Chapter 4

Know-How

Given my goal of developing a framework for multiagent systems, it is only natural that I should attempt to formalize the concept of know-how. There are several reasons for this. In light of the formal framework developed in Chapter 2 and used to formalize intentions in Chapter 3, one might ask under what conditions an agent with limited control would be guaranteed to succeed. Such conditions are technically studied in Chapter 5, but suffice it to note here that know-how is an important component of those conditions. An agent cannot be guaranteed to succeed with his intentions if he lacks the know-how to achieve them.

Know-How can also be used in specifying complex systems succinctly for the purposes of designing them or analyzing their behavior. We can require that a given agent under certain conditions have certain know-how. For example, a robot that is designed for helping a handicapped person should, when called upstairs, know how to climb the stairs. This requirement might, for a particular design, reduce to the requirement that the robot's batteries be fully charged, that its load be light, and that it have sufficiently extendible legs. These requirements can be further analyzed in designing the robot, e.g., we might decide that the robot must recharge itself every hour and that it carry only small packages on its stair-climbing missions. But, once the know-how of the robot is established, it can also be used in different circumstances as well, e.g., when the robot has to go upstairs on its own initiative to deliver an unexpected express mail package.

Requirements of the kind described above arise naturally when one gives a formal semantics to the communications between the agents in a multiagent system. This proposal is elaborated in Chapter 6; an example of its use was given in section 1.1. Such a semantics for communications provides a mechanism by which constraints on the interactions among agents, e.g., that directives to come upstairs are satisfied, are reduced to constraints on the design of individual agents, e.g., that the robot assistant has the know-how and the intention to come upstairs.

I base the proposed theory of know-how on the framework of actions and time that was developed in Chapter 2. In section 4.1, I discuss know-how and present some intuitions about how we should proceed. In section 4.2, I define and axiomatize ability for the case of purely reactive agents; in section 4.3, I define and axiomatize it relative to strategies. In section 4.4, I state and prove some theorems about the logical properties of ability and its interactions with the modalities of time and action. In section 4.5, I define and axiomatize know-how for the case of purely reactive agents; in section 4.6, I define and axiomatize know-how relative to strategies. These definitions involve extensions to dynamic logic [Kozen & Tiurzyn, 1990].

4.1 Intuitive Considerations on Know-How

I shall take it as a starting point that intelligence is intimately tied to action. It is an agent's ability or potential to take effective action, and the skills he exhibits in doing things that make us attribute intelligence to him. For this reason, a useful conception of knowledge for our purposes is *know-how*, i.e., the knowledge of how to act, or the knowledge of skills. Thus a theory of know-how is needed that gives a definition with an appropriate formal semantics and logic. This would allow us to capture our intuitions about know-how straightforwardly and to use that concept directly whenever we need to.

Traditionally, however, preeminence is given to know-that, or the knowledge of facts. And know-how is reduced to know-that. While, no doubt, there are profound connections between know-how and know-that, know-how cannot be trivially reduced to know-that. Such a reduction buries intuitions about know-how and its logical properties within those of know-that. Also, this reduction, which was designed for classical planning agents, is inappropriate in general. Indeed, it is inappropriate even for planning agents who perform many of their actions reactively. This is an important class of agents in current theory and practice [Mitchell, 1990; Spector & Hendler, 1991]. I present a formal theory of know-how that applies to a wide-range of agents, and especially those who plan *and* react.

Newell defines knowledge as "[w]hatever can be ascribed to an agent, such that its behavior can be computed according to the principle of rationality" [Newell, 1982, p. 105]. He sees this definition as corresponding to the common scientific practice in AI (p. 125). I agree. However, I submit that unless know-how is also considered, ascriptions of beliefs or knowledge, and intentions must necessarily seem contrived. For example, we cannot conclude from an agent's not opening his umbrella that either he does not believe that it is raining or intends to get wet. It could just as well be that he does not know how to open his umbrella. The point of this example is simply that one cannot assume, as is traditionally done, that know-that is theoretically more fundamental than know-how, and that the latter should be reduced to it. An independent study of know-how would be beneficial in obtaining a better understanding important aspects of intelligence and rationality.

The philosopher Ryle was one of the early proponents of the distinction between know-how and know-that [Ryle, 1949]. His primary motivation, however, was to debunk "the dogma of the Ghost in the Machine," which is Descartes' doctrine of the separation and independence of minds from bodies (p. 15). Ryle's argument runs roughly as follows: intelligence is associated more with know-how than know-that; know-how involves bodies; therefore, intelligence (a quality of minds) is not independent of bodies. He rejects the view that a performance of an action is preceded or accompanied by explicit consideration of different rules (p. 29). He also rejects the claim that every component of an intelligent action must be planned (p. 31). Thus the now popular view that intelligent agents must have reactive components is in agreement with, and indeed was anticipated by, Ryle [Mitchell, 1990; Spector & Hendler, 1991].

I shall show that the intuitions motivated above are naturally captured in the proposed framework, and it is good to have some philosophical support for them. However, Ryle provides no insights into how know-how may be formalized and what technical properties it must have.

4.1.1 Traditional Theories of Action

Moore's work is among the most well-known theories of knowledge and action [1984]. Moore's focus is not on know-how and, when he discusses it, it is as the know-how to execute a given action description. But he also considers the know-how of achieving a certain condition relative to an action description (p. 347). An agent knows how to achieve p by doing an action (description), P, iff the agent knows that P achieves p, and that he can execute P. He can execute P iff he can identify each of the actions in P, i.e., if he knows what their rigid designators are. Thus, according to this definition, the agent knows how to achieve p only if he knows that for some plan P, that it will yield p, and that he will be able to execute it.

Moore's work has been extended by Morgenstern to allow an agent's plan to include actions such as asking others for information [Morgenstern, 1987]. However, Morgenstern does not define know-how either, just *know-how-to-perform*, which corresponds to *can execute* in Moore's theory. Roughly, an agent knows-how-to-perform a plan iff he knows all the required rigid designators occurring in that plan. Morgenstern's main aim, like Moore's, is to give the know-that requirements for the execution of plans; she does not address know-how *per se.*

The approaches described above incorporate useful intuitions about the relationship between know-how and know-that. They provide an analysis of the knowledge requirements of plans. Unfortunately, however, they also embody some restrictive assumptions about actions. I hope to relax some of the assumptions of these approaches, while retaining their useful components.

Many of these assumptions were considered in Chapter 2. In particular, I have generalized the underlying model of actions by allowing concurrent and asynchronous ones. I have also explicitly considered time and related it to actions. Furthermore, I admit a reactive layer of the architecture. Strategies, which were defined in section 2.5, correspond to plans in traditional theories. They do not directly involve basic actions, but instead are macros over them.

4.1.2 The Proposed Definition

I now motivate a definition of know-how that, I submit, captures the important aspects of our pretheoretic intuitions about it. As argued in section 4.1, even if the traditional theories could not be improved upon, it would be useful to have an independent treatment of know-how. But, relaxing some of the traditional assumptions provides us with a nice opportunity to consider the definition anyway.

The notion of know-how has to do with the ability to perform actions in a changing world. The agent is *competing* with the world, as it were. However, unlike in board games, the simplistic notion of turn-taking does not apply here. Of course, the world includes other agents. Thus, while the environment need not be opposed to the agent, it may be so. The environment and the agent must act concurrently: although the agent may wait for some time, the environment will not wait for him. As explained in Chapter 2, the outcome of an agent's actions depends on other events that take place in the world. These events may include the actions of other agents. For example, if I slide a mug over a table, it will simply move to its new location, unless someone sticks a hand in the way, in which case its contents may spill. By performing his actions, an agent exercises limited control over what transpires in the world. An agent must have some control over the world to ever know how to do anything; however, in general he cannot have perfect control over it. This is because events can always occur that potentially influence the outcome of any of the agent's actions.

The proposed definition of know-how is as follows. An agent, x, knows how to achieve p, if he is able to bring about the conditions for p through his actions. The world may change rapidly and unpredictably, but x is able to *force* it into an appropriate state. He has only limited knowledge of his rapidly changing environment and too little time to decide on an optimal course of actions. He can succeed only if he has the required physical resources and is able to choose them correctly.

This definition can be formalized both in terms of basic actions and strategies. Therefore, it contrasts both with situated theories of know-that [Rosenschein, 1985], and informal, and exclusively reactive, theories of action [Agre & Chapman, 1987]. Strategies were defined in section 2.5 as abstract descriptions of an agent's behavior. In section 4.3, I give a strategic definition of ability and in section 4.6, a strategic definition of know-how.

I first formalize the concept of ability, which depends solely on the actions that an agent can do and the effects of those actions, and ignores the agent's knowledge. This concept is simpler than know-how; it is a useful first step to formalize it and examine its properties. It is also useful in its own right.

4.2 Reactive Ability

As will soon become clear, it is useful to define ability relative to a *tree* of actions. 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. Such a fragment is diagramed in Figure 2.1. Thus a tree includes *some* of the possible actions of the given agent. For technical simplicity, I shall assume that when a tree is defined at a given moment, it is derived from a fragment of the model rooted at that moment. Thus it would automatically be executable along some scenarios at that moment.

An agent is able to achieve p relative to a tree of actions, iff on *all* scenarios where he performs the radix of the tree, either p occurs or the agent can choose one of the subtrees of the tree to pursue, and thereby achieve p. The definition does not require p to occur on all scenarios where the radix is done, only on the scenarios corresponding to the selected subtree. The agent gets to choose one subtree after the radix has been done. This is to allow the choice to depend on how the agent's environment has evolved. However, modulo this choice, the agent must achieve p by forcing it to occur. For example, a sailor's tree might call on him to head straight for the equator and then, depending on the wind, to adjust his sails accordingly. He would be able to arrive at his destination if, on his arrival at the equator, there is a setting for his sails with which he can succeed.

It is important to note that the tree need not be explicitly symbolically represented by the agent. The tree simply encodes the selection function the agent uses in picking out his actions at each stage. When a tree is finite in depth, it puts a bound on the number of actions that an agent may have to perform to achieve something. Since, intuitively, for an agent to be able to achieve some proposition, we expect him to be able to achieve the required proposition in finite time, this restriction is imposed here.

Since trees encode the choices that an agent may make while perform actions, there is no loss of generality in requiring that the different nonempty subtrees of a tree have different radices, i.e., begin with different actions. Given a tree in which this is not the case, we can easily transform it into one in which this restriction is satisfied. Let $\tau' = \langle a; \tau'_1, \ldots, \tau'_m \rangle$ and $\tau'' = \langle a; \tau''_1, \ldots, \tau''_n \rangle$. Then define a new tree, $\tau = \langle a; \tau_1, \ldots, \tau_k \rangle$, where $\{\tau_1, \ldots, \tau_k\} = \{\tau'_1, \ldots, \tau'_m\} \cup \{\tau''_1, \ldots, \tau''_n\}$. Thus one can replace τ' and τ'' by τ in the original tree. τ encodes precisely the same choices as τ' and τ'' do together. This procedure can be generalized to any set of subtrees with a given radix. Intuitively, this results in a better tree than before. This is because the choices of the agent are required to be made one action at a time. The agent is not called upon to look ahead and predict the state in which his current action might lead him.

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

TREE-1. $\emptyset \in \Upsilon$

TREE-2. $a \in \mathcal{B}$ implies that $a \in \Upsilon$

TREE-3. $\tau_1, \ldots, \tau_m \in \Upsilon$, $\tau_1, \ldots, \tau_m$ have different radices, and $a \in \mathcal{B}$ implies that $\langle a; \tau_1, \ldots, \tau_m \rangle \in \Upsilon$

In the sequel, I shall always assume that τ is of one of the forms above.

The formal language of this chapter, $\mathcal{L}^h$, is an extension of $\mathcal{L}$. At this point, we require $\mathcal{L}^h$ to have the following operators. The operator $\{\!\!\{\ \}\!\!\}$ denotes ability relative to trees and the operator $\langle\!\vert\,\vert\!\rangle$ ability relative to strategies (see section 4.3). The operators K_{rab} and K_{sab} are, respectively, the reactive and strategic versions of ability. $\mathcal{L}^h_s$ is used later in this chapter.

SYN-19. All the rules for $\mathcal{L}$, with $\mathcal{L}^h$ substituted for $\mathcal{L}$

SYN-20. All the rules for $\mathcal{L}_s$, with $\mathcal{L}^h_s$ substituted for $\mathcal{L}_s$

SYN-21. All the rules for $\mathcal{L}_y$, with $\mathcal{L}^h_y$ substituted for $\mathcal{L}_y$

SYN-22. $p \in \mathcal{L}^h_s$ and $x \in \mathcal{A}$ implies that $(x\mathsf{K}_{rab}p), (x\mathsf{K}_{sab}p) \in \mathcal{L}^h$

SYN-23. $p \in \mathcal{L}^h_s$, $Y \in \mathcal{L}^h_y$, and $x \in \mathcal{A}$ implies that $(x\langle\!\vert Y\vert\!\rangle p) \in \mathcal{L}^h$

SYN-24. $\tau \in \Upsilon$, $x \in \mathcal{A}$, and $p \in \mathcal{L}^h$ implies that $x\{\!\vert\tau\vert\!\}p \in \mathcal{L}^h$

Other extensions will be described as needed.

Now $\mathcal{L}^h$ is powerful enough to express the definition of ability via the auxiliary operator, $\{\!\vert\;\vert\!\}$. Intuitively, $x\{\!\vert\tau\vert\!\}p$ is true iff the agent, x, can use τ as a selection function and, thereby, force p to become true. For the empty tree, p must already be true. For a nonempty tree, assuming p is not already true, the agent first does the radix of the tree. Depending on the other events that take place then, this action progresses along some scenario. If that is the only action of the tree, p may occur at any moment while it is being done. Otherwise, one of the subtrees is chosen in the state where the initial action ends. From there, the process iterates.

SEM-21. $M \models_t x\{\!\vert\emptyset\vert\!\}p$ iff $M \models_t p$

SEM-22. $M \models_t x\{\!\vert a\vert\!\}p$ iff $M \models_t (\mathsf{E}x\langle a\rangle\mathsf{true} \wedge \mathsf{A}x[a]p)$

SEM-23. $M \models_t x\{\!\vert\langle a;\tau_1,\ldots,\tau_m\rangle\vert\!\}p$ iff $(\exists S,t' : [S;t,t'] \in [\![a]\!]^x)$ and $(\forall S,t' : [S;t,t'] \in [\![a]\!]^x \Rightarrow (\exists\tau \in \{\tau_1,\ldots,\tau_m\} : M \models_{t'} x\{\!\vert\tau\vert\!\}p))$

Figure 4.1 shows how the above definition applies in our formal model. Assume that $\neg p$ and $\neg q$ hold everywhere other than as shown. Let us consider the agent whose actions are written first in the figure and see whether he has the ability to achieve p. Since $\neg p$ holds at t_0, the agent will have to do some actions to achieve p. Clearly, action b would not do him much good, since p never occurs on any scenario on which b is performed. If the agent does action a and the other agent does action d, then the state of the world is t_2, where p holds. Therefore, the agent is trivially able to achieve p at t_2. Thus at that moment, the agent would be done. However, action a could just as well lead

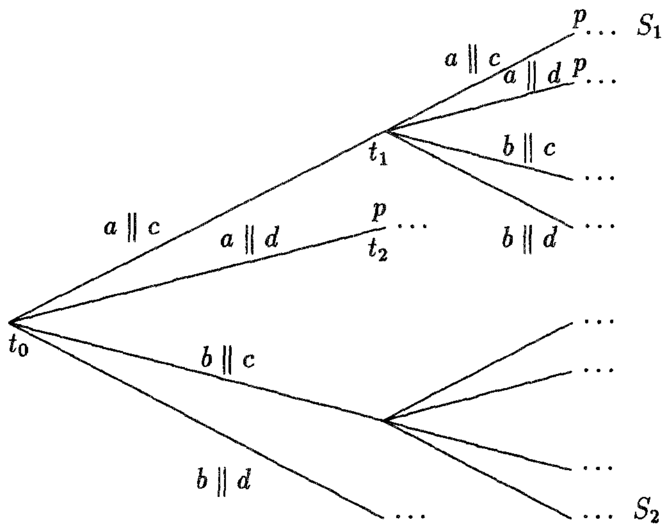

Figure 4.1: Ability

to moment t_1 (if the other agent does action b), where p does not hold. But fortunately, at t_1, if the agent does action a again, p is guaranteed. In other words, at t_1, the agent is able to achieve p. Thus the agent is able to achieve p even at t_0, since action a always leads to a state where he is able to achieve p. Thus the appropriate tree for this agent is $\langle a; a, \emptyset \rangle$.

Now reactive ability may be defined as follows:

SEM-24. $M \models_t x\mathsf{K}_{rab}p$ iff $(\exists \tau : M \models_t x\{\!|\tau|\!\}p)$

$x\mathsf{K}_{rab}p$ means that agent x can force p to occur by performing actions in whose resultant states he can perform further actions, and so on, until p occurs. $\mathsf{K}_{rab}p$ is vacuously true if p holds in the given state. For example, if p stands for "open(door)," then $\mathsf{K}_{rab}p$ holds if the door is already open. $\mathsf{K}_{rab}p$ also holds if p is inevitable, i.e., if p will eventually occur no matter what happens. For example, let p be "returns(Halley's comet)." Assuming it is inevitable that Halley's comet will return, $\mathsf{K}_{rab}p$ holds. I shall return to this point in section 4.4.

The above is a definition of raw ability. It attributes ability to an agent, even if he may not be able to make the right choices, just as long as he has the right set of basic actions available. The choices that an agent makes depend on his beliefs. Indeed, incorporating beliefs into the picture, as I attempt in section 4.5, yields a general theory of know-how itself.

Now I present an axiomatization for the definition of K_{rab} given above and a proof of its soundness and completeness. As for the case of intentions, discussed in section 3.2.2, axiom AX-AB-REACT-3 below is a way of relativizing this axiomatization to that of the underlying logic. Note that $\mathsf{E}\langle a\rangle\mathsf{true}$ means that a is a basic action of the agent at the given moment and is performed by him on at least one scenario.

AX-AB-REACT-1. $p \rightarrow x\mathsf{K}_{rab}p$

AX-AB-REACT-2. $(\bigvee a : \mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a](x\mathsf{K}_{rab}p)) \rightarrow x\mathsf{K}_{rab}p$

AX-AB-REACT-3. All substitution instances of the validities of the underlying logic

Theorem 4.1 Axioms AX-AB-REACT-1 through AX-AB-REACT-3 yield a sound and complete axiomatization for K_{rab}.

Proof.

Construct a branching-time model, M. The moments of M are notated as t and are maximally consistent sets of formulae that contain all the substitution instances of the validities of the underlying logic. The other components of the model, especially, $<$, $\mathbf{B}$, $\mathbf{R}$, and $[\![\,]\!]$ are constrained by the formulae that are true at the different moments. Furthermore, these sets are closed under the above two axioms for K_{rab}. We can ignore the agent symbol in the following discussion.

Soundness: For axiom AX-AB-REACT-1 above, soundness is trivial from the definition of $\{\!\!\{\emptyset\}\!\!\}p$. For axiom AX-AB-REACT-2, let $(\bigvee a : \mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K}_{rab}p)$ hold at t. Then $(\exists S, t' : [S; t, t'] \in [\![a]\!])$ and $(\forall S : (\forall t' \in S : [S; t, t'] \in [\![a]\!] \Rightarrow (\exists t'' : t < t'' \leq t'$ and $M \models_{t''} \mathsf{K}_{rab}p)))$. At each t'' in the preceding expression, $(\exists \tau' : M \models_{t''} \{\!\!\{\tau'\}\!\!\}p)$. Soundness may be shown by exhibiting a tree, τ, such that at moment t, $\{\!\!\{\tau\}\!\!\}p$ holds. Clearly, action a must be the radix of τ. To construct it, note that if $t'' = t'$, τ' must be a subtree of τ. And, if $t'' < t'$, then action a is being performed at moment t''. Thus the radix of τ' is a. But, in the definition of $\{\!\!\{\tau\}\!\!\}p$, we need to look at the subtrees of τ only at the moments where its radix, i.e., a, has ended. With this motivation, define $T_1 = \{\tau' | \tau'$ is the tree used to make $\mathsf{K}_{ab}p$ true at $t'\}$. Define $T_2 = \bigcup\{\tau'' | \tau''$ is the tree used to make $\mathsf{K}_{ab}p$ true at t'', where $t'' < t'\}$. Now define τ as $\langle a; \tau_1, \ldots, \tau_m\rangle$, where $\{\tau_1, \ldots, \tau_m\} = T_1 \cup T_2$. Thus $M \models_t \{\!\!\{\tau\}\!\!\}p$, or $M \models_t \mathsf{K}_{rab}p$. Hence, axiom AX-AB-REACT-2 is sound.

Completeness: The proof is by induction on the structure of formulae. Only the case of formulae of the form $\mathsf{K}_{rab}p$ is described below. Completeness means that $M \models_t \mathsf{K}_{rab}p$ entails $\mathsf{K}_{rab}p \in t$. $M \models_t \mathsf{K}_{rab}p$ iff $(\exists \tau : M \models_t \{\tau\}p)$. This proof is by induction inside the induction on the structure of formulae. This induction is on the structure of trees with which a formula of the form $\mathsf{K}_{rab}p$ is satisfied. One base case is the empty tree $\emptyset$. And $M \models_t \{\emptyset\}p$ iff $M \models_t p$. By AX-AB-REACT-3, $p \in t$. By axiom AX-AB-REACT-1 above, $\mathsf{K}_{rab}p \in t$, as desired. The other base case is for single-action trees: $M \models_t \{a\}p$ iff $(\exists S, t' : [S; t, t'] \in [\![a]\!])$ and $(\forall S : (\forall t' \in S : [S; t, t'] \in [\![a]\!] \Rightarrow (\exists t'' : t < t'' \leq t'$ and $M \models_{t''} p)))$. But, by axiom AX-AB-REACT-3, we have $(\bigvee a : \mathsf{E}\langle a \rangle \mathsf{true} \wedge \mathsf{A}[a]p)$. Thus by axiom AX-AB-REACT-2, we have $\mathsf{K}_{rab}p$.

For the inductive case, $M \models_t \{(a; \tau_1, \ldots, \tau_m)\}p$ iff $(\exists S, t' : [S; t, t'] \in [\![a]\!])$ and $(\forall S : (\forall t' \in S : [S; t, t'] \in [\![a]\!] \Rightarrow (\exists i : 1 \leq i \leq m$ and $M \models_{t'} \{\tau_i\}\mathsf{K}_{rab}p)))$. But since τ_i is a subtree of τ, we can use the inductive hypothesis on trees to show that this is equivalent to $(\exists t' : [S; t, t'] \in [\![a]\!])$ and $(\forall S : (\forall t' \in S : [S; t, t'] \in [\![a]\!] \Rightarrow M \models_{t'} \mathsf{K}_{rab}p))$. But it is easy to see that $(\exists t' : [S; t, t'] \in [\![a]\!])$ iff $\mathsf{E}\langle a \rangle \mathsf{true}$. And, using the definition of $[\,]$, we see that the second conjunct holds only if $\mathsf{A}[a]\mathsf{K}_{rab}p$. Thus $M \models_t \{\tau\}p$ only if $M \models_t (\bigvee b : \mathsf{E}\langle b \rangle \mathsf{true} \wedge \mathsf{A}[b]\mathsf{K}_{rab}p)$. But, by axiom AX-AB-REACT-3, $(\bigvee b : \mathsf{E}\langle b \rangle \mathsf{true} \wedge \mathsf{A}[b]\mathsf{K}_{rab}p) \in t$. Thus by axiom AX-AB-REACT-2, $\mathsf{K}_{rab}p \in t$. Hence we have completeness. □

4.3 Strategic Ability

The definition of ability given above involves trees of basic actions of a given agent. Strategies were introduced in section 2.5 as abstractions over agents' basic actions. I now define the ability of an agent relative to a strategy. Defining ability and know-how relative to a strategy not only shows us how abstractions over basic actions relate to the concepts considered, but also help make the connection to intentions clearer.

The operator $\langle\!\{\,\}\!\rangle$ is defined as follows. $x\langle\!\{Y\}\!\rangle p$ means that the agent, x, is able to perform all the substrategies of Y that he may need to perform, and furthermore that he can perform them in such a way as to force the world to make p true. Basically, this allows us to have the ability of an agent to achieve the conditions in different substrategies combined to yield the ability to achieve some composite condition. This is especially important from the point of view of designers and analyzers of agents, since they can take advantage of the abstraction provided by strategies to consider the ability of an agent in terms of his ability to achieve simpler conditions. Not just intuitively, but even formally, the ability to achieve simpler conditions as used here is purely reactive, as defined in section 4.2 (see Theorem 4.3 below for the technical justification).

In order to define $x\langle\!|Y|\!\rangle p$ formally, I need the auxiliary concept of the *ability-intension* of a tree. The ability-intension of a tree τ, for an agent x, and a strategy Y, is notated as $[\![\tau]\!]^x_Y$. This is the set of periods on which the given agent is able to achieve the given strategy, by following the given tree. Precisely those periods are included on which the success of the given strategy is assured or forced, and not fortuitous. The ability-intension of trees needs to be defined only for the $\downarrow$ of strategies, which are always of the forms, **skip** or **do**(q). As usual, the agent symbol is omitted when obvious from the context. Formally, we have

1. The empty strategy, **skip**, is achieved by the empty tree.
   $[S; t, t'] \in [\![\emptyset]\!]_{\mathbf{skip}}$ iff $t = t'$.
2. τ follows **do**(q) iff the agent can achieve q in doing τ.
   $[S; t, t'] \in [\![\tau]\!]_{\mathbf{do}(q)}$ iff
   (a) $\tau = \emptyset$ and $t = t'$ and $M \models_t q$;
   (b) $\tau = a$ and $M \models_{t'} q$ 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 $M \not\models_{t_2} q))$; or
   (c) $\tau = \langle a; \tau_1, \ldots, \tau_m\rangle$ and $M \models_{t'} q$ and $M \models_t \langle\!|\tau|\!\rangle q$ and $(\exists t_1 : [S; t, t_1] \in [\![a]\!]$ and $(\exists t_2, i : 1 \leq i \leq m$ 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 $M \not\models_{t_3} q)$

An important feature of this definition is that a period, $[S; t, t']$, is included in the ability-intension of **do**(q), only if no other period is available that begins at or before t and ends sooner than t'. This captures the intuition that, in order to achieve q, an agent has to act only till its first occurrence. Another consequence of this definition is that, for some strategies, the tree $\langle a; \emptyset\rangle$ may have a different ability-intension than the tree a. The former requires the relevant condition to hold when a ends, whereas the latter allows it to hold any time during the execution of a. This is not a problem, since, whenever the tree $\langle a; \emptyset\rangle$ is available to an agent, so is the tree a.

The ability-intension of a tree for a strategy of the form **do**(q) helps relate the notion of strategic ability as defined in this section to the notion of reactive ability as defined in section 4.2. In particular, a period is in the ability-intension of a tree relative to **do**(q) iff the tree can be used as a selection function by the agent to force q. Thus, we have the following result.

Lemma 4.2 Let $\tau = \langle a; \tau_1, \ldots, \tau_m \rangle$. Then, $M \models_t \{\tau\}q$ iff $(\exists S, t' : [S; t, t'] \in [\![\tau]\!]_{\mathbf{do}(q)})$

Proof.

The cases of the empty tree and single-action trees are trivial. The inductive case in the right to left direction is also trivial. In the left to right direction, let t_0 be the moment at which a is done. Then, by the definition of $\{\,\}$, $(\exists i : 1 \leq i \leq m$ and $M \models_t \{\langle \tau_i \rangle\}q)$. By the inductive hypothesis, $(\exists S, t' : [S; t_0, t'] \in [\![\tau_i]\!]_{\mathbf{do}(q)})$. Therefore, $(\exists S, t' : [S; t, t'] \in [\![\langle a; \tau_1, \ldots, \tau_m \rangle]\!]_{\mathbf{do}(q)})$ □

Using the definition of ability-intension, the satisfaction conditions for $(\!|\,|\!)$ can be given as below.

SEM-25. $M \models_t x(\!|\mathbf{skip}|\!)p$ iff $M \models_t p$

SEM-26. $M \models_t x(\!|Y|\!)p$ iff $(\exists \tau : (\exists S, t' : [S; t, t'] \in [\![\tau]\!]^x_{\downarrow_t Y})$ and $(\forall S, t' : [S; t, t'] \in [\![\tau]\!]^x_{\downarrow_t Y} \Rightarrow M \models_{t'} x(\!|\uparrow_t Y|\!)p))$

This definition says that an agent is able to achieve p relative to strategy Y iff there is a tree of actions for him such that he can achieve the current part of his strategy by following that tree, and that on all scenarios where he does so, he either achieves p or can continue with the rest of his strategy (and is able to achieve p relative to that). This definition allows the tree for the current part of Y to overlap with the tree for the rest of it. This is desirable, since we expect the strategy $\mathbf{do}(q);\mathbf{do}(q)$ to behave the same as $\mathbf{do}(q)$, which it would not if a different action were required for each substrategy.

Now $\mathsf{K}_{sab}p$ may be defined as given below.

SEM-27. $M \models_t x\mathsf{K}_{sab}p$ iff $(\exists Y : M \models_t x(\!|Y|\!)p)$

The execution of a strategy by an agent is equivalent to its being unraveled into a sequence of substrategies, each of the form $\mathbf{do}(q)$. The agent follows each substrategy by performing actions prescribed by some tree. Thus the substrategies serve as abstractions of trees of basic actions. In this way, the definition of ability exhibits a two-layered architecture of agents: the bottom layer determining how substrategies of limited forms are achieved, and the top layer how they are composed to form complex strategies.

Since strategies are structured, the axiomatization of ability relative to a strategy must involve their structure. This comes into the axiomatization of $(\!|Y|\!)p$. These axioms resemble those for standard dynamic logic modalities, but there are important differences. Just as for reactive ability, the axiomatization below is relativized to the underlying logic.

AX-AB-STRAT-1. $\langle\!|\mathbf{skip}|\!\rangle p \equiv p$

AX-AB-STRAT-2. $\langle\!|Y_1;Y_2|\!\rangle p \equiv \langle\!|Y_1|\!\rangle\langle\!|Y_2|\!\rangle p$

AX-AB-STRAT-3. $\langle\!|\mathbf{if}\ q\ \mathbf{then}\ Y_1\ \mathbf{else}\ Y_2|\!\rangle p \equiv (q\rightarrow \langle\!|Y_1|\!\rangle p) \wedge (\neg q\rightarrow \langle\!|Y_2|\!\rangle p)$

AX-AB-STRAT-4. $\langle\!|\mathbf{while}\ q\ \mathbf{do}\ Y_1|\!\rangle p \equiv (q\rightarrow \langle\!|Y_1|\!\rangle\langle\!|\mathbf{while}\ q\ \mathbf{do}\ Y_1|\!\rangle p) \wedge (\neg q\rightarrow p)$

AX-AB-STRAT-5. $\langle\!|\mathbf{do}(q)|\!\rangle p \equiv (q \wedge p) \vee (\neg q \wedge (\bigvee a : \mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\langle\!|\mathbf{do}(q)|\!\rangle p))$

AX-AB-STRAT-6. All substitution instances of the validities of the underlying logic

Theorem 4.3 Axioms AX-AB-STRAT-1 through AX-AB-STRAT-6 yield a sound and complete axiomatization of $\langle\!|Y|\!\rangle p$.

Proof.

Soundness and Completeness: The proofs of soundness and completeness are developed hand-in-hand. Only formulae of the form $\langle\!|Y|\!\rangle p$ are considered here. As in section 4.2, construct a model whose indices are maximally consistent sets of sentences of the language. Completeness means that $M \models_t \langle\!|Y|\!\rangle p$ entails $\langle\!|Y|\!\rangle p \in t$, the corresponding moment in the model, and soundness means that $\langle\!|Y|\!\rangle p \in t$ entails $M \models_t \langle\!|Y|\!\rangle p$. The proof is by induction on the structure of strategies.

$M \models_t \langle\!|\mathbf{skip}|\!\rangle p$ iff $M \models_t p$. But by axiom AX-AB-STRAT-1, $\langle\!|\mathbf{skip}|\!\rangle p \in t$ iff $p \in t$. Thus we simultaneously have soundness for axiom AX-AB-STRAT-1, and completeness for this case of strategies.

Similarly, $M \models_t \langle\!|\mathbf{if}\ q\ \mathbf{then}\ Y_1\ \mathbf{else}\ Y_2|\!\rangle p$ iff there exists a tree that follows $\downarrow_t(\mathbf{if}\ q\ \mathbf{then}\ Y_1\ \mathbf{else}\ Y_2)$ and some further properties hold of it. But the truth of q entails that $\downarrow_t(\mathbf{if}\ q\ \mathbf{then}\ Y_1\ \mathbf{else}\ Y_2) = \downarrow_t Y_1$ and $\uparrow_t(\mathbf{if}\ q\ \mathbf{then}\ Y_1\ \mathbf{else}\ Y_2) = \uparrow_t Y_1$. The corresponding condition holds for $\neg q$. Thus $M \models_t \langle\!|\mathbf{if}\ q\ \mathbf{then}\ Y_1\ \mathbf{else}\ Y_2|\!\rangle p$ iff $M \models_t (q\rightarrow \langle\!|Y_1|\!\rangle p)$ and $M \models_t (\neg q\rightarrow \langle\!|Y_2|\!\rangle p)$. But by axiom AX-AB-STRAT-3, $\langle\!|\mathbf{if}\ q\ \mathbf{then}\ Y_1\ \mathbf{else}\ Y_2|\!\rangle p \in t$ iff $(q\rightarrow \langle\!|Y_1|\!\rangle p) \in t$ and $(\neg q\rightarrow \langle\!|Y_2|\!\rangle p) \in t$. By induction, since Y_1 and Y_2 are structurally smaller than the above conditional strategy, and since we have axiom AX-AB-STRAT-6 (which applies for $\rightarrow$), we have that $\langle\!|\mathbf{if}\ q\ \mathbf{then}\ Y_1\ \mathbf{else}\ Y_2|\!\rangle p \in t$.

The case of $\langle\!|\mathbf{do}(q)|\!\rangle p$ is quite simple. This is because axiom AX-AB-STRAT-5 closely resembles the axioms for reactive ability given in section 4.2. Using the definition of $\downarrow_t$ and $\uparrow_t$ we have that $M \models_t \langle\!|\mathbf{do}(q)|\!\rangle p$ iff $(\exists \tau : (\exists S, t' : [S;t,t'] \in [\![\tau]\!]_{\mathbf{do}(q)})$ and $(\forall S : (\forall t' \in S : [S;t,t'] \in [\![\tau]\!]_{\mathbf{do}(q)} \Rightarrow M \models_{t'} p)))$. If

$\tau = \emptyset$, then the right hand expression reduces to $M \models_t (q \wedge p)$. By axiom AX-AB-STRAT-6, $q \wedge p \in t$, which by axiom AX-AB-STRAT-5 entails that $(\!|\mathbf{do}(q)|\!)p \in t$, as desired. If $\tau \neq \emptyset$, we can proceed as follows.

$(\exists S, t' : [S; t, t'] \in [\![\tau]\!]_{\mathbf{do}(q)})$ entails that $(\exists S, t'' : [S; t, t''] \in [\![a]\!])$, which means that $M \models_t \mathsf{E}\langle a \rangle \mathsf{true}$. It also entails that $M \not\models_t q$. Also, by Lemma 4.2, $(\exists S, t' : [S; t, t'] \in [\![\tau]\!]_{\mathbf{do}(q)})$ iff $M \models_t (\!|\tau|\!)q$. By definition, $M \models_t (\!|\tau|\!)q$ iff $(\forall S, t'' : [S; t, t''] \in [\![a]\!] \Rightarrow (\exists i : 1 \leq i \leq m \text{ and } M \models_{t''} (\!|\tau_i|\!)q))$. Moreover, $(\forall S, t' : [S; t, t'] \in [\![\tau]\!]_{\mathbf{do}(q)} \Rightarrow M \models_{t'} p)$ entails that $(\forall S, t'' : [S; t, t''] \in [\![a]\!] \Rightarrow (\exists i : 1 \leq i \leq m \text{ and } (\forall S, t' : [S; t'', t'] \in [\![\tau_i]\!]_{\mathbf{do}(q)} \Rightarrow M \models_{t'} p)))$. Thus the condition for $M \models_t \mathsf{A}[a]((\!|\mathbf{do}(q)|\!)p)$ is met. By axiom AX-AB-STRAT-6 (which applies for quantification over actions), we have $(\bigvee a : \mathsf{E}\langle a \rangle \mathsf{true} \wedge \mathsf{A}[a]((\!|\mathbf{do}(q)|\!)p))$. But by the given axiom, this entails $(\!|\mathbf{do}(q)|\!)p$, as desired. This proves completeness for strategies of the form $\mathbf{do}(q)$. It also proves soundness of axiom AX-AB-STRAT-5 in the left to right direction.

For soundness of axiom AX-AB-STRAT-5 in the right to left direction, we just need to note that for the first disjunct of axiom AX-AB-STRAT-5, we can use an empty tree to make $(\!|\mathbf{do}(q)|\!)p$ hold; for the second disjunct, using the action a for which the quantified expression holds and the trees corresponding to the occurrences of $(\!|\mathbf{do}(q)|\!)p$ at the moments it has been done, we can construct a tree that would cause the satisfaction of $(\!|\mathbf{do}(q)|\!)p$ at the given moment. This parallels the construction given in the proof of Theorem 4.1 in section 4.2 for reactive ability, and is not repeated here.

Surprisingly, the trickiest case in this proof turns out to be that of sequencing. When $\downarrow_t Y_1 = \mathbf{skip}$, the desired condition for axiom AX-AB-STRAT-2 follows trivially. But, when $\downarrow_t Y_1 \neq \mathbf{skip}$, the satisfaction condition for $(\!|Y_1;Y_2|\!)p$ recursively depends on that for $(\!|\uparrow_t Y_1;Y_2|\!)p$. However, this strategy does not directly feature in axiom AX-AB-STRAT-2. Also, it is not necessarily the case that $\uparrow_t Y_1;Y_2$ is a structurally smaller strategy than $Y_1;Y_2$, e.g., if Y_1 is an iterative strategy, $\uparrow_t Y_1$ may be structurally more complex than Y_1. However, we can use the fact that here, as in standard dynamic logic, iterative strategies are finitary or, in other words, lead only to a finite number of repetitions when executed at any time. Thus we can assume that for any strategy, $Y \neq \mathbf{skip}$ and moment t, the fragment of $<$ in the model restricted to the execution of Y has a finite *depth*. Here *depth* of a strategy at a moment is defined as the maximum number of recursive invocations of $\downarrow$ along any scenario at that moment. If Y is followed at t, then the $\uparrow_t Y$ is followed at those moments where $\downarrow_t Y$ has just been followed. The depth of $\uparrow_t Y$ equals (depth of Y) $- 1$. The depth of $\mathbf{skip}$ is 0. Thus the depth is a metric to do the necessary induction on. The remainder of the proof is quite simple.

The case of iterative strategies is now quite simple. Axiom AX-AB-

STRAT-4 directly captures the conditions for $\downarrow_t$ and $\uparrow_t$ of an iterative strategy. Using the above result for sequencing, and the fact that iterative strategies are finitary, we can perform induction on the depth of the strategy. This yields the desired result.

Thus for all cases in the definition of a strategy, $M \models_t (\!|Y|\!)p$ iff $(\!|Y|\!)p \in t$. This proves soundness and completeness. □

4.4 Results on Ability

The axioms for strategies of the form $(\!|\mathbf{do}(q)|\!)p$, which is the nontrivial base case for strategies, are similar to the axioms for reactive ability given in section 4.2. Indeed, the following theorem states that the two concepts are logically identical, even though they have differing significance in terms of implementations.

Theorem 4.4 $\mathsf{K}_{rab}p \equiv \mathsf{K}_{sab}p$

Proof.

For the left to right direction, $\mathsf{K}_{rab}p$ yields $(\!|\mathbf{do}(p)|\!)p$, which yields $\mathsf{K}_{sab}p$. For the other direction, associate with $(\!|Y|\!)p$ a fragment of the model at the root of which $(\!|Y|\!)p$ holds and at the leaves of which is the first occurrence of p after the root. From this, construct a tree as required in the definition of $\mathsf{K}_{rab}p$. The tree for **skip** is the empty tree. The definition given above unravels a strategy into a finite sequence of strategies of the form $\mathbf{do}(q)$. Consider the last such strategies that apply in different parts of the model fragment. For each of them, an appropriate tree may be obtained by working from the bottom up in the given model fragment. Since coherence constraint COH-5 of section 2.3 holds, for each of the leaves of the fragment, there is a last action that ends there. Consider the nodes at which those actions are begun. At those nodes, we have a set of trees, each consisting of precisely one action that begins there and whose consequences are entirely within the fragment. If the original fragment is well-formed with respect to ability, there must be at least one action that satisfies this requirement. Repeated applications of this yield a tree for each strategy of the form $\mathbf{do}(q)$, and with respect to which, the agent has the required ability. Because of constraint COH-5, only a finite number of applications of this step are required. At each of the nodes where these trees are defined, we have a condition of the form $(\!|\mathbf{do}(q)|\!)p$. Continuing further in this way, we obtain a tree for the entire strategy. □

It is convenient to refer to reactive and strategic ability jointly as K_{ab}. We are now able to state and prove some results characterizing the formal properties of ability.

Theorem 4.5 $\mathsf{K}_{ab}p \rightarrow \mathsf{EF}p$; and, therefore, $\neg\mathsf{K}_{ab}\mathsf{false}$

Proof.

Consider the two axioms for reactive ability. Note that $p \rightarrow \mathsf{EF}p$: this takes care of the base case. Also, $(\bigvee a : \mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{EF}p)$ entails $\mathsf{EF}p$: this takes care of the inductive case.

It is a trivial consequence of the definitions of E and F that $\neg\mathsf{EFfalse}$ is valid. □

Theorem 4.6 $\mathsf{K}_{ab}p \wedge \mathsf{AG}(p \rightarrow q) \rightarrow \mathsf{K}_{ab}q$

Proof. Consider the two axioms for reactive ability. From axiom Ax-Ab-React-1, $\mathsf{K}_{rab}p$ holds if p holds. Since $\mathsf{AG}(p \rightarrow q)$ is a premise, we also have q, which by axiom Ax-Ab-React-1 entails $\mathsf{K}_{ab}q$. From axiom Ax-Ab-React-2, $\mathsf{K}_{rab}p$ holds if $(\bigvee a : \mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K}_{rab}p)$. But for all actions, a, $\mathsf{AG}(p \rightarrow q) \rightarrow \mathsf{A}[a]\mathsf{AG}(p \rightarrow q)$. Therefore, we have $(\bigvee a : \mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a](\mathsf{K}_{rab}p \wedge \mathsf{AG}(p \rightarrow q)))$. By the inductive hypothesis, we can conclude $(\bigvee a : \mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K}_{rab}q)$ which, by axiom Ax-Ab-React-2, entails $\mathsf{K}_{rab}q$, as desired. □

Theorem 4.7 $\mathsf{AF}p \rightarrow \mathsf{K}_{ab}p$

Proof. Here, we need coherence condition Coh-5 of section 2.3, which states that each moment on each scenario may be reached by a finite number of actions. Consider the subtree of the model at the leaves of which we have the first occurrences of p, which make $\mathsf{AF}p$ true. At those moments, we have $\mathsf{K}_{rab}p$, due to axiom Ax-Ab-React-1. Assign a depth to each of these moments that equals the number of actions required to reach that moment. Begin with the deepest moments. Let their depth be n. Let the nth action on some scenario be a, and let it begin at t. At the moment where that action is begun, we have $\mathsf{E}\langle a\rangle\mathsf{true}$, since the action is done on that scenario. Since this is the last action begun in the given subtree of the model, we also have $\mathsf{A}[a]\mathsf{K}_{rab}p$. Hence, axiom Ax-Ab-React-2 applies, and we have $\mathsf{K}_{rab}p$ at t. The we simply use induction to repeat this step n times to obtain $\mathsf{K}_{rab}p$ at the root of the subtree, which is where $\mathsf{AF}p$ holds. □

Theorem 4.8 $\mathsf{K}_{ab}\mathsf{K}_{ab}p \rightarrow \mathsf{K}_{ab}p$

Proof. Using the definition of reactive ability, construct a single tree out of the trees for $\mathsf{K}_{rab}\mathsf{K}_{rab}p$. For the base case, simply use axiom Ax-Ab-React-1. $\mathsf{K}_{rab}\mathsf{K}_{rab}p$ holds if $\mathsf{K}_{rab}p$ does, which trivially implies $\mathsf{K}_{rab}p$. For the inductive case, $\mathsf{K}_{rab}\mathsf{K}_{rab}p$ holds if $(\bigvee a : \mathsf{E}\langle a\rangle\mathbf{true} \wedge \mathsf{A}[a]\mathsf{K}_{rab}\mathsf{K}_{rab}p)$. By the inductive hypothesis, we obtain $(\bigvee a : \mathsf{E}\langle a\rangle\mathbf{true} \wedge \mathsf{A}[a]\mathsf{K}_{rab}p)$ which, by axiom-AX-AB-REACT-2 implies $\mathsf{K}_{rab}p$, as desired. □

This seems intuitively quite obvious: if an agent can ensure that he will be able to ensure p, then he can already ensure p. But see the discussion following Theorem 4.10.

In section 4.2, I gave an example involving Halley's comet. If p is "returns(Halley's comet)," then assuming its return is inevitable, $\mathsf{K}_{ab}p$ holds. But this may seem strange relative to our pretheoretic intuitions about ability: we would not ordinarily state that an agent such as ourselves is able to make Halley's comet return. Intuitively, it seems that an agent can be said to be able to achieve something only if it is not inevitable anyway.

For this reason, it is useful to consider an alternative notion, namely, of *proper ability*. Let K^p_{ab} denote this concept. Then $\mathsf{K}^p_{ab}p$ holds only if p is not inevitable and does not hold currently. An obvious formalization of this is given next.

SEM-28. $M \models_t x\mathsf{K}^p_{ab}p$ iff $M \models_t (x\mathsf{K}_{ab}p)$ and $(\exists S : (\forall t' : t' \in S \rightarrow M \not\models_{t'} p))$

As a consequence of this definition, K^p_{ab} is a non-normal operator [Chellas, 1980, p. 114]. This means that $\mathsf{K}^p_{ab}p \wedge (p \rightarrow q)$ does not imply that $\mathsf{K}^p_{ab}q$. This is so, because q could be inevitable. For example, since **true** holds everywhere, we have $\neg\mathsf{K}^p_{ab}\mathbf{true}$. This complicates the logical properties of K^p_{ab}. Thus, whereas K^p_{ab} is the intuitively more reasonable sense of ability, K_{rab} is the technically more tractable one. This is one motivation for retaining both. Another motivation for K_{rab} is that in some applications, it is even intuitively the preferred notion (see section 4.9).

The operator K^p_{ab}, can now be axiomatized simply by adding the following axiom:

AX-AB-1. $\mathsf{K}^p_{ab}p \equiv (\mathsf{K}_{ab}p \wedge \neg\mathsf{AF}p)$

Theorem 4.9 $\neg\mathsf{K}^p_{ab}\mathbf{true}$

Proof. We trivially have $\mathsf{AF}\mathbf{true}$, which by axiom AX-AB-1 entails $\neg\mathsf{K}^p_{ab}\mathbf{true}$. □

Therefore, despite Theorem 4.6, the corresponding statement for K^p_{ab} fails. Indeed, we have

Theorem 4.10 $p \rightarrow \neg \mathsf{K}^p_{ab} p$

Proof. Trivially again, since $p \rightarrow \mathsf{AF}p$. □

Theorem 4.10 states that if p already holds then the agent cannot be felicitously said to be able to achieve it. By a simple substitution, we obtain $\mathsf{K}^p_{ab} p \rightarrow \neg \mathsf{K}^p_{ab} \mathsf{K}^p_{ab} p$, whose contrapositive is $\mathsf{K}^p_{ab} \mathsf{K}^p_{ab} p \rightarrow \neg \mathsf{K}^p_{ab} p$. This is in direct opposition to Theorem 4.8 for K_{ab}, and is surprising, if not counterintuitive. It says that an agent who is able to become able to achieve p is not able to achieve p. This too agrees with our intuitions about K^p_{ab} since we explicitly wish to exclude the ability to achieve inevitable propositions. The explanation for this unexpected observation is that when we speak of nested ability, which we do not do often in natural language, we use two different senses of ability: K^p_{ab} for the inner one and K_{ab} for the outer one. Thus the correct translation is $\mathsf{K}_{ab} \mathsf{K}^p_{ab} p$, which entails $\mathsf{K}_{ab} p$, as desired.

Theorem 4.11 $\mathsf{K}_{ab} \mathsf{K}^p_{ab} p \rightarrow \mathsf{K}_{ab} p$

Proof. $\mathsf{K}_{ab} \mathsf{K}^p_{ab} p \equiv \mathsf{K}_{ab}(\mathsf{K}_{ab} p \wedge \neg \mathsf{AF}p)$, by definition of K^p_{ab}. Since $\mathsf{AG}((r \wedge q) \rightarrow r)$, we can apply Theorem 4.6. Thus the left hand side implies $\mathsf{K}_{ab} \mathsf{K}_{ab} p$, which by Theorem 4.8 implies $\mathsf{K}_{ab} p$. □

We can sometimes do better than this. For example, if p describes a condition that persists over time, as many p's in natural language examples do, then we also have $\mathsf{K}^p_{ab} p$. Briefly, the persistence of a condition p can be described by the formula $\mathsf{AG}(p \rightarrow \mathsf{AG}p)$, which says that once p holds, it persists forever on all scenarios. For example, if p denotes that the carpet has an indelible stain, then once it holds it will continue to hold forever.

Theorem 4.12 $(\mathsf{K}_{ab} \mathsf{K}^p_{ab} p \wedge \mathsf{AG}(p \rightarrow \mathsf{AG}p)) \rightarrow \mathsf{K}^p_{ab} p$

Proof. Using the definition of K^p_{ab}, we can see that the left hand side expression is equivalent to $\mathsf{K}_{ab}(\mathsf{K}_{ab} p \wedge \neg \mathsf{AF}p) \wedge \mathsf{AG}(p \rightarrow \mathsf{AG}p)$. By Theorem 4.6, this implies $\mathsf{K}_{ab} \mathsf{K}_{ab} p \wedge \mathsf{K}_{ab} \neg \mathsf{AF}p \wedge \mathsf{AG}(p \rightarrow \mathsf{AG}p)$. Using Theorem 4.8 on the first conjunct and Theorem 4.5 on the second conjunct, we obtain $\mathsf{K}_{ab} p \wedge \mathsf{EF} \neg \mathsf{AF} p \wedge \mathsf{AG}(p \rightarrow \mathsf{AG}p)$. Now assume $\mathsf{AF}p$. Combining this with the last conjunct of the previous expression, we get $\mathsf{AFAG}p$. Note that $\mathsf{EF} \neg \mathsf{AF} p \equiv \mathsf{EFEG} \neg p$. This contradicts $\mathsf{AFAG}p$. Hence by reductio ad absurdum, $\neg \mathsf{AF}p$. Thus the left hand side of the statement of this theorem implies $\mathsf{K}_{ab} p \wedge \neg \mathsf{AF}p$, which is equivalent to $\mathsf{K}^p_{ab} p$. □

Theorem 4.12 can be weakened to apply even in the case of conditions that occur infinitely often, but do not persist forever. For example, the condition that a certain switch is on would not persist forever, simply because any agent could toggle it. However, it might reasonable to assume that the switch becomes on infinitely often. $\mathsf{AG}(p \rightarrow \mathsf{A}(\bigvee a : \langle a \rangle \mathsf{F}p))$ means that at all moments in the future if p holds, then for every scenario, there is a moment in its strict future at which $\mathsf{F}p$ holds (this is so because, as required in section 2.1.2, all actions take time). Thus for every occurrence of p, there is another occurrence of p in its strict future.

Theorem 4.13 $(\mathsf{K}_{ab}\mathsf{K}^{p}_{ab}p \wedge \mathsf{AG}(p \rightarrow \mathsf{A}(\bigvee a : \langle a \rangle \mathsf{F}p))) \rightarrow \mathsf{K}^{p}_{ab}p$

Proof. As before, using the definition of K^{p}_{ab}, we can see that the left hand side expression is equivalent to $\mathsf{K}_{ab}(\mathsf{K}_{ab}p \wedge \neg\mathsf{AF}p) \wedge \mathsf{AG}(p \rightarrow \mathsf{A}(\bigvee a : \langle a \rangle \mathsf{F}p))$. By Theorem 4.6, this implies $\mathsf{K}_{ab}\mathsf{K}_{ab}p \wedge \mathsf{K}_{ab}\neg\mathsf{AF}p \wedge \mathsf{AG}(p \rightarrow \mathsf{A}(\bigvee a : \langle a \rangle \mathsf{F}p))$. Using Theorem 4.8 on the first conjunct and Theorem 4.5 on the second conjunct, we obtain $\mathsf{K}_{ab}p \wedge \mathsf{EF}\neg\mathsf{AF}p \wedge \mathsf{AG}(p \rightarrow \mathsf{A}(\bigvee a : \langle a \rangle \mathsf{F}p))$. But $\mathsf{EF}\neg\mathsf{AF}p \equiv \mathsf{EFEG}\neg p$. In other words, there is a scenario, S, on which there is a moment, after which there are no more occurrences of p. Now assume $\mathsf{AF}p$. This implies that there is at least one occurrence of p on every scenario, hence there is an occurrence of p on S. Consider the last occurrence of p (this could be at the present time). Combining this with $\mathsf{AG}(p \rightarrow \mathsf{A}(\bigvee a : \langle a \rangle \mathsf{F}p))$, we conclude that there is an occurrence of p on S in the future of the given occurrence. This contradicts the assumption of a last occurrence of p. Hence we obtain $\neg\mathsf{AF}p$. Thus we have $\mathsf{K}_{ab}p \wedge \neg\mathsf{AF}p$, which is equivalent to $\mathsf{K}^{p}_{ab}p$. □

4.5 Incorporating Action Selection: Reactive Know-How

Ability as defined above considers the choices that an agent can exercise in principle. However, it finesses the problem with regard to the agent knowing enough to actually be in a position to make those choices. I now seek to complete this part of the picture, by explicitly considering an agent's beliefs, which influence the choices that he, in fact, makes. For example, if an agent is able to dial all possible combinations of a safe, then by the above definition 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.

I introduce the following notation into $\mathcal{L}^h$.

SYN-25. $p \in \mathcal{L}^h_s$ and $x \in \mathcal{A}$ implies that $(x\mathsf{K}_{\mathsf{hr}}p), (x\mathsf{K}_{\mathsf{hs}}p) \in \mathcal{L}^h$

SYN-26. $p \in \mathcal{L}^h_s$, $Y, Y' \in \mathcal{L}^h_y$, and $x \in \mathcal{A}$ implies that $(x[\![Y]\!]p), (x\lceil Y \rceil Y'), (x\lfloor Y \rfloor Y') \in \mathcal{L}^h$

SYN-27. $\tau \in \Upsilon$, $x \in \mathcal{A}$, and $p \in \mathcal{L}^h$ implies that $x[\![\tau]\!]p \in \mathcal{L}^h$

$x[\![\tau]\!]p$ denotes that agent x knows how to achieve p relative to tree τ. As usual, the agent symbol can be omitted, since it is obvious from the context. It reduces notational complexity to 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-29. $M \models_t [\![\emptyset]\!]p$ iff $M \models_t \mathsf{K_t}p$

SEM-30. $M \models_t [\![a]\!]p$ iff $M \models_t \mathsf{K_t}(\mathsf{E}\langle a \rangle \mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_t}p)$

SEM-31. $M \models_t [\![\langle a; \tau_1, \ldots, \tau_m \rangle]\!]p$ iff $M \models_t \mathsf{K_t}(\mathsf{E}\langle a \rangle \mathsf{true} \wedge \mathsf{A}[a](\bigvee_{1 \leq i \leq m} \tau_i : ([\![\tau_i]\!]p)))$

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 particular 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 at some moment during or 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, provided one has the necessary skills, i.e., the necessary actions, available.

SEM-32. $M \models_t x\mathsf{K_{hr}}p$ iff $(\exists \tau : M \models_t x[\![\tau]\!]p)$

Consider Figure 4.2 for an example. Let x be the agent whose actions are written first there. Assume for simplicity that each moment is its own unique alternative for x. Then, by the above definitions, $x\mathsf{K_t}p$ holds at t_3 and

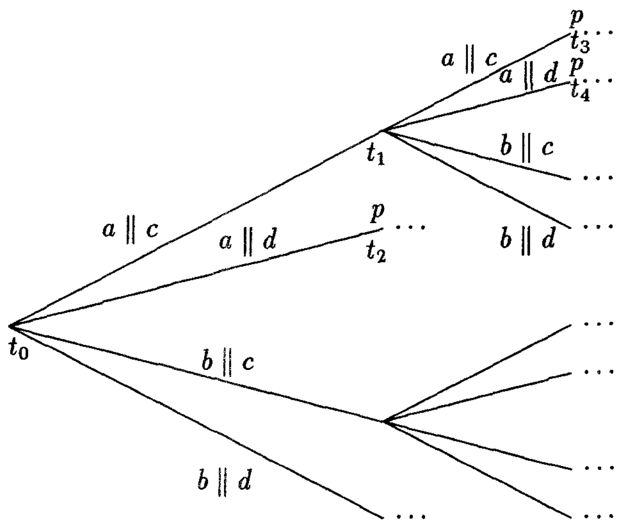

Figure 4.2: Know-how

t_4. Also, $x\mathsf{K}_{\mathsf{hr}}p$ 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 p at each moment where a ends. In other words, we can define a tree, $\langle a; a, \emptyset\rangle$, such that x can achieve p by properly executing that tree. Therefore, x knows how to achieve p at t_0.

I now propose the following axioms for K_{hr}. These axioms are motivated by analogy with the axioms for K_{rab} given previously.

AX-KH-REACT-1. $\mathsf{K_t}p \rightarrow \mathsf{K}_{\mathsf{hr}}p$

AX-KH-REACT-2. $(\bigvee a : \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K}_{\mathsf{hr}}p)) \rightarrow \mathsf{K}_{\mathsf{hr}}p$

AX-KH-REACT-3. All substitution instances of the validities of the underlying logic.

Theorem 4.14 Axioms AX-AB-REACT-1 through AX-KH-REACT-3 yield a sound and complete axiomatization for K_{hr}.

Proof.

Construct a branching-time model, M. The moments of M are notated as t and are maximally consistent sets of formulae that contain all the substitution instances of the validities of the underlying logic. The other components of the model, especially, $<$, $\mathbf{B}$, $\mathbf{R}$, and $[\![\,]\!]$, are constrained by the formulae that are true at the different moments. Furthermore, these sets are closed under the above two axioms for K_{h_r}. We can ignore the agent symbol in the following discussion.

Soundness: For axiom AX-KH-REACT-1 above, soundness is trivial from the definition of $[\![(\emptyset)]\!]p$. For axiom AX-KH-REACT-2, let $(\bigvee a : \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_{hr}}p))$ hold at t. Then, by semantic condition SEM-30, $[\![a]\!]p$ holds, which, by semantic condition SEM-32, entails $\mathsf{K_{hr}}p$. Hence, axiom AX-KH-REACT-2 is sound.

Completeness: The proof is by induction on the structure of formulae. Only the case of formulae of the form $\mathsf{K_{hr}}p$ is described below. Completeness means that $M \models_t \mathsf{K_{hr}}p$ entails $\mathsf{K_{hr}}p \in t$. $M \models_t \mathsf{K_{hr}}p$ iff $(\exists \tau : M \models_t [\![\tau]\!]p)$. This proof is by induction inside the induction on the structure of formulae. This induction is on the structure of trees with which a formula of the form $\mathsf{K_{hr}}p$ is satisfied. One base case is the empty tree $\emptyset$. And $M \models_t [\![(\emptyset)]\!]p$ iff $M \models_t \mathsf{K_t}p$. By AX-KH-REACT-3, $\mathsf{K_t}p \in t$. By axiom AX-KH-REACT-1 above, $\mathsf{K_{hr}}p \in t$, as desired.

The other base case is for single-action trees. $M \models_t [\![a]\!]p$ iff $M \models_t \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_t}p)$. This is equivalent to the following expression: $(\forall t_b : (t,t_b) \in \mathbf{B} \Rightarrow (\exists S_b, t' : [S_b; t_b, t'] \in [\![a]\!])$ and $(\forall S_b : (\forall t' \in S_b : [S_b; t_b, t'] \in [\![a]\!] \Rightarrow (\exists t'' : t_b < t'' \leq t'$ and $M \models_{t''} \mathsf{K_t}p)))$. But, by axiom AX-KH-REACT-1, $\mathsf{K_t}p \rightarrow \mathsf{K_{hr}}p$. And, by axiom AX-KH-REACT-3, we have that $(\forall t_b : (t,t_b) \in \mathbf{B} \Rightarrow (\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_{hr}}p))$. That is, $\mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_{hr}}p) \in t$, which trivially entails $(\bigvee a : \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_{hr}}p)) \in t$. Thus by axiom AX-KH-REACT-2, we have $\mathsf{K_{hr}}p \in t$.

For the inductive case, $M \models_t [\![\langle a; \tau_1, \ldots, \tau_m\rangle]\!]p$ iff $M \models_t \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a](\bigvee_{1 \leq i \leq m} \tau_i : ([\![\tau_i]\!]p)))$. This requires that , for some index i, $[\![\tau_i]\!]p$ holds at some appropriate moments. Let t_e be one such moment. Therefore, $M \models_{t_e} \mathsf{K_{hr}}p$. Further, by the inductive hypothesis, we have that $\mathsf{K_{hr}}p \in t_e$. Consequently, by axiom AX-KH-REACT-2, we obtain that $\mathsf{K_{hr}}p \in t$. Hence we have completeness of the above axiomatization. □

4.6 Strategic Know-How

The above formalization gives a reactive definition of know-how. It considers the beliefs of agents and their influence on the selection of actions by them. However, it still remains to be seen if we can incorporate strategies into the picture to give an abstract definition of know-how. I introduce an operator, $\langle\!\langle\, \rangle\!\rangle$, to denote an agent's know-how relative to a strategy. $x\langle\!\langle Y \rangle\!\rangle p$ means that x knows how to follow strategy Y and thereby to achieve p. Knowing how to follow a strategy presupposes knowing the right actions to perform in order to satisfy it.

Just as for the case of ability, I aim to show that the strategic definition of know-how builds on top of the reactive definition given previously. To this end, I define the *know-how-intension* of a tree, relative to a strategy, in analogy with the *ability-intension* of a tree defined in section 4.3. Let the *know-how-intension* of a tree, τ, relative to a strategy, Y, for an agent, x, be notated as $[\![\{\tau\}]\!]^x_Y$. This 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; mere ability is not sufficient. Just as for the ability-intension of trees, the know-how-intension of trees needs to be defined only for the $\downarrow$ of strategies, which are always of one of the forms, **skip** or **do**(q). Formally, we have the following cases in the definition of $[\![\{\tau\}]\!]^x_Y$.

AUX-10. The agent knows how to satisfy the empty strategy, **skip**, by doing nothing, i.e., by following the empty tree.

$[S;t,t'] \in [\![\{\emptyset\}]\!]^x_{\mathbf{skip}}$ iff $t = t'$

AUX-11. The agent may know how to satisfy the strategy **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 **do**(q) by following one of its subtrees. Thus we have:

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

Aux-1. $\tau = \emptyset$ and $t = t'$ and $M \models_t x\mathsf{K_t}q$;

Aux-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 $M \not\models_{t_2} q))$; or,

Aux-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 $(\exists t_1 : [S;t,t_1] \in [\![a]\!]$ and $(\exists t_2, i : 1 \leq i \leq m$ 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 $M \not\models_{t_3} q)$

By the above definition, $[S; t, t'] \in [\![\{\tau\}]\!]^x_{\mathbf{do}(q)}$ means that starting at moment t, moment t' is the earliest moment at which x knows how to make q happen by following it. As a result, for a scenario, S, and moments $t', t'' \in S$, we have that $[S; t, t'] \in [\![\{\tau\}]\!]^x_{\mathbf{do}(q)}$ and $[S; t, t''] \in [\![\{\tau\}]\!]^x_{\mathbf{do}(q)}$ implies that $t' = t''$. This agrees with the intuition behind constraint COH-1 of section 2.3 that an action begun at a moment can end at most one moment on each scenario. In other words, $[\![\{\tau\}]\!]^x_{\mathbf{do}(q)}$ denotes the intension of the abstract action performed by agent x. This is x's abstract action of achieving q by exercising his know-how.

The above is an important intuition about strategies as they have been used throughout here. In order to make it explicit, I extend the formal language by adding two operators on strategies: $\langle\,\rangle_h$ and $[\,]_h$. These operators are defined in analogy with the operators $\langle\,\rangle$ and $[\,]$, which were defined for basic actions in section 2.1.3. However, unlike those operators, $\langle\,\rangle_h$ and $[\,]_h$ involve the evaluation of the given condition at the final moment of the relevant period. Consequently, $\langle\,\rangle_h$ and $[\,]_h$ are duals of each other. Formally, I add the following rule to the syntax of $\mathcal{L}^h$.

SYN-28. $p \in \mathcal{L}^h_s$, $x \in \mathcal{A}$, and $Y \in \mathcal{L}^h_y$ implies that $x[Y]_h p, x\langle Y\rangle_h p \in \mathcal{L}^h_s$

Now I give the semantic conditions for the new operators.

SEM-33. $M \models_{S,t} x\langle\mathbf{do}(q)\rangle_h p$ iff $(\exists \tau, t' \in S : [S; t, t'] \in [\![\{\tau\}]\!]^x_{\mathbf{do}(q)}$ and $M \models_{S,t'} p)$

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

SEM-34. $M \models_{S,t} x[\mathbf{do}(q)]_h p$ 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. It is all right to quantify over all t''s here, since, as remarked above, there can be at most one t' (on the given scenario) at which the action has been completed. More importantly, we must quantify over all trees with which $\mathbf{do}(q)$ can be performed, because those trees are equally legitimate as ways to perform $\mathbf{do}(q)$. If only some of the ways of performing $\mathbf{do}(q)$ were acceptable, then $\mathbf{do}(q)$ would not be a reasonable abstraction to use in a strategy: one should, instead, have specified $\mathbf{do}(q')$, for an appropriately strong q'. In other words, all that is relevant about acceptable scenarios is specified in $\mathbf{do}(q)$ itself.

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. That is, the definition of know-how relative to a strategy should parallel the previous, reactive, definition of know-how. The only difference lies in the fact that the strategic definition employs the operators on abstract actions defined above. An agent knows how to achieve p by following the empty strategy, **skip**, if he knows that p. The justification for this is the same as the one for the case of the empty tree, i.e., SEM-29, considered in section 4.5.

The case of a general strategy is more interesting. 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. I introduce two new operators to capture what the agent does now and what he will need to do later. The formula $x\lfloor Y\rfloor Y'$ means that for the agent x to follow Y at the given moment, he must begin by following Y'. In light of the previous discussion, Y' must be of one of the forms, **skip** or **do**(q). Since we have stipulated that the agents' beliefs are true, $x\lfloor Y\rfloor Y'$ holds only if $Y' = \downarrow_t Y$. However, since the agents' beliefs may be incomplete, $x\lfloor Y\rfloor Y'$ may be false for all Y'. Assuming $x\lfloor Y\rfloor Y'$ as above, the formula $x\lceil Y\rceil Y''$ means that for the agent x to follow Y at the given moment, he must follow Y'' after he has followed Y'. As above, $x\lceil Y\rceil Y''$ holds only if $Y'' = \uparrow_t Y$. In other words, $\lfloor\rfloor$ and $\lceil\rceil$ capture knowledge on the agent's part of the $\downarrow$ and $\uparrow$ of a strategy.

SEM-35. $M \models_t x\lfloor\mathbf{skip}\rfloor\mathbf{skip}$

SEM-36. $M \models_t x\lfloor\mathbf{do}(q)\rfloor\mathbf{do}(q)$ iff $M \models_t \neg q$

SEM-37. $M \models_t x\lfloor\mathbf{do}(q)\rfloor\mathbf{skip}$ iff $M \models_t x\mathsf{K_t}q$

SEM-38. $M \models_t x\lfloor\mathbf{if}\ r\ \mathbf{then}\ Y_1\ \mathbf{else}\ Y_2\rfloor Y'$ iff $M \models_t (x\mathsf{K_t}r \wedge x\lfloor Y_1\rfloor Y') \vee (x\mathsf{K_t}\neg r \wedge x\lfloor Y_2\rfloor Y')$

SEM-39. $M \models_t x\lfloor Y_1;Y_2\rfloor Y'$ iff
(a) $Y' \neq \mathbf{skip}$ and $M \models_t x\lfloor Y_1\rfloor Y'$, or
(b) $M \models_t (x\lfloor Y_1\rfloor\mathbf{skip} \wedge x\lfloor Y_2\rfloor Y')$

SEM-40. $M \models_t x\lfloor\mathbf{while}\ r\ \mathbf{do}\ Y_1\rfloor Y'$ iff
(a) $M \models_t (x\mathsf{K_t}r \wedge x\lfloor Y_1\rfloor Y')$, or
(b) $Y' = \mathbf{skip}$ and $M \models_t x\mathsf{K_t}\neg r$

SEM-41. $M \models_t x\lceil\mathbf{skip}\rceil\mathbf{skip}$

SEM-42. $M \models_t x\lceil\mathbf{do}(q)\rceil\mathbf{skip}$

SEM-43. $M \models_t x\lceil$**if** r **then** Y_1 **else** $Y_2\rceil Y'$ iff $M \models_t (x\mathsf{K_t} r \wedge x\lceil Y_1\rceil Y') \vee (x\mathsf{K_t}\neg r \wedge x\lceil Y_2\rceil Y')$

SEM-44. $M \models_t x\lceil Y_1;Y_2\rceil Y'$ iff
(a) $M \models_t (x\lfloor Y_1\rfloor$**skip** $\wedge\, x\lceil Y_2\rceil Y')$, or
(b) $(\exists Y'', Y_0 : Y_0 \neq$ **skip** and $Y' = Y'';Y_2$ and $M \models_t (x\lfloor Y_1\rfloor Y_0 \wedge x\lceil Y_1\rceil Y''))$

SEM-45. $M \models_t x\lceil$**while** r **do** $Y_1\rceil$**skip** iff $M \models_t (x\mathsf{K_t}\neg r \vee x\lfloor Y_1\rfloor$**skip**$)$

SEM-46. $M \models_t x\lceil$**while** r **do** $Y_1\rceil Y'$ iff $(\exists Y_0 : Y_0 \neq$ **skip** and $Y' = Y_0;$(**while** r **do** Y_1) and $M \models_t (x\mathsf{K_t} r \wedge x\lceil Y_1\rceil Y_0))$

A consequence of these definitions is Lemma 4.15, which states that an agent can have at most one substrategy to perform at a given moment and at most one substrategy to perform on doing the first one. Another consequence is Lemma 4.16, which states that, if x knows what substrategy to follow at a given moment, he knows what substrategy to follow after the first substrategy is over. In analogy with Lemma 2.28, which states that $\downarrow_t Y =$ **skip** entails that $\uparrow_t Y =$ **skip**, we have Lemma 4.17, which states that $x\lfloor Y\rfloor$**skip** entails $x\lceil Y\rceil$**skip**. None of the lemmas mentioned above require the agent's beliefs to be true; they all just require them to be mutually consistent. Also, since **true** is valid, by the semantic definition of B (or $\mathsf{K_t}$), i.e., SEM-16 in section 2.6, we have that $x\mathsf{K_t}$**true** is valid. The only clause in the definition of $x\lfloor Y\rfloor Y'$ that allows $Y' =$ **do**(q), i.e., SEM-36, requires that $\neg q$ hold in the model, which cannot be the case for **true**. Therefore, $x\lfloor Y\rfloor Y'$ entails that $Y' \neq$ **do**(**true**).

Lemma 4.15 $(\forall Y', Y'' : M \models_t x\lfloor Y\rfloor Y' \wedge x\lfloor Y\rfloor Y''$ implies $Y' = Y'')$ and $(\forall Y', Y'' : M \models_t x\lceil Y\rceil Y' \wedge x\lceil Y\rceil Y''$ implies $Y' = Y'')$

Proof. It is easily seen that this claim holds for the base cases, i.e., for Y of one of the forms, **skip** and **do**(q). The conditions in the other semantic clauses are also mutually exclusive, so that at most one recursive invocation of $\lfloor\,\rfloor$ and $\lceil\,\rceil$, respectively, is possible in each case. □

Lemma 4.16 $(\exists Y' : M \models_t x\lfloor Y\rfloor Y')$ implies that $(\exists Y'' : M \models_t x\lceil Y\rceil Y'')$

Proof. The semantic clauses given above give the conditions that determine whether $(\exists Y' : M \models_t x\lfloor Y\rfloor Y')$ holds. Some of these conditions involve the knowledge of agent x at moment t. It can be seen by inspection that, except for the case where Y is of the form **do**(q), whenever $M \models_t x\lfloor Y\rfloor Y'$ holds, so does $M \models_t x\lceil Y\rceil Y''$, for some Y''. When Y is of the form **do**(q), then $M \models_t x\lceil Y\rceil$**skip** holds in all cases. This proves the above claim.

The converse of the above claim fails because $M \models_t x\lceil \mathbf{do}(q)\rceil \mathbf{skip}$ may be true even when $M \models_t q \wedge \neg x\mathsf{K_t}q$. In that case, neither condition SEM-36, nor condition SEM-37 applies and, therefore, $(\exists Y' : M \models_t x\lfloor \mathbf{do}(q)\rfloor Y')$ is false. □

Lemma 4.17 $x\lfloor Y\rfloor\mathbf{skip}$ entails $x\lceil Y\rceil\mathbf{skip}$

Proof. By inspection of the semantic conditions for $\lfloor\,\rfloor$ and $\lceil\,\rceil$. The lemma holds for the cases of **skip** and **do**(q) trivially. It can be proved for the other cases by checking their semantic conditions pairwise. □

4.7 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) $x\lfloor Y\rfloor Y'$ 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, **skip** or **do**(q), Y is progressively unraveled into a sequence of substrategies of those forms. Formally, we have

SEM-47. $M \models_t x\langle\!\langle \mathbf{skip}\rangle\!\rangle p$ iff $M \models_t x\mathsf{K_t}p$

SEM-48. $M \models_t x\langle\!\langle Y\rangle\!\rangle p$ iff $M \models_t x\mathsf{K_t}(\mathsf{E}x\langle\downarrow_t Y\rangle_h \mathsf{true} \wedge \mathsf{A}x[\downarrow_t Y]_h x\langle\!\langle \uparrow_t Y\rangle\!\rangle p)$ and $M \models_t x\lfloor Y\rfloor \downarrow_t Y$

An interesting observation about the above definition is that it 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.

In rough analogy with the axioms for ability, and with the knowledge of agents explicitly considered, we can come up with the following axioms for know-how.

AX-KH-STRAT-1. $x\langle\!\langle \mathbf{skip}\rangle\!\rangle p \equiv x\mathsf{K_t}p$

AX-KH-STRAT-2. $x\langle\!\langle Y_1;Y_2\rangle\!\rangle p \equiv x\langle\!\langle Y_1\rangle\!\rangle x\langle\!\langle Y_2\rangle\!\rangle p$

AX-KH-STRAT-3. $x\langle\!\langle \mathbf{if}\ q\ \mathbf{then}\ Y_1\ \mathbf{else}\ Y_2\rangle\!\rangle p \equiv$
$(x\mathsf{K_t}q \wedge x\langle\!\langle Y_1\rangle\!\rangle p) \vee (x\mathsf{K_t}\neg q \wedge x\langle\!\langle Y_2\rangle\!\rangle p)$

AX-KH-STRAT-4. $x[\!\langle \textbf{while } q \textbf{ do } Y_1 \rangle\!]p \equiv (x\mathsf{K_t}q \wedge x[\!\langle Y_1\rangle\!]x[\!\langle \textbf{while } q \textbf{ do } Y_1\rangle\!]p) \vee (x\mathsf{K_t}\neg q \wedge x\mathsf{K_t}p)$

AX-KH-STRAT-5. $x[\!\langle \textbf{do}(q)\rangle\!]p \equiv (\neg q \wedge (\bigvee a : x\mathsf{K_t}(\mathsf{E}x\langle a\rangle\mathsf{true} \wedge \mathsf{A}x[a][\!\langle \textbf{do}(q)\rangle\!]p))) \vee x\mathsf{K_t}(q \wedge p)$

AX-KH-STRAT-6. All substitution instances of the validities of the underlying logic

Theorem 4.18 Axioms AX-KH-STRAT-1 through AX-KH-STRAT-6 yield a sound and complete axiomatization of $x[\!\langle Y\rangle\!]p$.

Proof.

Soundness and Completeness: The proofs of soundness and completeness are developed together. Only formulae of the form $x[\!\langle Y\rangle\!]p$ are considered here. As before, construct a model whose indices are maximally consistent sets of sentences of the language. Completeness means that $M \models_t x[\!\langle Y\rangle\!]p$ entails $x[\!\langle Y\rangle\!]p \in t$ and soundness means that $x[\!\langle Y\rangle\!]p \in t$ entails $M \models_t x[\!\langle Y\rangle\!]p$. The proof is by induction on the structure of strategies.

$M \models_t x[\!\langle \textbf{skip}\rangle\!]p$ iff $M \models_t x\mathsf{K_t}p$. But, by axiom AX-KH-STRAT-1, $x[\!\langle \textbf{skip}\rangle\!]p \in t$ iff $x\mathsf{K_t}p \in t$. Thus, we simultaneously have soundness for axiom AX-KH-STRAT-1, and completeness for strategies of the form **skip**.

Let $Y =$ **if** q **then** Y_1 **else** Y_2. Then, $M \models_t x[\!\langle Y\rangle\!]p$ iff x first performs Y' and then Y'', where $M \models_t (x\lfloor Y\rfloor Y' \wedge x\lceil Y\rceil Y'')$. But this holds only if $x\mathsf{K_t}q$ or $x\mathsf{K_t}\neg q$ holds at t. By the definitions SEM-38 and SEM-43, $M \models_t (x\mathsf{K_t}q \wedge x\lfloor Y_1\rfloor Y' \wedge x\lceil Y_1\rceil Y'') \vee (x\mathsf{K_t}\neg q \wedge x\lfloor Y_2\rfloor Y' \wedge x\lceil Y_2\rceil Y'')$. Therefore, using the definition of $[\!\langle\,\rangle\!]$, we obtain $M \models_t x[\!\langle Y\rangle\!]p$ iff $M \models_t (x\mathsf{K_t}q \wedge x[\!\langle Y_1\rangle\!]p) \vee (x\mathsf{K_t}\neg q \wedge x[\!\langle Y_2\rangle\!]p)$. Thus, we simultaneously have soundness for axiom AX-KH-STRAT-3, and completeness for conditional strategies.

Let $Y = \textbf{do}(q)$. Using definitions SEM-37, SEM-36, and SEM-42 and the definition of $[\!\langle \textbf{skip}\rangle\!]p$ (SEM-47), we have that $M \models_t x[\!\langle \textbf{do}(q)\rangle\!]p$ iff $M \models_t (x\mathsf{K_t}q \wedge x\mathsf{K_t}p)$ or $M \models_t \neg q \wedge x\mathsf{K_t}(\mathsf{E}x\langle \textbf{do}(q)\rangle_h\mathsf{true} \wedge x[\textbf{do}(q)]_h x\mathsf{K_t}p)$. The first case is taken care of by one disjunct of axiom AX-KH-STRAT-5. Let t_b be a moment such that $(t, t_b) \in \mathbf{B}(x)$. Let τ be a tree that makes $\mathsf{E}x\langle \textbf{do}(q)\rangle_h\mathsf{true}$ hold at t_b. Consider the same tree in the definition of $[\,]_h$.

The rest of the proof of this case is by induction on the structure of trees. Initially, since $\neg q$ holds, $\tau \neq \emptyset$. However, $\tau = \emptyset$ is considered as a base case of the induction. At any moment, t', if $\tau = \emptyset$ satisfies $\mathsf{E}x\langle \textbf{do}(q)\rangle_h\mathsf{true}$, then $\mathsf{E}x\langle \textbf{do}(q)\rangle_h\mathsf{true} \wedge x[\textbf{do}(q)]_h x\mathsf{K_t}p$ implies that $x\mathsf{K_t}(q \wedge p)$ which, by axiom AX-KH-STRAT-5, entails $x[\!\langle \textbf{do}(q)\rangle\!]p$ holds at t'. If $\tau = a$ then, since the first occurrence of q is relevant, $\neg q$ must hold at the given moment. Also, $\mathsf{E}x\langle \textbf{do}(q)\rangle_h\mathsf{true} \wedge$

$x[\mathbf{do}(q)]_h x\mathsf{K_t} p$ entails $\mathsf{E}x\langle a\rangle\mathsf{true} \wedge x[a]x\mathsf{K_t}(q \wedge p)$. By the definition of $\mathsf{K_t}$, we obtain $x\mathsf{K_t}(\mathsf{E}x\langle a\rangle\mathsf{true} \wedge x[a]x\mathsf{K_t}(q \wedge p))$ at moment t. This trivially entails $(\bigvee a : x\mathsf{K_t}(\mathsf{E}x\langle a\rangle\mathsf{true} \wedge x[a]x\mathsf{K_t}(q \wedge p)))$. By axiom AX-KH-STRAT-5, $x\mathsf{K_t}(q \wedge p)$ entails $x\langle\!\langle\mathbf{do}(q)\rangle\!\rangle p$. Thus the previous expression yields $(\bigvee a : x\mathsf{K_t}(\mathsf{E}x\langle a\rangle\mathsf{true} \wedge x[a]x\langle\!\langle\mathbf{do}(q)\rangle\!\rangle p))$. Since $\neg q$ also holds at t, we have $x\langle\!\langle\mathbf{do}(q)\rangle\!\rangle p$ by axiom AX-KH-STRAT-5.

The case when $\tau = \langle a; \tau_1, \ldots, \tau_m\rangle$ follows quite simply by induction. The tree τ follows $\mathbf{do}(q)$ over a period iff (a) the period ends at the first occurrence of q and $x\mathsf{K_t}q$ also holds at that moment; and (b) the radix, a, is done in a prefix of the period and one of the τ_i follows $\mathbf{do}(q)$ over the rest of the period. By the inductive hypothesis applied to τ_i, $x\langle\!\langle\mathbf{do}(q)\rangle\!\rangle p$ holds at each of the moments at which a is performed. Axiom AX-KH-STRAT-5, then, entails that $x\langle\!\langle\mathbf{do}(q)\rangle\!\rangle p$ holds at the moment at which τ was begun. The rest of the argument is the same as for single-action trees. This proves completeness for strategies of the form, $\mathbf{do}(q)$. It also proves soundness of axiom AX-KH-STRAT-5 in the left to right direction.

For soundness of axiom AX-KH-STRAT-5 in the right to left direction, note that for the second disjunct of axiom AX-KH-STRAT-5, the empty tree makes $x\langle\!\langle\mathbf{do}(q)\rangle\!\rangle p$ hold wherever $x\mathsf{K_t}(q \wedge p)$ holds. For the first disjunct, let t be the given moment. Using the action for which the quantified expression holds and the trees corresponding to the occurrences of $x\langle\!\langle\mathbf{do}(q)\rangle\!\rangle p$ at the moments that action has been done, we can construct a tree at each alternative moment of t that makes $x\langle\!\langle\mathbf{do}(q)\rangle\!\rangle p$ true at t. This parallels the construction given in the proof of Theorem 4.1 in section 4.2, and is not repeated here.

Now let $Y = Y_1;Y_2$. If $\downarrow_t Y_1 = \mathbf{skip}$, the desired condition for axiom AX-KH-STRAT-2 follows trivially. But, if $\downarrow_t Y_1 \neq \mathbf{skip}$, the satisfaction condition for $x\langle\!\langle Y_1;Y_2\rangle\!\rangle p$ recursively depends on that for $x\langle\!\langle\uparrow_t Y_1;Y_2\rangle\!\rangle p$. Therefore, as in the proof of Theorem 4.3 in section 4.3, we use the fact that strategies are finitary. That is, they have a finite depth or, in other words, require only a finite number of applications of $\downarrow$ when performed at any time. Thus we can assume that for any moment t and strategy Y, such that $\downarrow_t Y \neq \mathbf{skip}$, the fragment of the model restricted to the execution of Y has a finite *depth*. If Y is followed at t, then the $\uparrow_t Y$ is followed at those moments where $\downarrow_t Y$ has just been performed. The depth of $\uparrow_t Y$ equals (depth of Y) $- 1$. The depth of **skip** is 0. Thus the depth is a metric to do the necessary induction on. The remainder of the proof is quite simple.

Finally, let $Y = \mathbf{while}\ q\ \mathbf{do}\ Y_1$. Axiom AX-KH-STRAT-4 captures the conditions for the $\lfloor\rfloor$ and $\lceil\rceil$ of Y. Using the above result for sequencing, and the fact that iterative strategies are finitary, we can perform induction on the depth of the strategy. This yields the desired result.

Thus for all cases in the definition of a strategy, $M \models_t x[\!\langle Y \rangle\!]p$ iff $x[\!\langle Y \rangle\!]p \in t$. This proves soundness and completeness of the proposed axiomatization. □

SEM-49. $M \models_t x\mathsf{K_{hs}}p$ iff $(\exists Y : M \models_t x[\!\langle Y \rangle\!]p)$

4.8 Results on Know-How

Just as for the case of ability, the strategic definition of know-how exploits its previous, reactive, definition. The following theorem states that strategic and reactive know-how are logically identical.

Theorem 4.19 $\mathsf{K_{hr}}p \equiv \mathsf{K_{hs}}p$

Proof.

The left to right direction is trivial. In the other direction, associate with $[\!\langle Y \rangle\!]p$ a fragment of the model whose root satisfies $[\!\langle Y \rangle\!]p$ and whose leaves satisfy the first occurrence of $\mathsf{K_t}p$ since the root. From this, construct a tree as required for $\mathsf{K_{hr}}p$. The details of this construction are identical to those in the proof of Theorem 4.4. □

It is often convenient to refer to reactive and strategic know-how jointly as $\mathsf{K_h}$. Below, we state and prove some results about know-how and its interaction with time and knowledge. These properties help us better delineate the concept of know-how as captured by the formalization presented above.

Theorem 4.20 $\mathsf{K_h}p \rightarrow \mathsf{K_t}\mathsf{K_h}p$

Proof.

Consider the two axioms for reactive know-how one by one. $\mathsf{K_t}p$ entails $\mathsf{K_t}\mathsf{K_t}p$. Thus, by the base axiom, we get $\mathsf{K_t}\mathsf{K_h}p$. For the inductive axiom, $(\bigvee a : \mathsf{K_t}(\mathsf{E}\langle a \rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_h}p))$ entails $(\bigvee a : \mathsf{K_t}\mathsf{K_t}(\mathsf{E}\langle a \rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_h}p))$ by introspection. And, that entails $\mathsf{K_t}(\bigvee a : \mathsf{K_t}(\mathsf{E}\langle a \rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_h}p))$. Thus we obtain $\mathsf{K_t}\mathsf{K_h}p$. □

Theorem 4.21 $\mathsf{K_h}p \rightarrow \mathsf{K_t}\mathsf{EF}p$; consequently, $\neg\mathsf{K_h}\mathsf{false}$

Proof.

It sufficient to consider the two axioms for reactive know-how. Since $p \rightarrow \mathsf{EF}p$ is valid, we have $\mathsf{K_t}p \rightarrow \mathsf{K_t}\mathsf{EF}p$ by axioms AX-BEL-3 and AX-BEL-4. This takes care of the base case. Also, $(\bigvee a : \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_t}\mathsf{EF}p))$ entails $\mathsf{K_t}\mathsf{E}\langle a\rangle\mathsf{EF}p$, which entails $\mathsf{K_t}\mathsf{EF}p$: this takes care of the inductive case.

It is a trivial consequence of the definitions of E and F that $\neg\mathsf{EFfalse}$ is valid. Hence, by definition of $\mathsf{K_t}$, $\neg\mathsf{K_t}\mathsf{EFfalse}$ is valid. Thus, $\neg\mathsf{K_h}\mathsf{false}$ holds. □

Theorem 4.22 $\mathsf{K_h}p \wedge \mathsf{K_t}\mathsf{AGK_t}(p\rightarrow q)\rightarrow \mathsf{K_h}q$

Proof. It sufficient to consider the two axioms for reactive know-how. From axiom Ax-Kh-React-1, $\mathsf{K_{hr}}p$ holds if $\mathsf{K_t}p$ holds. $\mathsf{K_t}\mathsf{AGK_t}(p\rightarrow q)$ entails $\mathsf{K_t}(p\rightarrow q)$. Thus, by axiom Ax-Bel-4, we have $\mathsf{K_t}q$, which by axiom Ax-Kh-React-1 entails $\mathsf{K_h}q$. From axiom Ax-Kh-React-2, $\mathsf{K_{hr}}p$ holds if $(\bigvee a : \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_{hr}}p))$. But, as a consequence of the definitions of A and G, we have that for all actions, a, $\mathsf{K_t}\mathsf{AGK_t}(p\rightarrow q)\rightarrow \mathsf{K_t}\mathsf{A}[a]\mathsf{AGK_t}(p\rightarrow q)$. Therefore, we have $(\bigvee a : \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a](\mathsf{K_{hr}}p \wedge \mathsf{AGK_t}(p\rightarrow q))))$. By the inductive hypothesis, we can conclude $(\bigvee a : \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true}\wedge\mathsf{A}[a]\mathsf{K_{hr}}q))$ which, by axiom Ax-Kh-React-2, entails $\mathsf{K_{hr}}q$, as desired. □

For the next theorem, we need to add an assumption about knowledge. The relations, $\mathbf{B}(x)$, which were assumed to be reflexive and transitive in section 2.6, are now additionally assumed to be symmetric. This validates the following axiom of negative introspection [Chellas, 1980].

Ax-Bel-5. $\neg x\mathsf{K_t}p\rightarrow x\mathsf{K_t}\neg x\mathsf{K_t}p$

I shall assume this axiom in the rest of this section.

Theorem 4.23 $\mathsf{K_h}\mathsf{K_h}p\rightarrow \mathsf{K_h}p$

Proof. Using the definition of reactive know-how, construct a single tree out of the trees for $\mathsf{K_{hr}}\mathsf{K_{hr}}p$. For the base case, simply use axiom Ax-Kh-React-1. $\mathsf{K_{hr}}\mathsf{K_{hr}}p$ holds if $\mathsf{K_{hr}}p$ does, which trivially implies $\mathsf{K_{hr}}p$. For the inductive case, $\mathsf{K_{hr}}\mathsf{K_{hr}}p$ holds if $(\bigvee a : \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_{hr}}\mathsf{K_{hr}}p))$. By the inductive hypothesis, we obtain $(\bigvee a : \mathsf{K_t}(\mathsf{E}\langle a\rangle\mathsf{true} \wedge \mathsf{A}[a]\mathsf{K_{hr}}p))$ which, by axiom-Ax-Kh-React-2 implies $\mathsf{K_{hr}}p$, as desired. □

This seems intuitively quite obvious: if an agent can ensure that he will be able to ensure p, then he can already ensure p. But see the discussion following Theorem 4.25.

If p is "returns(Halley's comet)," then assuming the agent knows that he will come to know that it has returned, $\mathsf{K_h}p$ holds. The condition here is stronger than for ability, but is perhaps too weak relative to our pretheoretic intuitions. For this reason, it is useful to consider an alternative notion, ***proper know-how***, notated as K^p_h that prevents this inference. Its semantic condition and axiomatization are as follows.

SEM-50. $M \models_t x\mathsf{K}^p_\mathsf{h}p$ iff $M \models_t (x\mathsf{K}_\mathsf{h}p) \wedge (\neg x\mathsf{K}_\mathsf{t}\mathsf{AF}x\mathsf{K}_\mathsf{t}p)$

AX-AB-2. $\mathsf{K}^p_\mathsf{h}p \equiv (\mathsf{K}_\mathsf{h}p \wedge \neg\mathsf{K}_\mathsf{t}\mathsf{AFK}_\mathsf{t}p)$

Theorem 4.24 $\neg\mathsf{K}^p_\mathsf{h}\mathsf{true}$

Proof. We trivially have $\mathsf{K}_\mathsf{t}\mathsf{true}$, which entails $\mathsf{AFK}_\mathsf{t}\mathsf{true}$, which entails $\mathsf{K}_\mathsf{t}\mathsf{AFK}_\mathsf{t}\mathsf{true}$. By axiom AX-AB-2, that entails $\neg\mathsf{K}^p_\mathsf{h}\mathsf{true}$. □

Therefore, despite Theorem 4.22, the corresponding statement for K^p_h fails. Indeed, we have

Theorem 4.25 $\mathsf{K}_\mathsf{t}p \rightarrow \neg\mathsf{K}^p_\mathsf{h}p$

Proof. $\mathsf{K}_\mathsf{t}p$ entails $\mathsf{K}_\mathsf{t}\mathsf{K}_\mathsf{t}p$, which entails $\mathsf{K}_\mathsf{t}\mathsf{AFK}_\mathsf{t}p$. Hence, $\neg\mathsf{K}^p_\mathsf{h}p$. □

Theorem 4.25 states that if p is known then the agent does not properly know how to achieve it. By substitution, we obtain $\mathsf{K}^p_\mathsf{h}p \rightarrow \neg\mathsf{K}^p_\mathsf{h}\mathsf{K}^p_\mathsf{h}p$, whose contrapositive is $\mathsf{K}^p_\mathsf{h}\mathsf{K}^p_\mathsf{h}p \rightarrow \neg\mathsf{K}^p_\mathsf{h}p$. This is in direct opposition to Theorem 4.23 for K_h, and is surprising. The explanation for this observation is that when we speak of nested know-how, which we do not do often in natural language, we use two different senses of know-how: K^p_h for the inner one and K_h for the outer one. Thus the correct translation is $\mathsf{K}_\mathsf{h}\mathsf{K}^p_\mathsf{h}p$, which entails $\mathsf{K}_\mathsf{h}p$, as desired.

Theorem 4.26 $\mathsf{K}_\mathsf{h}\mathsf{K}^p_\mathsf{h}p \rightarrow \mathsf{K}_\mathsf{h}p$

Proof. $\mathsf{K}_\mathsf{h}\mathsf{K}^p_\mathsf{h}p \equiv \mathsf{K}_\mathsf{h}(\mathsf{K}_\mathsf{h}p \wedge \neg\mathsf{K}_\mathsf{t}\mathsf{AFK}_\mathsf{t}p)$, by definition of K^p_h. Since $\mathsf{K}_\mathsf{t}\mathsf{AGK}_\mathsf{t}((r \wedge q) \rightarrow r)$, we can apply Theorem 4.22. Thus the left hand side implies $\mathsf{K}_\mathsf{h}\mathsf{K}_\mathsf{h}p$, which by Theorem 4.23 implies $\mathsf{K}_\mathsf{h}p$. □

4.9 Conclusions

I presented two sound and complete logics for ability and know-how that were developed in the same framework as used for intentions earlier in this work. This formalization reveals interesting properties of ability and know-how and helps clarify some of our intuitions about them. Capturing these intuitions is an important first step in applying these concepts rigorously in the design and analysis of intelligent agents. The process of formalization helps uncover certain technical nuances that might never have been brought to the fore otherwise. This is a valuable service. Interestingly, many proofs from the proposed axioms turn out to be much simpler than those that might be given using purely model-theoretic reasoning.

For some purposes, the proper notion of know-how may be preferred, since it excludes cases where the given condition holds and is known to hold already. For most purposes, however, the general notion of know-how is preferred. For example, a household robot should know how to get upstairs when called; in this case, there is no problem if it knows that it is already upstairs.

Of special technical interest are the operators ⦃ ⦄ and [⟨ ⟩], which differ from those in standard dynamic logic. These operators provide a viable formal notion with which to capture the ability and know-how of an agent whose behavior is abstractly characterized in terms of strategies. The differences between the reactive and strategic definitions of these concepts lie mainly in the complexity of the agents to whom they may be attributed. As explained in section 2.5, the strategic definition lets an agent be specified and reasoned about using something akin to macros over reactive actions.

This approach complements previous work on knowledge and action [Moore, 1984; Morgenstern, 1987] in some respects. The details of the conditions that are achieved have been abstracted out, but could be filled in. For example, definite descriptions could be included easily: an agent's strategy can be **do**(q), where q stands for "dialed(combination of the safe)."

At this point, I have formalized intentions and know-how within the same general framework of actions and time. Now the question arises as to whether we can pull these formalizations together to prove the kinds of results we are most interested in. The next chapter answers this question with a yes. And the chapter after the next reaffirms that for the problem of specifying communications among agents.