MUNINDAR P. SINGH

KNOW-HOW

In the knowledge lies the power.
Frederick Hayes-Roth

In the know-how lies the power.
Carl Hewitt

The study of knowledge is crucial to the science of rational agency. This fact is well-recognized in artificial intelligence (AI) and related fields. However, most often the form of knowledge that is studied and formalized is the knowledge of (putative) facts. We refer to this form of knowledge as *know-that*. Know-that has proved an extremely successful concept in AI, being the basis of a large number of AI systems, which are therefore termed *knowledge-based*.

There is great need, however, for other notions of knowledge as well. In particular, since rational agency is intimately related to actions, it is important also to consider the form of knowledge that is about actions and procedures. We refer to this form of knowledge as *know-how*. Intuitively, we might think of the distinction between know-how and know-that as reflecting the distinction between rational agents and on the one hand, and disembodied minds, such as knowledge-based expert systems, on the other.

This chapter introduces know-how and allied concepts from a conceptual standpoint. It presents a formalization of two accounts of know-how borrowed from [33]. It also and compares reviews a selected subset of the approaches available in the literature. It seeks to provide the background with which one may understand the details of the different technical approaches.

Historical Remarks. The noted British philosopher Gilbert Ryle is widely regarded as having been the first, at least in modern times, to have argued for the fundamental difference between knowing how and knowing that. Ryle devotes a chapter of his famous 1949 book *The Concept of Mind* to this distinction, arguing among other things for the key difference between (a) stupidity, that is, not knowing how, and (b) ignorance, that is, not knowing that [28]. He argues that the two are fundamentally separate notions, because often an agent may know how to perform certain complex actions, yet not know that he does a certain specific sequence. This distinction is interesting and related to one discovered years later

in the study of reactive systems in AI, for example, by Agre & Chapman [1] and others.

There has been much work in planning right from the early days of AI. To a large extent, Agre & Chapman and others were rebutting the centrality of planning when they proposed reactive architectures. However, the planning literature did not address the logical notion of know-how *per se*, although it considered the mechanics through which it could be realized. In fact, even the more traditional know-that was not always studied formally in AI, although systems that reasoned with it were abundant.

There was, however, considerable work on logics of know-that in the late 1970s and early 1980s. This was based on previous work in philosophical logic. Building on ideas from McCarthy & Hayes [20], Robert Moore developed a formal logic of knowledge that was essentially an S4 modal logic, but captured in terms of its first-order metatheory [22]. With slight modifications, this is the logic we present in section 2.5. Moore and others did not study know-how *per se*, but rather the knowledge required to execute plans. For example, to execute a conditional plan requires knowing whether its condition is true. Know-How has only recently begun to be studied intensively in AI. Indeed, Jaakko Hintikka, who gave the first formalization of knowledge using ideas from modal logic [14], observed that the logic of knowing how had proved difficult to develop [15] (cited by McCarthy & Hayes [20, p. 447]).

We began looking at know-how as a first-class topic of investigation in the late 1980s. Around the same time, Werner worked on a general theory that included abilities, but not at the present level of detail [41]. Independently, Meyer and associates studied capabilities from a perspective that included other concepts captured as modal operators [38]. Some philosophical work on this subject carried out over roughly the same time-frame by Brown [6], Belnap & Perloff [2], Chellas [8], and Segerberg [31].

Although we take the notion of know-how seriously, we confess that in developing formal theories of it, we shall not be supporting all of the associated philosophical positions. In particular, the very definition of know-how has not much to do—pro or con—with the doctrine of strict reactivity, as evinced in the works of Ryle and Agre & Chapman. This is because the notion of know-that, which we also discuss, can be taken as describing the knowledge of an agent in an explicit conscious sense, or in an implicit sense. Indeed, the specific formal theories we describe mostly develop an implicit notion of knowledge. The distinction between explicit and implicit knowledge, however, is not definitional or logical, but related to the computational power available to an agent. In other words, it will be just as acceptable to us that an agent can describe his know-how as that he cannot.

Organization. This chapter provides a conceptual introduction to several different variants of know-how. Although it includes some technical description to

give a flavor of how such formalizations bring together ideas from temporal and dynamic logics, the cited works should be read for their technical details.

The rest of this chapter is organized as follows. Section 1 introduces the key concepts and motivates the study of know-how. Section 2 describes our technical framework, including formalizations of the background concepts of time, actions, and know-that. Section 3 presents our definition of know-how. Section 4 discusses other approaches known in the literature, and relates them to the approach of section 3. Section 5 concludes with a discussion and pointers to some future directions.

1 MOTIVATION

Agency is inherently about performing actions. Because of the intimate relationship between agency and actions, the formal study of rational agents has involved the development of a number of folk concepts of which several relate to actions. Two such key concepts are intentions and desires studied by Rao & Georgeff [27] among others. Of the two, intentions have the closer and more direct relationship with actions, and we consider them in more detail. Intentions are generally understood as having a causal relationship with actions—they not only lead an agent to select suitable actions, but also to perform those actions. As a consequence, intentions have another role in rational agency, namely as explanations of actions, which can be used by designers and analyzers to reason about some agent's behavior, or by the agents themselves to reason about the behavior of other agents.

One of the ways in which intentions are applied in rational agency is as specifying the ends an agent has chosen to pursue. These intentions lead to deliberation by the agent, leading him to adopt additional, more specific intentions as means to his original ends. This process can iterate several times, resulting (if successful) in intentions that the agent can act on directly. This view of deliberation is shared by many researchers, including the philosophers Bratman [4] and Brand [3].

The successful use of intentions in theories of rational agency, therefore, relies upon their linkage to actions. For instance, a natural question is to determine under what circumstances an intention may be taken to lead to success. The simple answer is that an intention can lead to success when it is held long enough, is acted upon, and when the agent has the requisite know-how. This, in our mind, is the single most important motivation for the study of know-how, and was the basis for the work reported in [33].

There are obvious connections between intentions and know-how, some negative. For example, intentions do not entail know-how—you cannot always do what you intend to. Similarly, know-how does not entail intentions—you don't always intend what you have the know-how to do. Although intentions are not formally discussed in this chapter, it will be helpful to keep these connections, at least informally, in mind.

1.1 Actions

When talking of actions, it is conventional to define *basic* actions as those that an agent can perform atomically with a single choice. Philosophers have spent considerable energy in attempting to give necessary and sufficient conditions for when an observed event counts as an agent's action. As for other important topics, there is profound disagreement among the philosophers! Some approaches, exemplified by Searle's work, define actions in terms of what he calls intentions-in-action [29]. Roughly, what this means is that the agent should have the intention to do the given action he is in fact doing, and that his intention should play some causal role in the performance of that action. Another interesting theory is the STIT approach, due to Belnap & Perloff, which states that the actions of an agent are what he has seen to [2]. In a similar vein, Brown argues that actions are exercised abilities [6]. Both of the latter approaches are discussed below.

Somewhat in sympathy with these approaches, the theories of most interest to computer science simply assume that the basic actions are given in the model. We follow this approach in our treatment below. We assume that basic actions can be performed through a single choice by an agent. In other words, the basic actions correspond to the atomic abilities of the agent. Because we do not require that the set of basic actions of an agent is unvarying, the agent must choose from among the basic actions available in the given situation. In this way, there is a component of opportunity wired into the actions. This is quite realistic. For example, a robot can move forward in a hall, but not when pushing against a wall. We could alternatively model the attempt to move as an action in its own right, and leave the success of the move as something to be determined *post hoc*. As far as our theory is concerned, this is not a major step—all we require is that there is a set of basic actions.

A natural extension is to high-level actions. High-level actions can be specified indirectly as propositions that an agent can achieve through a combination of lower-level basic actions. Indeed, many actions can be specified naturally only through the corresponding propositions. This idea too has been long been recognized in the philosophy of actions, for example, by von Wright [39] (cited by Segerberg [31, p. 327]). Although basic actions can be performed directly if the agent has the corresponding physical ability, performing complex high-level actions frequently requires not only the physical ability to perform the underlying basic actions, but also the knowledge to select the appropriate actions to perform at each stage of the complex action.

Thus know-how, when applied to high-level actions, inherently includes or supervenes on the notion of know-that. This is an important connection between the two notions. This connection was recognized in the early work on formalizations of knowledge in AI, for example, by Moore, but framed in terms of the knowledge required to perform specific plans of actions, where the plans, which include conditional actions, correspond to high-level actions. The treatment of high-level actions leads to another view of know-how, which is of course related to intentions

as well. Some of the approaches described below will exploit this connection.

Intentions and know-how (or ability) and indeed even plans in general have usually been viewed as being directed toward the achievement of specific conditions. It is equally natural, and in many cases better, to consider not only the achievement of conditions but also their *maintenance*. We have developed an approach to maintenance in [34]. When intentions are similarly expanded, we would expect a similar relationship between intentions to maintain and the know-how to maintain, as between intentions to achieve and know-how to achieve. However, this subject has not yet been thoroughly studied in the literature and the bounds of the expected similarity are not known. For this reason, while acknowledging its importance, we discuss maintenance to a lesser degree here.

1.2 Separating versus Combining Ability and Opportunity

We informally describe two main classes of approaches to know-how. Ability refers to the intrinsic capability of an agent to do something reliably (if he knows what to do). Opportunity refers to the specific openings that an agent may have in specific situations to apply his ability. Know-how refers to the knowledge of how to achieve certain conditions, that is, to perform high-level actions. Intuitively, it is ability combined with the know-that to determine what actions to perform.

Ability and Opportunity

It is traditionally common to distinguish between *ability* and *opportunity*. This understanding is quite natural with respect to the natural language meanings of the two terms. With this understanding, ability refers to the reliable performance of an action by an agent, where the reliability is assessed over all possible situations.

Although natural, this account adds some complexity to the formal treatment. This is because to tease apart the definition of ability from the definition of opportunity requires that we consider *counterfactual* conditions of the following form: for an agent to have an ability means that he would succeed in achieving the given condition *if* he has the opportunity and carries out his actions.

Such approaches makes a subtle distinction between what an agent has the ability for and what he can do now. Importantly, the agent may have the ability but not actually succeed, because he lacks the opportunity. This is technically difficult, because to establish the above conditional statement requires modeling the situations in such a manner as to enable moving from the actual situation (where the agent does not have the opportunity) to a counterfactual situation (where the agent has the opportunity). In doing so, we must ensure that acquiring the opportunity has caused the ability neither to emerge nor to be lost. If either of those is the case, then the opportunity is not independent of the ability, and therefore neither concept is really coherent in itself. This complexity makes this class of approaches less tractable conceptually.

Situated Know-How

There is another class of approaches that do not separate the ability from the opportunity. Therefore, these approaches apply to a given specific situation in which an agent may find himself. In this situation, he has certain abilities and certain opportunities, but we consider them together rather than separately. With this understanding, ability refers to the reliable performance of an action by an agent, where the reliability is assessed only in the given situation.

As a result, the problem of identifying the abilities that cannot be exercised is avoided. Conversely, the accuracy of the concepts studied as formalizations of natural language concepts may be reduced. However, this trade-off is acceptable in coming up with formal concepts that may not be perfect realizations of the folk terms, but are nevertheless useful and more technically tractable than the folk concepts they formalize. We believe that this is just an instance of a pattern that one encounters repeatedly in the formalization of the folk concepts underlying rational agency.

1.3 Possible Worlds versus Representational Approaches

In general, there are a number of possible analyses of informal cognitive concepts, such as knowledge. In particular, for knowledge, there is a family of approaches based on modal logics, which is contrasted with the family of approaches based on sentential logics. The key intuitive difference between these two families is that the modal approaches support a number of inferences, including some inferences that are counterintuitive for humans and other resource-bounded agents.

Modal Approaches

The modal approaches are based on the so-called *possible worlds* approaches that consider alternative sets of situations [7]. These approaches postulate an *alternativeness* relation on situations. This relation is used to give a semantics to the modalities of *necessity* ($\Box$) and *possibility* ($\Diamond$). A proposition is necessarily true at a situation if it holds in all situations that are alternatives of the given situation. A proposition is possibly true if it holds in some alternative situation of the given situation. This definition becomes interesting when additional requirements are stated on the alternativeness relation, for instance, whether it is reflexive, symmetric, transitive, and so on.

The variations among these definitions don't concern us here. However, all of the simpler definitions support the inference of *consequential closure:* if p is necessary, then so are all its logical consequences. The argument is quite simple. Suppose p is necessary. Then p is true in all alternative situations. If q is a logical consequence of p, then q is also true in each of those situations. Hence, q is also necessary. Modal approaches that satisfy consequential closure are termed *normal*.

Some of the more sophisticated modal approaches are *non-normal*. In these approaches, the alternativeness relation relates a situation to a set of situations, and necessity is defined in terms of truth within the alternative set. As a consequence, non-normal modal logics avoid consequential closure. However, such models must validate *closure under logical equivalence:* if p is necessary, then so are all propositions logically equivalent to it.

Possible worlds approaches to knowledge were introduced by Hintikka [14] and developed by several others. These approaches model knowledge as a necessity modal operator. In this case, the underlying relation is one of *epistemic alternativeness*—there is a different relation for each agent. Neither consequential closure nor closure under logical equivalence is acceptable in describing the knowledge of a computationally limited agent. However, sometimes knowledge can be understood from the perspective of an objective designer, in which case it is the designer's capacities of reasoning that are postulated.

Sentential Approaches

The *representational* (typically, *sentential*) approaches contrast with the modal approaches in which an agent is said to have an explicit set of representations (typically, sentences in a formal language) that describe his cognitive state. An agent knows a condition (expressed in a particular sentence) if that sentence is among those in the set of sentences describing his cognitive state. The advantage of this approach is that it is explicit about the agent's knowledge. If a logical consequence of a given sentence is not included in the set of sentences that define an agent's cognitive state, then there is no implication that the logical consequence is known. This is certainly more accurate when describing the knowledge states of agents. However, in restricting ourselves to precisely the sentences that are included in the set of sentences, we also prevent all kinds of other inferences that might be viable. In this sense, the sentential approaches preclude all general inferences; this observation limits their usability in reasoning about agents. A representational approach was developed by Konolige [18].

Hybrid Approaches

There are also some hybrid approaches, which seek to use possible worlds approaches for their semantic ease, in conjunction with some representations to characterize how a computationally bounded agent may reason about his knowledge. These approaches prevent the problematic inferences of the possible worlds approaches, but give a semantic basis for the inferences they do support. Two example approaches are those of Fagin & Halpern [13] and Singh & Asher [35]. The latter also considers intentions in the same framework as beliefs. Although promising, these approaches are technically quite complex, and have not drawn as much attention in the literature as perhaps they deserve.

Consequently, the modal approaches are by far the most common ones in the literature. Accordingly, we primarily consider such approaches below.

2 TECHNICAL FRAMEWORK

It is clear and widely agreed that any formal treatment of any of the shades of know-how requires a mathematical framework that includes actions as primitives. Usually there is also the need for a separate notion of time to help capture other associated intuitions. Traditional approaches, which consider commonsense situations, and especially those that are geared to the natural language meanings of the above terms, include some notion of Newtonian time, usually in terms of a date-based language and semantics. This enables them to express facts such as whether an agent can catch the bus by 3:00 PM, where 3:00 PM is defined independently of any specific course of events, given as it were by a Newtonian clock. This is convenient enough in many cases, so we allow the assignment of real date values to different situations, although the rest of our framework involves a branching time model and considers as primary a qualitative ordering among moments.

2.1 Branching-Time Models

In conceptualizing about actions and know-how, it is important to recognize the choices that the agents can exercise as they go about their business. Intuitively, the world can evolve in several different ways, but the agents constrain it to evolve in a way that suits them by performing appropriate actions. To the extent that they can achieve what they want they can be said to have the requisite ability.

The need to represent choices translates into the requirement of representing multiple courses of events in our technical framework so that our formal definitions can exploit that multiplicity. There are a number of ways of capturing this requirement. One way that is intuitively quite direct is to construct *branching models* of time. There is a large variety of these models; at the very least, because of our need to represent and reason about multiple actions, we must allow the branching to take place into the future. For simplicity, we consider models that are linear in the past. This captures the idea that the past can in principle be fully known, but the future is nondeterministic as long as the agents' choices are open. The ignorance that some agent may have about the past is captured by the general mechanism of beliefs.

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, that is, 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.

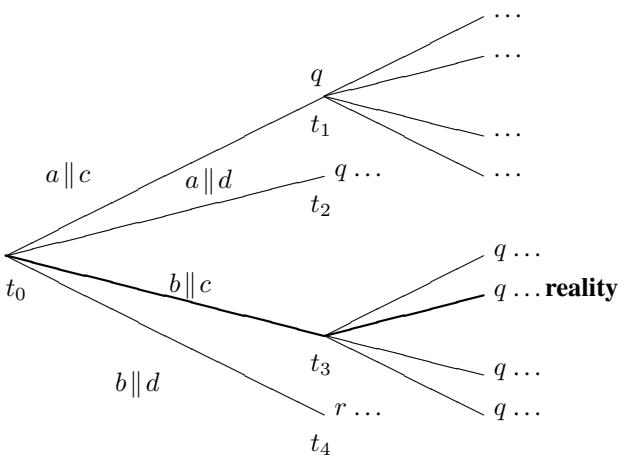

Figure 1. An Example Formal Model

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 2.2). 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. 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 1 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.

A lot of good research has been carried out on temporal and dynamic logics

and models of time and action. We encourage the reader to peruse at least the following works: Emerson [12], Kozen & Tiurzyn [19], van Benthem [36, 37], and Prior [25, 26].

2.2 The Formal Language

Given especially the branching-time models described above, it is convenient to adopt as our formal language one that includes not only traditional propositional logic, but also certain operators borrowed from temporal and dynamic logics. In doing so, we can emphasize the intellectual heritage of the present approaches on research into logics of program, developed for and applied on problems in computing at large. Consequently, our language includes a capacity for expressing conditions, actions, and branching futures. Time is intimately related to actions.

We use a qualitative temporal language, $\mathcal{L}$, based on CTL* [12]. Our language captures the essential properties of actions and time that are of interest in specifying rational 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.

SYN-1. $\psi \in \Phi$ implies that $\psi \in \mathcal{L}$

SYN-2. $p, q \in \mathcal{L}$ and $x \in \mathcal{A}$ implies that $p \wedge q$, $\neg p$, $\mathsf{P}p$, $(\bigvee a : p)$, $(x\mathsf{K}_{\mathsf{t}}p)$, $(x\mathsf{K}_{\mathsf{h}}p), (x\mathsf{K}_{\mathsf{m}}p) \in \mathcal{L}$

SYN-3. $\mathcal{L} \subseteq \mathcal{L}_s$

SYN-4. $p, q \in \mathcal{L}_s$, $x \in \mathcal{A}$, and $a \in \mathcal{B}$ implies that $p \wedge q$, $\neg p$, $p\mathsf{U}q$, $x[a]p$, $x\langle a\rangle p$ $\in \mathcal{L}_s$

SYN-5. $p \in \mathcal{L}_s$ implies that $\mathsf{A}p, \mathsf{R}p \in \mathcal{L}$

SYN-6. $p \in (\mathcal{L}_s - \mathcal{L})$ and $a \in \mathcal{X}$ implies that $(\bigvee a : p) \in \mathcal{L}_s$

2.3 Informal Description

The formulas in $\mathcal{L}$ refer to moments in the model. Each moment has a state corresponding to a possible snapshot of the system. The formulas in $\mathcal{L}_s$ refer to scenarios in the model, that is, to specific computations of the system. Note that $\mathcal{L} \subseteq \mathcal{L}_s$. However, our formal semantics, given in section 2.4, ensures that the formulas in $\mathcal{L}$ are given a unique meaning even if interpreted as being in $\mathcal{L}_s$.

Recall that the semantics of a formal language is given by stating rules through which the interpretation of syntactically acceptable formulae can be determined.

This is carried out in the context of some *model*, that is, a description of the world where the formal language is being applied. In logic, the term *model* is used with a specific technical meaning. A model is not just a description of reality, but one that is fine-tuned with respect to the given logical language. Thus our formal model should capture the structure exhibited in Figure 1.

The boolean operators are standard. We introduce two abbreviations. For any $p \in \Phi$: $\mathsf{false} \stackrel{\text{def}}{=} (p \wedge \neg p)$ and $\mathsf{true} \stackrel{\text{def}}{=} \neg\mathsf{false}$.

The temporal and action formulas explicitly consider the evolution of the system's state—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, $\mathsf{E}p \Leftrightarrow \neg\mathsf{A}\neg p$.

Example 2 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 helps tie together intuitions about what may and what will happen.

Example 3 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 [19], 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. Under these assumptions, in each of $[\,]$ and $\langle\,\rangle$, p holds at the moment where the action ends. Thus, $x[a]p \Leftrightarrow \neg x\langle a\rangle\neg p$, that is, $[\,]$ and $\langle\,\rangle$ are duals.

Example 4 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 for some action p becomes true. The action symbol a typically occurs in p and is replaced by the specific action which makes p true. The construct $(\bigwedge a : p)$ abbreviates $\neg(\bigvee a : \neg p)$. This means that for all actions p becomes true.

Example 5 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 other important construct is $x\mathsf{K_h}p$. $x\mathsf{K_h}p$ is interpreted to mean that agent x *knows how* to achieve p. The formal definition of these operators is the subject of this paper.

2.4 The Formal Model

Let $M = \langle \mathbf{T}, <, [\![\,]\!], \mathbf{R}, \mathbf{K}\rangle$ 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. The binary relation $<$ is a partial order over $\mathbf{T}$, and 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. We further assume that $<$ is discrete and finitely branching. $[\![\,]\!]$ 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 alluded to above, and which is highlighted by a bold line in Figure 1.

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 given by Emerson [12]. 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. $M \models_t (\bigvee a : p)$ iff $(\exists b : b \in \mathcal{B}$ and $M \models_t p|^a_b)$, where $p \in \mathcal{L}$

SEM-8. $M \models_{S,t} (\bigvee a : p)$ iff $(\exists b : b \in \mathcal{B}$ and $M \models_{S,t} p|^a_b)$, where $p \in (\mathcal{L}_s - \mathcal{L})$

SEM-9. $M \models_{S,t} p\mathsf{U}q$ 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-10. $M \models_{S,t} x[a]p$ 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-11. $M \models_{S,t} x\langle a\rangle p$ 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-12. $M \models_{S,t} p \wedge q$ iff $M \models_{S,t} p$ and $M \models_{S,t} q$

SEM-13. $M \models_{S,t} \neg p$ iff $M \not\models_{S,t} p$

SEM-14. $M \models_{S,t} p$ iff $M \models_t p$, where $p \in \mathcal{L}$

The above definitions do not include the postulates for know-that and know-how on purpose. We introduce them after further technical motivation in the sections below.

2.5 Know-That

We discuss know-that as part of the technical framework, because logics of know-that are standard, but provide a key basis for the study of know-how. As explained in section 1.3, the basic idea of know-that or knowledge as captured in most common formalisms is that the knowledge of an agent helps the agent discriminate among possible states of the world.

$\mathbf{K}$ 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 $\mathbf{K}$ is an equivalence relation, resulting in $\mathsf{K_t}$ being an S5 modal logic operator [7], which grants both positive and negative introspection.

SEM-15. $M \models_t x\mathsf{K_t}p$ iff $(\forall t' : (t, t') \in \mathbf{K}(x) \Rightarrow M \models_{t'} p)$

3 FORMALIZATION

We now use the above technical framework to present some of the common approaches to know-how. In some cases, we modify the details of the approaches a little to facilitate the exposition.

We propose that an agent, x, knows how to achieve p, if he is able to bring about p through his actions, that is, to 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. (Let's assume, for simplicity, that trying a wrong combination precludes the success of any future attempts.)

3.1 Trees

To formalize know-how, we define the auxiliary notion of a *tree* of actions. A tree 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. Intuitively, a tree encodes the selection function that the agent may use in choosing his actions at each moment. A tree should be bushy enough to cover all the cases.

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 $\langle a; \tau_1, \ldots, \tau_m \rangle \in \Upsilon$

Sometimes it is convenient to just write a as a shorthand for the tree $\langle a; \emptyset \rangle$. Now we extend the formal language with an auxiliary construct.

SYN-7. $\tau \in \Upsilon$, $x \in \mathcal{A}$, 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 the 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-16. $M \models_t [\![(\emptyset)]\!]p$ iff $M \models_t \mathsf{K_t}p$

SEM-17. $M \models_t [\![(a)]\!]p$ iff $M \models_t \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_t}p)$

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

The denotation of a tree, that is, its *know-how denotation*, is implicit in this definition. We need to make the corresponding denotation explicit when we consider maintenance.

3.2 Plain Know-How

Thus an agent knows how to achieve p by following the empty tree, that is, 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-19. $M \models_t x\mathsf{K_h}p$ iff $(\exists\tau : M \models_t x[\![(\tau)]\!]p)$

Example 6 Consider Figure 2. 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.

Now we present a recursive characterization of know-how. This characterization, which is remarkably simple, forms the basis of the mu-calculus approach developed in [34].

Lemma 1 $\mathsf{K_t}p \vee (\bigvee \alpha : \mathsf{K_t}(\exists\langle\alpha\rangle\mathsf{true} \wedge \forall[\alpha]\mathsf{K_h}p)) \Leftrightarrow \mathsf{K_h}p$

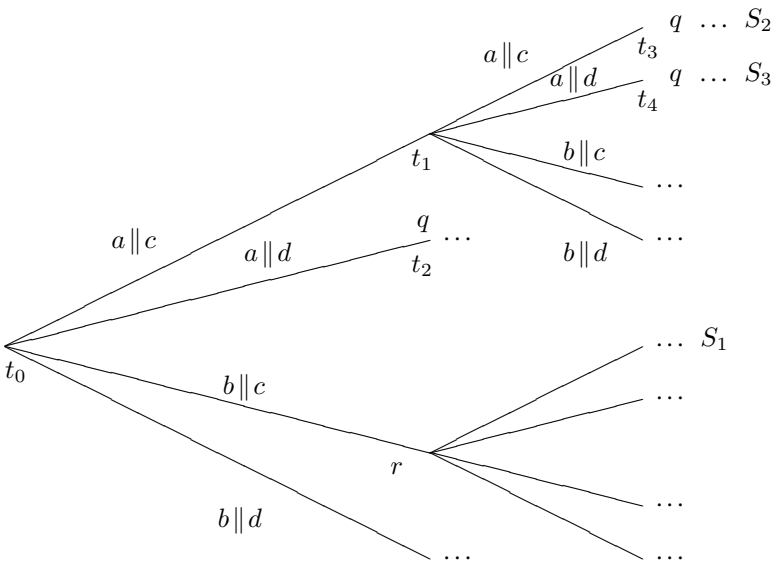

Figure 2. Know-How

3.3 Reliable Know-How

The above treatment of know-how captures it essentially as any other modal operator, albeit one that combines temporal and dynamic aspects. The truth and falsity of this operator are determined at a specified moment. The traditional, natural language use of the term know-how, however, includes a greater sense of reliability. In effect, reliability requires looking not only at the given moment, but also at other moments. Once the finer notion has been formalized, its reliable version proves fairly natural. To simplify our presentation, let us assume that 0 is the unique initial moment in $\mathbf{T}$. We also add an operator K_{rh} (meaning reliably knows-how) to the formal language. Then we can simply state that

SEM-20. $M \models_t x\mathsf{K}_{\mathsf{rh}}p$ iff $M \models_0 x\mathsf{K}_{\mathsf{rh}}p$, where $0 < t$

SEM-21. $M \models_0 x\mathsf{K}_{\mathsf{rh}}p$ iff $M \models_0 \mathsf{AG}x\mathsf{K}_{\mathsf{h}}p$

This states that reliable know-how is obtained if the agent has the restricted know-how in every possible state. Alternative versions of reliable know-how can be readily formulated. In particular, those that select some relevant moments to a given

moment would correspond more to natural language, but would also be technically more complex.

4 ALTERNATIVE APPROACHES

We now consider some leading approaches from the computer science and philosophy literatures.

4.1 Actions as Exercised Abilities

Brown distinguishes between ability and opportunity. He formalizes his approach in a modal logic with operators for necessity $\Box$ and possibility $\Diamond$ [6], which builds on his previous work on ability [5]. Brown captures ability and opportunity as nested applications of these operators.

A strength of Brown's approach is its intuitive treatment of the interplay between action and ability. He shows how each can be defined in terms of the other, and how they share various logical inferences, and how they differ. Brown presents a number of interesting axioms and inference rules for his modalities, and studies which of them must be validated by different interpretations of those modalities. For example, $\Box p \Rightarrow p$ is satisfied when $\Box p$ is interpreted as "the agent does p," and not as "the agent is able to do p."

A key notion is that an agent has so acted as to bring about the truth of a given condition. The main intuition is that ability is about the reliable performance of actions. Conversely, actions are exercised abilities (p. 101). Only the reliable consequences of one's actions are counted as actions (p. 96). It also appears that Brown counts all the reliable consequences of an action as actions, which may be too strong, when talking about (intentional) action in general.

Tentatively considering ability as the possibility operator of modal logic, Brown argues that his modal logic be *non-normal* (p. 98), as described in section 1.3. This means that it need not support the inference that an agent who is able to achieve $A \vee B$ is able to achieve A or achieve B. This would clearly be undesirable. Brown's proposed interpretation of ability is stronger than mere modal possibility. Yet the same reason applies for giving it a non-normal semantics.

In Brown's formulation, a *relevance relation* is postulated that relates a possible world to subsets of possible worlds that are somehow "relevant" to it. Each subset is called a *cluster*. The agent is said to be able to achieve p iff there is a relevant cluster such that each world within it satisfies p. Intuitively, each cluster corresponds to the possible outcomes of an action. This is intuitively similar to our definition, and indeed most other definitions of know-how, in that the agent selects an action, such that in each resulting state, the given condition holds.

However, this approach is a purely modal approach, with no reference to any epistemic or temporal aspect. Thus, the agent's knowledge is not taken into ac-

count. This agrees with the common definition of ability. Our remark is not meant as a criticism of Brown, but to highlight what is nevertheless an important point of difference with know-how. Even actions are not explicitly modeled. Consequently, although models and a semantics are given and have a direct connection with the logical inferences under discussion, the models are not obviously related to our intuitions about actions and ability. Brown does not offer any reasonable intuitive interpretations of the relevant clusters. Are they actions, action sequences, routines, or regular programs? Let's assume that they are composite actions of some sort.

Brown describes two interesting properties of the relevant clusters. First, he requires that they are *weakly-centered*, meaning that the given world is always a member of each relevant cluster. This effectively means that we are looking at the case where the given action is in fact performed. In other words, the action can take the world in question along its real scenario.

Second, Brown states that the relevant clusters are *closed under pairwise intersection*. Roughly, this means that the "parallel composition" of two actions is also an action. Or, more strongly (in the presence of weak-centering), real actions can be composed to yield another real action. If the agents perform one basic action at a time, the composition can be effected in terms of either interleaving the component actions, or by having one action be a subsequence of the other.

4.2 STIT: Seeing To It That

STIT refers to the *seeing to it that* approaches developed initially by Belnap & Perloff [2], and refined and explained by Chellas [8], from whose exposition we benefited a lot. Perloff compares the STIT approach to leading philosophical approaches in [24]. The STIT approaches seek to characterize the notion of ability in which an agent sees to it that a certain condition is obtained. This presupposes continual actions by the agent leading up to success in achieving the given condition. Informally, an agent sees to it that p if p is not already true, is not inevitable, and he can select and perform certain actions leading up to the truth of p. The STIT approaches are also naturally expressed in branching-time models.

Intuitively, STIT is about the actions that have just been performed. In fact, we find the progressive misleading, and believe a better gloss for STIT would be *has just seen to it that*. This gives its formal logic some characteristics different from the logics of ability or opportunity. Indeed, the concept is better understood as a form of high-level action.

Just like in the approach of section 3 above, Belnap & Perloff consider histories with linear past and branching future (pp. 189–192). The moments are ordered qualitatively, as described above. Belnap & Perloff also assume, as we did above, that each agent can act in different ways, but the future depends on the combination of the actions of the agents and events in the environment. The choices of each

agent partition the set of future branches (like $\mathbf{S}_t$ in our framework). Intuitively, each choice-set corresponds to the result of performing some (sequence of) actions.

With this setup, Belnap & Perloff state (p. 191) that an agent x STITs p at moment m iff there is a past moment m_0, such that

- x had a choice set at m_0 such that at every branch in the choice set, p holds at the moments that are alternatives to m, that is, x had a choice that guaranteed p
- x had a choice in which p was not guaranteed.

The definition as stated has a bug in it. We must also ensure that the given moment m itself lies on one of the branches in the choice set being used. The version given by Chellas, however, fixes the bug.

Chellas, in his approach–termed the *imperative approach*, considers linear histories (like scenarios), but relates them intuitively to branching time, so the effect is practically indistinguishable. However, he assumes that a metric time is given with which states in the histories can be identified. Chellas has the notion of an *instigative alternative (IA)* of a history at a time. A (linear) history h' is an IA to h at t if h' is under the control of, or responsive to, the actions of the agent. In this way, actions are defined indirectly via the IAs. The agent's high-level actions are defined in terms of what holds on all of the IAs at the given history and time.

Chellas assumes *historical relevance* of the IAs meaning that the IAs of a history agree with it up to the given time. He also assumes *reflexivity* meaning that a history is an IA to itself.

When relating the IAs to actions, it is not clear if the IAs are the actions the agent is instigating or may instigate. We would expect a set of set of IAs, as in Belnap & Perloff's approach, not a single set. In conjunction with reflexivity, this suggests that the IAs are in fact the chosen IAs that the agent is pursuing on the given history as well.

Neither Chellas nor Belnap & Perloff mention knowledge explicitly, although their intuitive descriptions seem to call for it. An agent could not see to it that something without knowing what he was doing.

The STIT approaches are geared more toward the natural language uses of the term *seeing to it that* than toward the technical definition *per se*. A point where this focus of the STIT approaches is reflected is in their attempt at capturing the felicity of natural language statements involving an agent seeing to it that something obtain. For example, they require that the given condition does not already hold and is not inevitable (independent of the agent's actions). Although these restrictions are appropriate when you announce that a given agent can see to it that something happens, they are not necessarily appropriate as intrinsic components of the concept itself. We believe that these are extrinsic properties that are based on the pragmatics of communication, rather than the semantics of the underlying

concept. Indeed, these properties can be thought of as specific Gricean inferences on the *report* of what an agent can see to.

4.3 Strategic Know-How

The approach to know-how described in section 3 considers the actions of the agents directly, although organized into trees. A natural extension is to consider higher level compositions of the actions, which result in a more realistic treatment of know-how [33]. This extension uses *strategies*, which describe at a high level the actions that an agent may perform. Strategies have long been studied in AI and cognitive science. Mention of them goes back to Kochen & Galanter [17] (cited in [21, p. 17]), McCarthy & Hayes [20], and Brand [3].

Strategies do not add any special capability to the agents. They simply help us, designers and analyzers, better organize the skills and capabilities that agents have anyway. Hierarchical or partial plans of agents, thus, turn out to be good examples of strategies. The formal notion of strategies here is based on regular programs, as studied in dynamic logic [19], with an enhancement to allow high-level actions instead of atomic programs, and restricting the language to only allow deterministic programs. The first column of Table 1 shows the syntax. Intuitively, the strategy $\mathbf{do}(q)$ denotes an abstract action, namely, the action of achieving q. It could be realized by any sequence of basic actions that yields q. The remaining constructs are standard.

Y	$\downarrow_t Y$	$\uparrow_t Y$
skip	**skip**	**skip**
$\mathbf{do}(q)$	if $M \models_t \neg q$ then $\mathbf{do}(q)$ else **skip**	**skip**
$Y_1;Y_2$	if $\downarrow_t Y_1 \neq \mathbf{skip}$ then $\downarrow_t Y_1$ else $\downarrow_t Y_2$	if $\downarrow_t Y_1 \neq \mathbf{skip}$ then $(\uparrow_t Y_1);Y_2$ else $\uparrow_t Y_2$
if q **then** Y_1 **else** Y_2	if $M \models_t q$ then $\downarrow_t Y_1$ else $\downarrow_t Y_2$	if $M \models_t q$ then $\uparrow_t Y_1$ else Y_2
while q **do** Y_1	if $M \models_t \neg q$ then **skip** else $\downarrow_t Y_1$	if $M \models_t \neg q$ then **skip** else if $\downarrow_t Y_1 \neq \mathbf{skip}$ then $(\uparrow_t Y_1);Y$ else $\uparrow_t Y_2$

Table 1. Strategies: Syntax and Definitions of Current and Rest

It is useful to define two functions, *current* $\downarrow$ and *rest* $\uparrow$, on strategies. These functions depend on the moment at which they are evaluated. Let Y be a strategy. $\downarrow_t Y$ denotes the part of Y up for execution at moment t, and $\uparrow_t Y$ the part of Y that would remain after $\downarrow_t Y$ has been done. Assume that strategies are normalized with respect to the following constraints: (a) $\mathbf{skip};Y = Y$ and (b) $Y;\mathbf{skip} = Y$.

Then the $\downarrow_t Y$, which can be either $\mathbf{skip}$ or $\mathbf{do}(q)$. This helps unravel a strategy for acting on.

Strategies as Abstract Actions

The strategic definition of know-how builds on the definition given previously. To this end, we define $[\!\{\tau\}\!]^x_Y$ as the *know-how denotation* of a tree, τ, relative to a strategy, Y, for an agent, x. $[\!\{\tau\}\!]^x_Y$ is the set of periods on which the given agent knows how to achieve Y by following τ. Precisely those periods are included on which the agent has the requisite knowledge to force the success of the given strategy. The know-how denotation needs to be defined only for the base case of $\downarrow_t Y$. Formally, we have the following cases in the definition of $[\!\{\tau\}\!]^x_Y$.

The agent knows how to satisfy the empty strategy, $\mathbf{skip}$, by doing nothing, that is, by following the empty tree.

The agent may know how to satisfy the strategy $\mathbf{do}(q)$ in one of three ways: (a) by doing nothing, if he knows that q holds; (b) by following a single action tree, if he knows that it will force q; or, (c) by following a general tree, if doing the radix of that tree will result in a state in which he knows how to satisfy $\mathbf{do}(q)$ by following one of its subtrees. Thus we have:

$[S; t, t'] \in [\!\{\tau\}\!]^x_{\mathbf{do}(q)}$ iff

1. $\tau = \emptyset$ and $t = t'$ and $M \models_t x\mathsf{K_t}q$
2. $\tau = a$ and $M \models_t [\![\tau]\!]q$ and $M \models_{t'} x\mathsf{K_t}q$ and $(\exists t_1 : t < t' \leq t_1$ and $[S; t, t_1] \in [\![a]\!]$ and $(\forall t_2 : t \leq t_2 < t'$ implies $M \not\models_{t_2} q))$
3. $\tau = \langle a; \tau_1, \ldots, \tau_m\rangle$ and $M \models_{t'} x\mathsf{K_t}q$ and $M \models_t [\![\tau]\!]q$ and $(\exists t_1, t_2, i :$ $[S; t, t_1] \in [\![a]\!]$ and $1 \leq i \leq m$ and $[S; t_1, t_2] \in [\![\tau_i]\!]_{\mathbf{do}(q)}$ and $t_1 \leq t' \leq t_2)$ and $(\forall t_3 : t \leq t_3 < t'$ implies $M \not\models_{t_3} q)$

Intuitively, $[\!\{\tau\}\!]^x_{\mathbf{do}(q)}$ corresponds to the denotation of the abstract action performed by agent x of achieving q by exercising his know-how. Based on the above, we extend the formal language by allowing the operators $\langle\,\rangle$ and $[\,]$ to apply on strategies. Now we give the semantic conditions for the new operators. We must quantify over trees with which $\mathbf{do}(q)$ can be performed, because those trees are equally legitimate as ways to perform $\mathbf{do}(q)$.

SEM-22. $M \models_{S,t} x\langle\mathbf{do}(q)\rangle p$ iff $(\exists \tau, t' \in S : [S; t, t'] \in [\!\{\tau\}\!]^x_{\mathbf{do}(q)}$ and $M \models_{S,t'} p)$

This means that $\mathbf{do}(q)$ can be knowingly and forcibly performed on the given scenario and p holds at the moment at which it ends.

SEM-23. $M \models_{S,t} x[\mathbf{do}(q)]p$ iff $(\forall \tau, t' \in S : [S; t, t'] \in [\!\{\tau\}\!]^x_{\mathbf{do}(q)} \Rightarrow M \models_{S,t'} p)$

> This means that if the abstract action $\mathbf{do}(q)$ is knowingly and forcibly performed on the given scenario, then at the moment at which it is over, condition p holds.

The notion of know-how relative to a strategy can now be formalized to explicitly reflect the idea that strategies are abstractions over basic actions. An agent knows how to achieve p by following the empty strategy, $\mathbf{skip}$, if he knows that p. The justification for this is the same as the one for the case of the empty tree.

For a general strategy, not only must the agent know how to perform the relevant substrategies of a given strategy, he must know what they are when he has to perform them. We introduce two new operators to capture the agent's knowledge of the $\downarrow$ and $\uparrow$ of a strategy. The formula $x[Y]Y'$ means that for the agent x to follow Y at the given moment, he must begin by following Y', which is either $\mathbf{skip}$ or $\mathbf{do}(q)$; $x\overline{[Y]}Y''$ means that he must continue with Y''.

Thus, $x[Y]Y'$ holds only if $Y' = {\downarrow_t}Y$. However, since the agents' beliefs may be incomplete, $x[Y]Y'$ may be false for all Y'. Assuming $x[Y]Y'$, $x\overline{[Y]}Y''$ means that for the agent x to follow Y at the given moment, he must follow Y'' after he has followed Y'. As above, $x\overline{[Y]}Y''$ holds only if $Y'' = {\uparrow_t}Y$. We include only some sample definitions, and refer the reader to [33] for additional details.

SEM-24. $M \models_t x[\mathbf{do}(q)]\mathbf{skip}$ iff $M \models_t x\mathsf{K_t}q$

SEM-25. $M \models_t x[\mathbf{if}\ r\ \mathbf{then}\ Y_1\ \mathbf{else}\ Y_2]Y'$ iff $M \models_t (x\mathsf{K_t}r \wedge x[Y_1]Y') \vee (x\mathsf{K_t}\neg r \wedge x[Y_2]Y')$

SEM-26. $M \models_t x\overline{[\mathbf{do}(q)]}\mathbf{skip}$

Strategic Know-How Defined

An agent, x, knows how to achieve a proposition p by following a strategy Y, if there is a strategy Y' such that (a) $x[Y]Y'$ holds; (b) he knows how to perform Y'; and, (c) he knows that, in each of the states where Y' is completed, he would know how to achieve p relative to ${\uparrow_t}Y$. Since Y' is always of one of the forms, $\mathbf{skip}$ or $\mathbf{do}(q)$, Y is progressively unraveled into a sequence of substrategies of those forms. Formally, we have

SEM-27. $M \models_t x[\langle\mathbf{skip}\rangle]p$ iff $M \models_t x\mathsf{K_t}p$

SEM-28. $M \models_t x[\langle Y\rangle]p$ iff $M \models_t x\mathsf{K_t}(\exists x\langle{\downarrow_t}Y\rangle\mathsf{true} \wedge \forall x[{\downarrow_t}Y]x[\langle{\uparrow_t}Y\rangle]p)$ and $M \models_t x[Y]\downarrow_t Y$

The above definition requires an agent to know what substrategy he must perform only when he has to begin acting on it. The knowledge prerequisites for executing different strategies can be read off from the above semantic definitions. For example, a conditional or iterative strategy can be executed only if the truth-value of the relevant condition is known.

SEM-29. $M \models_t x\mathsf{K}_{\mathsf{hs}}p$ iff $(\exists Y : M \models_t x[\![Y]\!]p)$

4.4 Capabilities

van der Hoek *et al.* develop a theory of ability and opportunity [38]. This theory too is based on dynamic logic. However, they separate the actions of dynamic logic from the ability to perform them. van der Hoek *et al.* also use a deterministic variant of dynamic logic. In their notation, $\text{do}_i(\alpha)$ is the *event* corresponding to the agent i doing action α. Here do applies to actions, not to propositions. (We will elide the agent symbol below.) As in dynamic logic, $\langle \text{do}(\alpha)\rangle p$ means that the agent does α and p holds at the end. This is taken to mean that all prerequisites for performing α are satisfied, and the agent performs it. That is, the agent has the opportunity to perform α. $\mathbf{A}(\alpha)$ is a separate operator that denotes that the agent can perform α. This is a primitive, and not formally defined.

van der Hoek *et al.* define $\mathbf{can}(\alpha, p)$ to mean the agent knows that the agent does α resulting in p and has the ability to do α (p. 5). Conversely, **cannot**(α, p) means that the agent knows he cannot perform α resulting in p or lacks the ability to do α.

van der Hoek *et al.* define action transformations as ways to manipulate one action (description) into another. They state a number of rules that preserve equivalence of the actions under transformation. The simplest example is that skip; α is equivalent to α. A more complex example involves unraveling a while statement by one loop, but we won't get into the details here. van der Hoek *et al.* then show that if α is equivalent to α', then (a) $[\text{do}(\alpha)]p \Leftrightarrow [\text{do}(\alpha')]p$, (b) $\phi \Leftrightarrow \phi|^{\alpha}_{\alpha'}$, and (c) $\mathbf{A}(\alpha) \Leftrightarrow \mathbf{A}(\alpha')$. As a result, the equivalence of actions satisfy the expected kinds of results. In other words, the definitions are well-formed model-theoretically.

In studying the ability to perform actions, this work generalizes over Moore's analysis of knowing how to perform a plan [22]. Although the idea of separating abilities is interesting, we find the specific definitions a little awkward. For example, $\mathbf{can}(\alpha, p)$ entails not only that the agent can do α, but in fact does α. Clearly, there can be lots of things that an agent can do that he does not. Know-how, in our view, does not entail performance.

4.5 Bringing It About

Segerberg developed a theory of *bringing it about*, which deals with how an agent brings about a particular condition. He proposes a logic of achievements, also in the framework of dynamic logic [19]. Segerberg bases his conceptual account on the notion of *routines*, which roughly are scripts of actions that agents might follow in order to bring things about, that is, to perform high-level actions [30, 31].

Segerberg defines an operator δ, which takes a condition and yields an action, namely, the action of bringing about that condition. Segerberg uses actions of the form δq as the primitive actions in his variant of dynamic logic—he has no

other atomic programs. In this respect, Segerberg's work is similar to strategic know-how. The δ operator is intuitively quite close to strategies of the form $\mathbf{do}(q)$. Segerberg defines the denotations of propositions quite in the manner of section 2.4 above. However, he defines the denotation of δp for a proposition p as the set of periods that can result from some program all of whose periods result in the given proposition p.

Segerberg's main intuitions about $[\![\delta p]\!]$ are that it is (a) *reliable* meaning that any of the periods in it will satisfy p, and (b) *maximal* if all periods corresponding to the different executions of a program at a state satisfy p, then all of those periods must be included (p. 329). Segerberg's definition is also similar to STIT in requiring choices to be made that are guaranteed to succeed. However, Segerberg's definitions are forward-looking and do not have the negative condition that is a part of STIT. In this way, Segerberg's definitions are closer to strategic know-how.

However, there are some important dissimilarities from strategic know-how as well. First, Segerberg acknowledges the importance of considering only periods that are "optimal" in some sense, such as being minimal in satisfying the given program. This is in fact done in [33]. However, to keep his approach simple, Segerberg does not make any assumption of minimality. Optimality of this sort is important in considering executions of strategies and in relating know-how with intentions, because it tells us just how far the current substrategy of a strategy will be executed before the rest of it kicks in.

Second, Segerberg does not consider basic actions at all in his framework, only actions that are derived from propositions. While his results are appealing in terms of their analysis of high-level actions, they lack a connection to the physical actions with which an agent may actually bring something about. In other words, it is not obvious where the semantics is grounded.

Third, Segerberg does not consider the knowledge of agents. Thus the effects of agents' knowledge on their choices cannot be considered. Such choices arise in Segerberg's logic as tests on conditions and in the present approach in conditional and iterative strategies.

The reader might consult Elgesem's paper for a critical review of Segerberg's research program [11].

4.6 Maintenance

Most of the work on know-how and related concepts of interest here has focused on the achievement of different conditions. Sometimes it is important not only to achieve conditions, but to maintain the conditions that hold. Maintenance in this style has not been intensively studied in the literature, but it has recently begun to draw some attention [9, 34].

The following discussion follows the presentation in [34]. Although it bears some resemblance to achievement, maintenance is not easily derived from achieve-

ment. For example, simple kinds of duality results between achievement and maintenance do not hold. An agent knows how to maintain a condition if he can continually and knowingly force it to be true, that is, if he can always perform an action that would counteract the potentially harmful actions of other agents. This entails that not only must the actions of other agents not cause any immediate damage, but the given agent should also ensure that they do not lead to a state where he will not be able to control the situation. A key difference with knowing how to achieve some condition is that achievement necessarily requires a bounded number of steps, whereas maintenance does not.

As the base case, we require that the agent know that the given condition holds in the present state. Further, to know how to maintain p, the agent must be able to respond to all eventualities that might cause p to become false. The agent must choose his action such that no combination of the other agents' actions can violate p. Not only must the agent's chosen action maintain p, it should also maintain his ability to maintain p further.

Following the style of the definition for knowing how to achieve, we state that an agent maintains p over an empty tree if he knows that p holds currently. He maintains p over a single action, a, if he knows that he can perform a in the given state and p holds where a begins and where it ends. An agent maintains p over a general tree if he maintains it over its initial action and then over some applicable subtree.

We define $[\![\tau]\!]_{t,p}$, the *maintenance denotation* of a tree τ, as the set of periods beginning at t over which p is maintained by τ. These are the periods over which the agent can knowingly select the right actions. $[\![\tau]\!]_{t,p} = \{\,\}$ means that p cannot be maintained using τ.

- $[\![\emptyset]\!]_{t,p} \stackrel{\text{def}}{=}$ (if $M \models_t \mathsf{K_t}p$, then $\{[S;t,t]\}$ else $\{\,\}$)
- $[\![a]\!]_{t,p} \stackrel{\text{def}}{=} \{[S;t,t'] : M \models_t \mathsf{K_t}p$ and $(\forall t_k : (t,t_k) \in \mathbf{K}(x) \Rightarrow (\exists S_k, t'_k :$ $[S_k;t_k,t'_k] \in [\![a]\!]$ and $[S;t,t'] \in [\![a]\!]$ and $(\forall S_k, t'_k : [S_k;t_k,t'_k] \in [\![a]\!] \Rightarrow$ and $M \models_{t'_k} \mathsf{K_t}p)))\}$
- $[\![\langle a;\tau_1,\ldots,\tau_m\rangle]\!]_{t,p} \stackrel{\text{def}}{=} \{[S;t,t''] : (\forall t_k : (t,t_k) \in \mathbf{K}(x) \Rightarrow (\exists S_k, t'_k :$ $[S_k;t_k,t'_k] \in [\![a]\!]_{t_k,p}$ and $(\forall S_k, t'_k : [S_k;t_k,t'_k] \in [\![a]\!]_{t_k,p} \Rightarrow (\exists t''_k, j :$ $[S_k;t'_k,t''_k] \in [\![\tau_j]\!]_{t'_k,p}))))$ and $(\exists t', t'', j : [S;t,t'] \in [\![a]\!]_{t,p}$ and $[S;t',t''] \in$ $[\![\tau_j]\!]_{t',p})\}$

In other words, the agent maintains p over $[S;t,t'']$ iff the agent knows at t that he will maintain p over a, that is, till t', and then maintain p till t'' using some subtree.

An agent maintains p to depth i if there is a tree of depth i over which he maintains p. An agent maintains p if he can maintain it to all depths.

SEM-30. $M \models_t \mathsf{K_m}^i p$ iff $(\exists \tau : \text{depth}(\tau) = i$ and $[\![\tau]\!]_{t,p} \neq \{\,\})$

SEM-31. $M \models_t \mathsf{K_m}p$ iff $(\forall i : M \models_t \mathsf{K_m}^i p)$

Now we present a recursive characterization of maintenance. This characterization resembles the one given in section 3.2, and is also used in [34], where we develop an approach based on the mu-calculus for computing know-how and maintenance.

Lemma 2 $\mathsf{K}_{\mathsf{t}}p \wedge (\bigvee \alpha : \mathsf{K}_{\mathsf{t}}(\exists\langle\alpha\rangle\mathsf{true} \wedge \forall[\alpha]\mathsf{K}_{\mathsf{m}}p)) \Leftrightarrow \mathsf{K}_{\mathsf{m}}p$

5 CONCLUSIONS

We discussed a number of variants of the broad concept of know-how that has been studied in the literature on theoretical aspects of rational agency. These variants fill an essential need in the theories of agency that relate the intentions and knowledge of agents with their actions.

To summarize briefly, our initial approach, Brown, Chellas, Belnap & Perloff do not use dynamic logic, whereas our latter approach (strategic know-how), van der Hoek *et al.*, and Segerberg do use dynamic logic. Branching time is explicit in some and implicit in the other approaches, but is a key unifying theme. While there remain important differences, it is remarkable that the different approaches, although developed independently, share many important intuitions. We take this as a promising sign that this subarea of rational agency is maturing.

There are important problems that require additional study. One problem is to sensitize the know-how to the real-time aspects of decision-making in practical settings, both in terms of being able to achieve the desired conditions in bounded time, and to determine the appropriate actions with bounded reasoning. A step in this direction would be develop computational techniques for know-how that are related to planning.

Another challenge is to give a probabilistic account of know-how, which can give a more realistic treatment of the notion of reliability. We believe that such an account will preserve many of the intuitions of the qualitative approaches discussed above.

Another set of issues is opened up when we turn our attention to multiagent settings. If the agents can cooperate with each other, they can together achieve more than any of them can individually. There has been some work on this problem, for example, [32], but additional research is needed to relate the know-how of agents with the structures of the organizations in which they exist.

The foregoing should have made it clear that there are considerable overlaps and similarities between approaches to rational agency in computer science and philosophy. There are also some important differences. Unfortunately, the relationships are not always as well understood as they ought to be.

After all is said and done, have we understood know-how as that term is commonly used, for example, in the quotation by Hewitt given at the beginning of this paper? In its entirety, we believe, not. However, good progress has been made in

this small community of computer scientists and philosophers. We encourage the reader to participate in the program of research described above. Its challenges remain important, and provide a fertile ground on which to explore the key concepts of both philosophy and computer science.

ACKNOWLEDGMENTS

This paper being expository in nature is based on previous work by us and others. This work is supported by the NCSU College of Engineering, the National Science Foundation under grants IRI-9529179 and IRI-9624425, and IBM corporation.

REFERENCES

[1] Philip Agre and David Chapman. Pengi: An implementation of a theory of activity. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 268–272, 1987.
[2] Nuel Belnap and Michael Perloff. Seeing to it that: A canonical form for agentives. *Theoria*, 54(3):175–199, 1988.
[3] Myles Brand. *Intending and Acting*. MIT Press, Cambridge, MA, 1984.
[4] Michael E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA, 1987.
[5] Mark A. Brown. On the logic of ability. *Journal of Philosophical Logic*, 17:1–26, 1988.
[6] Mark A. Brown. Action and ability. *Journal of Philosophical Logic*, 19:95–114, 1990.
[7] Brian F. Chellas. *Modal Logic*. Cambridge University Press, New York, 1980.
[8] Brian F. Chellas. Time and modality in the logic of agency. *Studia Logica*, 51(3/4):485–517, 1992.
[9] Giuseppe De Giacomo and Xiao Jun Chen. Reasoning about nondeterministic and concurrent actions: A process algebra approach. In *Proceedings of the National Conference on Artificial Intelligence*, pages 658–663, 1996.
[10] Yves Demazeau and Jean-Pierre Müller, editors. *Decentralized Artificial Intelligence, Volume 2*. Elsevier/North-Holland, Amsterdam, 1991.
[11] Dag Elgesem. Intentions, actions and routines: A problem in Krister Segerberg's theory of actions. *Synthese*, 85:153–177, 1990.
[12] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. North-Holland, Amsterdam, 1990.
[13] Ronald Fagin and Joseph Y. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34:39–76, 1988.
[14] Jaakko Hintikka. *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Cornell University Press, Ithaca, 1962.
[15] Jaakko Hintikka. Alternative constructions in terms of the basic epistemological attitudes. In *[23]*. 1972.
[16] Michael N. Huhns and Munindar P. Singh, editors. *Readings in Agents*. Morgan Kaufmann, San Francisco, 1997.
[17] Manfred Kochen and Eugene Galanter. The acquisition and utilization of information in problem solving and thinking. *Information and Control*, 1:267–288, 1958.
[18] Kurt Konolige. *A Deduction Model of Belief*. Morgan Kaufmann, 1986.
[19] Dexter Kozen and Jerzy Tiurzyn. Logics of program. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 789–840. North-Holland, Amsterdam, 1990.
[20] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*. American Elsevier, 1969. Page numbers from the version reprinted in [40].
[21] George A. Miller, Eugene Galanter, and Karl Pribram. *Plans and the Structure of Behavior*. Henry Holt, New York, 1960.

[22] 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, Norwood, NJ, 1984.
[23] Raymond E. Olson and Anthony M. Paul, editors. *Contemporary Philosophy in Scandinavia*. Johns Hopkins Press, Baltimore, 1972.
[24] Michael Perloff. *Stit* and the language of agency. *Synthese*, 86(3):379–408, 1991.
[25] Arthur N. Prior. *Time and Modality*. Clarendon Press, Oxford, 1957.
[26] Arthur N. Prior. *Past, Present and Future*. Clarendon Press, Oxford, 1967.
[27] Anand S. Rao and Michael P. Georgeff. Modeling rational agents within a BDI-architecture. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, pages 473–484, 1991. Reprinted in [16].
[28] Gilbert Ryle. *The Concept of Mind*. Hutchinson's University Library, London, 1949.
[29] John R. Searle. *Intentionality: An Essay in the Philosophy of Mind*. Cambridge University Press, Cambridge, UK, 1983.
[30] Krister Segerberg. Routines. *Synthese*, 65:185–210, 1985.
[31] Krister Segerberg. Bringing it about. *Journal of Philosophical Logic*, 18:327–347, 1989.
[32] Munindar P. Singh. Group ability and structure. In *[10]*, pages 127–145. 1991.
[33] Munindar P. Singh. *Multiagent Systems: A Theoretical Framework for Intentions, Know-How, and Communications*. Springer-Verlag, Heidelberg, 1994.
[34] Munindar P. Singh. Applying the mu-calculus in planning and reasoning about action. *Journal of Logic and Computation*, 1998. In press.
[35] Munindar P. Singh and Nicholas M. Asher. A logic of intentions and beliefs. *Journal of Philosophical Logic*, 22(5):513–544, October 1993.
[36] J. F. A. K. van Benthem. Temporal logic. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3. Oxford University Press, Oxford, 1990.
[37] Johan F. A. K. van Benthem. *The Logic of Time: A Model-Theoretic Investigation into the Varieties of Temporal Ontology and Temporal Discourse*, volume 152 of *Synthese Library*. Kluwer, Dordrecht, Holland, 2nd edition, 1991.
[38] Wiebe van der Hoek, Bernd van Linder, and John-Jules Ch. Meyer. A logic of capabilities. TR IR-330, Vrije Universiteit, Amsterdam, 1993.
[39] Georg Henrik von Wright. *Norm and Action*. Routledge & Kegan Paul, London, 1963.
[40] Bonnie L. Webber and Nils J. Nilsson, editors. *Readings in Artificial Intelligence*. Morgan Kaufmann, 1981.
[41] Eric Werner. A unified view of information, intention and ability. In *[10]*, pages 109–125, 1991.