Ortony, Clore, and Collins, 1988, page 69].
In this appendix we will describe in more detail the context in which this research has been performed.

## B.1 The Boon Companion Project

This research has been performed in the context of the *Boon Companion* project.\(^1\) This project was instigated by the French toy company Berchet, with the aim of producing a robot which could function as a playmate and, more importantly, as a tutor for children. Producing such a robot called for many innovations in the programming of robots, in particular with respect to making robots sociable and capable of creating learning opportunities for children. Therefore many European research groups were attracted to participate in this project. The Boon Companion project was funded from the ITEA (Information Technology for European Advancement) program.

Shortly after funding for the Boon Companion project was approved, Berchet was acquired by Smoby, another French toy company, which withdrew Berchet’s involvement in the Boon Companion project. Renault Trucks temporarily offered to step in as client for the Boon Companion project, suggesting that instead of a robotic companion for children, a companion for truckers could be developed. Nevertheless, the Dutch research partners in the project decided to continue in the spirit of the Boon Companion project in a home setting, but with slightly revised means and objectives. Section B.1.1 below details the approach and expected results of the *Dutch Companion* project, as the branch of the Boon Companion project involving the Dutch partners was named.

---
\(^1\)Many non-native English speakers appear to be unfamiliar with the expression “boon companion,” but it is synonymous with “best friend.”
B.1.1 The Dutch Companion Project

Below follows a copy of the project description by D-CIS, the chair of the Dutch Companion project. (D-CIS is an open research partnership of Thales, TNO, the University of Amsterdam and the Delft University of Technology.)

This work is supported by SenterNovem, Dutch Companion project grant nr: IS053013. The Dutch consortium consists of Thales (coordinator), Philips, Sound Intelligence, University of Groningen, and University of Utrecht [sic].

Most of today’s information processing systems interact with their users via built-in, pre-defined rule-based interfaces. They have no or little knowledge of the environment or their user(s). In fact these systems assume a static user environment, which can seldom be guaranteed. In the best of cases the interaction models contain some built-in knowledge of the user’s capabilities, but lack any knowledge of the current state of the user’s tacit or background knowledge, the user’s situational awareness, skills, and actual “state-of-mind.” This contrasts heavily with the standard human-human interaction, which contain more elaborate patterns of use and situational awareness, thus grounding the belief that the conception and design of complex information systems that are based on cognitive capabilities will be the key differentiator between future products.

Approach

The main focus of Dutch Companion is on improving the effectiveness of actor-agent interaction through making an agent aware of human emotions and expected social interactions. An agent can then respond not only with appropriate content but also with appropriate emotional state.

The agent will have the embodiment of Philips’ iCat. The iCat is a plug & play desktop user-interface robot, capable of mechanically rendering facial expressions. The iCat will be situated inside the kitchen. It learns to take preferences and emotional state into account when suggesting meals or activities to their inhabitants.

The project concentrates on the following:

- The ability to focus attention on the user’s needs for information and assistance as a function of the user’s situation, goals and current (cognitive) capabilities and emotional states;
- The ability to adapt in real time to the behavior and responses beyond the mere use of some built-in static user model.

The first point will be addressed by looking at the use of sound recognition and analysis to enhance its situational awareness. The classification

---

2 In Section B.1.3 we will give a detailed description of the iCat.
of particular sounds from the environment will be used by the iCat to alert the user, depending on the current user context. Via deliberation it determines whether or not the user should be alerted.

The second point will be addressed by detection of emotion in the user’s voice (isolated from the overall sound input), combining the emotional output from other modalities, such as facial recognition and gestures. The perceived emotional state can then be used as a controlling input for the iCat’s deliberation. The personal and social based deliberation designs and implementations will be based on agent-oriented programming, in particular 3APL.\(^3\) This language enables one to program the agent’s behavior based on its beliefs, goals and plans.

A central element of 3APL is the way the agent deliberates how to perform plans given certain goals and beliefs, in particular which goals and plans to select and pursue, and which goals and plans to revise under the circumstances given. This so-called ‘deliberation cycle’ has to be extended to also include the effect of emotional states, as well the influence of social aspects such as norms and obligations arising from the social context. In addition we will couple the emotion recognition with the communication and dialogue iCat module, thus changing the nature of the dialogue and the content of the communication depending on the emotional state of the user.

Note that interacting with agents requires integration of different fields in AI, cognitive science, computer science, etc. For example, speech recognition is still difficult in uncontrolled environments. Equally challenging still is computer vision. If we foresee a future of actor agent communities, all these issues on the sensor/input side, as well as cognitive and actuator side need to be cross-disciplinary resolved. Starting by building a small system and deploy it in an operational setting, will uncover new unforeseen research issues. Furthermore, note that such cognitive systems will always serve specific purposes and not yet provide generic solutions to ‘universal’ problems. In different situations or context, people have different tasks, or make more or less use of certain abilities, so artificial systems should know what to support in which context. For example, a human driving a car needs different support than human inside its home.

**Expected Results**

The Dutch Companion project is expected to deliver concrete results based on experimentation and research regarding embodied emotional agents and natural actor-agent interaction. The expected short term results are summarized below.

---

\(^3\)3APL ("An Abstract Agent Programming Language" [Dastani et al., 2004]) is a programming platform developed at Utrecht University. 3APL allows *agents* to be programmed in terms high-level constructs such as ‘beliefs’, ‘goals’, and ‘plans’.
1. **System Architecture.** From an overall actor agent system’s point of view, we foresee Dutch Companion to deliver implemented new functional system architectural insights by combining (i) sound recognition, (ii) emotion recognition, (iii) affect controlled deliberation, (iv) concept formation, (v) affect controlled communication and dialogue, and (vi) cognitive system architectures.

2. **Validating Experiments.** We develop and deploy a demonstrator framework for autonomous cognitive systems (ACSs) that show the added value of actor agent system architecture. In particular, for these purposes we assess the iCat demonstrator with its users. This assessment will be mainly conducted in the Home Lab of Philips Research in Eindhoven. The iCat will be placed in the kitchen of the Home Lab and connected to the home-network. We test how users will interact with the iCat, focusing on the effectiveness, enjoyment and acceptance of using such a domestic companion to assist and accompany a user doing (kitchen) tasks.

### B.1.2 The Partners

With these new objectives, the following groups participated in the Dutch Companion project.

D-CIS functioned as chair of the Dutch Companion project. As such, D-CIS was responsible for the overall architecture of the companion robot’s software. Furthermore, because of their expertise in knowledge representation, D-CIS was responsible for providing the domain knowledge and databases required for the project’s application.

Philips Research provided the iCat robot such that the Dutch Companion project could focus entirely on the software side of companion robots. It should be noted that Philips had already taken care of most of the low-level programming issues by shipping a programming platform with the iCat. The iCat robot itself is described in detail in Section B.1.3; see also Figure B.1.

The University of Groningen together with the company Sound Intelligence was responsible for the sound awareness of the companion robot. Sound awareness means recognizing what kinds of sounds are picked up by the robot’s microphones and what they mean. Note that this means that not only speech could be recognized by the robot, but also other sounds such as the doorbell, screams, slamming of doors, overflying airplanes, etc. It should also be noted that recognizing the fact that speech is present is just that; it does not include any processing of the speech patterns.

Acapela Group is a company that provided text-to-speech synthesis and automatic speech recognition software. This software allowed us to make the robot read out any text (many different languages and voices were supported) and convert things spoken to the robot to flat text. Note that this does not involve any interpreting or reasoning about speech. Composing the desired text to be pronounced by the robot is a wholly different issue, as is interpreting and reasoning about any speech converted to text by the automatic speech recognition software.
We at Utrecht University were then responsible for two things: providing the reasoning capabilities of the companion robot, and providing a dialogue system such that the companion robot could make sensible conversations. The latter task then became the dissertation research of Nieske Vergunst, while I was assigned to the former task. All actual programming tasks were performed by Christian Mol. The reasoning system that I designed had to support reasoning about social situations, take into account the users’ emotions, and be appropriately responsive (run in ‘real-time’). This dissertation particularly focuses on one of these aspects (namely emotion).

B.1.3 The iCat

As has been mentioned several times above, the robot that was used as a platform for the ‘Dutch Companion’ was the iCat, developed and provided by Philips. As the name suggests, the iCat is a robotic cat; however, it has several human-like features. The iCat has a cartoon-like appearance because it has human-like lips, eyes, eyelids, and eyebrows, but it is fixed in a sphinx-like position, so the iCat cannot walk or drive around. Figure B.1 shows the iCat in relation to a person (it stands about 38 cm tall) and illustrates some of the expressions which it is capable of making. Philips Research have also used the iCat themselves as an experimentation platform for human-robot interaction [van Breemen, 2005].

The iCat is equipped with a number of sensors to give it the potential to be aware of its environment. These sensors are summarized below.

- iCat has a small camera in its ‘nose’ with a resolution of $640 \times 480$ pixels at 60 frames per second.

- There is a microphone in each of iCat’s two ‘paws’. This makes it possible for the iCat to determine the direction of a sound source. When appropriate, the
Figure B.2: This picture shows the final version of the Philips iCat robot. At the back are an on/off switch, three USB plugs for motor, video, and audio data, a DC 13V plug, and mini jack in and out plugs so that an external microphone and speakers can be connected to the iCat.

iCat can thus be made to look in the direction of a sound source, adding to its appearance of awareness.

- The left ‘paw’ houses an infrared proximity sensor. Although the camera can also be used to determine where objects and persons are, the proximity sensor makes it possible to determine with accuracy the distance at which someone is standing in front of the iCat. Since it is likely that a person addressing the iCat will naturally go and stand in front of the robot, the proximity sensor can be a useful tool in dialogues.

- On top of the ‘paws’ and just behind the ‘ears’ are four capacitive touch sensors. This allows the iCat to respond to petting.

Although the iCat is stationary, it can move its body, neck, and face to make a wide range of expressions. A complete list of the robot’s actuators is provided below.

- The iCat has thirteen degrees of freedom: one in the body for turning left and right, one in the neck for moving the head up and down, three in the eyes for looking up and down (together) and left and right (independently), two in the eyebrows (they can be rotated independently), two in the eyelids (they can be closed independently), and four in lips (each lip corner can rotate independently).

- The top of each ‘paw’ and the front of each ‘ear’ contains a LED which can show red, green, or blue light. These lights can serve many purposes, such as showing agreement (e.g., by flashing green) or error (e.g., by flashing red), or to give a hint of the robot’s mood, or to acknowledge that the co-located touch sensor has been activated.
A loudspeaker is located between the iCat’s ‘paws’, i.e., in its ‘chest’. Although the iCat can play back any sound, it should be noted that the programming platform provided by Philips supports automatic lip syncing when text is spoken by the robot.

It should be noted that the iCat is only a shell with motors, sensors, and a sound card, i.e., without processor(s). The iCat has to be connected by USB to a computer running the software necessary to drive the robot’s motors. Indeed, this implies that the iCat’s ‘brain’ (i.e., the reasoning software) is located outside the robot. Thus the robot is no more than a physical user-interface. Of course, the whole idea of the iCat is that it is a most intuitive user-interface, because everyone already knows how to talk to a face.

The iCat is shipped with a programming platform called OPPR (“Open Platform for Personal Robotics”). Among the software provided with the iCat is also an Animation Editor with which it is easy to design animations involving any of the iCat’s actuators. Moreover, the scripting language Lua can be used to make even more flexible animations.

B.2 A Companion Robot as Research Context

In this section we will sketch the difficulties and complexities involved in building a companion robot. Many of the issues mentioned here can probably be used to fill an entire dissertation project; indeed, one of them (i.e., emotion) has been used as the topic of the current dissertation. Still, in this section we will discuss the possible requirements and design of a generic companion robot or virtual character in order to put this research into context. This section is based on a paper titled “A Generic Architecture for a Companion Robot” [Steunebrink et al., 2008b].

B.2.1 The Need for a Generic Architecture

Companion robots are supposed to exhibit sociable behavior and perform several different kinds of tasks in cooperation with a human user. Typically, they should proactively assist users in everyday tasks and engage in intuitive, expressive, and affective interaction. Moreover, they usually have multiple sensors and actuators that allow for rich communication with the user. Of course, the task of designing and building a companion robot is highly complex. This holds for both the hardware and software of the robot. In our case, the hardware was provided to us by Philips in the form of the iCat robot, but the software design challenges remain.

Companion robots and affective virtual characters have already been built up to quite advanced stages. However, teams wishing to research companion robots often have to start from scratch on the software design part, because it is hard to distill a firm basis from the literature to work from. Such a basis could be a detailed specification of an architecture. An architecture shows which components there are and how they are connected. Although many figures representing architectures have been published, it remains difficult to find out how existing companion robots really work internally because the presented architectures are not specified in enough detail. This may be
due to most publications focusing on test results of the overall behaviors rather than on explaining their architectures in the level of detail required for replication.

The lack of emphasis on architectures may be caused by much of the research on companion robots being driven by the teams' research goals, resulting in their architectures mostly being designed to support just the desired behaviors instead of being generic for companion robots. If there were a good generic architecture for companion robots, a (simple) default implementation could be made, providing (new) researchers with a framework that they could use as a starting point. Depending on the application domain and research goals, some default implementations of modules constituting the architecture may be replaced to achieve the desired custom behavior, while other modules can just be readily used to complete the software of the companion robot.

In this section we introduce an architecture which is generic for companion robots and explain it in as much detail as possible in this limited space. This architecture contains the components necessary to produce reasonably social behavior given the multimodality of a companion robot's inputs and outputs. We do not claim that the proposed architecture represents the ultimate companion robot architecture. Rather, it is used to identify the issues and choices involved in designing the software of a companion robot and to function as a context for the rest of this dissertation.

### B.2.2 Possible Requirements for a Companion Robot

In order to come up with a generic architecture suitable for companion robots, we must first investigate the possible requirements for a companion robot. These requirements are optional, meaning that only the 'ultimate' companion robot would satisfy them all. In practice however, a companion robot does not have to. The actual requirements depend on various factors, such as the application area of the robot and its hardware configuration. However, below we compile a list, as exhaustive as possible, of possible requirements which a generic architecture must take into account.

- First of all, a companion robot should be able to perceive the world around it, including auditory, visual, and tactile information. The multimodality of the input creates the need for synchronization (e.g., visual input and simultaneously occurring auditory input are very likely to be related), and any input inconsistent over different modalities should be resolved. Moreover, input processors can be driven by expectations from a reasoning system to focus the robot's attention to certain signals. Of course, any incoming data must be checked for relevancy and must be categorized if it is to be stored (e.g., to keep separate models of the environment, its users, and domain knowledge).

- A companion robot should be able to communicate with the user in a reasonably social manner. This means not only producing sensible utterances, but also taking into account basic rules of communication (such as topic consistency). In order to maintain a robust interaction, a companion robot must always be able to keep the conversation going (except of course when the user indicates that he is done with the conversation). This also involves real-time aspects; for
example, to avoid confusing or boring the user, long silences should not occur in a conversation.

- Additionally, a companion robot is likely to be designed for certain specific tasks, besides communicating with its users. Depending on, e.g., the domain for which the companion robot is designed and the type of robot and the types of tasks involved, this may call for capabilities involving planning, physical actions such as moving around and manipulating objects, or electronic actions (e.g., performing a search on the internet or programming a DVD recorder). Proactiveness on part of the robot is often desirable in tasks involving cooperation with the user.

- A companion robot should also exhibit some low-level reactive behaviors that do not (have to) enter the reasoning loop, such as blinking and following the user’s face, and fast reactive behaviors such as startling when subjected to a sudden loud noise. To make the interactions more natural and intuitive, a companion robot should also be able to form and exhibit emotions. These emotions can be caused by cognitive-level events, such as plans failing (disappointment), goal achievement (joy), and perceived emotions from the user (pitying the user if he or she is sad). Reactive emotions like startle or disgust can also influence a robot’s emotional state. Moreover, emotions can manifest themselves in many different ways, for example in facial expressions, speech prosody, selecting or abandoning certain plans, etc.

- Finally, a companion robot should of course produce coherent and sensible output over all available modalities. Because different processes may produce output concurrently and because a companion robot typically has multiple output modalities, there should be a mechanism to synchronize, prioritize, and/or merge these output signals. For example, speech should coincide with appropriate lip movements, which should overrule the current facial animation, but only the part that concerns the mouth of the robot (provided it has a mouth with lips, like our iCat).

Figure B.3 then presents a generic architecture for companion robots which accounts for the requirements described above. Note that we abstract from specific robot details, making the architecture useful for different types of companion robots. We emphasize again that this is an architecture for an ‘ultimate’ companion robot; in practice, some modules can be left out or implemented empty. For example, many companion robots exhibit emotions, but not all companion robots do so.

Of course, anyone wishing to build the software of a companion robot can just start up his/her favorite programming environment and try to deal with problems when they occur, but obviously this is not a very good strategy to follow. Instead, designing and discussing an architecture beforehand raises interesting issues and allows questions to be asked that otherwise remain hidden. Indeed, there are many non-trivial choices that have to be made, pertaining to, for example: distribution and assignment of control among processes; synchronization of concurrent processes; which process is to convert what data into what form; where to store data and in
Figure B.3: A generic architecture for a companion robot. The architecture takes into account the possible (or rather, probable) existence of multiple input modalities, multiple input preprocessing modules for each input modality, databases for filtering, storing, and querying relevant information, action selection engines for complex, goal-directed, long-term processes such as conversing, planning, and locomotion, an emotion synthesizer producing emotions that influence action selection and animations, multiple (reactive) low-level behaviors that can compete for output control, multiple output preprocessing modules including a conflict manager, and finally, multiple output modalities. Straight boxes stand for data storages, rounded boxes for processes, and ovals for sensors/actuators. The interfaces (arrows) between different modules indicate flow of data or control; the connections and contents are made more precise in the text. Note that only the ‘ultimate’ companion robot would fully implement all depicted modules; a typical companion robot implementation will probably leave out some modules or implement them empty, awaiting future work.
what form; which process has access to which stored data; which process/data influences which other process and how; the types of action abstractions that can be distinguished (e.g., strategic planning actions, dialogue actions, locomotion actions); the level of action abstraction used for reasoning; who converts abstract actions into control signals; how are conflicts in control signals resolved; what are the properties of a behavior emerging from a chosen wiring of modules; what defines the character/personality of a companion robot (is it stored somewhere, can its parameters be tweaked, or does it emerge from the interactions between the modules?). Answers to these and many other questions may not be obvious when presented with a figure representing an architecture, but these issues can be made explicit by proposing and discussing one.

B.2.3 Functional Components Constituting the Architecture

In this section we describe the ‘blocks’ that constitute the proposed architecture. The interfaces (‘arrows’) between the components are explained in the next section.

To begin with, the architecture is divided into eight functional components (i.e., the larger boxes encompassing the smaller blocks). Each functional component contains several modules that are functionally related. Modules drawn as straight boxes represent data storages, the rounded boxes represent processes, and the ovals represent sensors and actuators. Each process is allowed to run in a separate thread, or even on a different, dedicated machine.

No synchronization is forced between these processes by the architecture; they can simply send information to each other (see the next section), delegating the task of making links between data coming in from different sources to the processes themselves.

Below, each of the eight functional components is described, together with the modules they encompass.

Input Modalities A companion robot typically has a rich arsenal of input modalities or sensors. These are grouped in the lower left corner of Figure B.3, but only partially filled in. Of course, different kinds of companion robots can have different input modalities, of which a camera and a microphone are probably the most widely occurring. Other sensors may include touch, (infrared) proximity, accelerometer, etc.

Input Preprocessing It is impractical for a reasoning engine to work directly with most raw input data, especially raw visual and auditory data. Therefore, several input preprocessing modules must exist in order to extract salient features from these raw inputs and convert these to a suitable data format. Some input modalities may even require multiple preprocessing modules; for example, one audio processing module may extract only speech from an audio signal and produce text, while another audio processing module may extract other kinds of sounds to create a level of ‘sound awareness’ for the companion robot. Note that some of these input preprocessing modules may be readily available as off-the-shelf software (most notably, speech recognizers), so a generic architecture must provide a place for them to be plugged in.
Furthermore, there may be need for an input synchronizer that can make links between processed data from different modalities, in order to pass it as a single event to another module. The input synchronizer may initially be implemented empty; that is, it simply passes all processed data unchanged to connected modules. The input synchronizer can also be used to dispatch expectations that are formed by the action selection engines to the input preprocessing modules, which can use these expectations to facilitate feature recognition.

Low-level Behaviors Low-level behaviors are autonomous processes that compete for control of actuators in an emergent way. Some behaviors may also influence each other and other modules. Examples of low-level behaviors include face tracking and gaze directing, blinking, breathing, and other ‘idle’ animations, homeostasis such as the need for interaction, sleep, and ‘hunger’ (low battery power), and reactive emotions such as startle and disgust.

Action Selection Engines The ‘heart’ of the architecture is formed by the action selection engines. These are cognitive-level processes that select actions based on collections of data, goals, plans, events, rules, and heuristics. The outputs that they produce can generally not be directly executed by the actuators, but will have to be preprocessed first to appropriate control signals. Note that the interpreters of the action selection engines are depicted as layered to indicate that they can be multi-threaded.

The reasoning engine may be based on the BDI theory of beliefs, desires, and intentions [Bratman, 1987], deciding which actions to take based on percepts and its internal state. It should be noted that in terms of the BDI theory, the databases component plus the working memories of the action selection engines constitute the robot’s beliefs. An action selected by the reasoning engine may be sent to an output preprocessing module, but it can also consist of a request to initiate a dialogue. Because dialogues are generally complex and spread over a longer period of time, a dedicated action selection engine may be needed to successfully have a conversation. This dialogue engine contains an extra process called an utterance formulator; the task of this module is to convert an illocutionary act to fully annotated text, i.e. the exact text to utter together with information about speed, emphasis, tone, etc. This text can then be converted to audio output by the text-to-speech module (in the output preprocessing component).

A similar discussion about separating dialogues and (strategic) planning can be held for locomotion. In our research we have worked with stationary companion robots that focus on dialogues and facial animations. But there can of course be companion robots with advanced limbs and motions. For such robots there may be need for a third action selection engine, dedicated to motion planning. In the proposed architecture, there is room for additional dedicated engines in the functional component of action selection engines.

Finally, the architecture provides for a module called heuristics / timing rules. This is a collection of heuristics for balancing control between the different action selection engines, as they are assumed to be autonomous processes. The
different engines will get priorities in different cases. For example, the plans of the dialogue engine will get top priority if a misunderstanding needs to be repaired. On the other hand, if the dialogue engine does not have any urgent issues, the reasoning engine will get control over the interaction in order to address its goals. Furthermore, it can verify whether the goals of the different action selection engines adhere to certain norms that apply to the companion robot in question, as well as provide new goals based on timing rules. For example, to avoid long silences, the robot should always say something within a few seconds, even if the reasoning engine is still busy.

**Databases** The architecture contains a distinct functional component where data is stored in different forms. This data includes domain knowledge, ontologies, situation models, and profiles of the robot itself and of other agents. The ontologies and domain knowledge are (possibly static) databases that are used by the input preprocessing modules to find data representations suitable to the action selection engines and databases. The agent profiles store information about other agents, such as the robot’s interaction histories with these modeled agents, the common grounds between the robot and each modeled agent, and the presumed beliefs, goals, plans, and emotions of each modeled agent. These agent models also include one of the robot itself, which enables it to reason about its own emotions, goals, etc.

In order to provide a consistent interface to these different databases, a query manager must be in place to handle queries, originating from the action selection engines. A special situation arises when the robot queries its own agent model, for there already exist modules containing the goals, plans, and emotions of the robot itself. So the query manager should ensure that queries concerning these types of data can get their results directly from these modules.

Finally, a relevant data extractor takes care of interpreting incoming data in order to determine whether it can be stored in a more suitable format. For example, if visual and auditory data from the input preprocessing component provides new (updated) information about the environment of the robot, it is interpreted by the relevant data extractor and stored in the situation model. Moreover, simple spatial-temporal reasoning may be performed by the relevant data extractor. If advanced spatial-temporal reasoning is needed for some companion robot, it may be better to delegate this task to a separate input preprocessing module.

**Emotion Synthesizer** Typically, companion robots must show some level of affective behavior. This means responding appropriately to emotions of a (human) user, but also includes experiencing emotions itself in response to the current situation and its internal state. The emotions that concern this functional component are those of the companion robot itself and are at a cognitive level, i.e., at the level of the action selection engines. Examples of emotions are joy when a goal is achieved, disappointment when a plan fails, resentment when another agent (e.g., a human user) gains something at the expense of the robot, etc. More reactive emotions (e.g., startle) can be handled by a low-level behavior.
The emotion component consists of three parts. The appraiser is a process that triggers the creation of emotions based on the state of the action selection engines. The intensity of triggered emotions is influenced by the robot’s mood (the representation of which may be as simple as a single number) and a database of previously triggered emotions. This database of emotions then influences the action selection engines (by way of their emotional heuristics module) and the animations of the robot, e.g., by showing a happy or sad face.

Output Preprocessing Different processes may try to control the robot’s actuators at the same time; obviously, this calls for conflict management and scheduling of control signals. Moreover, some modules may produce actions that cannot be directly executed, but instead these abstract actions need some preprocessing to convert them to the low-level control signals expected by the robot’s actuators. For example, the dialogue engine may want some sentence to be uttered by the robot, but this must first be converted from text to a sound signal before it can be sent to the loudspeaker. This functionality is provided by the text-to-speech module, which is also assumed to produce corresponding lip sync animations.

For companion robots with a relatively simple motor system, it probably suffices to have a single module for animation control which converts abstract animation commands to low-level control signals. This can be done with the help of an animation database containing sequences of animations that can be invoked by name and then readily played out. (Philips has provided such a component with the iCat, called the ‘Animation Module’.) This animation database may also contain more elaborate animations that can be played on a user’s request, e.g., to perform a preprogrammed dance. For companion robots with a complex motor system, the animation control module may be replaced by a motion engine (which is placed among the other action selection engines), as discussed above. In this case, an animation database may still fulfill an important role as a storage of small, commonly used sequences of motor commands.

Finally, actuator control requests may occur concurrently and be in conflict with each other. It is the task of the conflict manager to provide the actuators with consistent control signals. This can be done by choosing between conflicting requests, scheduling concurrent requests, or merging them. These choices are made on a domain-dependent basis.

Output Modalities All output modalities or actuators are grouped in the lower right corner of Figure B.3. Similarly with the input modalities, these will be different for different kinds of companion robots, but a typical companion robot will probably have at least some motors (for, e.g., facial expressions and locomotion) and a loudspeaker. Other actuators may include lights, radio, control of other electronic devices, etc.
B.2.4 Interfaces Between Functional Components

In this section, we explain the meaning of the interfaces between the functional components. For cosmetic reasons, the ‘arrows’ in Figure B.3 appear to lead from one functional component to another, while they actually connect one or more specific modules inside a functional component to other modules inside another functional component. References to arrows in Figure B.3 are marked in boldface.

Raw data that is obtained by the input sensors is sent to the input preprocessing component for processing. Needless to say, data from each sensor is sent to the appropriate processing module. For example, input from the camera is sent to the vision processing and facial emotion recognition modules, while input from the microphone is sent to the sound awareness and speech recognition modules. Any module inside the low-level behaviors component is also allowed to access all raw input data if it wants to perform its own feature extraction. In addition to raw data, low-level behaviors also have access to the Processed data from the modules inside the input preprocessing component. After the processed data is synchronized (or not) by the input synchronizer, it is sent to the action selection engines, where it is placed in the events modules inside the engines. The processed data is also sent to the databases, where the relevant data extractor will process and dispatch relevant data to each of the databases. For example, context-relevant features are added to the situation model, while emotions, intentions and attention of a user that are recognized by the various input preprocessing modules are put in the appropriate agent model. Furthermore, the action selection engines can form Expectations about future events. These expectations are sent from the action selection engines back to the input synchronizer, which subsequently splits up the expectations and sends them to the appropriate input processing modules. They can then use these expectations to facilitate processing of input.

All processing modules in the input preprocessing component have access to Ontology information, which they might need to process raw data properly. For example, the vision processing module might need ontological information about a perceived object in order to classify it as a particular item. This also ensures the use of consistent data formats. The processing modules can obtain this ontological information via the query manager in the databases component, which takes care of all queries to the databases. Updates to the databases can be performed by the action selection engines. The updates are processed by the relevant data extractor, which places the data in a suitable format in the appropriate database, in the same way as the processed data from the input preprocessing component. Query results can be requested by the action selection engines from the databases. The query manager processes the query and searches the proper database(s), guaranteeing a coherent interface to all databases.

(De)activate signals can be sent from the action selection engines to the low-level behaviors component. These signals allow the action selection engines some cognitive control over the robot’s reactive behavior. For example, if needed, the face tracker can be activated or deactivated, or in some special cases the reactive emotions can be turned off. Urges arising from the low-level behaviors can be made into goals for the action selection engines. For example, if the homeostasis module detects a
low energy level, a goal to go to the nearest electricity socket can be added to the
goals of the motion engine.

The action selection engines provide their Cognitive state to the emotion syn-
thesizer. The cognitive state can be used by the appraiser to synthesize appropriate
emotions. In addition to the cognitive state, a Primitive emotional state is also sent
to the appraiser, where it can influence the intensity of cognitive-level emotions and
the robot’s mood. The current Emotional state, which is a compilation of the col-
lection of triggered emotions, is sent to the action selection engines. The emotional
heuristics inside the action selection engines then determine how the interpreter
is influenced by these emotions. The animation control module inside the output
preprocessing component also receives the emotional state of the agent, so that it
can select a suitable facial expression from the animation database representing the
current emotional state.

Output requests are sent from the interpreters inside action selection engines
to the output preprocessing component. Of course, the different kinds of output
requests are sent to different modules inside the output preprocessing. For example,
(annotated) utterances from the dialogue engine’s utterance formulator are sent to
the text-to-speech module, while any actions from the action selection engines that
involve motors are sent to the animation control module. The synchronization of all
output signals is taken care of by the conflict manager, as explained in the previous
section. Finally, Control signals are gathered and synchronized by the conflict man-
ger inside the output preprocessing component and sent to the appropriate output
modality.

B.2.5 Related Work

We do not claim that the presented architecture is perfect, and although we claim
that it is generic for companion robots, it is probably not unique. Another team
setting out to make a generic companion robot architecture will probably come up
with a different figure. However, we expect the level of complexity of alternative
architectures to resemble that of the one presented here, as it takes many components
and processes to achieve reasonably social behavior. It should be noted that the
intelligence of the system may not lie within the modules, but rather in the wiring (the
‘arrows’). The presentation of an architecture should therefore include a discussion
on the particular choice of interfaces between modules. We draw confidence in our
architecture from the fact that mappings can be found between this one and the
architectures of existing companion robots and affective virtual characters, several of
which we discuss next.

Breazeal [2002] uses competing behaviors for the robot Kismet in order to achieve
an emerging overall behavior that is sociable as a small child. Kismet has a number
of different response types with activation levels that change according to Kismet’s
interaction with a user. In Kismet’s architecture, the Behavior System and Motiva-
tion System can be mapped on our low-level behaviors; however, it lacks cognitive
reasoning (obviously this was not necessary for its application), which is provided
by our action selection engines. Other components in Kismet’s architecture pertain
to input and output processing, which map to corresponding preprocessing modules
in our architecture.

Max, the “Multimodal Assembly eXpert” developed at the University of Bielefeld [Kopp et al., 2003], can also be mapped to our architecture. For example, it uses a reasoning engine that provides feedback to the input module to focus attention to certain input signals, which is similar to our expectations. It also has a lower-level Reactive Behavior layer that produces direct output without having to enter the reasoning process, and a Mediator that performs the same task as our conflict manager (i.e. synchronizing output). However, Max only has one (BDI) reasoning engine, where we have provided for two or more action selection engines.

B.2.6 Some Final Remarks

It should be noted that the presented architecture does not have to be fully implemented in all cases. Some of the functional components can be left out, simplified, or even extended (depending on the application) or programmed empty (awaiting future work). For yet other modules, off-the-shelf or built-in software can be used. Furthermore, we do not claim that the discussed architecture should be the foundation of every companion robot; rather, this architecture has been presented here in order to make many of the issues encountered when programming a companion robot explicit, so that these issues can be appropriately investigated. This way it puts the current dissertation in a proper context. However, a full investigation of each component and interface is not feasible within a single Ph.D. project. Indeed, there is probably a dissertation in each box and in each arrow of Figure B.3.
References


René Descartes. *Passions of the Soul (Les passions de l’âme)*. 1649.


Andrew Ortony, October 2009. Personal communication.

Andrew Ortony and Gerald L. Clore, April–June 2009. Personal communication.


Acknowledgments

Honest criticism is hard to take, particularly from a relative, a friend, an acquaintance, or a stranger.
– Franklin P. Jones

Doing research on emotions is a surefire way to ensure that you always have a conversation topic at conferences and other kinds of parties. Everyone seems to be able to talk along about emotions, because everyone knows what an emotion is. At least, until you ask them to give a definition. (See the quote at the start of Chapter 1.) I never get to ask this question though, because people always ask me first to give a definition of emotion. On more than one occasion, my answer has been met with dissatisfaction, after which the other wondered aloud what is the status of any results I achieve when they are based on concepts that are so vague (at least to them…)! Fortunately, I have also met plenty of people who surprised themselves by finding out that they actually find this stuff interesting. The reactions of these people to my talks were often of the form “Gee, I had never thought of emotions in this way.” They have all experienced emotions (I hope), but few people have stood and thought about the mechanisms underlying their emotions. I too belong to this group of people; at least, until the moment that I started on this Ph.D. project.

So first I thank my promotor John-Jules Meyer and co-promotor Mehdi Dastani for piquing my interest in the fascinating topic that is emotion. What started as just a part of the project has—thanks to their stimulation and enthusiasm about my ideas—grown to become the main theme of my dissertation. I am satisfied with the result, proud of my work, and relieved to have made it this far (to name a few emotions that for me are associated with this dissertation). I will also fondly remember our meetings; always constructive, stimulating, and with plenty room for laughter! Very important to me has been their continuing support and belief in me, especially when I decided to restart my formalization from scratch with less than one year left. I also thank my third supervisor, Frank Dignum, for the many stimulating discussions and for keeping me see the bigger picture.

The colleagues that I have spent most time with are of course those with whom I
shared my office, room A119: Nieske “red pan” Vergunst, Christian “who goes there about?” Mol, and for a significant time also Michal “wèèesp” Sindlar. I already miss the absurd discussions, the Dunglish, the random news flashes, and the whiteboard with memorable quotes; thanks for the great times!

I also thank my other fellow Ph.D. students at the Intelligent Systems group, in no particular order: Susan “brownie” van den Braak, Maaike Harbers, Hado van Hasselt, Geert Jonker, Eric Kok, Nick Tinnemeier, Paolo “lateral raise” Turrini, Bob van der Vecht, Tom van der Weide, and Joost Westra. Also, the ‘old’ Ph.D. students: Huib Aldewereld, Jurriaan van Diggelen, Davide Grossi, and Birna van Riemsdijk. Special thanks go to Huib and Davide for their tips on the makeup of the dissertation and for generously sharing \LaTeX{} style files. I thank all for the good times in general, and the infamous “weekends of pleasure” in particular.

Thanks to the rest of the staff: Jan Broersen, Henry Prakken, and Gerard Vreeswijk. Special thanks go to Marco Wiering for making me realize I wanted to be a researcher in the first place, for notifying me of the availability of this Ph.D. position, and for supporting my bid to become a postdoc at IDSIA in Lugano.

Thanks to my reading committee members: Cristiano Castelfranchi, Gerald Clore, Jaap van den Herik, Wiebe van der Hoek, and Rineke Verbrugge. I thank them for the critical examination of my dissertation, their constructive comments, and, of course, their approval.

Last but definitely not least, I thank the home front. Meike has provided me with everything I could wish for, most importantly love and food. Meike, also thanks for your patience and for putting up with my occasional pre-deadline late-nighters! Thanks to my parents Jan and Marjolijn for always looking after me and for their unconditional support.
Summary

Even though emotions sometimes lead us astray, there is mounting evidence from psychology and neurology that emotions have—on the whole—a positive effect on intelligent decision making and acting. Emotions help both overtly and covertly by focusing a person’s attention to what is important and pruning unpromising directions of reasoning. Contrary to what might be expected, people who cannot experience emotions (e.g., due to extensive damage to the frontal lobes) are not ‘superrational’. Instead, the inability to experience emotions coincides with a crippling tendency to get stuck deliberating on unimportant details, as well as a devastating inability to foresee or care about the personal and social consequences of one’s actions.

In cognitive psychology, an emotion is described as a valenced reaction in response to perceiving an event, action, or object. The valence is either positive or negative and has a certain intensity which usually decreases over time, dissipating within seconds or minutes, in extreme cases lasting in the range of hours. Emotions are distinguished from other affective phenomena such as moods (which are more diffuse and longer lasting) and impulses (which are quite inflexible responses to hunger, lust, and pain). The ‘purpose’ of emotions is to function as innate and learned heuristics for selecting sensible and timely actions.

Like humans, artificial agents—such as robots and virtual characters—have to act intelligently under resource constraints. A deep understanding of how emotions facilitate this feat in humans can help us in designing more effective artificial agents. Even if one does not want artificial agents to behave ‘emotionally’, it will still be useful to make these agents have knowledge of human emotions, so that they can take these into account when interacting or cooperating with humans.

In order to incorporate emotions in artificial agents, a bridge must be built from psychological models of human emotions to computer science. A formal specification of emotions from a computer science perspective will be much easier to implement in a robot or virtual character than an informal psychological specification. This is because formal specification languages used by computer scientists are much closer to programming code—and thus easier to translate to programming code—than the natural language used in psychological descriptions of emotions.
In this dissertation a formalization of emotions is presented that can serve as a foundation for the implementation of emotions in artificial agents. This formalization proceeds in three stages. First, the conditions that trigger emotions are investigated and formally defined. For this we follow a psychological model commonly called the OCC model. Second, the concept of emotional experience is investigated and formalized in terms of triggered emotions and emotional intensity. Third, it is investigated how experienced emotions influence the selection of actions. This completes the circle, because performing actions leads to new situations and new emotions.

The formalization of emotions presented in this dissertation is not only useful for possible implementations of emotions in artificial agents; it also enables us to formally analyze properties of the psychological models on which the formalization is based. Ultimately, a precise analysis facilitated by this formalization can lead to a more accurate understanding of the workings of human emotions.

Of course, the presented formalization of emotions is not the final word on the matter. Emotion is a huge subject and a topic of ongoing research in psychology, neurology, and many other fields. For this reason, the presented formalization has been constructed with extensibility and modifiability in mind. The work in this dissertation should therefore be viewed as a starting point for further analysis of human emotions and incorporation of emotions in artificial agents.
Samenvatting

Hoewel emoties soms kwaad doen komen er steeds meer aanwijzingen vanuit de psychologie en de neurologie dat emoties—over het algemeen—een positief effect hebben op het nemen van intelligente beslissingen en het intelligent handelen. Emoties helpen zowel merkbaar als onmerkbaar door de aandacht te richten op wat belangrijk is en door weinigbelovende redeneerpaden af te kappen. In tegenstelling tot wat men misschien zou verwachten worden mensen die geen emoties kunnen ervaren (bijvoorbeeld door grote schade aan de voorhoofdskwabben) niet ‘super-rationeel’. Het onvermogen om emoties te ervaren valt juist samen met een verlamende neiging om te verzeild in onbelangrijke details, alsmede een desastreus onvermogen om de gevolgen van handelingen voor zichzelf of anderen in acht te nemen.

In de cognitieve psychologie wordt een emotie gezien als een waardeoordeel met betrekking tot een gebeurtenis, actie of object. De waarde is of positief of negatief en heeft een bepaalde intensiteit die meestal na verloop van tijd afneemt en binnen enkele seconden of minuten, of in extreme gevallen binnen uren, verdwijnt. Emoties worden onderscheiden van andere affectieve fenomenen zoals stemmingen (welke duurder zijn) en impulsen (welke vrij inflexibele reacties op honger, lust en pijn zijn). Het ‘doel’ van emoties is om te fungeren als aangeboren en aangeleerde heuristieken voor het tijdig selecteren van verstandige acties.


Om emoties bruikbaar te maken voor kunstmatige agenten moet er een brug gebouwd worden van psychologische modellen van menselijke emoties naar de informatica. Het is eenvoudiger om een formele specificatie van emoties vanuit een informaticaperspectief te implementeren in robots en virtuele personages dan een
informele psychologische specificatie. Dit komt doordat de formele specificatietalen die door informatici gebruikt worden veel dichter bij programmacode liggen—en dus gemakkelijker te vertalen zijn naar programmacode—dan de natuurlijke taal die wordt gebruikt in psychologische beschrijvingen van emoties.

In dit proefschrift wordt een formalisering van emoties gepresenteerd die als basis kan dienen voor het verwerken van emoties in kunstmatige agenten. Deze formalisering wordt in drie fasen geconstrueerd. Ten eerste worden de condities die emoties doen ontstaan onderzocht en formeel gedefinieerd. Hiervoor volgen we het zogenaamde OCC model, afkomstig uit de cognitieve psychologie. Ten tweede wordt het concept van emotionele ervaring onderzocht en geformaliseerd in termen van ontstane emoties en emotie-intensiteit. Ten derde wordt er onderzocht hoe ervaren emoties actieselectie beïnvloeden. Hiermee is de cirkel rond, want het uitvoeren van acties leidt tot nieuwe situaties en nieuwe emoties.

De in dit proefschrift gepresenteerde formalisering van emoties is niet alleen nuttig voor mogelijke implementaties van emoties in kunstmatige agenten, maar stelt ons ook in staat eigenschappen van de psychologische modellen waarop de formalisering is gebaseerd formeel te analyseren. Uiteindelijk kan een nauwgezette, door deze formalisering vergemakkelijkte analyse leiden tot een beter inzicht in de aard van menselijke emoties.

Natuurlijk is de gepresenteerde formalisering van emoties niet het laatste woord over de kwestie. Emotie is een enorm onderwerp waar actief onderzoek naar wordt gedaan in de psychologie, neurologie en vele andere gebieden. Daarom is de gepresenteerde formalisering geconstrueerd met uitbreidbaarheid en aanpasbaarheid in het achterhoofd. Het werk in dit proefschrift moet daarom worden gezien als een startpunt voor verdere analyse van de menselijke emoties en verwerking van emoties in kunstmatige agenten.
SIKS Dissertation Series

1998

Johan van den Akker, DEGAS - An Active, Temporal Database of Autonomous Objects, CWI, 1998-1

Floris Wiesman, Information Retrieval by Graphically Browsing Meta-Information, UM, 1998-2

Ans Steuten, A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective, TUD, 1998-3

Dennis Breuker, Memory versus Search in Games, UM, 1998-4

E.W. Oskamp, Computerondersteuning bij Straftoemeting, RUL, 1998-5

1999

Mark Sloof, Physiology of Quality Change Modelling: Automated modelling of Quality Change of Agricultural Products, VU, 1999-1

Rob Potharst, Classification using decision trees and neural nets, EUR, 1999-2

Don Beal, The Nature of Minimax Search, UM, 1999-3

Jacques Penders, The practical Art of Moving Physical Objects, UM, 1999-4


Niek J.E. Wijngaards, Re-design of compositional systems, VU, 1999-6

David Spelt, Verification support for object database design, UT, 1999-7

Jacques H.J. Lenting, Informed Gambling: Conceptual and Analytical Suggestion and Analysis of a Multi-Agent Mechanism for Discrete Reallocation, UM, 1999-8

2000

Frank Niessink, Perspectives on Improving Software Maintenance, VU, 2000-1

Koen Holtman, Prototyping of CMS Storage Management, TUE, 2000-2

Carolien M.T. Metselaar, Sociaal-organisatorische gevolgen van kennisverwerkingskunde: een processbenadering en actorkernperspectief, UVA, 2000-3

Geert de Haan, ETAG, A Formal Model of Competence Knowledge for User Interface Design, VU, 2000-4

Ruud van der Pol, Knowledge-based Query Formulation in Information Retrieval, UM, 2000-5

Rogier van Eijk, Programming Languages for Agent Communication, UU, 2000-6

Niels Peek, Decision-theoretic Planning of Clinical Patient Management, UU, 2000-7

Veerle Coup, Sensitivity Analysis of Decision-Theoretic Networks, EUR, 2000-8

Florian Waas, Principles of Probabilistic Query Optimization, CWI, 2000-9


Jonas Karlsson, Scalable Distributed Data Structures for Database Management, CWI, 2000-11
2001

Silja Renooij, Qualitative Approaches to Quantifying Probabilistic Networks, UU, 2001-1

Koen Hindriks, Agent Programming Languages: Programming with Mental Models, UU, 2001-2

Maarten van Someren, Learning as problem solving, UvA, 2001-3

Evgueni Smirnov, Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets, UM, 2001-4


Martijn van Welie, Task-based User Interface Design, VU, 2001-6

Bastiaan Schonhage, Diva: Architectural Perspectives on Information Visualization, VU, 2001-7

Pascal van Eck, A Compositional Semantic Structure for Multi-Agent Systems Dynamics, VU, 2001-8

Pieter Jan 't Hoen, Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes, RUL, 2001-9

Maarten Sierhuis, Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design, UvA, 2001-10

Tom M. van Engers, Knowledge Management: The Role of Mental Models in Business Systems Design, VUA, 2001-11

2002

Nico Lassing, Architecture-Level Modifiability Analysis, VU, 2002-01

Roelof van Zwol, Modelling and searching web-based document collections, UT, 2002-02

Henk Ernst Blok, Database Optimization Aspects for Information Retrieval, UT, 2002-03

Juan Roberto Castelo Valdueza, The Discrete Acyclic Digraph Markov Model in Data Mining, UU, 2002-04

Radu Serban, The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents, VU, 2002-05

Laurens Mommers, Applied legal epistemology: Building a knowledge-based ontology of the legal domain, UL, 2002-06

Peter Boncz, Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications, CWI, 2002-07

Jaap Gordijn, Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas, VU, 2002-08

Willem-Jan van den Heuvel, Integrating Modern Business Applications with Objectified Legacy Systems, KUB, 2002-09

Brian Sheppard, Towards Perfect Play of Scrabble, UM, 2002-10

Wouter C.A. Wijngaards, Agent Based Modelling of Dynamics: Biological and Organisational Applications, VU, 2002-11

Albrecht Schmidt, Processing XML in Database Systems, UVA, 2002-12

Hongjing Wu, A Reference Architecture for Adaptive Hypermedia Applications, TUE, 2002-13

Wieke de Vries, Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems, UU, 2002-14

Rik Eshuis, Semantics and Verification of UML Activity Diagrams for Workflow Modelling, UT, 2002-15


Stefan Manegold, Understanding, Modeling, and Improving Main-Memory Database Performance, UVA, 2002-17

2003

Heiner Stuckenschmidt, Ontology-Based Information Sharing In Weakly Structured Environments, VU, 2003-1

Jan Broersen, Modal Action Logics for Reasoning About Reactive Systems, VU, 2003-02

Martijn Schuemie, Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy, TUD, 2003-03

Milan Petkovic, Content-Based Video Retrieval Supported by Database Technology, UT, 2003-04

Jos Lehmann, Causation in Artificial Intelligence and Law - A modelling approach, UVA, 2003-05

Boris van Schooten, Development and specification of virtual environments, UT, 2003-06

Machiel Jansen, Formal Explorations of Knowledge Intensive Tasks, UvA, 2003-07

Yongping Ran, Repair Based Scheduling, UM, 2003-08

Rens Kortmann, The resolution of visually guided behaviour, UM, 2003-09

Andreas Lincke, Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture, UvT, 2003-10
Simon Keizer, Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks, UT, 2003-11
Roeland Ordelman, Dutch speech recognition in multimedia information retrieval, UT, 2003-12
Jeroen Donkers, Nosce Hostem - Searching with Opponent Models, UM, 2003-13
Stijn Hoppenbrouwers, Freezing Language: Conceptualisation Processes across ICT-Supported Organisations, KUN, 2003-14
Mathijs de Weerdt, Plan Merging in Multi-Agent Systems, TUD, 2003-15
Menzo Windhouwer, Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses, CWI, 2003-16
David Jansen, Extensions of Statecharts with Probability, Time, and Stochastic Timing, UT, 2003-17
Levente Kocsis, Learning Search Decisions, UM, 2003-18

2004
Virginia Dignum, A Model for Organizational Interaction: Based on Agents, Founded in Logic, UU, 2004-01
Lai Xu, Monitoring Multi-party Contracts for E-business, UvT, 2004-02
Perry Groot, A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving, VU, 2004-03
Chris van Aart, Organizational Principles for Multi-Agent Architectures, UVA, 2004-04
Viara Popova, Knowledge discovery and monotonicity, EUR, 2004-05
Elise Boltjes, Voorbeeldig onderwijs; voorbeeldge-stuurd onderwijs, een opslag naar abstract denken, vooral voor meisjes, UM, 2004-07
Joop Verbeek, Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politièele gegevensuitwisseling en digitale expertise, UM, 2004-08
Martin Caminada, For the Sake of the Argument; explorations into argument-based reasoning, VU, 2004-09
Suzanne Kabel, Knowledge-rich indexing of learning-objects, UVA, 2004-10
Michel Klein, Change Management for Distributed Ontologies, VU, 2004-11
The Duy Bui, Creating emotions and facial expressions for embodied agents, UT, 2004-12
Paul Harrenstein, Logic in Conflict. Logical Explorations in Strategic Equilibrium, UU, 2004-14
Arno Knobbe, Multi-Relational Data Mining, UU, 2004-15
Federico Divina, Hybrid Genetic Relational Search for Inductive Learning, VU, 2004-16
Mark Winands, Informed Search in Complex Games, UM, 2004-17
Vania Bessa Machado, Supporting the Construction of Qualitative Knowledge Models, UvA, 2004-18
Thijs Westerveld, Using generative probabilistic models for multimedia retrieval, UT, 2004-19
Madelon Evers, Learning from Design: facilitating multidisciplinary design teams, Nyenrode, 2004-20

2005
Floor Verdenius, Methodological Aspects of Designing Induction-Based Applications, UVA, 2005-01
Erik van der Werf, AI techniques for the game of Go, UM, 2005-02
Franc Groothjen, A Pragmatic Approach to the Conceptualisation of Language, RUN, 2005-03
Nirvana Meratnia, Towards Database Support for Moving Object data, UT, 2005-04
Gabriel Infante-Lopez, Two-Level Probabilistic Grammars for Natural Language Parsing, UVA, 2005-05
Pieter Spronck, Adaptive Game AI, UM, 2005-06
Flavius Frasincar, Hypermedia Presentation Generation for Semantic Web Information Systems, TUE, 2005-07
Richard Vdovjak, A Model-driven Approach for Building Distributed Ontology-based Web Applications, TUE, 2005-08
Jeen Broekstra, Storage, Querying and Inferencing for Semantic Web Languages, VU, 2005-09
Anders Bouwer, Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments, UVA, 2005-10
Elth Ogston, Agent Based Matchmaking and Clustering - A Decentralized Approach to Search, VU, 2005-11
Csaba Boer, Distributed Simulation in Industry, EUR, 2005-12
Fred Hamburg, Een Computermodel voor het Onder-
Borys Omelayenko, Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics, VU, 2005-14

Tibor Bosse, Analysis of the Dynamics of Cognitive Processes, VU, 2005-15

Joris Graaumans, Usability of XML Query Languages, UU, 2005-16

Boris Shishkov, Software Specification Based on Re-usable Business Components, TUD, 2005-17

Danielle Sent, Test-selection strategies for probabilistic networks, UU, 2005-18

Michel van Dartel, Situated Representation, UM, 2005-19

Cristina Coteanu, Cyber Consumer Law, State of the Art and Perspectives, UL, 2005-20

Wijnand Derks, Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics, UT, 2005-21


Cristina Chisalita, Contextual issues in the design and use of information technology in organizations, VU, 2006-02

Noor Christoph, The role of metacognitive skills in learning to solve problems, UVA, 2006-03

Marta Sabou, Building Web Service Ontologies, VU, 2006-04

Cees Pierik, Validation Techniques for Object-Oriented Proof Outlines, UU, 2006-05


Marko Smiljanić, XML schema matching – balancing efficiency and effectiveness by means of clustering, UT, 2006-07

Eelco Herder, Forward, Back and Home Again – Analyzing User Behavior on the Web, UT, 2006-08

Mohamed Wahdan, Automatic Formulation of the Auditor’s Opinion, UM, 2006-09

Ronny Siebes, Semantic Routing in Peer-to-Peer Systems, VU, 2006-10

Joeri van Ruth, Flattening Queries over Nested Data Types, UT, 2006-11

Bert Bongers, Interactivation – Towards an e-cology of people, our technological environment, and the arts, VU, 2006-12

Henk-Jan Lebbink, Dialogue and Decision Games for Information Exchanging Agents, UU, 2006-13

Johan Hoorn, Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change, VU, 2006-14

Rainer Malik, CONAN: Text Mining in the Biomedical Domain, UU, 2006-15

Carsten Riggelsen, Approximation Methods for Efficient Learning of Bayesian Networks, UU, 2006-16

Stacey Nagata, User Assistance for Multitasking with Interruptions on a Mobile Device, UU, 2006-17

Valentin Zhizhkon, Graph transformation for Natural Language Processing, UVA, 2006-18


Marina Velikova, Monotone models for prediction in data mining, UvT, 2006-20

Bas van Gils, Aptoness on the Web, RUN, 2006-21

Paul de Vrieze, Fundaments of Adaptive Personalisation, RUN, 2006-22

Ion Juvina, Development of Cognitive Model for Navigating on the Web, UU, 2006-23

Laura Hollink, Semantic Annotation for Retrieval of Visual Resources, VU, 2006-24

Madalina Drugan, Conditional log-likelihood MDL and Evolutionary MCMC, UU, 2006-25

Vojkan Mihajlovic, Score Region Algebra: A Flexible Framework for Structured Information Retrieval, UT, 2006-26

Stefano Bocconi, Vox Populi: generating video documentaries from semantically annotated media repositories, CWI, 2006-27

Borkur Sigurbjornsson, Focused Information Access using XML Element Retrieval, UVA, 2006-28

2007

Kees Leune, Access Control and Service-Oriented Architectures, UvT, 2007-01

Wouter Teepe, Reconciling Information Exchange and Confidentiality: A Formal Approach, RUG, 2007-02

Peter Mika, Social Networks and the Semantic Web, VU, 2007-03


Bart Schermer, Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for
Agent-enabled Surveillance, UL, 2007-05
Gilad Mishne, Applied Text Analytics for Blogs, UVA, 2007-06
Natasa Jovanovic, To Who It May Concern - Addressee Identification in Face-to-Face Meetings, UT, 2007-08
Mark Hoogendoorn, Modeling of Change in Multi-Agent Organizations, VU, 2007-09
David Mobach, Agent-Based Mediated Service Negotiation, VU, 2007-09
Huib Aldewereld, Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols, UU, 2007-10
Marcel van Gerven, Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty, RUN, 2007-12
Rutger Rienks, Meetings in Smart Environments: Implications of Progressing Technology, UT, 2007-13
Niek Bergboer, Context-Based Image Analysis, UM, 2007-14
Joyca Lacroix, NIM: a Situated Computational Memory Model, UM, 2007-15
Davide Grossi, Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems, UU, 2007-16
Theodore Charitos, Reasoning with Dynamic Networks in Practice, UU, 2007-17
Bart Orriens, On the development an management of adaptive business collaborations, UT, 2007-18
David Levy, Intimate relationships with artificial partners, UM, 2007-19
Slinger Jansen, Customer Configuration Updating in a Software Supply Network, UU, 2007-20
Zlatko Zlatev, Goal-oriented design of value and process models from patterns, UT, 2007-22
Peter Barna, Specification of Application Logic in Web Information Systems, TUE, 2007-23
Georgina Ramirez Camps, Structural Features in XML Retrieval, CWI, 2007-24
Katalin Boer-Sorbn, Agent-Based Simulation of Financial Markets: A modular,continuous-time approach, EUR, 2008-01
Vera Hollink, Optimizing hierarchical menus: a usage-based approach, UVA, 2008-03
Ander de Keijzer, Management of Uncertain Data - towards unattended integration, UT, 2008-04
Bela Mutschler, Modeling and simulating causal dependencies on process-aware information systems from a cost perspective, UT, 2008-05
Arjen Hommersom, On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective, RUN, 2008-06
Peter van Rosmalen, Supporting the tutor in the design and support of adaptive e-learning, OU, 2008-07
Janneke Bolt, Bayesian Networks: Aspects of Approximate Inference, UU, 2008-08
Christof van Nimwegen, The paradox of the guided user: assistance can be counter-effective, UU, 2008-09
Wauter Bosmegen, Discourse oriented summarization, UT, 2008-10
Vera Kartseva, Designing Controls for Network Organizations: A Value-Based Approach, VU, 2008-11
Jozsef Farkas, A Semiotically Oriented Cognitive Model of Knowledge Representation, RUN, 2008-12
Caterina Carraciolo, Topic Driven Access to Scientific Handbooks, UVA, 2008-13
Arthur van Bunningen, Context-Aware Querying: Better Answers with Less Effort, UT, 2008-14
Henriette van Vugt, Embodied agents from a user’s perspective, VU, 2008-16
Martin Op ‘t Land, Applying Architecture and Ontology to the Splitting and Allying of Enterprises, TUD, 2008-17
Guido de Croon, Adaptive Active Vision, UM, 2008-18
Henning Rode, From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search, UT, 2008-19
Rex Arendsen, Geen bericht, goed bericht. Een on-
derzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven, UVA, 2008-20

Krisztian Balog, People Search in the Enterprise, UVA, 2008-21

Henk Koning, Communication of IT-Architecture, UU, 2008-22

Stefan Visscher, Bayesian network models for the management of ventilator-associated pneumonia, UU, 2008-23

Zharko Alekovski, Using background knowledge in ontology matching, VU, 2008-24


Marijn Huijbregts, Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled, UT, 2008-26

Hubert Vogten, Design and Implementation Strategies for IMS Learning Design, OU, 2008-27

Ildiko Flesch, On the Use of Independence Relations in Bayesian Networks, RUN, 2008-28

Dennis Reidasma, Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans, UT, 2008-29

Wouter van Atteveldt, Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content, VU, 2008-30

Loes Braun, Pro-Active Medical Information Retrieval, UM, 2008-31


Frank Terpstra, Scientific Workflow Design; theoretical and practical issues, UVA, 2008-33

Jeroen de Knijf, Studies in Frequent Tree Mining, UU, 2008-34

Ben Torben Nielsen, Dendritic morphologies: function shapes structure, UvT, 2008-35

2009

Rasa Jurgelenaite, Symmetric Causal Independence Models, RUN, 2009-01

Willem Robert van Hage, Evaluating Ontology-Alignment Techniques, VU, 2009-02

Hans Stol, A Framework for Evidence-based Policy Making Using IT, UvT, 2009-03

Josephine Nabukunya, Improving the Quality of Organisational Policy Making using Collaboration Engineering, RUN, 2009-04

Sietse Overbeek, Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality, RUN, 2009-05

Muhammad Subianto, Understanding Classification, UU, 2009-06

Ronald Poppe, Discriminative Vision-Based Recovery and Recognition of Human Motion, UT, 2009-07

Volker Nannen, Evolutionary Agent-Based Policy Analysis in Dynamic Environments, VU, 2009-08

Benjamin Kanagwa, Design, Discovery and Construction of Service-oriented Systems, RUN, 2009-09

Jan Wielemaker, Logic programming for knowledge-intensive interactive applications, UVA, 2009-10

Alexander Boer, Legal Theory, Sources of Law & the Semantic Web, UVA, 2009-11

Peter Massuth, Perating Guidelines for Services, TUE, Humboldt-Universitaet zu Berlin, 2009-12

Steven de Jong, Fairness in Multi-Agent Systems, UM, 2009-13

Maksym Korotkiy, From ontology-enabled services to service-enabled ontologies, VU, 2009-14

ontologies work in e-science with ONTO-SOA

Rinke Hoekstra, Ontology Representation - Design Patterns and Ontologies that Make Sense, UVA, 2009-15

Fritz Reul, New Architectures in Computer Chess, UvT, 2009-16

Laurens van der Maaten, Feature Extraction from Visual Data, UvT, 2009-17

Fabian Groffen, Armada, An Evolving Database System, CWI, 2009-18

Valentin Robu, Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets, CWI, 2009-19

Bob van der Vecht, Adjustable Autonomy: Controlling Influences on Decision Making, UU, 2009-20

Stijn Vanderlooy, Ranking and Reliable Classification, UM, 2009-21

Pavel Serdyukov, Search For Expertise: Going beyond direct evidence, UT, 2009-22

Peter Hofgesang, Modelling Web Usage in a Changing Environment, VU, 2009-23

Annerieke Heuvelink, Cognitive Models for Training Simulations, VUA, 2009-24

Alex van Ballegooij, RAM: Array Database Management through Relational Mapping, CWI, 2009-25

Fernando Koch, An Agent-Based Model for the De-
development of Intelligent Mobile Services, UU, 2009-26

Christian Glahn, Contextual Support of social Engagement and Reflection on the Web, OU, 2009-27

Sander Evers, Sensor Data Management with Probabilistic Models, UT, 2009-28

Stanislav Pokraev, Model-Driven Semantic Integration of Service-Oriented Applications, UT, 2009-29

Marcin Zukowski, Balancing vectorized query execution with bandwidth-optimized storage, CWI, 2009-30

Sofiya Katrenko, A Closer Look at Learning Relations from Text, UVA, 2009-31

Rik Farenhorst and Remco de Boer, Architectural Knowledge Management: Supporting Architects and Auditors, VU, 2009-32

Khiet Truong, How Does Real Affect Affect Recognition In Speech?, UT, 2009-33

Inge van de Weerd, Advancing in Software Product Management: An Incremental Method Engineering Approach, UU, 2009-34

Wouter Koelewijn, Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling, UL, 2009-35

Marco Kalz, Placement Support for Learners in Learning Networks, OUN, 2009-36

Hendrik Drachslcr, Navigation Support for Learners in Informal Learning Networks, OUN, 2009-37

Riina Vuorikari, Tags and self-organisation: a metadata ecology for learning resources in a multilingual context, OUN, 2009-38

Christian Stahl, Service Substitution – A Behavioral Approach Based on Petri Nets, TUE, Humboldt-Universität zu Berlin, 2009-39

Stephan Raajmakers, Multinomial Language Learning: Investigations into the Geometry of Language, UvT, 2009-40

Igor Berezhnyy, Digital Analysis of Paintings, UvT, 2009-41

Toine Bogers, Recommender Systems for Social Bookmarking, UvT, 2009-42

Virginia Nunes Leal Franqueira, Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Agents, UT, 2009-43

Roberto Santana Tapia, Assessing Business-IT Alignment in Networked Organizations, UT, 2009-44

Jilles Vreeken, Making Pattern Mining Useful, UU, 2009-45

Loredana Afanasiev, Querying XML: Benchmarks and Recursion, UvA, 2009-46

2010

Matthijs van Leeuwen, Patterns that Matter, UU, 2010-01

Ingo Wassink, Work Flows in Life Science, UT, 2010-02


Olga Kulyk, Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments, UT, 2010-04

Claudia Hauff, Predicting the Effectiveness of Queries and Retrieval Systems, UT, 2010-05

Sander Bakkes, Rapid Adaptation of Video Game AI, UvT, 2010-06

Wim Fikkert, A Gesture interaction at a Distance, UT, 2010-07

Krzysztof Siewicz, Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments, UL., 2010-08

Hugo Kielman, A Politiegegevensverwerking en Privacy, Naar een effectieve waarborging, UL, 2010-09

Rebecca Ong, Mobile Communication and Protection of Children, UL, 2010-10

Adriaan Ter Mors, The world according to MARP: Multi-Agent Route Planning, TUD, 2010-11

Susan van den Braak, Sensemaking software for crime analysis, UU, 2010-12

Gianluigi Folino, High Performance Data Mining using Bio-inspired techniques, RUN, 2010-13

Sander van Splunter, Automated Web Service Reconfiguration, VU, 2010-14

Lianne Bodenstaff, Managing Dependency Relations in Inter-Organizational Models, UT, 2010-15

Sicco Verwer, Efficient Identification of Timed Automata, theory and practice, TUD, 2010-16

Spyros Kotoulas, Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications, VU, 2010-17

Charlotte Gerritsen, Caught in the Act: Investigating Crime by Agent-Based Simulation, VU, 2010-18

Henriette Cramer, People’s Responses to Autonomous and Adaptive Systems, UvA, 2010-19

Ivo Swartjes, Whose Story Is It Anyway? How Improv Informs Agency and Authorship in Emergent Narrative, UT, 2010-20

Harold van Heerde, Privacy-aware data management
by means of data degradation, UT, 2010-21

Michiel Hildebrand, *End-user Support for Access to Heterogeneous Linked Data*, CWI, 2010-22

Bas Steunebrink, *The Logical Structure of Emotions*, UU, 2010-23