Logical Foundations of Agent-Based Computing

Wiebe van der Hoek[1,2]

[1] Institute of Information and Computing Sciences,
Department of Philosophy
Utrecht University, PO box 80.089 3508 TB Utrecht
The Netherlands
wiebe@cs.uu.nl
http://www.cs.uu.nl/~wiebe
[2] Department of Computer Science
The University of Liverpool
United Kingdom

Abstract. Logics for agents are useful when specifying, implementing and verifying agent programs. We show that modal logic provides a nice tool to define informational, motivational and dynamic aspects of agents. We conclude by showing how an agent programming language can also benefit from this modal approach.

1 Introduction

There is an increasing interest in applying logical tools to formarlize and analyze agent systems. These tools play a role in at least three levels of software engineering: at the level of *specification*, of *programming* and the level of *verification*. One of the benefits of a logical approach to agents, is that it can unify these three approaches in some way.

Specification As computer-programs and architectures for intelligent systems become more and more complex, in order to understand the behaviour of such systems, designers and users frequently make use of *folk-* or *common psychology*. Using terms like 'belief', 'wish' and 'anger', we are equipped to predict and explain behaviour of ourselves and others. The philosopher Dennett used the phrase *intentional system* to refer to an entity that is best understood in terms of folk-psychology notions such as beliefs, desires, and the like [4]. This was also what Hofstadter was referring to already in '81, when one of his characters, Sandy, puts the following forward in a Coffee House Conversation ([11]):

> But eventually, when you put enough feelingless calculations together in a huge coordinated organization, you'll get something that has properties on another level. You can see it – in fact you *have* to see it– not as a bunch of little calculations, but as a system of tendencies and desires and beliefs and so on. When things get complicated enough, you're forced to change your level of description. To some extend that' already happening, which is why we use words such as "want," "think," "try," and "hope," to describe chess programs and other attempts at mechanical thought.

M. Luck et al. (Eds.): ACAI 2001, LNAI 2086, pp. 50–73, 2001.

The intentional stance has been widely discussed in the literature –let us just remark here that Sandy of the Coffeeshop Conversation claims that the really interesting things in AI will only begin to happen, 'when the program *itself* adopts the intentional stance towards itself'– and it is not our intention to add to this debate; see [16] for a discussion and references. What is important here, is that the intentional stance is an abstraction tool, allowing us to specify and reason about complex systems, without having to go into their internal structure. The logics we are going to present in Section 2 are designed with this purpose.

Programming By definition ('agre' means 'actor'), agents are the producers of action. Agents perform actions in order to shape and modify the environment they inhabit. This in fact is already true for mere (intelligent) systems. In 'classical AI' for example, systems are designed according to the so-called *Sense-Plan-Act*-cycle, in which the system observes the state of the environment, then does some deliberation or planning in which a decision about the next action is taken, which is then exercised, after which the effects of it can be observed again, etc. Now, one way to conceive agents is that they enrich the notion of *state* in this cycle; apart from the state of the environment, or the real world, in the agent approach one also assumes a *mental state*.

The most appealing components of such an internal state define the agent's *informational* (like knowledge and belief) and *motivational* (desires, wishes, intentions) attitudes, but one could also think about *normative* or *social* attitudes like obligations and norms or *emotional* attitudes (anger, pain, fear).

It is assumed that agents typically have some kind of awareness or introspective capabilities, they can inspect (parts of) their own mental state. Thus, an agent may know that it does not know some fact, or believe that is has the permission to do some action, or know that it wishes to be without pain. In other words, agents not only (as in the SPA-cycle) observe the environment, but also inspect their own mental state.

Moreover, also the domain of action is no longer restricted to the real world. Agents are pretty well capable of *changing their mind* for instance: they revise their beliefs, modify their goals and can ask for permission. Authors like Shoham ([15]) put it even more bluntly: (the semantics of) an agent action, or program, is nothing else than a transition from one state to another. That some physical agents, like robots, also happen to affect the real world, is in such a conception not more than a 'side effect' of the mental state, or its dynamics.

Verification Having specified a solution for a problem, and implemented a system that should do the job, one needs to show that the implemented specification is correct. Or, more modestly, one is sometimes satisfied if the implemented system satisfies certain properties. Verification often involves temporal properties. Examples of general properties of programs are *safety* (it will never be the case that some undesirable property becomes true) and *liveness* (eventually, some requirements will be met). There are several ways to approach the problem of verification. We give an illustration of this in Section 4.

The rest of this tutorial paper is organised af follows. In Section 2 we make our case for modal logic. Zooming in on epistemic logic, we discuss a number of technical issues. In Section 3, we give an overview of three main approaches to the logic of agent-systems. In Section 4 we demonstrate a toy Agent Programming Language that has more or less separate constructs that address the agents' informational or motivational attitudes, and for which the logics of Section 3 supply at least tools for a neat *semantics* of such programming languages.

2 Intensional Logic

Intensional logic, and, in particular modal logic, have proven to be popular when modelling agents. To explain this, let us consider one characteristic property of classical logic, and that intensional logic typically wants to avoid.

Observation 1 (Extensionality) Let $[q/p]\varphi$ denote the formula φ, but with (an arbitrary number of occurrences of) the subformula p replaced by q. Then, classical logic encompasses the following property:

$$\models (p \leftrightarrow q) \rightarrow (\varphi \leftrightarrow [q/p]\varphi)$$

In words, extensionality says that, to determine the truth-value of a formula φ, it is only the truth-value of its subformulas that counts: if we replace any occurrence of a subformula p by another formula q *with the same meaning*, then this does not matter for the value of the value as a whole. Since the truth-value, or the meaning of a formula is sometimes also denoted as its extension, we can rephrase Observation 1 loosely as: the extension of a complex formula is determined by the extension of its subformulas, not by their form.

To give an example, let p denote that the summerschool is in Prague, and let q be the statement that it is in July. We then quickly recognize that p and q are both true, and thus are equivalent: $(p \leftrightarrow q)$. Furthermore, let l denote that logic is important (true) and w that the summerschool is one week (false). Then, according to extensionality, we have that $(w \rightarrow q)$ is equivalent to $(w \rightarrow p)$, and $l \vee (q \wedge w)$ is equivalent to $l \vee (p \wedge w)$. Combining complex assertions and then calculating their truth-value, is done by substituting the values (extension) for the subformulas. $\mathrm{Ext}(l \vee (q \wedge w)) == \mathrm{Ext}(l \vee (p \wedge w))$

Having established that classical logic satisfies the property of extensionality, one may wonder whether this is desirable, or whether there are constructs in natural language that do not satisfy this principle. It appears there are many. Let c denote that the summerschool is in Czech Republic. Then 'c, because p' is obviously true, whereas 'c, because of q' makes no sense. Noting that, by extensionality, $p \rightarrow c$ and $q \rightarrow c$ are equivalent, we obtain two conclusions: 'B because of A' cannot be modelled by $A \rightarrow B$, and, even stronger, 'because of' cannot be modelled in propositional logic at all (since 'because of' is not extensional, whereas classical logic is). This observation was in fact one of the motivations to develop modal logic.

But there are more example of constructs that are not extensional. For instance 'I wish the summerschool to be in Prague' is not the same as 'I wish the summerschool to be in July'. Also, knowing p is different from knowing q. Compare 'last year, q', with 'last year, p'. 'When postponing the school for a month, $\neg q$' does not necessarily mean 'when postponing the school for a month, $\neg p$'. Thus, when reasoning with motivational attitudes (wishing), informational attitudes (knowing), temporal properties (last year) or hypothetical events (when postponing), we don't have extensionality. Ergo: we cannot use classical logic to deal with them.

Modal logic is one attempt to circumvent these problems. In a nutshell, the modal language adds one or more unary operators $\Box$ to the language, where $\Box\varphi$ can be used to model 'φ is known', or 'φ is always the case', 'φ is a desire' or 'φ is a result of executing program π'. The intuition of such formulas is best explained by looking at the semantics of $\Box$. Given a situation s (for the moment, think of it as a truth-assignment to atoms), $\Box\varphi$ is true if φ is true in all situations t that are relevant for s. For instance, is p denotes 'sunny in Prague' and q 'sunny in Amsterdam', and s is the situation where I am in Prague, where it is sunny, then for me, two situations are relevant, if compatibility with my knowledge is concerned: t_1 in which $p \wedge q$ is true, and t_2 in which $(p \wedge \neg q)$ is. Since in all my alternatives p is true, I know p ($\Box p$), but we also have $\neg\Box q$ and $\neg\Box\neg q$) (see Figure 1). Note that this perfectly solves our problem of extensionality: in s, we have $(p \leftrightarrow q)$ but also $(\Box p \wedge \neg\Box q)$.

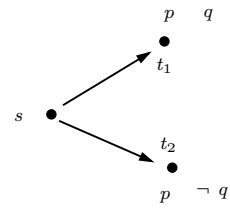

Fig. 1. A Kripke model for knowledge without extensionality

2.1 Modal Logic

All the formalisms in this overview are in fact some kind of modal logic. Mostly, they are *multi-modal*, having several modal operators for different agents, or to capture different mental attitudes. One can have modal operators $K_i\varphi$ for 'agent i knows that φ, or $B_i\varphi$, for agent i believes that φ (such epistemic operators are the focus of Section 2.2), but in $X\varphi$, the operator X can also be a goal, a wish or a desire for the agent, or even a program (see 3.3).

That is, a Kripke model M will be a tuple $M = \langle S, \pi, \bigcup_{X\in\mathcal{X}} R_X\rangle$ where S is a non-empty set of worlds or states s, π gives, for every state s the truth-value $\pi(s)(p)$ for every atom p, and R_X is a set of accessibility relations, one for each

$X \in \mathcal{X}$. In order to determine whether a formula $\varphi \in \mathcal{L}$ is true in w (if so, we write $(M, w) \models \varphi$), we look at the structure of φ:

$$\begin{array}{ll} M, s \models p & \text{iff } \pi(s)(p) = true \\ M, s \models (\varphi_1 \wedge \varphi_2) & \text{iff } M, s \models \varphi_1 \text{ and } M, s \models \varphi_2 \\ M, s \models \neg\varphi & \text{iff not} M, s \models \varphi \\ M, s \models X\varphi & \text{iff } \forall\, s(R_X st \Rightarrow M, t \models \varphi) \end{array}$$

Under such a definition, we say that X is the necessity operator for an accessibility relation R_X. The clause for $X\varphi$ is sometimes also written in a functional way: $\forall\, t \in R_X(s), M, t \models \varphi$. A formula φ is true in a model, written $M \models \varphi$, if $M, s \models \varphi$ for all $s \in S$. If $\mathcal{M}$ is a class of models, φ is said to be *valid on* $\mathcal{M}$, if for all $M \in \mathcal{M}, M \models \varphi$. This is an interesting notion in modal logic: it appears that many modal properties correspond to some restriction on Kripke models. For instance, the formula $X\varphi \rightarrow \varphi$ is valid on the class of models in which R_X is reflexive, i.e., if for all $s \in S, R_X ss$.

If modal logic is used for so many agent attitudes, it is an interesting question what the properties of any modal logic are, i.e., properties φ that are valid in every model. For such φ, we write $\models \varphi$. They are sometimes referred to as instances of the problem of *Logical Omniscience*, since, when interpreted as knowledge, they express that agents are omniscient: they are perfect reasoners.

Definition 2. Let φ, ψ be modal formulae, and let $\mathbf{X}$ be some operator.

- $\models \mathbf{X}\varphi \wedge \mathbf{X}(\varphi \rightarrow \psi) \rightarrow \mathbf{X}\psi$ $LO1$
- $\models \varphi \Rightarrow \models \mathbf{X}\varphi$ $LO2$
- $\models \varphi \rightarrow \psi \Rightarrow \models \mathbf{X}\varphi \rightarrow \mathbf{X}\psi$ $LO3$
- $\models \varphi \leftrightarrow \psi \Rightarrow \models \mathbf{X}\varphi \leftrightarrow \mathbf{X}\psi$ $LO4$
- $\models (\mathbf{X}\varphi \wedge \mathbf{X}\psi) \rightarrow \mathbf{X}(\varphi \wedge \psi)$ $LO5$
- $\models \mathbf{X}\varphi \rightarrow \mathbf{X}(\varphi \vee \psi)$ $LO6$
- $\models \neg(\mathbf{X}\varphi \wedge \mathbf{X}\neg\varphi)$ $LO7$

If X denotes knowledge, $LO1$ for example says that knowledge is closed under consequences. $LO2$ expresses that agents know all validities. If X has to model being a goal, then all the properties (maybe except $LO4$) are rather questionable. We will see later on how in some systems these properties are nevertheless accepted; in others, there are some more complicated constructs for goals. In the next section, we will see examples of modal properties that one can add on top of the properties of Definition 2.

2.2 Epistemic Logic

Epistemic logic is the logic of knowledge, which is of importance to researchers in philosophy, computer science, AI, and game theory. The material in this subsection is completely covered in the text books [5,12]. For the simplest kind of epistemic logic, it is sufficient to enrich the language of classical propositional logic by unary operators K_i, where $K_i\varphi$ stands for "agent i knows φ". Here, an

agent may be a human being, a robot, a machine, or simply a 'process'. Before looking at an application, let us note that the meaning (or truth value) of $K_i\varphi$ cannot be given by any propositional truth table in terms of the truth value of φ: knowledge is not extensional. Why are these knowledge operators useful? The derivation and correctness proofs of *communication protocols* form a nice example.

Example 1 (alternating bit protocol). There are two processes, let us say a 'Sender S' and a 'Receiver R'. The goal is for S to read a tape $X = \langle x_0, x_1, \ldots \rangle$, and to send all the inputs it has read to R over a communication channel. R in turn writes his received messages on an output tape Y. Unfortunately the channel is not trustworthy: there is no guarantee that all messages arrive. On the other hand, *some* messages will not get lost: if a message is sent repeatedly, an instance of it will arrive eventually. Now the question is whether one can write a protocol (program) that satisfies the following two constraints:

- *safety:* at any moment, Y is a prefix of X;
- *liveness:* every x_i will eventually be written on Y.

In the protocol below, $K_S(x_i)$ means that Sender knows that the i-th element of X is equal to x_i.

PROTOCOL FOR S:

$S1$ `i :=0`
$S2$ `while true do`
$S3$ `begin read` x_i;
$S4$ `send` x_i `until` $K_S K_R(x_i)$;
$S5$ `send` "$K_S K_R(x_i)$" `until` $K_S K_R K_S K_R(x_i)$
$S6$ `i := i + 1`
$S7$ `end`

PROTOCOL FOR R:

$R1$ `when` $K_R(x_0)$ `set i :=0`
$R2$ `while true do`
$R3$ `begin write` x_i;
$R4$ `send` $K_R(x_i)$ `until` $K_R K_S K_R(x_i)$;
$R5$ `send` "$K_R K_S K_R(x_i)$" `until` $K_R(x_{i+1})$
$R6$ `i := i + 1`
$R7$ `end`

An important aspect of the protocol is that Sender at line $S5$ does not continue reading X and does not yet add 1 to the counter i. We will show why this is crucial for guaranteeing safety. For, suppose that the lines $S5$ and $R5$ would be absent, and that instead line $R4$ would read as $R4'$: `send` $K_R(x_i)$ `until`

$K_R(x_{i+1})$; Suppose also, as an example, that $X = \langle a, a, b, \ldots \rangle$. Sender starts by reading x_0, an a, and sends it to R. We know that an instance of that a will arrive at a certain moment, and so by line $R3$ it will be written on Y. Receiver then acts as it should and sends an acknowledgement ($R4'$) that will also arrive eventually, thus Sender continues with $S6$ followed by $S3$: once again it reads an a and sends it to Receiver. The latter will eventually receive an instance of that a, but will not know how to interpret it: "is this a a repetition of the previous one, because Sender does not know that I know what x_0 is, or is this a the next element of the input tape, x_1"? This would clearly endanger safety.

As a final remark on the protocol, let us note that it is possible to rewrite the protocol without using any knowledge operators. The result is known as the 'alternating bit protocol'.

For negotiations and games it is not only important for participants to know what the others do know, but even more to know what the others do *not* know. Thus, your ignorance can provide me with useful information. A well-known example of this phenomenon is the *wise men* puzzle, in which one wise person can derive the color of his hat from the fact that his colleagues have said they don't know the color of their hats. A somewhat more complex variant of this phenomenon is the muddy children example, discussed below.

It is interesting to investigate notions of group knowledge for multiple agents. For example, for a group of n agents $\{1, \ldots, n\}$, one can define 'Everybody Knows' ($E\varphi$) by $E\varphi \equiv K_1\varphi \wedge \cdots \wedge K_n\varphi$. Another intriguing notion of group knowledge is 'Common Knowledge' ($C\varphi$), that should mean something like $E\varphi \wedge EE\varphi \wedge EEE\varphi \wedge \ldots$. (Unfortunately, such an infinite conjunction is not allowed in the language of epistemic logic.) Common Knowledge is a very strong notion, which therefore holds only of very weak propositions φ. In general it is very difficult to establish Common Knowledge, especially in situations like the following, where communication is not generally known to be totally reliable.

Example 2 (Byzantine generals). Imagine two allied generals, A and B, standing on two mountain summits, with their enemy in the valley between them. It is commonly known that A and B together can easily beat the enemy, but if only one of them attacks, he will certainly lose the battle.

A sends a messenger to B with the message b (= "I propose that we attack on the first day of the next month at 8 PM sharp"). It is not guaranteed, however, that the messenger will arrive. Suppose that the messenger does reach the other summit and delivers the message to B. Then $K_B b$ holds, and even $K_B K_A b$. Will it be a good idea to attack? Certainly not, because A wants to know for certain that B will attack as well, and he does not know that yet. Thus, B sends the messenger back with an 'okay' message. Suppose the messenger survives again. Then $K_A K_B K_A b$ holds. Will the generals attack now? No, because B does not know whether his 'okay' has arrived, so $K_B K_A K_B b$ does not hold, and Common Knowledge of b has not yet been established. One proves that in order to start a coordinated attack, Common Knowledge can never be established in this way using a messenger.

Example 3 (The muddy children). In this example the principal players are a father and k children, of whom m (with $m \leq k$ have mud on their forehead. The father wants to have a serious talk with the muddy children. Thus, he calls all the children together. None of them knows whether it is muddy or not, but they can all accurately perceive the other children and judge whether they are muddy. Moreover, all this is general knowledge; it is also Common Knowledge that all children are perfect logical reasoners and have even successfully finished a course on epistemic logic. Now father has a very simple announcement ψ to make:

> *At least one of you is muddy. If you know that you are muddy, please come forward.*

After this, nothing happens (except in case $m = 1$). When the father notices this, he literally repeats the announcement ψ. Once again, nothing happens (except in case $m = 2$). The announcement and subsequent silence are repeated until the father's m-th announcement. Suddenly all m muddy children step forward! It would go too far to explain the logical techniques needed to give a sound explanation, but one gets a good idea when investigating what happens in the cases $m = 1, 2$. Thus, suppose $m = 1$ and father just announced ψ, then the only muddy child knows it is muddy, because it does not see any muddy companions. It duly steps forward. Now suppose $m = 2$, and call the muddy children m_1 and m_2. Let us follow m_2's reasoning. After the first 'ψ', m_2 reasons about m_1 just like we did in the previous case: "I don't know whether I'm muddy. If not, m_1 wouldn't see any muddy companions and would step forward". At the father's second 'ψ', m_2 knows that m_1 has not in fact stepped forward, so:"m_1 must have seen another muddy child. I don't, so that must have been me". Now m_2 steps forward, and m_1 as well (by a symmetrical argument). Note, finally, that however many children are muddy, there is no Common Knowledge that there is even at least one muddy child before the father makes his first announcement! For example, in case $m = 2$, child m_1 holds it to be possible that it is not muddy and that simultaneously m_2 holds it for possible that m_2 is not muddy either. During the course, we will study the dynamics of the knowledge states of the children involved, and, for a particular case, see how we end up with an appropriate Kripke model that explains why the muddy children step forward.

Epistemics: a formal treatment For convenience, we now give a formal definition of the general logic for knowledge in a group of m agents, starting by giving its language.

Definition 3. Let P be a non-empty set of propositional variables, and $m \in \mathbb{N}$ be given. The *language* L is the smallest superset of P such that

$$\varphi, \psi \in \mathsf{L} \Rightarrow \neg\varphi, (\varphi \wedge \psi), K_i\varphi, C\varphi, D\varphi, E\varphi \in \mathsf{L} \ (i \leq m).$$

We also assume to have the usual definitions for $\vee, \leftarrow$ and $\leftrightarrow$ as logical connectives, as well as the special formula $\bot =_{\text{def}} (p \wedge \neg p)$. In the sequel, we will use $\Box$ as a variable over the operators $\mathsf{OP} = \{K_1, \ldots, K_m, C, D, E\}$. Indices i and j will range over $\{1, \ldots m\}$.

The intended meaning of $K_i\varphi$ is 'agent i knows φ', $D\varphi$ means 'φ is distributed knowledge', or 'φ is implicit knowledge of the m agents'. $E\varphi$ has to be read as 'everybody knows φ' and $C\varphi$ is 'it is common knowledge that φ'. Distributed knowledge is the knowledge that is implicitly present in a group, and which might become explicit if the agents were able to communicate . For instance, it is possible that no agent knows the assertion ψ, while at the same time $D\psi$ may be derived from $K_1\phi \wedge K_2(\phi \rightarrow \psi))$. A common example of distributed knowledge in a group is for instance the fact whether two members of that group have the same birthday. The meaning of 'everybody knows ϕ' is simply that all members of the group know that ϕ, and Common knowledge of ϕ is supposed to be $E\phi \wedge EE\phi \wedge EEE\phi \wedge \ldots$. The following definition establishes the exact properties of and relations between the notions mentioned.

Definition 4. *The logic $S5_m(CDE)$, or* **L** *for short, has the following axioms:*
$A1$ *any axiomatization for propositional logic*
$A2$ $(K_i\varphi \wedge K_i(\varphi \rightarrow \psi)) \rightarrow K_i\psi$
$A3$ $K_i\varphi \rightarrow \varphi$
$A4$ $K_i\varphi \rightarrow K_iK_i\varphi$
$A5$ $\neg K_i\varphi \rightarrow K_i\neg K_i\varphi$
$A6$ $E\varphi \leftrightarrow (K_1\varphi \wedge \cdots \wedge K_m\varphi)$

$A7$ $C\varphi \rightarrow \varphi$
$A8$ $C\varphi \rightarrow EC\varphi$
$A9$ $(C\varphi \wedge C(\varphi \rightarrow \psi)) \rightarrow C\psi$
$A10$ $C(\varphi \rightarrow E\varphi) \rightarrow (\varphi \rightarrow C\varphi)$

$A11$ $K_i\varphi \rightarrow D\varphi$
$A12$ $(D\varphi \wedge D(\varphi \rightarrow \psi)) \rightarrow D\psi$
$A13$ $D\varphi \rightarrow \varphi$
$A14$ $D\varphi \rightarrow DD\varphi$
$A15$ $\neg D\varphi \rightarrow D\neg D\varphi$

On top of that, we assume the following derivation rules:

$R1$ $\vdash \varphi, \vdash \varphi \rightarrow \psi \ \Rightarrow \vdash \psi$
$R2$ $\vdash \varphi \ \Rightarrow \vdash K_i\varphi$, *for all* $i \leq m$
$R3$ $\vdash \varphi \ \Rightarrow \vdash C\varphi$

In words, we assume a logical system $(A1, R1)$ for rational agents, (that the agents are taken to be rational, is perhaps best reflected by the fact that we have the properties $(\Box\varphi \wedge \Box(\varphi \rightarrow \psi)) \rightarrow \Box\psi$ for all $\Box \in \mathsf{OP}$ —which follows from $A2, A9, A12$ and, in the case of E, with a simple calculation using $A6$). Individual knowledge, common knowledge and distributed knowledge are all supposed to be *veridical* ($A3$, $A7$ and $A13$, respectively). The agents are assumed to be *fully introspective*: they are supposed to have *positive* ($A4$) as well as *negative* ($A5$) introspection; properties we also ascribe to distributed knowledge ($A14$ and $A15$, respectively). Both properties of introspection can be shown to hold for common knowledge as well. Axiom $A6$ can be understood as the definition of E, whereas

$A8$ says that all common knowledge is known by everybody as such. Axiom $A10$ is also known as the *induction axiom*.

What are the appropriate models for this system? Kripke models of the form $M = \langle S, \pi, R_1, \ldots, R_m, R_D, R_E, R_C \rangle$ where S and π are as before, and the accessibility relations are given suggestive names indicating to which operators they belong. To achieve the properties of Definition 4 we then require:

- For $i \leq m$, every R_i is an *equivalence* relation. The logic comprising the properties for the individual agents is often referred to as the system **S5**.
- $R_E = \bigcup_{i \leq m} R_i$. φ is known by everybody if no agent considers $\neg\varphi$ possible.
- $R_X = \bigcap_{i \leq m} R_i$. φ is distributed knowledge when true in all the states eliminated by no agent.
- R_C is the transitive reflexive closure of R_E: φ is commonly known if there is no path in the model leading to a $\neg\varphi$ world.

3 Logics for Agency

3.1 Intention Logic

In [3] Cohen and Levesque aim to specify what they call the "rational balance" between the beliefs, goals, plans, intentions, commitments and actions of autonomous agents. The main focus of their paper is on intention, for which they mention the following prerequisite: agents should act on their intentions, not in spite of them; adopt intentions that are believed to be feasible; keep intentions, but not forever; discharge intentions when believed to be satisfied; alter intentions when relevant beliefs change; and adopt intentions during plan formation.

Cohen and Levesque do not want to define intentions in terms of beliefs and desires, but, instead, introduce a third mental state to model them. Following Bratman ([2]), they ascribe to intentions the following functional roles:

(1) Intentions normally pose problems for the agent, who needs to determine ('plan') a way to achieve them;
(2) Intentions provide a "screen of admissibility" for adopting other intentions: one cannot adopt intentions that are incompatible with existing ones;
(3) Agents "track" the success of their attempts to achieve their intentions, giving rise to possibly replanning by the agent.

In their formal language, Cohen and Levesque assume a predicate logic to reason about other agents, a given domain and about time. To this, they add operators: $\mathsf{HAPPENS}\ \alpha$ means that α happens next, $\mathsf{DONE}\ \alpha$ that action α has just happened, $\mathsf{AGT}\ i\ \alpha$ that i is the agent of action α, $\mathsf{BEL}\ i\ \varphi$ that φ is believed and, finally, $\mathsf{GOAL}\ i\ \varphi$ that φ is one of agent i's goals. Actions are either atomic, or a test $\varphi?$ or composed as $\cdot;\ \cdot$ (sequential composition), $\cdot \mid \cdot$ (nondeterministic choice or $\cdot^*$ (iteration). Finally, there are also expressions that refer to time, such as $2:30\mathsf{PM}$.

Without giving the full formal semantics, it is helpful to realise that formulas are interpreted on quadruples M, σ, v, n, where M is a Kripke model with accessibility relations B for the beliefs and G for the goals, v gives the interpretation of

variables, n is an integer denoting time, and, finally, σ is a sequence of events. To get a feeling for the semantics, we explain the interpretation of several formulas in state $\langle M, \sigma, \mathsf{v}, n\rangle$. First, actions α are interpreted as $[\![\alpha]\!]$ which are sequences: $M, \sigma, \mathsf{v}, n[\![\alpha]\!](n+m)$ means that α is a sequence of events $e_1, e_2, \ldots e_m$ 'complying' with σ: $\sigma(n+i) = e_i$. Action constructs are then interpreted as expected: $M, \sigma, \mathsf{v}, n[\![\varphi?]\!]n$ if $M, \sigma, \mathsf{v}, n \models \varphi$. For sequential composition, $M, \sigma, \mathsf{v}, n[\![\alpha;\ \beta]\!]m$ holds iff there is a k such that $M, \sigma, \mathsf{v}, n[\![\alpha]\!]k$ and $M, \sigma, \mathsf{v}, k[\![\beta]\!]m$.

1. HAPPENS α is true if for some $m \geq n$, we have $M, \sigma, \mathsf{v}, n[\![\alpha]\!]m$, i.e, the meaning $[\![\alpha]\!]$ of α describes a sequence that happens after n. There is also a two-placed HAPPENS, HAPPENS $i\ \alpha$ = HAPPENS $a \wedge$ AGT $i\ \alpha$ denoting that i was the actor.
2. DONE α is true if there for some $m \leq n$, $M, \sigma, \mathsf{v}, m[\![\ \alpha\ ]\!]n$. There is also a two-placed done, DONE $i\ \alpha$ denoting that i was the actor.
3. BEL $i\ \varphi$ is true if for all σ' for which $\langle \sigma, n\rangle Bi\sigma'$ one has $M, \sigma', \mathsf{v}, n \models \varphi$.
4. GOAL $i\ \varphi$ is true if for all σ' for which $\langle \sigma, n\rangle Gi\sigma'$ one has $M, \sigma', \mathsf{v}, n \models \varphi$.

Beliefs and goals are ordinary modal operators, and hence satisfy the logical omniscience properties of Definition 2. This, of course, is most questionable for goals. Also, not that goals are *declarative*, not *procedural*: agents have a goal that something is the case, and not to do some action. By additional constraints on the model, [3] guarantee that furthermore, for belief, agents have positive and negative introspection, and always have consistent beliefs: $\models$ (BEL $i\ p$) $\rightarrow \neg$(BEL $i\ \neg p$). The connection between the informational and motivational attitude is given by a property called *realism*: $\models$ (BEL $i\ p$) $\rightarrow$ (GOAL $i\ p$). The motivation Cohen and Levesque give for this property is '... if an agent believes p now, he cannot now want it currently false; agents do not choose what they cannot change. Conversely, if p is now true in all the agent's chosen worlds, then the agent does not believe it is currently false' ([3, page 234]). It seems that this property is better modelled by imposing $\models$ BEL $i\ p \rightarrow \neg$GOAL $i\ \neg p$. Another property in this system is *expected consequences*, denoted by $\models$ (GOAL $i\ p \wedge$ (BEL $i\ (p \rightarrow q)$) $\rightarrow$ (GOAL $i\ q$). Thus, goals are closed under the believed consequences of the agent.

The system leaves room to define temporal modalities. Let *eventually* φ be defined as $\Diamond\varphi \equiv \exists\, x$(HAPPENS x; φ?). Note how such a concept can be defined by quantifying over actions: φ is eventually true if it holds after some sequence of events. *Always* φ, $\Box\varphi$ is then defined as the dual of eventually φ: $\Box\varphi = \neg\Diamond\neg\varphi$: always φ holds if there is no sequence of events leading to $\neg\varphi$. Another concept that is defined by quantification is BEFORE $p\ q = \forall\, c$(HAPPENS c; q? $\rightarrow$ ($\exists\, a(a \leq c) \wedge$ HAPPENS a; p?: p is true before q if for every event c that makes q true, there is another event, happening not later than c, that leads to p. As a last temporal modality, LATER p denotes ($\neg p \wedge \Diamond p$) These temporal operators can be used to express specific persistence properties of agents w.r.t. their goals. For instance, achievement goals are defined as

$$\mathsf{A-GOAL}\ i\ p = \mathsf{GOAL}\ i\ \mathsf{LATER}\ p \wedge \mathsf{BEL}\ i\ \neg p$$

That is, agent i believes that p is currently false, but in its desired worlds, p is eventually true. Next, agents do not try to achieve the same goal forever: $\models \Diamond\neg(\mathsf{GOAL}\ i\ \mathsf{LATER}\ P)$. Thus, agents eventually drop their achievements goals. The framework is rich enough to capture fanatic agents, having persistent goals:

$$\begin{aligned}\mathsf{P-GOAL}\ i\ p = \quad & \mathsf{GOAL}\ i\ \mathsf{LATER}\ p \wedge \mathsf{BEL}\ i\ \neg p \wedge \\ & \mathsf{BEFORE}\ (\mathsf{BEL}\ i\ p \vee \mathsf{BEL}\ i\ \Box\neg p)\ (\neg\mathsf{GOAL}\ i\ \mathsf{LATER}\ p)\end{aligned}$$

In words: p is a persistent goal if it is an achievement goal with the property that, if the agent drops it as a goal, then before that, he came either to believe that the goal is fulfilled, or that p will never become true anymore. Using the second conjunct of the definition of persistent goals, we immediately derive:

$$\mathsf{P-GOAL}\ i\ q \rightarrow \Diamond[\mathsf{BEL}\ i\ q \vee \mathsf{BEL}\ i\ \Box\neg q] \tag{1}$$

That is, persistent goals persist until they are believe to achieved, or believed to have become unachievable. Let us have a look at how intentions are defined. Here, we only look at a procedural definition:

$$\mathsf{INTEND}\ i\ a = \mathsf{P-GOAL}\ i\ [\mathsf{DONE}\ i\ (\mathsf{BEL}\ i\ (\mathsf{HAPPENS}\ a)?);\ a]$$

that is, action a is intended if it is a persistent goal of the agent that he has done a, but not by accident, no, before he does a he is aware of it: he believes, just before doing a, that a will happen.

Have we now met the desiderata of Bratman? Equation (1) guarantees that intentions cause the agent some problems; as long as the agent does not believe that the intended goal is achieved, he believes that it will be brought about. Looking at the specific structure of the action that is intended, the agent will always know which action has to be taken next. For instance, when the action is sequential, say $a;\ b$, then one easily verifies that if i intends to do $a;\ b$, then he intends to do a. Also, he intends to do $\mathsf{DONE}\ a?b$, which means that i just intends to do the second step of his plan just at the right time.

Moving to Bratman's second requirement, that intentions provide the agent with a screen of admissibility, one has:

$$\mathsf{INTEND}\ i\ b \wedge \Box(\mathsf{BEL}\ i\ [\mathsf{DONE}\ \ i\ a \rightarrow \Box\neg(\mathsf{DONE}\ \ i\ b]) \rightarrow \neg\mathsf{INTEND}\ i\ a;\ b \tag{2}$$

In words: if i intends to do b and he believes that doing a is incompatible with doing b, then the agent will not intend to do $a;\ b$. Finally, Bratman demands that agents track the success of their attempts to achieve their intentions. This is expressed by

$$\begin{aligned}&\mathsf{DONE}\ [(\mathsf{INTEND}\ i\ a) \wedge (\mathsf{BEL}\ i\ (\mathsf{HAPPENS}\ a))]?;\ e \wedge \\ &(\mathsf{BEL}\ i\ \neg(\mathsf{DONE}\ \ i\ a)) \wedge \neg(\mathsf{BEL}\ i\ \Box\neg(\mathsf{DONE}\ \ i\ a)) \rightarrow \\ &\qquad\qquad\qquad\qquad\qquad\qquad \mathsf{INTEND}\ i\ a\end{aligned} \tag{3}$$

(3) says: if the agent intends to do a and he believes that he is doing a but some other event e happens, then, if the agent believes he has not done a, but doing a is still possible, then he persists in intending to do a.

3.2 BDI-Agents

In their paper [14], Rao and Georgeff want to give a possible-worlds formalism for agent systems in which attitudes such as Beliefs, Desires and Intentions play a prominent role. A main difference with the approach of Cohen and Levesque form Section 3.1 is that, rather than taking linear sequences of events as the semantic building blocks, the basic semantic entities in [14] are temporal structures, i.e. trees, where the branching within a tree models the alternative choices that the agent has. Rao and Georgeff mention three crucial elements of their formalism:

- Intentions are treated as first-class citizens on a par with beliefs and goals. Thus, one can define different strategies of commitment with respect to the agent's intentions;
- They distinguish between the choice that the agent has over the actions he can perform, and the possibly different outcomes of an action, which are not under control of the agent;
- An interrelationship between beliefs, goals and intentions is specified that avoids many of the problems usually associated with a possible-world formalism.

To understand the formalism of this BDI-approach, one has to distinguish between a local and a global level. Locally, a world in a Kripke model is not a single point without structure anymore, but instead, it is a branching time tree. To be more precise, a *situation* is a point in a time tree, with a branching time future and a single, linear past. One such a situation is depicted in Figure 2. Given a situation, we do not reason about the history, only about the current situation and the future. In the latter, it is important to make the difference between truth on a path, and truth over paths. For instance, one can express that a certain formula φ is true at *some* point in *every* branch. By varying the quantifiers, one thus obtains 4 possible modalities referring to future.

In the syntax, we can make this more precise by distinguishing between local, *state* formulas and *path* formulas. State formulas (evaluated at a state s) are predicates, and, if ϕ, ϕ_1, ϕ_2 are state formulas, so are $\exists\,\phi$, $(\phi_1 \wedge \phi_2)$ and $\neg\phi$ and the modal formulas $\mathsf{BEL}\phi, \mathsf{GOAL}\phi$ and $\mathsf{INTEND}\phi$. Moreover, for every event e, the formulas *succeeds(e), fails(e), done(e)* etc. are also state formulas. Finally, if ψ is a path-formula, then *optional*ψ is a state formula, expressing that there is a path starting from s, in which the path formula ψ is true. A path-formula ψ (evaluated at a path p) is either a state formula ϕ or obtained from path formulas ψ, ψ_1, ψ_2 by applying the Boolean connectives, or the following constructs: $\Diamond\psi$ (in some future state on the path p, ψ holds) $\bigcirc\psi$ (ψ holds in the next state in the branch) and $\psi_1\ \mathsf{U}\ \psi_2$ (either ψ_1 remains true for ever on the path, or ψ_2 will become true somewhere and until that point, ψ_1 is true). Furthermore, *inevitable*ψ (in all paths starting in s, ψ holds) is defined as $\neg$*optional*$\neg\psi$ and $\Box\psi$ (in every future point along every path, ψ is defined as $\neg\Diamond\neg\psi$.

On a global level, such situations are accessible to each other on the basis of what the agent believes, desires or intends. Thus, on the global level, in the syntax we have operators $\mathsf{BEL}, \mathsf{GOAL}$ and INTEND which are $\Box$-operators with

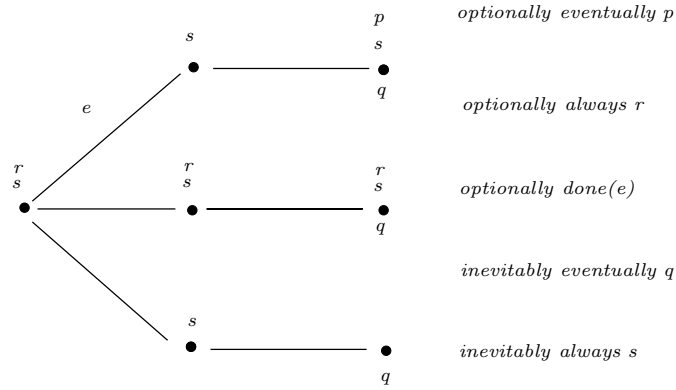

Fig. 2. Path and state formulas, events

respect to relations, or, rather functions $\mathcal{B}, \mathcal{G}$ and $\mathcal{I}$, respectively. To be more precise, let w_t be a situation at time t, with a single past and a branching future. Let X be one of our operators $\mathsf{BEL}, \mathsf{GOAL}$ and INTEND, and, given a situation w_t, $\mathcal{X}_t^w$ its semantic counterpart. Then the truth-definition reads:

$$M, v, w_t \models \mathsf{X}\varphi \text{ iff } \forall\, w' \in \mathcal{X}_t^w, M, v, w'_t \models \varphi$$

Note that agents are aware of the time: they only /believe/desire/intend worlds to be possible with the same timestamp t. Which doe not mean that they can have beliefs, desires and intentions concerning the future: φ can of course refer to a branch or a point later than t. Also note that these beliefs, desires and intentions are defined as modal operators, and thus, suffer from the logical omniscience properties mentioned in 2. By putting constraints on the semantic functions $\mathcal{X}_t^w$, it is guaranteed that beliefs satisfy the KD45 axioms, and for $\mathsf{X} = \mathsf{GOAL}$ or INTEND we have $(\mathsf{X}\varphi \rightarrow \psi) \rightarrow (\mathsf{X}\varphi \rightarrow \mathsf{X}\psi)$ and $(\neg \mathsf{X}\bot)$. As an example of the interrelation between these attitudes, [14] assumes the following scheme for $\langle \mathsf{X}, \mathsf{Y} \rangle \in \{\langle \mathsf{BEL}, \mathsf{INTEND} \rangle, \langle \mathsf{BEL}, \mathsf{GOAL} \rangle, \langle \mathsf{GOAL}, \mathsf{INTEND} \rangle\}$:

$$\mathsf{X}\varphi \rightarrow \mathsf{YX}\varphi \tag{4}$$

Saying that agents are aware of their goals and intentions and, if the agent intends to achieve φ, it has the goal to intend this. Semantically a constraint like (4) is guaranteed by imposing (5)

$$\forall\, w' \in \mathcal{Y}_t^w \,\forall\, w'' \in \mathcal{X}_t^w,\ w'' \in \mathcal{X}_t^{w'} \tag{5}$$

Concerning goal-intention compatibility, the following constraint (6) is imposed for $\bigcirc$-formulas α, that is, for formulas that have only positive references to optional formulas. It captures the idea that an agent only intends to achieve some of its goals:

$$\mathsf{INTEND}\alpha \rightarrow \mathsf{GOAL}\alpha \tag{6}$$

It is remarkable how Rao and Georgeff divert from Cohen and Levesque. Instead of the realism of the latter, Rao and Georgeff require, for $\bigcirc$-formulas α:

$$\mathsf{GOAL}\alpha \rightarrow \mathsf{BEL}\alpha \tag{7}$$

This property of *belief-goal compatibility* expresses that if an agent has a goal that α –say $\Diamond p$– holds, i.e, eventually p, then he believes that this is feasible: the agent thinks that p will indeed eventually be achieved.

Let us end this short overview of BDI agents by showing how several agents w.r.t. their commitments can be defined. First, we define a *blindly* committed agent, an agent that maintains his intentions until he actually believes that he has achieved them. Equation (8) reads as follows: if the agent intends that eventually φ holds, then the agent will maintain this intention until it believes she believes φ (has been achieved):

$$\mathsf{INTEND}(\mathit{inevitable}\Diamond\varphi) \rightarrow \mathit{inevitable}(\mathsf{INTEND}(\mathit{inevitable}\Diamond\varphi)\ \mathsf{U}\ \mathsf{BEL}\varphi) \tag{8}$$

One can relax the blind-commitment strategy to a so-called *single-minded* commitment. In (9), the agent does not hold on to his intentions for ever or until he believes he has achieved them; instead, he maintains his intentions only as long he believe they are still options:

$$\mathsf{INTEND}(\mathit{inevitable}\Diamond\varphi) \rightarrow \mathit{inevitable}(\mathsf{INTEND}(\mathit{inevitable}\Diamond\varphi)\ \mathsf{U}\ (\mathsf{BEL}\varphi \vee \neg\mathsf{BEL}\mathit{optional}\varphi)) \tag{9}$$

Finally, for an *open minded* agent, it is allowed to drop an intention φ already if he drops it as a goal, no matter he believes in φ's implementability:

$$\mathsf{INTEND}(\mathit{inevitable}\Diamond\varphi) \rightarrow \mathit{inevitable}(\mathsf{INTEND}(\mathit{inevitable}\Diamond\varphi)\ \mathsf{U}\ (\mathsf{BEL}\varphi \vee \neg\mathsf{GOAL}\mathit{optional}\varphi)) \tag{10}$$

One can then investigate various properties of the agents that we have just defined. For instance, one can prove that a blindly committed agent satisfies $\mathsf{INTEND}(\mathit{inevitable}\Diamond\varphi) \rightarrow \mathit{inevitable}(\Diamond\mathsf{BEL}\varphi)$: if such an agent intends that φ will eventually become true, it will inevitably be the case that he eventually will believe that φ holds.

3.3 KARO

We now briefly discuss the KARO framework of van der Hoek, van Linder and Meyer ([10]). This system combines the notion of Knowledge and belief with that of Abilities, Results and Opportunities to do actions. All operators are defined on Kripke models, even Goals, without obtaining the LO-properties for them. In order to successfully complete an action, both the opportunity and the ability

to perform the action are necessary. Although these notions are interconnected, they are surely not identical: the abilities of agents comprise mental and physical powers, moral capacities, and physical possibility, whereas the opportunity to perform actions is best described by the notion of circumstantial possibility.

The abilities of agents are formalised via the $\mathbf{A}_i$ operator; the formula $\mathbf{A}_i\alpha$ denotes the fact that agent i has the ability to do α. When using the descriptions of opportunities and results as given above, the framework of (propositional) dynamic logic provides an excellent means to formalise these notions. Using events $\mathrm{do}_i(\alpha)$ to refer to the performance of the action α by the agent i, we consider the formulae $\langle \mathrm{do}_i(\alpha)\rangle\varphi$ and $[\mathrm{do}_i(\alpha)]\varphi$. As we shall only encounter deterministic actions in this paper, $\langle \mathrm{do}_i(\alpha)\rangle\varphi$ is the stronger of these formulae; it represents the fact that agent i has the opportunity to do α and that doing α leads to φ. The formula $[\mathrm{do}_i(\alpha)]\varphi$ is noncommittal about the opportunity of the agent to do α but states that if the opportunity to do α is indeed present, doing α results in φ.

Definition 5. Let denumerable sets $\mathrm{A} = \{1, \ldots, n\}$ of agents, Π of propositional symbols and At of atomic actions be given. The language L is the smallest superset of Π such that:

- if $\varphi, \psi \in \mathrm{L}, i \in \mathrm{A}, \alpha \in \mathrm{Ac}$ then $\neg\varphi, \varphi \vee \psi, \mathbf{K}_i\varphi, \langle \mathrm{do}_i(\alpha)\rangle\varphi, \mathbf{A}_i\alpha \in \mathrm{L}$

where Ac is the smallest superset of At such that if $\varphi \in \mathrm{L}$, $\alpha, \alpha_1, \alpha_2 \in \mathrm{Ac}$ then

- $\varphi? \in \mathrm{Ac}$ *tests*
- $\alpha_1;\ \alpha_2 \in \mathrm{Ac}$ *sequential composition*
- $\mathtt{if}\,\varphi\,\mathtt{then}\,\alpha_1\,\mathtt{else}\,\alpha_2\,\mathtt{fi} \in \mathrm{Ac}$ *conditional composition*
- $\mathtt{while}\,\varphi\,\mathtt{do}\,\alpha\,\mathtt{od} \in \mathrm{Ac}$ *repetitive composition*

Actions and knowledge are all defined on the same states. With this, one can for instance formulate properties like

- $K_i[\mathrm{do}_i(\alpha)]\varphi \rightarrow [\mathrm{do}_i(\alpha)]K_i\varphi$ *(perfect recall)*
- $[\mathrm{do}_i(\alpha)]K_i\varphi \rightarrow K_i[\mathrm{do}_i(\alpha)]$ *(no learning)*

In the KARO framework,the mental attitudes of the agents are *dynamic*: apart form actions that change the world (i.e., the truth-value of atomic propositions), they also can affect the state of the agent. We give an example of the effects two mind-changing actions. By expanding his beliefs, the agent just adds a formula to his beliefs, but when revising them, he is careful that his beliefs remain consistent, whenever possible.

Proposition 1. *For all propositional formulas φ, ψ, ϑ we have:*

- $\models [\mathrm{do}_i(\mathtt{revise}\,\varphi)]\mathsf{B}\varphi$
- $\models [\mathrm{do}_i(\mathtt{revise}\,\varphi)]\mathsf{B}\vartheta \rightarrow [\mathrm{do}_i(\mathtt{expand}\,\varphi)]\mathsf{B}\vartheta$
- $\models \neg\mathsf{B}\neg\varphi \rightarrow ([\mathrm{do}_i(\mathtt{expand}\,\varphi)]\mathsf{B}\vartheta \leftrightarrow [\mathrm{do}_i(\mathtt{revise}\,\varphi)]\mathsf{B}\vartheta)$
- $\models \mathbf{K}_i\neg\varphi \leftrightarrow [\mathrm{do}_i(\mathtt{revise}\,\varphi)]\mathsf{B}\bot$
- $\models \mathbf{K}_i(\varphi \leftrightarrow \psi) \rightarrow ([\mathrm{do}_i(\mathtt{revise}\,\varphi)]\mathsf{B}\vartheta \leftrightarrow [\mathrm{do}_i(\mathtt{revise}\,\psi)]\mathsf{B}\vartheta)$

The first clause of Proposition 1 states that agents believe φ as the result of revising their beliefs with φ. Secondly, a revision with φ results in the agent believing at most the formulae that it would believe after expanding its beliefs with φ. Instead of rephrasing all the properties, let us here remark that KARO provides utilities to express the well-known postulates ([1]) for belief revision *within the language*.

To formalise the knowledge of agents on their practical possibilities, [10] introduces the so-called Can-predicate and Cannot-predicate.

Definition 6. *The Can-predicate and the Cannot-predicate are, for all agents i, actions α and formulae φ, defined as follows.*

- $\mathbf{PracPoss}_i(\alpha, \varphi) =^{\text{def}} \langle \text{do}_i(\alpha) \rangle \varphi \wedge \mathbf{A}_i \alpha$
- $\mathbf{Can}_i(\alpha, \varphi) =^{\text{def}} \mathbf{K}_i \mathbf{PracPoss}_i(\alpha, \varphi)$
- $\mathbf{Cannot}_i(\alpha, \varphi) =^{\text{def}} \mathbf{K}_i \neg \mathbf{PracPoss}_i(\alpha, \varphi)$

Thus the Can-predicate and the Cannot-predicate express the agent's knowledge about its practical possibilities and impossibilities, respectively. Therefore these predicates are important for the agent's planning of actions.

In KARO, an agent's goals are not primitive but induced by his wishes. Basically, an agent selects among its (implicit and passive) wishes those that it (explicitly and actively) aims to fulfil. Given the rationality of agents, these selected wishes should be both unfulfilled and implementable. In KARO, wishes ($\mathbf{W}_i \varphi$) are defined as an ordinary modal operator, and the practical possibility $\Diamond_i \varphi$ to obtain φ is defined as:

$$\text{M}, s \models \Diamond_i \varphi \Leftrightarrow \exists\, k \in \mathbb{N}\, \exists\, a_1, \ldots, a_k \in \text{At}(\text{M}, s \models \mathbf{PracPoss}_i(a_1;\ \ldots;\ a_k, \varphi))$$

A goal $\mathbf{Goal}_i \varphi$ in KARO is defined as a wish that is selected ($\mathbf{C}_i \varphi$), not fulfilled yet but implementable:

$$\mathbf{Goal}_i \varphi =^{\text{def}} \mathbf{W}_i \varphi \wedge \neg \varphi \wedge \Diamond_i \varphi \wedge \mathbf{C}_i \varphi$$

This definition of goals guarantees that none of the logical omniscience problems of Definition 2 applies. Also for the motivational attitudes, KARO provides for means to update them. That is, agents can commit and uncommit themselves to execute certain tasks. Technically, in the semantics, this is taken care of by adding the notion of an *agenda* for every agent, in every state. Rather than going into the details, we mention some obtained properties of these dynamics:

Proposition 2. *For all $i \in \text{A}$, $\alpha, \beta \in \text{Ac}$ and $\varphi \in \text{L}$ we have:*

1. $\models \mathbf{Committed}_i \alpha \rightarrow \neg \mathbf{A}_i \texttt{commit\_to}\, \beta$
2. $\models \mathbf{Committed}_i \alpha \leftrightarrow \langle \text{do}_i(\texttt{uncommit}\, \alpha) \rangle \neg \mathbf{Committed}_i \alpha$
3. $\models (\mathbf{C}_i \varphi \leftrightarrow \mathbf{K}_i \mathbf{C}_i \varphi) \rightarrow (\mathbf{A}_i \texttt{uncommit}\, \alpha \leftrightarrow \mathbf{K}_i \mathbf{A}_i \texttt{uncommit}\, \alpha)$
4. $\models \mathbf{Committed}_i \alpha \wedge \neg \mathbf{Can}_i(\alpha, \top) \rightarrow \mathbf{Can}_i(\texttt{uncommit}\, \alpha, \neg \mathbf{Committed}_i \alpha)$

In the first item it is stated that being committed prevents an agent from having the ability to (re)commit. Item 2 states that being committed is a necessary and sufficient condition for having the opportunity to uncommit. In item 4 it is stated that agents are (morally) unable to undo commitments to actions that are still known to be correct and feasible to achieve some goal. In item 5 it is formalised that agents know of their abilities to uncommit to some action. The last item states that whenever an agent is committed to an action that is no longer known to be practically possible, it knows that it can undo this impossible commitment.

The following proposition formalises some of the desiderata for the statics of commitments that are valid in KARO:

Proposition 3. *For all* $i \in \mathrm{A}$, $\alpha, \alpha_1, \alpha_2 \in \mathrm{Ac}$ *and all* $\varphi \in \mathrm{L}$ *we have:*

1. $\models \mathbf{Committed}_i\alpha \rightarrow \mathbf{K}_i\mathbf{Committed}_i\alpha$
2. $\models \mathbf{Committed}_i(\alpha_1;\ \alpha_2) \rightarrow \mathbf{Committed}_i\alpha_1 \wedge \mathbf{K}_i[\mathrm{do}_i(\alpha_1)]\mathbf{Committed}_i\alpha_2$
3. $\models \mathbf{Committed}_i\mathtt{if}\ \varphi\ \mathtt{then}\ \alpha_1\ \mathtt{else}\ \alpha_2\ \mathtt{fi} \wedge \mathbf{K}_i\varphi \rightarrow \mathbf{Committed}_i(\varphi?;\ \alpha_1)$

4 Agent Programming

We have now seen how the concept of agents is made more precise by means of *logical systems*. The exact relation of these logics with more practical approaches remains unclear, however, to this day. Several efforts to bridge the gap have been attempted. In particular, a number of *agent programming languages* have been developed for this purpose [13]. These languages show a clear family resemblance with one of the first agent programming languages Agent-0 [15,8], and also with the language ConGolog [6]. Here, we present a brief description of the language GOAL [7]; with its declarative notion of a goal, it fits well in this overview.

As in most agent programming languages, GOAL agents select actions on the basis of their current mental state. A mental state is a pair $\langle \sigma, \gamma \rangle$ where σ are the beliefs and γ are the goals of the agent. Constraint on mental states insist that an agent cannot have a goal to achieve ϕ if the agent already believes that ϕ is the case. Formally, the constraint on mental states $\langle \sigma, \gamma \rangle$ means that no $\psi \in \gamma$ can be inconsistent nor can ψ be entailed by σ ($\sigma \not\models \psi$), and σ must be consistent.

To express conditions on mental states, the language $\mathcal{L}_M$ of mental state formulas is introduced. The language $\mathcal{L}_M$ consists of boolean combinations of the basic mental state formulas $\mathsf{B}\phi$, which expresses that ϕ is believed to be the case, and $\mathsf{G}\psi$, which expresses that ψ is a goal of the agent.

Besides beliefs and goals, a third basic concept in GOAL is that of an agent *capability*. The capabilities of an agent consist of a set of so called *basic actions* which are interpreted as updates on the agent's belief base. An example of a capability is the action $\mathsf{ins}(\phi)$ which inserts ϕ in the belief base. The capabilities of an agent do not modify the agent's goals. Two special actions $\mathsf{adopt}(\phi)$ and $\mathsf{drop}(\phi)$ are introduced to respectively adopt a new goal or drop an old goal. We

use *Bcap* to denote the set of all belief update capabilities of an agent. The set of all capabilities is then defined by: $Cap = Bcap \cup \{\mathsf{adopt}(\phi), \mathsf{drop}(\phi) \mid \phi \in \mathcal{L}_0\}$.

A *GOAL agent* is a triple $\langle \Pi, \sigma_0, \gamma_0 \rangle$ that consists of the specification of an *initial mental state* $\langle \sigma_0, \gamma_0 \rangle$ and a set of actions built from the capabilities associated with the agent. Actions derived from the capabilities are *conditional actions* of the form $\varphi \rightarrow do(\mathsf{a})$, where $\mathsf{a} \in Cap$, and $\varphi \in \mathcal{L}_M$ is a mental state condition. The mental state condition specifies when the action a may be considered for execution by the agent. Note the most salient differences between GOAL agents and, for example, AgentSpeak or 3APL agents: whereas AgentSpeak and 3APL agents have planning capabilities (by means of plan rules), GOAL agents do not; whereas GOAL agents have declarative goals, neither AgentSpeak nor 3APL has such goals.

One of the key ideas in the semantics for GOAL is to incorporate into the semantics a particular *commitment strategy* (cf. Section 3). The semantics is based on a particularly simple and transparent commitment strategy, called *blind commitment*. An agent that acts according to a blind commitment drops a goal if and only if it believes that that goal has been achieved. By incorporating this commitment strategy into the semantics of GOAL, a default commitment strategy is built into agents. It is, however, only a default strategy and a programmer can overwrite this default strategy by means of the drop action. It is not possible, however, to adopt a goal ψ in case the agent believes that ψ is already achieved.

Formally, the update on the current mental state - and not just the belief base - due to an action is derived from a given partial transition function $\mathcal{T}$ of type : $Bcap \times \wp(\mathcal{L}) \rightarrow \wp(\mathcal{L})$. $\mathcal{T}$ specifies how a capability updates a belief base. The update function $\mathcal{M}$ on mental states $\langle \sigma, \gamma \rangle$ is derived from $\mathcal{T}$. Like $\mathcal{T}$, $\mathcal{M}$ is a *partial* function representing the fact that an action may not be executable or *enabled* in some mental states. The semantic function $\mathcal{M}$ maps an agent capability and a mental state to a new mental state. The capabilities of an agent thus are *mental state transformers*.

$\mathcal{M}(\mathsf{a}, \langle \sigma, \gamma \rangle) = \langle \mathcal{T}(\mathsf{a}, \sigma), \gamma \setminus \{\psi \in \gamma \mid \mathcal{T}(\mathsf{a}, \sigma) \models \psi\} \rangle$
for $\mathsf{a} \in Bcap$ if $\mathcal{T}(\mathsf{a}, \sigma)$ is defined,
$\mathcal{M}(\mathsf{a}, \langle \sigma, \gamma \rangle)$ is undefined for $\mathsf{a} \in Bcap$ if $\mathcal{T}(\mathsf{a}, \sigma)$ is undefined,
$\mathcal{M}(\mathsf{drop}(\phi), \langle \sigma, \gamma \rangle) = \langle \sigma, \gamma \setminus \{\psi \in \gamma \mid \psi \models \phi\} \rangle$,
$\mathcal{M}(\mathsf{adopt}(\phi), \langle \sigma, \gamma \rangle) = \langle \sigma, \gamma \cup \{\phi\} \rangle$ if $\sigma \not\models \phi$ and $\not\models \neg\phi$,
$\mathcal{M}(\mathsf{adopt}(\phi), \langle \sigma, \gamma \rangle)$ is undefined if $\sigma \models \phi$ or $\models \neg\phi$.

The second idea incorporated into the semantics concerns the *selection of conditional actions*. A conditional action $\varphi \rightarrow do(\mathsf{a})$ may specify conditions on the beliefs as well as conditions on the goals of an agent. As is usual, conditions on the beliefs are taken as a precondition for action execution: only if the agent's current beliefs entail the belief conditions associated with φ the agent will select a for execution. The goal condition, however, is used in a different way. To make this discussion more precise, we introduce a formal definition of a *formula ϕ that partially fulfils a goal in a mental state* $\langle \sigma, \gamma \rangle$, notation: $\phi \rightsquigarrow_\sigma \gamma$. By definition, we have $\phi \rightsquigarrow_\sigma \gamma$ iff for some $\psi \in \gamma : \psi \models \phi$ and $\sigma \not\models \phi$ Formally, for a mental state $\langle \sigma, \gamma \rangle$, the semantics of a mental state formula is defined by:

$\langle \sigma, \gamma \rangle \models \mathsf{B}\phi$ iff $\sigma \models \phi$ $\quad$ $\langle \sigma, \gamma \rangle \models \mathsf{G}\psi$ iff $\psi \rightsquigarrow_\sigma \gamma$
$\langle \sigma, \gamma \rangle \models \neg\varphi$ iff $\langle \sigma, \gamma \rangle \not\models \varphi$ $\langle \sigma, \gamma \rangle \models \varphi_1 \wedge \varphi_2$ iff $\langle \sigma, \gamma \rangle \models \varphi_1$ and $\langle \sigma, \gamma \rangle \models \varphi_2$.

Now we know what it means that a mental condition holds in a mental state, we are able to formalise the selection and execution of a conditional action by an agent. In the definition below, we assume that the action component Π of an agent $\langle \Pi, \sigma_0, \gamma_0 \rangle$ is fixed. The execution of an action gives rise to a *computation step* formally denoted by the transition relation $\xrightarrow{b}$ where b is the conditional action executed in the computation step. More than one computation step may be possible in a current state and the step relation $\longrightarrow$ thus denotes a *possible* computation step in a state. A computation step updates the current state and yields the next state of the computation. Note that because $\mathcal{M}$ is a partial function, a conditional action can only be successfully executed if both the condition is satisfied and the basic action is enabled.

Definition 7. (action selection) Let $\langle \sigma, \gamma \rangle$ be a mental state and $b = \varphi \rightarrow do(\mathsf{a}) \in \Pi$. Then, as a rule, we have that if the mental condition φ holds in $\langle \sigma, \gamma \rangle$, i.e. $\langle \sigma, \gamma \rangle \models \varphi$, and a is enabled in $\langle \sigma, \gamma \rangle$, i.e. $\mathcal{M}(\mathsf{a}, \langle \sigma, \gamma \rangle)$ is defined, then $\langle \sigma, \gamma \rangle \xrightarrow{b} \mathcal{M}(\mathsf{a}, \langle \sigma, \gamma \rangle)$ is a possible computation step. The relation $\longrightarrow$ is the smallest relation closed under this rule.
We say that a conditional action b is *enabled* in a mental state $\langle \sigma, \gamma \rangle$ in case $\mathcal{M}(\mathsf{a}, \langle \sigma, \gamma \rangle)$ is defined.

Temporal Logic for GOAL The semantics of GOAL agents is derived directly from the operational semantics as presented above. The meaning of a GOAL agent consists of a set of so-called *traces*, infinite computation sequences of consecutive mental states and actions performed in those mental states.

Definition 8. A trace s is an infinite sequence $s_0, b_0, s_1, b_1, s_2, \ldots$ where s_i are states, b_i are conditional actions, and for every i we have: $s_i \xrightarrow{b_i} s_{i+1}$, or b_i is not enabled in s_i and $s_i = s_{i+1}$.

An important assumption in the programming logic for GOAL is a *fairness* assumption. In a fair trace, there always will be a future time point at which an action is scheduled (considered for execution) and so a fair trace implements the weak fairness assumption. However, note that the fact that an action is scheduled does not mean that the action also is enabled (and therefore, the selection of the action may result in an idle step which does not change the state).

By definition, the *meaning of a GOAL agent* $\langle \Pi, \sigma_0, \gamma_0 \rangle$ is the set of *fair* traces S such that for $s \in S$ we have $s_0 = \langle \sigma_0, \gamma_0 \rangle$.

Basic Action Theories and Hoare Triples The specification of basic actions provides the basis for the programming logic and, as we will show below, is all we need to prove properties of agents. Because they play such an important role in the proof theory of GOAL, the specification of the basic agent capabilities requires special care. In the proof theory of GOAL, Hoare triples of the form

$\{\varphi\}\ b\ \{\psi\}$, where φ and ψ are *mental state formulas*, are used to specify actions. The use of Hoare triples ([9]) in a formal treatment of traditional assignments is well-understood: such triples are used to specify the *preconditions*, the *postconditions* (effects) and the *frame conditions* of actions.

Definition 9. A *Hoare triple for conditional actions* $\{\varphi\}\ b\ \{\psi\}$ means that for all traces $s \in S_A$ and time points i, we have that $(\varphi[s_i] \wedge b = b_i \in s) \Rightarrow \psi[s_{i+1}]$ where $b_i \in s$ means that action b_i is taken in state i of trace s.
A *Hoare triple for basic capabilities* $\{\varphi\}\ \mathsf{a}\ \{\psi\}$ means that for all σ, γ

- if $\langle \sigma, \gamma \rangle \models \varphi$ and a is enabled in $\langle \sigma, \gamma \rangle$, then $\mathcal{M}(\mathsf{a}, \langle \sigma, \gamma \rangle) \models \psi$, and
- if $\langle \sigma, \gamma \rangle \models \varphi$ and a is *not* enabled in $\langle \sigma, \gamma \rangle$, then $\langle \sigma, \gamma \rangle \models \psi$.

We first list some general properties of the belief and goal modalities. First, from the definition of the semantics of the goal modality, it is easy to see that $\mathsf{B}\phi \rightarrow \neg\mathsf{G}\phi$ is valid. In particular, an agent never has the goal to achieve a tautology, that is, $\neg\mathsf{G}(\mathsf{true})$ is valid. By definition of a mental state, an agent also cannot have an inconsistent goal. As a consequence, we have that $\neg\mathsf{G}(\mathsf{false})$ is an axiom.

The goal modality is a weak logical operator. In particular, it does not distribute over implication, and we do not have $(\mathsf{G}p \wedge \mathsf{G}(p \rightarrow q)) \rightarrow \mathsf{G}q$. This is due to the fact that two independent goals $\gamma = \{p, p \rightarrow q\}$ are adopted or it is due to the fact that the agent believes q. In the case that $\mathsf{B}q$, we even do not have $\mathsf{G}(p \wedge (p \rightarrow q)) \rightarrow \mathsf{G}q$ because $\mathsf{B}q \rightarrow \neg\mathsf{G}q$. From a given goal $\mathsf{G}\phi$ it is possible to conclude that the agent also has goal $\mathsf{G}\psi$ if ψ is entailed by ϕ and the agent does not already believe that ψ is the case (otherwise the axiom $\mathsf{B}\phi \rightarrow \neg\mathsf{G}\phi$ would be contradicted). Finally, we *cannot* conclude from two goals $\mathsf{G}\phi$ and $\mathsf{G}\psi$ that an agent has the conjunctive goal $\mathsf{G}(\phi \wedge \psi)$. That is, $(\mathsf{G}\phi \wedge \mathsf{G}\psi) \rightarrow \mathsf{G}(\phi \wedge \psi)$ is *not* valid. In sum, most of the usual problems that many logical operators for motivational attitudes suffer from do not apply to our G operator (cf. also Sections 2 and 3).

Conditional actions and capabilities are formalised by means of Hoare triples in our framework. A set of rules to derive Hoare triples for capabilities from *Cap* and conditional actions is listed below.

Rule for Infeasible Capabilities:
$$\frac{\varphi \rightarrow \neg \mathit{enabled}(\mathsf{a})}{\{\varphi\}\ \mathsf{a}\ \{\varphi\}}$$

Rule for Conditional Actions:
$$\frac{\{\varphi \wedge \psi\}\ \mathsf{a}\ \{\varphi'\}, (\varphi \wedge \neg\psi) \rightarrow \varphi'}{\{\varphi\}\ \psi \rightarrow \mathit{do}(\mathsf{a})\ \{\varphi'\}}$$

Consequence Rule:
$$\frac{\varphi' \rightarrow \varphi, \{\varphi\}\ \mathsf{a}\ \{\psi\}, \psi \rightarrow \psi'}{\{\varphi'\}\ \mathsf{a}\ \{\psi'\}}$$

Conjunction Rule:
$$\frac{\{\varphi_1\}\ b\ \{\psi_1\}, \{\varphi_2\}\ b\ \{\psi_2\}}{\{\varphi_1 \wedge \varphi_2\}\ b\ \{\psi_1 \wedge \psi_2\}}$$

Disjunction Rule:
$$\frac{\{\varphi_1\}\ b\ \{\psi\}, \{\varphi_2\}\ b\ \{\psi\}}{\{\varphi_1 \vee \varphi_2\}\ b\ \{\psi\}}$$

The default commitment strategy can also be captured by a Hoare triple. In case $\mathsf{a} \neq \mathsf{drop}(\psi)$, we have that $\{\mathsf{G}\phi\}\ \varphi \rightarrow do(\mathsf{a})\ \{\mathsf{B}\phi \vee \mathsf{G}\phi\}$ which expresses that after execution of an action an agent either believes it has achieved ϕ or it still has the goal ϕ in case ϕ was a goal before action execution. The next Hoare triple formalises a similar kind of statement for the absence of goals. In principle, no other action than an adopt action can add a goal to the goals of an agent. However, in case an agent believes that ϕ has been achieved before an action is executed, but after execution no longer believes this to be the case, it may adopt ϕ as a goal again. Formally, we have $\{\neg\mathsf{G}\phi\}\ b\ \{\neg\mathsf{B}\phi \vee \neg\mathsf{G}\phi\}$. Adopting goals again when it is believed they are not established anymore, provides for a mechanism similar to that of maintenance goals.

The remaining axioms and derivation rules concern the special actions drop and adopt. Neither of these actions changes anything with respect to the current beliefs of an agent. This is captured by the following four Hoare triples:

- $\{\mathsf{B}\phi\}\ \mathsf{adopt}(\psi)\ \{\mathsf{B}\phi\}$, $\{\neg\mathsf{B}\phi\}\ \mathsf{adopt}(\psi)\ \{\neg\mathsf{B}\phi\}$,
- $\{\mathsf{B}\phi\}\ \mathsf{drop}(\psi)\ \{\mathsf{B}\phi\}$, $\{\neg\mathsf{B}\phi\}\ \mathsf{drop}(\psi)\ \{\neg\mathsf{B}\phi\}$.

Concerning the changes to goals, if an agent does not believe ψ and ψ is not a contradiction, then $\mathsf{adopt}(\psi)$ results in a (new) goal $\mathsf{G}\psi$. Formally, $\{\neg\mathsf{B}\psi\}\ \mathsf{adopt}(\psi)\ \{\mathsf{G}\psi\}$. An adopt action does not have any effect on current goals of the agent. That is, if ϕ is a goal before the execution of an adopt action, it is still a goal after the execution of the adopt action: $\{\mathsf{G}\phi\}\ \mathsf{adopt}(\psi)\ \{\mathsf{G}\phi\}$. On the other hand, an $\mathsf{adopt}(\psi)$ action does not result in the adoption of a new goal ϕ in case ϕ is not entailed by ψ (similar rules can be formulated for the drop-action):

$$\frac{\not\models \psi \rightarrow \phi}{\{\neg\mathsf{G}\phi\}\ \mathsf{adopt}(\psi)\ \{\neg\mathsf{G}\phi\}}$$

Temporal logic On top of the Hoare triples for specifying basic actions, a temporal logic is used to specify and verify properties of a GOAL agent. The temporal logic language $\mathcal{L}_T$ based on $\mathcal{L}$ is defined by: (i) **init** $\in \mathcal{L}_T$, (ii) if $\phi \in \mathcal{L}$, then $\mathsf{B}\phi, \mathsf{G}\phi \in \mathcal{L}_T$, (iii) if $\varphi, \psi \in \mathcal{L}_T$, then $\neg\varphi, \varphi \wedge \psi \in \mathcal{L}_T$, (iv) if $\varphi, \psi \in \mathcal{L}_T$, then φ **until** $\psi \in \mathcal{L}_T$.

init is a proposition which states that the agent is at the beginning of execution, i.e. nothing has happened yet. The **until** operator is a weak until operator. φ **until** ψ means that ψ eventually becomes true and φ is true until ψ becomes true, or ψ never becomes true and φ remains true forever. The usual abbreviations for the propositional operators $\vee$, $\rightarrow$, and $\leftrightarrow$ are used. In case we just write false as a formula, this should be taken as an abbreviation for $\mathsf{B}(p \wedge \neg p)$ for some p. The *always* operator $\Box\varphi$ is an abbreviation for φ **until** false, and the *eventuality* operator $\Diamond\varphi$ is defined as $\neg\Box\neg\varphi$ as usual. As already was explained in the previous section, the atoms $\mathsf{B}\phi$, $\mathsf{G}\psi$ and any other state formula are evaluated with respect to mental states. The semantics of temporal formulas (φ **until** ψ is defined as in Section 3.2), relative to a trace s and time point i is defined by:

$$\boxed{\begin{array}{l} s,i \models \textbf{init} \text{ iff } i=0 \quad s,i \models \mathsf{B}\phi \text{ iff } \mathsf{B}\phi[s_i]\ s,i \models \mathsf{G}\phi \text{ iff } \mathsf{G}\phi[s_i] \\ s,i \models \neg\varphi \text{ iff } s,i \not\models \varphi,\ s,i \models \varphi \wedge \psi \text{ iff } (s,i \models \varphi \text{ and } s,i \models \psi) \end{array}}$$

For a set of traces S, we define: (i) $S \models \varphi$ iff $\forall\, s \in S, i(s,i \models \varphi)$, and (ii) $\models \varphi$ iff $S \models \varphi$ where S is the set of all traces. Temporal formulas evaluated with respect to the traces of a GOAL agent express properties of that agent. Let A be a GOAL agent and S_A be the set of traces associated with A. Then, if $S_A \models \varphi$, φ is a property of A.

In general, two important types of temporal properties are distinguished. Temporal properties are divided into *liveness* and *safety* properties. Liveness properties concern the progress that an agent makes and express that a (good) state eventually will be reached. Safety properties, on the other hand, express that some (bad) states are never entered. In the rest of this section, we discuss a number of specific liveness and safety properties of an agent $A = \langle \Pi_A, \sigma_0, \gamma_0 \rangle$ and show how these properties can be proven on the basis of the program text only. The fact that proofs can be constructed from just the program text is important because it avoids the need to reason about individual traces of a program. Reasoning from the program text is more economical since the number of traces associated with a program in general is exponential in the size of the program.

The first property we discuss concerns a safety property. Informally, the property states that if φ ever becomes true, then it remains true until ψ becomes true. Formally, this property can be written as $\varphi \rightarrow (\varphi$ **until** $\psi)$, which is abbreviated as: φ **unless** $\psi = \varphi \rightarrow (\varphi$ **until** $\psi)$. **unless** properties of an agent A can be proven by proving Hoare triples for conditional actions in Π_A only. In case we can prove that after execution of an arbitrary action either φ persists or ψ becomes true, we can conclude that φ **unless** ψ.

Theorem 1. $\forall\, b \in \Pi_A(\{\varphi \wedge \neg\psi\}\ b\ \{\varphi \vee \psi\})$ *iff* $S_A \models \varphi$ **unless** ψ

An important special case of an **unless** property is φ **unless** false, which expresses that if φ ever becomes true, it will remain true. The Hoare triples which are needed to prove φ **unless** false simplify to $\{\varphi\}\ b\ \{\varphi\}$. In case we also have **init** $\rightarrow \varphi$, where **init** denotes the initial starting point of execution, φ is always true and φ is an *invariant* of the program.

Liveness properties involve eventualities stating that some state will be reached given some condition. To express a special class of such properties, we introduce the operator φ **ensures** ψ which informally means that condition φ guarantees the realisation of ψ. The operator **ensures** is defined by: φ **ensures** $\psi = \varphi$ **unless** $\psi \wedge (\varphi \rightarrow \diamond\psi)$. ¿From this operator, below we derive a somewhat less constrained operator 'leads to'. Again, we can show that **ensures** properties can be derived by inspecting the program text only.

Theorem 2. $\forall\, b \in \Pi_A(\{\varphi \wedge \neg\psi\}\ b\ \{\varphi \vee \psi\}) \wedge \exists\, b \in \Pi_A(\{\varphi \wedge \neg\psi\}\ b\ \{\psi\}) \Rightarrow S_A \models \varphi$ **ensures** ψ

5 Conclusion

We have argued that Modal logic is a convenient tool to reason about, specify, implement and verify intelligent agents. As an illustrative case, we discussed epistemic modal logic. We have described three logical approaches to agents, all dealing with informational and motivational attitudes of agents.

We then presented a simple agent programming language that uses many of the constructs and ideas introduces earlier. Admittedly, there still is a gap between the logical approaches on the one hand, and the agent programming languages on the other. However, first steps are made to bridge this gap.

References

1. C.E. Alchourrón, P. Gärdenfors and D. Makinson, On the logic of theory change: partial meet contraction and revision functions, in *Journal of Symbolic Logic*, **50**, 1985 pp. 510–530.
2. M. Bratman, Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, MA, 1987.
3. P. Cohen and H. Levesque, Intention is Choice with Commitment, in *Artificial Intelligence*, **42** pp. 213–261 (1990).
4. D.C. Dennet, *The Intentional Stance*, MIT Press, 1987.
5. R. Fagin, J.Y. Halpern, Y. Moses and M.Y. Vardi, *Reasoning About Knowledge*, MIT Press, 1995.
6. G. De Giacomo, Y Lespérance and H. Levesque, ConGolog, a Concurrent Programming Language Based on the Situation Calculus, in *Artificial Intelligence*, accepted for publication.
7. K.V. Hindriks, F.S. de Boer, W. van der Hoek and J.-J.Ch. Meyer, *Agent Programming with Declarative Goals* To appear in the proceedings of ATAL'2000.
8. K. Hindriks, F. de Boer, W. van der Hoek and J.-J.Ch. Meyer, Agent Programming in 3APL, in *Autonomous Agents and Multi-Agent Systems*, **2**:4, pp. 357-401 1999.
9. C.A.R. Hoare, *Communicating Sequential Processes* Prentice Hall, 1985.
10. W. van der Hoek, B. van Linder and J.-J. Ch. Meyer, 'An integrated Modal Approach to Rational Agents', in M. Wooldridge and A. Rao (eds.) *Foundations of Rational Agency* Kluwer, Dordrecht, 1999, pp. 37 - 75.
11. D.R. Hofstadter, "Metamagical Themas: A coffeehouse conversation on the Turing test to determine if a machine can think", in *Scientific American*, (1981), pp. 15–36.
12. J.-J.Ch. Meyer and W. van der Hoek, *Epistemic Logic for AI and Computer Science*, Cambridge University Press, 1995.
13. A.S. Rao, AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language, in W. van der Velde and J.W. Perram (eds), *Agents Breaking Away*, 1996.
14. A.S. Rao and M.P. Georgeff. Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall (eds) *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pp. 473–484. Morgan Kaufmann, 1991.
15. Y. Shoham, Agent-oriented programming, in *Artificial Intelligence*, **60**, pp. 51-92, 1993.
16. M. Wooldridge, *Reasoning about Rational Agents*, MITP, 2000.