Semantical Considerations on Some Primitives for Agent Specification

Munindar P. Singh

Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206
USA

singh@ncsu.edu

Abstract. Intelligent agents, invented in artificial intelligence (AI), are finding application in a number of traditional areas. Classical AI notions such as knowledge and intentions can serve as natural primitives for the specification of agents. However, in order for them to live up to their promise, these notions must be given rigorous definitions. We propose formal definitions for intentions, knowledge, and know-how in a general model of actions and time. Our definitions are conceptually simple and are designed to be modular, in the sense of being orthogonal to one another. Using these definitions, we are able to prove a success result for agents that is akin to the notion of liveness in traditional computing. Others have been able to prove similar results only with the support of rather strong additional assumptions.

1 Introduction

Agents have garnered much research interest of late. Like many previous artificial intelligence (AI) ideas, as agent technology matures, it is moving into more traditional computing applications. These include old chestnuts such as information management, enterprise integration over heterogeneous databases, concurrent engineering, and so on. The rapidly advancing infrastructure for computing and communications has also opened up several new applications, for which also agents are a natural technology. These include accessing and using open information systems, agile manufacturing, electronic commerce, and so on. Lastly, certain applications that have been known for several years are picking up steam and are again natural targets for agent technology. These include robotics, space and aeronautics systems, and control systems in general.

What makes agents applicable in such varied applications is that they are natural loci of autonomous activity. While numerous definitions of agents have been proposed. We like to use the following operational definition that includes their essential properties: agents are entities with a persistent identity that are the loci of perception and behavior (actions and communications) [26]. The autonomy of agents is captured in the fact that they are the loci of actions. Agents can in addition be mobile [27], reflective [10], and lightweight threads [28]. The above is a useful definition for agents in general.

However, in many cases of interest, the agent metaphor is the most applicable when the agents are given high-level cognitive specifications. This is described as taking an

intentional stance toward agents [14] or viewing them at the *knowledge level* [17]. The high-level cognitive specifications take the form of concepts such as beliefs, knowledge, desires, and intentions. They are high-level, because they enable us to define the current state of an agent, what the agent might do, and how the agent might behave in different situations without regard to how the agent is implemented. These are perhaps the most significant of the AI contributions to agents is the notion of high-level specifications derived from cognitive notions such as beliefs and intentions.

Such high-level specifications serve as natural scientific abstractions for agents. They promise to simplify the capture of requirements on agents and of their interactions with one another. However, to be used effectively, cognitive notions must be given rigorous definitions in general models of actions and time. Our contributions must meet the standards of traditional disciplines such as distributed computing if they are to find application there.

Our research has sought to address this challenge. First, we give the simplest possible formal definitions that capture the key pretheoretic intuitions behind the various primitives. Second, our definitions are conceptually mutually "orthogonal" in that the various primitives are kept independent of each other as far as possible. This enables a greater variety of situations to be perspicuously modeled. Third, our underlying model of time and action is more general than in previous research. This not only extends the applicability of our theories, but also yields greater accuracy, since irrelevant properties of the underlying models do not affect the conceptually significant properties of our primitives. There is a well-known tension in assigning cognitive properties to physical systems. By assuming our models to be *weakly deterministic*, we can capture the notion of state just as in traditional logics of program approaches [5], and yet give a nonvacuous semantics for our primitives.

Like previous approaches, we make knowledge a primitive of our approach. But, whereas traditional approaches consider *know-that* or the knowledge of facts solely, we also formalize *know-how* or the knowledge of actions to achieve different conditions. Our third primitive is that of *intentions*, also studied before, but to a smaller extent than knowledge. Our work seeks to enhance both traditional and AI approaches.

Our enhancements to the traditional distributed computing approaches are chiefly in adding the concepts of know-how and intentions, because know-that and time are well-studied there [2, 5]. Know-How and intentions prove crucial because they enable the separation of what an agent *might* perform from what the agent *will* perform given certain intentions, knowledge, and abilities. Temporal logic has been used for several years in reasoning about distributed systems [5]. It enables us to distinguish between the conditions that are or are not attained in the computations of a distributed system. *Linear time* temporal logic considers specific computations; *branching time* temporal logic considers all possible computations. Although temporal logic has been used for traditional distributed systems, it has certain weaknesses when dealing with agents that are understood as acting autonomously. Temporal logic does not distinguish between the conditions the agent is trying and failing to achieve and the conditions that fail to hold for any other reason; similarly, it does not distinguish between conditions that hold accidentally and those that the agent intends. These distinctions are important when intelligent agents are involved, because they may be required to choose their actions

appropriately given their intentions and abilities. Indeed, the intentions adopted by different agents can be a part of the specification of correct behavior. For example, we might require our agents to intend to keep their promises even if they cannot guarantee success—this is a reasonable communication constraint in the sense of [22]. Under temporal logic, intended and unintended failures are equally unacceptable.

Our enhancements to past AI approaches, notably, [15, 4, 20] are chiefly in (a) formalizing know-how, which has typically been ignored and (b) obtaining results similar to the liveness and safety properties in a framework based on intentions and know-how. We have pursued this program of research for several years. A more leisurely description of our greater motivations is presented in [24]. The technical framework of branching time and the formalization of know-how are shared with our previous work. The present paper has a new formalization of intentions, of the constraints on models, and of the key results.

In formalizing various cognitive concepts, the inferences that are invalidated are often as important as the inferences that are validated. By carefully separating intentions from know-how and know-that, our approach prevents various spurious inferences, e.g., that intentions presuppose know-how or that intentions presuppose absence of know-how to the contrary.

Example 1. Consider a simplistic automatic teller machine (ATM). Traditionally, a formalization of its behavior involves specifying the conditions under which it produces money for a customer. One way to capture this requirement would be to state that when a customer inserts a card, the card is found valid, and the ATM has enough money, then the ATM gives out money (the desired amount).

The above kind of a constraint is naturally captured in temporal logic [8, 18]. However, it conflates the issues of whether the ATM *intends* to give the money and whether it *can* given the money. If a valid card is found, the ATM may intend to give money below some limit, but may not be able to do so, because of various reasons—e.g., it is out of money, the rollers are jammed, or whatever. If an ATM may have multiple ways of satisfying an intention, then it would be inappropriate to specify each of them—it should be left to the implementer to decide those.

High-level specifications also enable negotiation. For example, if the ATM infers the customer's intention to obtain money and is unable to fulfill the specific amount requested, it can offer to disburse a smaller amount. ∎

Example 2. Consider a simple household helper robot that can perform some typical chores. How can we specify the behavior of this robot? How can we arrange to give commands to this robot without having to program in some procedural or low-level language? The robot is naturally thought of as having beliefs and intentions, being able to perform actions, and so on. With these primitives, we can specify the expected behavior of the robot while leaving the details of how to build it open to the implementer. Further, we can tell the robot to perform certain high-level tasks. ∎

Section 2 presents our formal framework, highlighting our core temporal language and its semantics. Section 3 formalizes intentions in the above framework and shows some useful properties of this formalization. Section 4 presents a formal semantics for

know-how. Section 5 relates know-how and intentions to derive the key success result of this paper.

2 Technical Framework

The proposed formal model is based on a set of *moments* with a strict partial order, which denotes temporal precedence. Each moment is associated with a possible state of the world, which is identified by the atomic conditions or propositions that hold at that moment. A *scenario* at a moment is any maximal set of moments containing the given moment, and all moments in its future along some particular branch. Thus a scenario is a possible course of events, i.e., a specific, possible computation of the system. It is useful for capturing many of our intuitions about the choices and abilities of agents to identify one of the scenarios beginning at a moment as the *real* one. This is the scenario on which the world progresses, assuming it was in the state denoted by the given moment. Constraints on what should or will happen can naturally be formulated in terms of the real scenario.

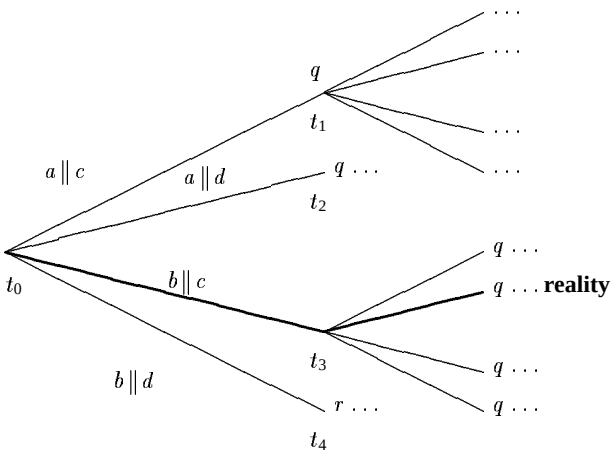

Fig. 1. An Example Formal Model

Figure 1 has a schematic picture of the formal model. Each point in the picture is a moment. Each moment is associated with a possible state of the world, which is identified by the atomic conditions or propositions that hold at that moment (atomic propositions are explained in section 2.1). With each moment are also associated the knowledge and intentions of the different agents. A condition p is said to be achieved when a state is attained in which p holds. There is a partial order on moments that denotes temporal precedence. In general, time may branch into the future—in any interesting application,

it does. Since the past is determined at each moment, the temporal precedence relation is taken to be linear in the past. The ignorance that some agent may have about the past is captured by the general mechanism of beliefs. A *scenario* at a moment is any maximal set of moments containing the given moment, and all moments in its future along some particular branch.

Example 3. Figure 1 is labeled with the actions of two agents. Each agent influences the future by acting, but the outcome also depends on other events. For example, in Figure 1, the first agent can constrain the future to some extent by choosing to do action a or action b. If he does action a, then the world progresses along one of the top two branches out of t_0; if he does action b, then it progresses along one of the bottom two branches. ∎

The important intuition about actions is that they correspond to the granularity at which an agent can make his choices. The agent cannot control what exactly transpires, but he can influence it to some extent through his actions.

Example 4. In Figure 1, the first agent can choose between t_1 and t_2, on the one hand, and between t_3 and t_4, on the other hand. However, he can choose neither between t_1 and t_2, nor between t_3 and t_4. ∎

Example 5. We formalize part of Example 2. Referring to Figure 1 again, let us interpret proposition q to mean "the room is warm" and r to mean "the room has a breeze." Let us interpret actions a, b, c, and d as "turn on high heat," "turn on medium heat," "open window," and "turn on light," respectively. If our robot turns on high heat, then the room becomes warm irrespective of the other agent's actions, whereas if he turns on medium heat, it becomes warm quickly only if the window is not opened. ∎

2.1 The Formal Language

We use a qualitative temporal language, $\mathcal{L}$, based on CTL* [5]. This captures the essential properties of actions and time that are of interest in specifying intelligent agents. Formally, $\mathcal{L}$ is the minimal set closed under the rules given below. Here $\mathcal{L}_s$ is the set of "scenario-formulas," which is used as an auxiliary definition. Φ is a set of atomic propositional symbols, $\mathcal{A}$ is a set of agent symbols, $\mathcal{B}$ is a set of basic action symbols, and $\mathcal{X}$ is a set of variables. We give intuitive meanings of the constructs of our formal language after the following syntactic definitions.

L1. $\psi \in \Phi$ implies that $\psi \in \mathcal{L}$
L2. $p, q \in \mathcal{L}$ and $x \in \mathcal{A}$ implies that $p \wedge q$, $\neg p$, $\mathsf{P}p$, $(\bigvee a : p)$, $(x\mathsf{K}_{\mathsf{t}}p) \in \mathcal{L}$
L3. $\mathcal{L} \subseteq \mathcal{L}_s$
L4. $p, q \in \mathcal{L}_s$, $x \in \mathcal{A}$, and $a \in \mathcal{B}$ implies that $p \wedge q$, $\neg p$, $p\mathsf{U}q$, $x[a]p$, $x\langle a\rangle p \in \mathcal{L}_s$
L5. $p \in \mathcal{L}_s$ implies that $\mathsf{A}p, \mathsf{R}p \in \mathcal{L}$
L6. $p \in (\mathcal{L}_s - \mathcal{L})$ and $a \in \mathcal{X}$ implies that $(\bigvee a : p) \in \mathcal{L}_s$
L7. $p \in \mathcal{L}_s$ and $x \in \mathcal{A}$ implies that $(x|p), (x\mathsf{K}_{\mathsf{h}}p) \in \mathcal{L}$

The formulas in $\mathcal{L}$ refer to moments in the model, which describe states or snapshots of the system. The formulas in $\mathcal{L}_s$ refer to scenarios in the model, i.e., to specific computations of the system. Note that $\mathcal{L} \subseteq \mathcal{L}_s$. However, our formal semantics, given in section 2.2, ensures that the formulas in $\mathcal{L}$ are given a unique meaning even if interpreted as being in $\mathcal{L}_s$.

The atomic propositions and boolean combinations of them are used to describe states of the system. They do not consider how the system may evolve or has been evolving. Two useful abbreviations are $\mathsf{false} \equiv (p \wedge \neg p)$, for any $p \in \Phi$, and $\mathsf{true} \equiv \neg\mathsf{false}$. The temporal and action formulas explicitly consider the evolution of the system—the scenario-formulas along a specific scenario and the other formulas along all or some of the possible scenarios. $p\mathsf{U}q$ is true at a moment t on a scenario, iff q holds at a future moment on the given scenario and p holds on all moments between t and the selected occurrence of q. $\mathsf{F}p$ means that p holds sometimes in the future on the given scenario and abbreviates $\mathsf{true}\mathsf{U}p$. $\mathsf{G}p$ means that p always holds in the future on the given scenario; it abbreviates $\neg\mathsf{F}\neg p$. $\mathsf{P}q$ means that q held in a past moment (we assume a linear past). The branching-time operator, A, denotes "in *all* scenarios at the present moment." Here "the present moment" refers to the moment at which a given formula is evaluated. A useful abbreviation is E, which denotes "in *some* scenario at the present moment." In other words, $\mathsf{E}p \equiv \neg\mathsf{A}\neg p$.

Example 6. In Figure 1, $\mathsf{EF}r$ and $\mathsf{AF}(q \vee r)$ hold at t_0, since r holds on some moment on some scenario at t_0 and q holds on some moment on each scenario. ∎

The *reality* operator, R, denotes "in the *real* scenario at the present moment." R is novel to our approach and helps tie together intuitions about what may and what will happen.

Example 7. In Figure 1, $\mathsf{RF}q$ holds at t_0, since q holds on some moment on the real scenario identified at t_0. ∎

$\mathcal{L}$ also contains operators on actions. These are adapted and generalized from dynamic logic [13], in which the action operators behave essentially like state-formulas. Our operators can capture the traditional operators. For an action symbol a, an agent symbol x, and a formula p, $x[a]p$ holds on a given scenario S and a moment t on it, iff, if x performs a on S starting at t, then p holds at some moment while a is being performed. The formula $x\langle a\rangle p$ holds on a given scenario S and a moment t on it, iff, x performs a on S starting at t and p holds at some moment while a is being performed. These definitions require p to hold at any moment in the (left-open and right-closed) period in which the given action is being performed. These definitions generalize naturally to variable length actions, although we restrict our attention in this paper to unitlength actions over discrete time.

Example 8. In Figure 1, $\mathsf{E}\langle b\rangle r$ and $\mathsf{A}[a]q$ hold at t_0, since r holds at the end of b on one scenario, and q holds at the end of a on each scenario. Similarly, $\mathsf{A}[d](q \vee r)$ also holds at t_0. Also, $\mathsf{A}[e]\mathsf{true}$ holds at t_0, because action e does not occur at t_0. ∎

The construct $(\bigvee a : p)$ means that there is an action under which p becomes true. The action symbol a typically occurs in p and is replaced by the specific action which makes p true.

Example 9. In Figure 1, $(\bigvee e : \mathsf{E}x\langle e\rangle\mathsf{true} \wedge \mathsf{A}x[e]q)$ holds at t_0. This means there is an action, namely, a, such that x performs it on some scenario starting at t_0 and on all scenarios on which it is performed, it results in q being true. In other words, some action is possible that always leads to q. This paradigm is used in our formalization of know-how. ∎

The formula $x\mathsf{K_t}p$ means that the agent x *knows that* p. The two other important constructs are $x\mathsf{I}p$ and $x\mathsf{K_h}p$. $x\mathsf{I}p$ is interpreted to mean that agent x intends to bring about p ('I' is a sans serif 'I'). $x\mathsf{K_h}p$ is interpreted to mean that agent x *knows how* to achieve p. The formal definition of these operators is the subject of this paper.

2.2 The Formal Model

Let $M = \langle \mathbf{T}, <, [\![\,]\!], \mathbf{R}, \mathbf{B}, \mathbf{I}\rangle$ be a formal model. $\mathbf{T}$ is the set of moments. Each moment is associated with a possible state of the system—this includes the physical state as identified by the atomic propositions that hold there, as well as the states of the agents described through their beliefs and intentions. $<$ is a partial order over $\mathbf{T}$, which is interpreted as the temporal order among the moments of $\mathbf{T}$. Therefore, $<$ must be transitive and asymmetric; it typically branches into the future; we assume it is linear in the past. $[\![\,]\!]$ gives the denotation of the various atomic propositions and of the action symbols. For an atomic proposition, p, $[\![p]\!]$ is the set of moments where p is interpreted as holding; for an action a and an agent x, $[\![a]\!]^x$ is the set of periods over which a is performed by x. These periods are notated as $[S; t, t']$ such that a begins at t and ends at t', where $t, t' \in S$.

$\mathbf{R}$ picks out at each moment the *real* scenario at that moment. This is the notion of relativized reality that we alluded to above, and which is highlighted by a bold line in Figure 1. $\mathbf{B}$ assigns to each agent at each moment the moments that the agent implicitly considers as equivalent to the given moment. This is used in the formal semantics for know-that in the traditional manner. For simplicity, we assume that $\mathbf{B}$ is an equivalence relation, resulting in $\mathsf{K_t}$ being an S5 modal logic operator [3], which grants both positive and negative introspection. For most purposes, an S4 operator would suffice, which only has positive introspection [15]. $\mathbf{I}$ assigns to each agent at each moment the scenarios that the agent prefers. This is explained further in section 3, where it is used to give a formal meaning to intentions.

For $p \in \mathcal{L}$, $M \models_t p$ expresses "M satisfies p at t." For $p \in \mathcal{L}_s$, $M \models_{S,t} p$ expresses "M satisfies p at moment t on scenario S" (we require $t \in S$). We say p is *satisfiable* iff for some M and t, $M \models_t p$. The satisfaction conditions for the temporal operators are adapted from those in [5]. For simplicity, we assume that each action symbol is quantified over at most once in any formula. Below, $p|^a_b$ is the formula resulting from the substitution of all occurrences of a in p by b. We also assume that agent symbols are mapped to unique agents throughout the model. Formally, we have:

SEM-1. $M \models_t \psi$ iff $t \in [\![\psi]\!]$, where $\psi \in \Phi$
SEM-2. $M \models_t p \wedge q$ iff $M \models_t p$ and $M \models_t q$
SEM-3. $M \models_t \neg p$ iff $M \not\models_t p$
SEM-4. $M \models_t \mathsf{A}p$ iff $(\forall S : S \in \mathbf{S}_t \Rightarrow M \models_{S,t} p)$

SEM-5. $M \models_t \mathsf{R}p$ iff $M \models_{\mathbf{R}(t),t} p$

SEM-6. $M \models_t \mathsf{P}p$ iff $(\exists t' : t' < t$ and $M \models_{t'} p)$

SEM-7. $M \models_t x\mathsf{K_t}p$ iff $(\forall t' : (t,t') \in \mathbf{B}(x,t) \Rightarrow M \models_{t'} p)$

SEM-8. $M \models_t (\bigvee a : p)$ iff $(\exists b : b \in \mathcal{B}$ and $M \models_t p|_b^a)$, where $p \in \mathcal{L}$

SEM-9. $M \models_{S,t} (\bigvee a : p)$ iff $(\exists b : b \in \mathcal{B}$ and $M \models_{S,t} p|_b^a)$, where $p \in (\mathcal{L}_s - \mathcal{L})$

SEM-10. $M \models_{S,t} p\mathsf{U}q$ iff $(\exists t' : t \leq t'$ and $M \models_{S,t'} q$ and $(\forall t'' : t \leq t'' \leq t' \Rightarrow M \models_{S,t''} p))$

SEM-11. $M \models_{S,t} x[a]p$ iff $(\forall t' \in S : [S;t,t'] \in [\![a]\!]^x$ implies that $(\exists t'' : t < t'' \leq t'$ and $M \models_{S,t''} p))$

SEM-12. $M \models_{S,t} x\langle a\rangle p$ iff $(\exists t' \in S : [S;t,t'] \in [\![a]\!]^x$ and $(\exists t'' : t < t'' \leq t'$ and $M \models_{S,t''} p))$

SEM-13. $M \models_{S,t} p \wedge q$ iff $M \models_{S,t} p$ and $M \models_{S,t} q$

SEM-14. $M \models_{S,t} \neg p$ iff $M \not\models_{S,t} p$

SEM-15. $M \models_{S,t} p$ iff $M \models_t p$, where $p \in \mathcal{L}$

The above semantic definitions may be viewed as fairly standard or at least noncontroversial. The main contribution of this paper is in operations such as intentions (|) and know-how ($x\mathsf{K_h}$). These are discussed at some length in the sections below, wherein their formal definitions are developed and shown to have certain desirable technical properties.

Various coherence constraints may be stated on the above models. One especially useful constraint is *weak determinism*, which roughly means that the range of possible futures at a moment depends on the atomic propositions that hold there. The knowledge and intentions of agents could, of course, be different at moments that have the same range of possible futures. This constraint can be understood as stating that the physical state of a system is given by the atomic propositions and that the physical state determines all that might happen. What actually happens depends on the agents' intentions. Thus there is a close dependence between $\mathbf{R}$ on the one hand, and the agents' intentions, know-how, and actions on the other. This is highlighted in section 5. Suffice it to state here that we achieve a clean separation of all that is possible and what is possible given the agents' intentions and how they act on them.

3 Intentions

What does it mean to intend? This question has been studied for centuries in philosophy and psychology, and recently in AI. Some of the literature is reviewed in [24, chap. 3]. However, existing approaches do not validate some of the properties that are crucial to the kinds of reasoning that we must perform about agents in general. One, they preclude a useful characterization of liveness, because they explicitly assume that agents can succeed in achieving some intended condition by fiat. They assume that all intentions are always dropped—a reasonable constraint on how agents modify their intentions. But they also add a constraint that agents will succeed before their intentions are dropped. This assumption is invalid, because it makes no reference to whether the given agent has the know-how required to succeed and whether the agent acts on his intentions. Consequently, success is achieved trivially for any intention—this is clearly unintuitive

and reduces the concept of intentions to something quite meaningless for real-life agents. Details of this argument were presented in [21].

Further, traditional theories do not provide any means to capture the distinction between what an agent will do given his intentions and what he might have done. The formal models are unconstrained, so that certain inferences that are clearly valid are not captured by these theories. We will show how these problems can be avoided here.

At each moment in the model, the model component $\mathbf{I}$ assigns to each agent a set of scenarios that the agent is interpreted as having selected or preferred. Roughly, our definition of intentions is that *intentions are the conditions that inevitably hold on each of the selected scenarios.* Here we consider achievement intentions in that these intentions are about achieving various conditions. However, intentions can be defined for the maintenance of conditions as well. Whereas achievement intentions are useful for liveness reasoning, maintenance intentions are useful for safety reasoning. For reasons of space, we will not discuss the latter in this paper.

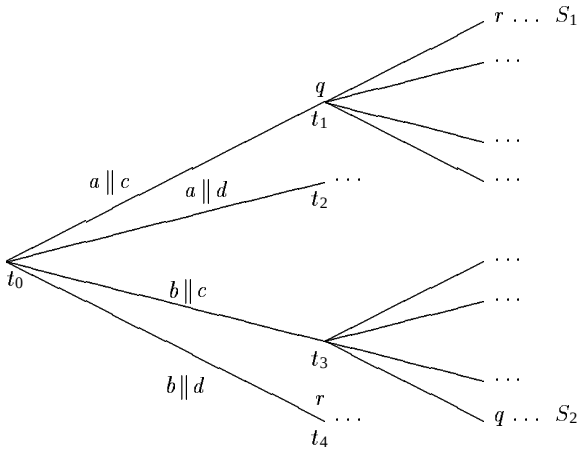

Fig. 2. Intentions

Example 10. Consider Figure 2. Assume that $\neg r$ and $\neg q$ hold everywhere other than as shown. Let the agent x (whose actions are written first in the figure) at moment t_0 prefer the scenarios S_1 and S_2. Then, by the informal definition given above, we have that x intends q (because it occurs eventually on both the preferred scenarios) and does not intend r (because it never occurs on S_2). Thus, to follow up on Example 5, our robot intends to warm the room, but not to create a breeze. ▮

We now turn to the fairly simple formal definition of achievement intentions:

SEM-16. $M \models_t x|p$ iff $(\forall S : S \in \mathbf{I}(x,t) \Rightarrow M \models_{S,t} \mathsf{F}p)$

The above definition validates several useful properties of intentions. We discuss some of these next. Some of these were obtained with an entirely different formal structure in [24]—the present development uses fewer conceptual primitives but ignores certain abstraction issues.

IC1. **Satisfiability:**
$x|p \rightarrow \mathsf{E}\mathsf{F}p$
This says that if p is intended by some agent, then it occurs eventually on some scenario. That is, the given intention is satisfiable on some future. This does not hold in general, since the sets of scenarios assigned by $\mathbf{I}$ may be empty. Formally, we must additionally constrain our models as follows:
$\mathbf{I}(x,t) \neq \emptyset$

IC2. **Temporal Consistency:**
$(x|p \wedge x|q) \rightarrow x|(\mathsf{F}p \wedge \mathsf{F}q)$
This says that if an agent intends p and intends q, then he (implicitly) intends achieving them in some undetermined temporal order: p before q, q before p, or both simultaneously. This holds because the function $\mathbf{I}$ assigns exactly one set of scenarios to each agent at each moment. Thus if both p and q, which are scenario-formulas, occur on all selected scenarios, then they occur in some temporal order on each of those scenarios. The formula $(\mathsf{F}p \wedge \mathsf{F}q)$ is true at a moment on a scenario precisely when p and q are true at (possibly distinct) future moments on the given scenario.

IC3. **Persistence does not entail success:**
$\mathsf{E}\mathsf{G}((x|p) \wedge \neg p)$ is satisfiable
This is quite obvious intuitively: just because an agent persists with an intention does not mean that he will succeed. Technically, two main ingredients are missing. The agent must know-how to achieve the intended condition and must act on his intentions. We include this here to point out that in the theory of [4], persistence is sufficient for success (p. 233). This is a major conceptual weakness, since it violates the usual understanding that intentions do not entail know-how [21]. The need to state the conditions under which an agent can succeed with his intentions is one of the motivations for the concept of know-how.

IC4. **Persist while succeeding:**
This constraint is a possible restriction on the architectures of agents. It requires that agents desist from revising their intentions as long as they are able to proceed properly. As stated, this is stronger than necessary, but we avoid getting into more appropriate weaker versions for reasons of space. The formal constraint below states that if an agent selects some scenarios, then at future moments on those scenarios, he selects from among the future components of those scenarios:
$(S \in \mathbf{I}(x,t)$ and $[S;t,t'] \in [\![a]\!]^x) \Rightarrow (\forall S' \in \mathbf{I}(x,t') \Rightarrow (\exists S'' \in \mathbf{I}(x,t)$ and $S' \subseteq S''))$

Other important constraints on intentions validated by our approach include (a) the absence of closure of intentions under beliefs, (b) the consistency of intentions with beliefs about reality, and (c) the non-entailment of beliefs about reality. Of these, (a) and

(b) are jointly termed the *asymmetry thesis* by Bratman [1, p. 38]. He argues that they are among the more basic constraints on the intentions and beliefs of rational agents.

Intentions have an obvious connection with actions—agents act to satisfy their intentions. However, intentions do not ensure success.

Example 11. Consider Figure 2. At t_0, x may do either action a or action b, since both can potentially lead to one of the preferred scenarios being realized. However, if the other agent does action d, then no matter which action x chooses, he will not succeed with his intentions, because none of his preferred scenarios will be realized. ∎

4 Know-How

It is intuitively obvious that the mere fact of having an intention does not guarantee an agent's success; item IC3 above showed that even persistence is not sufficient for success. A key ingredient is know-how. An agent can succeed with his intentions only if he has the requisite know-how. We discuss the other prerequisites in section 5. Here we concentrate on formalizing know-how.

We propose that an agent, x, knows how to achieve p, if he is able to bring about p through his actions, i.e., force p to occur. The agent's beliefs or knowledge must be explicitly considered, since these influence his decision. For example, if an agent is able to dial all possible combinations of a safe, then he is able to open that safe: for, surely, the correct combination is among those that he can dial. On the other hand, for an agent to really know how to open a safe, he must not only have the basic skills to dial different combinations on it, but also know which combination to dial.

A tree of actions consists of an action, called its *radix*, and a set of subtrees. The idea is that the agent does the radix action initially and, then, picks out one of the available subtrees to pursue further. In other words, a tree of actions for an agent is a projection to the agent's actions of a fragment of **T**. Thus a tree includes *some* of the possible actions of the given agent, chosen to force a given condition.

Let Υ be the set of trees. $\emptyset$ is the empty tree. Then Υ is defined as follows.

T1. $\emptyset \in \Upsilon$
T2. $a \in \mathcal{B}$ implies that $a \in \Upsilon$
T3. $\{\tau_1, \ldots, \tau_m\} \subseteq \Upsilon$, $\tau_1, \ldots, \tau_m$ have different radices, and $a \in \mathcal{B}$ implies that $\langle a; \tau_1, \ldots, \tau_m \rangle \in \Upsilon$

Now we extend the formal language with an auxiliary construct. This extension is only meant to simplify our definitions.

L8. $\tau \in \Upsilon$, $x \in \mathcal{A}$, and $p \in \mathcal{L}$ implies that $x[\![\tau]\!]p \in \mathcal{L}$

$x[\![\tau]\!]p$ denotes that agent x knows how to achieve p relative to tree τ. As usual, the agent symbol can be omitted when it is obvious from the context. To simplify notation, we extend $\bigvee$ to apply to a given range of trees. Since distinct trees in each such range have distinct radix actions, the extension of $\bigvee$ from actions to trees is not a major step.

SEM-17. $M \models_t [\![\emptyset]\!]p$ iff $M \models_t \mathsf{K_t}p$

SEM-18. $M \models_t [\![(a)]\!]p$ iff $M \models_t \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_t}p)$
SEM-19. $M \models_t [\![\langle a; \tau_1, \ldots, \tau_m\rangle]\!]p$ iff
$M \models_t \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a](\bigvee_{1\le i\le m} \tau_i : ([\![\tau_i]\!]p)))$

Thus an agent knows how to achieve p by following the empty tree, i.e., by doing nothing, if he knows that p already holds. As a consequence of his knowledge, the agent will undertake no specific action to achieve p. The nontrivial base case is when the agent knows how to achieve p by doing a single action: this would be the last action that the agent performs to achieve p. In this case, the agent has to know that he will know p immediately after the given action.

It is important to require knowledge in the state in which the agent finally achieves the given condition, because it helps limit the actions selected by the agent. If p holds, but the agent does not know this, then he might select still more actions in order to achieve p.

Lastly, an agent knows how to achieve p by following a nested tree if he knows that he must choose the radix of this tree first and, when it is done, that he would know how to achieve p by following one of its subtrees. Thus know-how presupposes knowledge to choose the next action and confidence that one would know what to do when that action has been performed.

SEM-20. $M \models_t x\mathsf{K_h}p$ iff $(\exists\tau : M \models_t x[\![\tau]\!]p)$

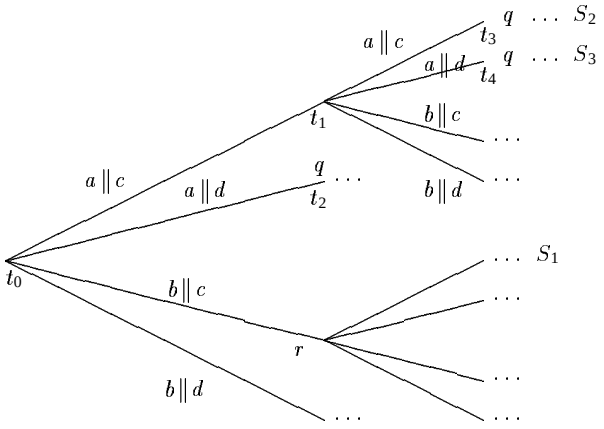

Fig. 3. Know-How

Example 12. Consider Figure 3. Let x be the agent whose actions are written first there. Assume for simplicity that each moment is its own unique alternative for x (this is

tantamount to assuming that x has perfect knowledge—our formal definitions do not make this assumption). Then, by the above definitions, $x\mathsf{K_t}q$ holds at t_3 and t_4. Also, $x\mathsf{K_h}q$ holds at t_1 (using a tree with the single action a) and at t_2 (using the empty tree). As a result, at moment t_0, x knows that if he performs a, then he will know how to achieve q at each moment where a ends. In other words, we can define a tree, $\langle a; a, \emptyset \rangle$, such that x can achieve q by properly executing that tree. Therefore, x knows how to achieve q at t_0. In terms of Example 5, our robot knows how to warm the room in the situation described by Figure 3. ∎

5 Success

Now that we have formalized our primitives, what can we use them for? Although we kept the primitives independent of each other, they can—and should—naturally be pulled together to reason about intelligent agents. One of the results we can obtain has to do with success conditions for agents. But first we mention some technically simple, but conceptually significant, results:

Lemma 1. $x\mathsf{I}p$ does not entail $x\mathsf{K_h}p$.
Consider Figure 3 and the assumptions of Example 12. If S_1 is the sole preferred scenario at t_0, then $x\mathsf{I}r$ holds at t_0. However, $x\mathsf{K_h}r$ clearly does not hold at t_0. ∎

Lemma 2. $x\mathsf{K_h}p$ does not entail $x\mathsf{I}p$.
This also follows directly from Figure 3, where $x\mathsf{K_h}q$ holds at t_0, but not $x\mathsf{I}q$. ∎

Suppose the agent both intends and know how to achieve something. Then would the intended condition eventually be realized? (This is akin to a liveness property from traditional computing when adapted to intelligent agents.) The obvious answer with the above assumptions is no! A lot more is required for success. The agent must not only intend a condition and knows how to achieve it, the agent must also act on his intention in a manner that exploits his know-how. Since agents act on their intentions, this requires that the agent persist with his intention long enough.

We have formalized most of the key premises of the above liveness result. We now formalize the remaining premise, which is that an agent acts on his intentions. Instead of attempting to formalize this in general, we consider the case where there is at least one action that would not lead the agent astray from his intentions. If the agent performs one such action, he can stay on one of the intended scenarios. By repeatedly performing such actions, the agent would eventually arrive at a state where the intended condition holds. Would such actions be available in all cases? Clearly, not. However, if the agent has the requisite know-how, then such actions are available.

IC5. **Perform sure action, if one is available:** If there is an action that guarantees that the agent stays on course with the intended scenarios, then the agent should perform that (or another such) action. Formally,
$(\exists a : (\forall S, t' : [S; t, t'] \in [\![a]\!]^x \Rightarrow S \in \mathbf{I}(x, t))) \Rightarrow (\exists a : (\forall S, t' : [S; t, t'] \in [\![a]\!]^x \Rightarrow S \in \mathbf{I}(x, t))$ and $[\mathbf{R}(x, t); t, t'] \in [\![a]\!]^x)$

Theorem 3. Under IC5 and IC4, $(x\mathsf{I}p \wedge x\mathsf{K}_{\mathsf{h}}p) \rightarrow \mathsf{RF}p$ ∎

This theorem gives us the key success property we require. It precisely formalizes the intuition that an agent who

- intends a condition,
- sticks to his intention long enough,
- has the necessary know-how, and
- acts rationally given his intentions and know-how

will succeed in realizing the intended condition.

Example 13. Consider Figure 3 interpreted as in Example 5, where the robot prefers scenarios S_2 and S_3. Then, the robot intends and knows how to make the room warm. If he acts according to the above postulate, either S_2 or S_3 will be realized (i.e., become the real scenario). Thus, the room will become warm! ∎

6 Other Approaches

Some of the primitives discussed above have attracted a fair amount of attention from researchers in AI and traditional computer science. Knowledge, understood as know-that, was extensively studied in the AI and distributed computing literatures. Three main classes of approaches can be identified. The simplest are the modal approaches, on which our present framework is based [2, 9, 6, 15]. These approaches are simple, but incorrectly predict that agents know the logical consequences of their knowledge. Another class of approaches are the sentential ones [11], which avoid the above problem, but do not facilitate many positive inferences involving knowledge. The third kind are hybrid approaches, which seek to avoid both extremes but at the price of a greater technical and conceptual complexity [7, 25]. In our present work, we have not taken advantage of our previous research [25] primarily because we wish to highlight orthogonal issues in the simplest possible framework.

The above classes of research would extend naturally into intentions. However, most work on intentions has been on modal approaches—the few exceptions include [12, 25]. The formal literature on intentions includes reference to time and beliefs or know-that [4, 20]. However, know-how is not considered. The above approaches consequently cannot prove success results such as we exhibited here. They require additional and, in our view spurious assumptions [21]. A common assumption is that agents will necessarily drop their intentions. Since agents are additionally assumed to do so only upon success, success is guaranteed. Many approaches also end up constraining the formal models so that the various primitives are extraneously tied to each other. [20] have shown how to remove some of these restrictions; we believe we have additional results, but lack the space to elaborate here.

Curiously, there has been little work on know-how, although knowledge preconditions for actions and plans were studied in [15, 16]. The recent STIT (for "seeing to it that") approaches [29] appear to embody similar intuitions, although they also mingle intentions and know-how, as we have identified those concepts.

We believe that the primitives developed herein will pay off when they begin to be incorporated into tools for reasoning about agents. Some useful results have been obtained by [19, 23, 29]. Complexity issues remain a challenge. This research area has been focused on conceptual issues in terms of the expressiveness to capture various cases, to obtain useful and avoid pernicious inferences. As these aspects are better understood, it will become appropriate to look for restricted sublanguages that have more efficient decision procedures or model checking algorithms. We expect that techniques such as the above when enhanced and applied to carefully engineered tractable formal languages will lead to sophisticated systems for designing, implementing, and validating powerful intelligent agents.

7 Conclusions

AI supplies a number of cognitive concepts for specifying intelligent agents. Although such anthropomorphic terms are attractive in various respects, they must be given a rigorous formal meaning in order to be technically useful in computer science. The concepts of knowledge and belief have been formalized in the computing literature. Whereas they admit certain forms of reasoning about computational systems, they do not allow other, equally essential, forms. We showed how a more powerful set of cognitive primitives can be formalized, in such as way as to capture more of the relevant AI *and* distributed computing intuitions. Further, by focusing on agents as computational entities, we can simplify our task somewhat. Indeed, certain technical properties such as consequential closure, which are weaknesses if one wishes to model human cognition, are quite acceptable in computing at large. Our approach naturally captures key computing concepts, such as liveness, in an AI-like approach. This kind of synthesis between AI and traditional intuitions is crucial to the further expansion of the agent metaphor.

References

1. Michael E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA, 1987.
2. K. M. Chandy and Jayadev Misra. How processes learn. *Distributed Computing*, 1:40–52, 1986.
3. Brian F. Chellas. *Modal Logic*. Cambridge University Press, New York, NY, 1980.
4. Philip R. Cohen and Hector J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990.
5. E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B. North-Holland Publishing Company, Amsterdam, The Netherlands, 1990.
6. Ronald Fagin, Joseph Y. Halpern, and Moshe Y. Vardi. What can machines know? on the epistemic properties of machines. In *AAAI*, pages 428–434, 1986.
7. Ronald Fagin, Joseph Y. Halpern, and Moshe Y. Vardi. A nonstandard approach to the logical omniscience problem. In *Proceedings of the Third Conference on Theoretical Aspects of Reasoning About Knowledge*. Morgan Kaufmann Inc., 1990.

8. R. Gotzhein and F. H. Vogt. The design of a temporal logic for open distributed systems. In *Proceedings of the International Conference on Open Distributed Processing*. Elsevier Science Publishers B. V., 1992.
9. Joseph Y. Halpern. Reasoning about knowledge. In Joseph Y. Halpern, editor, *Theoretical Aspects of Reasoning About Knowledge*, pages 1–26, 1986.
10. Carl Hewitt, C. Manning, Jeff Inman, and Gul Agha, editors. *Toward Open Information Systems Science*. MIT Press, Cambridge, MA, 1991.
11. Kurt Konolige. *A Deduction Model of Belief*. Morgan Kaufmann, Inc., 1986.
12. Kurt G. Konolige and Martha E. Pollack. A representationalist theory of intentions. In *IJCAI*, 1989.
13. Dexter Kozen and Jerzy Tiurzyn. Logics of program. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*. North-Holland Publishing Company, Amsterdam, The Netherlands, 1990.
14. John McCarthy. Ascribing mental qualities to machines. In Martin Ringle, editor, *Philosophical Perspectives in Artificial Intelligence*. Harvester Press, 1979. Page nos. from a revised version, issued as a report in 1987.
15. Robert C. Moore. A formal theory of knowledge and action. In Jerry R. Hobbs and Robert C. Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358. Ablex Publishing Company, Norwood, NJ, 1984.
16. Leora Morgenstern. A theory of knowledge and planning. In *IJCAI*, 1987.
17. Allen Newell. The knowledge level. *Artificial Intelligence*, 18(1):87–127, 1982.
18. A. H. H. Ngu, R. Meersman, and H. Weigand. Specification and verification of communication constraints for interoperable transactions. In *Proceedings of the 2nd International Conference on Cooperative Information Systems (CoopIS)*, 1994.
19. Anand S. Rao. Decision procedures for propositional linear-time belief-desire-intention logics. In *IJCAI Workshop on Agent Theories, Architectures, and Languages*, August 1995.
20. Anand S. Rao and Michael P. Georgeff. Asymmetry thesis and side-effect problems in linear-time and branching-time intention logics. In *IJCAI*, 1991.
21. Munindar P. Singh. A critical examination of the Cohen-Levesque theory of intentions. In *10th European Conference on Artificial Intelligence*, August 1992.
22. Munindar P. Singh. A semantics for speech acts. *Annals of Mathematics and Artificial Intelligence*, 8(I–II):47–71, 1993.
23. Munindar P. Singh. Maintenance and prevention: Formalization and fixpoint characterization. In *ECAI Workshop on Logic and Change*, August 1994.
24. Munindar P. Singh. *Multiagent Systems: A Theoretical Framework for Intentions, Know-How, and Communications*. Springer Verlag, Heidelberg, Germany, 1994.
25. Munindar P. Singh and Nicholas M. Asher. A logic of intentions and beliefs. *Journal of Philosophical Logic*, 22:513–544, 1993.
26. Munindar P. Singh and Michael N. Huhns. Cooperative information systems: Tutorial notes, 1995. Tutorial given at the International Conference on Distributed Computing Systems and at the International Joint Conference on Artificial Intelligence.
27. James White. TeleScript technology: The foundation for the electronic marketplace, 1994. White paper.
28. Darrell Woelk and Christine Tomlinson. The InfoSleuth project white-paper. Technical Report InfoSleuth-95-01, Microelectronics and Computer Technology Corporation, Austin, TX, January 1995.
29. Michael Wooldridge. Time, knowledge, and choice. In *IJCAI Workshop on Agent Theories, Architectures, and Languages*, August 1995.

This article was processed using the LaTeX macro package with LLNCS style