Chapter 2

Technical Framework

In light of the goals of this research, there is need for a formal model that incorporates at least the following features:

- time,
- concurrent actions by more than one agent,
- a notion of *choice* so that intentions can be captured, and
- a notion of *control* so that know-how can be captured.

Just such a model is developed here. Only the basic model and formal language are described in this chapter. Further components of the model and extensions to the language are motivated and introduced as needed in succeeding chapters.

The formal model of this work is based on possible worlds, which are well-known from modal logic [Chellas, 1980]. The possible worlds here, in the technical sense of the term, are possible *moments*. That is, each moment plays the role of a *world* in standard modal logic. However, I shall use *moment* in the technical sense and *world* only informally. Each moment is associated with a possible state of the world, which is identified by the atomic conditions or propositions that hold at that moment (atomic propositions are explained in section 2.1.1). A condition is said to be achieved when a state in which it holds is attained. At each moment, environmental events, and agents' actions occur. The same physical state may occur at different moments. A *scenario* at a moment is any maximal set of moments containing the given moment, and all moments in its future along some particular branch.

Figure 2.1 shows a schematic picture of the formal model. Each point in the picture is a moment. There is a partial order on moments that denotes

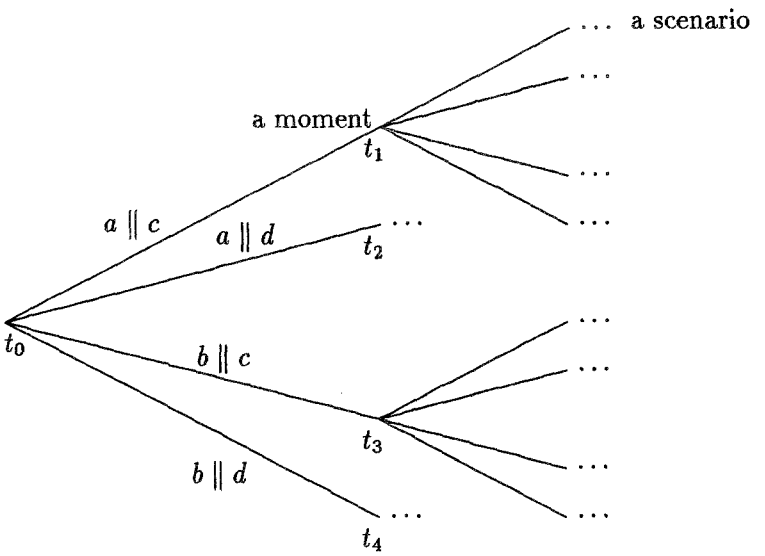

Figure 2.1: The Formal Model

temporal precedence: following usual conventions, time is shown flowing towards the right of the picture. In general, time may branch into the future. Indeed, in any interesting application, it will branch into the future. Since the past is determined at each moment, the temporal precedence relation is taken to be linear in the past. The ignorance that some agent may have about the past is captured by the general mechanism of beliefs, which I discuss in section 2.6. Figure 2.1 is labeled with the actions of two agents. Each agent influences the future by acting, but the outcome also depends on other events. For example, in Figure 2.1, the first agent can constrain the future to some extent by choosing to do action a or action b. If he does action a, then the world progresses along one of the top two branches out of t_0; if he does action b, then it progresses along one of the bottom two branches. However, the agent cannot control what exactly transpires. For example, if he does action a, then whether t_1 or t_2 becomes the case cannot be controlled by him, but rather depends on the actions of the second agent. Both *choice* and *limited control* can thus be captured in this model.

The important intuition about actions is that they are package deals. They correspond to the granularity at which an agent can make his choices. In the above example, the first agent can choose between t_1 and t_2, on the

one hand, and between t_3 and t_4, on the other hand. However, he can choose neither between t_1 and t_2, nor between t_3 and t_4.

It is useful for capturing many of our intuitions about the choices and abilities of agents to identify one of the scenarios beginning at a moment as the *real* one. This is the scenario on which the world can be seen to have progressed, assuming it was in the state denoted by the given moment. The real scenario is determined by the choices of the agents and events in the environment. Thus the reality of a scenario is relativized to the moment at which it is considered. In classical modal logic, a distinguished world is sometimes identified as being the real one; however, unlike in the present approach, that world is the unique real world for the entire model. Here, the real scenarios at different moments may have no moment in common. Of course, the real scenarios for moments on the real scenario of a preceding moment must be suffixes of that scenario.

The rest of this chapter proceeds as follows. I present the core formal language, formal model, and semantics in successive subsections of the next section. Section 2.2 is about the temporal and action operators that I define. The temporal operators have standard definitions but the actions operators are quite novel. Their semantics involve subtleties, so that the same definitions can apply in a variety of models, from discrete to continuous time. Section 2.3 motivates and formalizes the several coherence constraints needed in this approach. These are required to eliminate counterintuitive models and simplify the presentation, so that expected results can still be obtained. Section 2.4 presents some simple results relating the temporal and the action operators: these show why the subtleties of some of our definitions and some of the coherence constraints were required. Section 2.5 presents *strategies* as abstract descriptions of actions necessary to understanding complex systems. Section 2.6 presents a standard modal view of belief and knowledge, which are required to complete the present theory. Section 2.7 discusses theories of actions in linguistics, philosophy, and artificial intelligence. Lastly, section 2.8 briefly gives a rationale for why the simpler approach of qualitative temporal logic is appropriate for our purposes.

2.1 The Core Formal Framework

2.1.1 The Formal Language

The proposed formal language, $\mathcal{L}$, is based on CTL*, which is a well-known propositional branching-time logic [Emerson, 1990]. $\mathcal{L}$ also includes the operators $[]$ and $\langle\rangle$, and permits quantification over basic action symbols. The

operators [] and $\langle\rangle$ depend on basic actions. Formally, $\mathcal{L}$ is the minimal set closed under the following rules. Here $\mathcal{L}_s$ is the set of "scenario-formulae," which is used as an auxiliary definition. The formulae in $\mathcal{L}$ are evaluated relative to moments; those in $\mathcal{L}_s$ are evaluated relative to scenarios and moments. In the following,

- Φ is a set of atomic propositional symbols,
- $\mathcal{A}$ is a set of agent symbols,
- $\mathcal{B}$ is a set of basic action symbols, and
- $\mathcal{X}$ is a set of variables.

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

SYN-2. $p, q \in \mathcal{L}$ implies that $p \wedge q \in \mathcal{L}$

SYN-3. $p \in \mathcal{L}$ implies that $\neg p \in \mathcal{L}$

SYN-4. $\mathcal{L} \subseteq \mathcal{L}_s$

SYN-5. $p, q \in \mathcal{L}_s$ implies that $p \wedge q \in \mathcal{L}_s$

SYN-6. $p \in \mathcal{L}_s$ implies that $\neg p \in \mathcal{L}_s$

SYN-7. $p \in \mathcal{L}_s$ implies that $\mathsf{A}p, \mathsf{R}p \in \mathcal{L}$

SYN-8. $p \in \mathcal{L}$ implies that $\mathsf{P}p \in \mathcal{L}$

SYN-9. $p \in \mathcal{L}$ and $a \in \mathcal{X}$ implies that $(\bigvee a : p) \in \mathcal{L}$

SYN-10. $p \in (\mathcal{L}_s - \mathcal{L})$ and $a \in \mathcal{X}$ implies that $(\bigvee a : p) \in \mathcal{L}_s$

SYN-11. $p, q \in \mathcal{L}_s$ implies that $p \mathsf{U} q \in \mathcal{L}_s$

SYN-12. $p \in \mathcal{L}_s$, $x \in \mathcal{A}$, and $a \in \mathcal{B}$ implies that $x[a]p, x\langle a\rangle p, x[\![a]\!]p \in \mathcal{L}_s$

The atomic propositional symbols denote the primitive propositions or conditions of our models. Conditions of interest to the given application, e.g., whether a given runway is busy or free, are mapped to different propositional symbols. Similarly, the basic actions symbols denote the elementary actions, e.g., landing or taking off, that are important in the given application. Choosing the right atomic propositions and basic actions is an important component of constructing useful formal models of a given applications. However, we shall not study this task in any detail in the present work.

2.1.2 The Formal Model

Let $M = \langle F, N \rangle$ be a model for the language $\mathcal{L}$, where $F = \langle \mathbf{T}, <, \mathbf{A} \rangle$ is a frame, and $N = \langle [\![ ]\!], \mathbf{Y}, \mathbf{B}, \mathbf{R} \rangle$ is an interpretation. Here $\mathbf{T}$ is a set of possible moments ordered by $<$. $\mathbf{A}$ assigns agents to different moments; i.e., $\mathbf{A} : \mathbf{T} \mapsto \wp(\mathcal{A})$. As described below, $[\![ ]\!]$ assigns intensions to atomic propositions and to pairs of agent symbols and action symbols. $\mathbf{Y}$ assigns *strategies* (to be defined in section 2.5) to the agents at each moment. $\mathbf{B}$ assigns alternative moments to the agents at each moment. As explained in section 2.6, these are the moments that denote states of affairs that the agents imagine to be the case. $\mathbf{B}$ is used to give the semantics of belief and know-that. $\mathbf{R}$ assigns a scenario to each moment, which is interpreted as the *real* scenario at that moment.

The relation, $<$, which is a subset of $\mathbf{T} \times \mathbf{T}$, is a strict partial order. It models time as linear in the past. Time may or may not branch in the future; however, if it branches at a moment, the branches cannot join again. Indeed, if two branches join at some moment, then the linear past requirement would be violated at that moment. This makes it possible to identify periods (defined below) uniquely by their endpoints. Formally, the following properties hold of the relation $<$.

- *Transitivity:* $(\forall t, t', t'' \in \mathbf{T} : (t < t' \wedge t' < t'') \Rightarrow t < t'')$
- *Asymmetry:* $(\forall t, t' \in \mathbf{T} : t < t' \Rightarrow t' \not< t)$
- *Irreflexivity:* $(\forall t \in \mathbf{T} : t \not< t)$

It may be intuitively helpful for the reader to think of the connected components of $\mathbf{T}$ induced by $<$ as different possible worlds, in the classical sense, i.e., as entities that evolve over time. By the definition of connectedness, the actions of agents cannot begin at a moment in one such component and end in another. However, more than one such component is needed for many of the technical definitions given here, since they explicitly involve alternative states of affairs. The reader may consult [Emerson, 1990] for an introduction to temporal logic and to models of time and [Chellas, 1980] for a textbook level introduction to modal logic.

In earlier versions of this work, I also assumed that models were linear past. However, that assumption was needed only to simplify the notation for periods. If the past at each moment is linear, then branches of time never merge. Hence, periods of time can be uniquely identified by their beginning and ending moments. Since this assumption had no substantive effect on the theory, I have now decided to remove it altogether. Indeed, in determining compact

representations for the proposed models, it would help to collapse moments that were in some sense the same: this process would result in models that were directed graphs rather than trees and possibly be directed graphs with cycles. This point will become clearer in section 2.3 in the discussion of weak determinism.

A scenario at a moment is any single branch of the relation $<$ that begins at the given moment, and contains *all* moments in some linear subrelation of $<$. Different scenarios correspond to different ways in which the world may develop in the future, as a result of the actions of agents and events in the environment. Only one scenario can be realized. This property is not used in the formal theory in Chapters 3 and 4, but is used in Chapter 5. It is intuitively useful throughout for understanding many of the definitions. Formally, a scenario at moment t is a set $S \subseteq \mathbf{T}$ of which the following conditions hold.

- *Rootedness:* $t \in S$
- *Linearity:* $(\forall t', t'' \in S : (t' = t'') \vee (t' < t'') \vee (t'' < t'))$
- *Relative Density:* $(\forall t', t'' \in S, t''' \in \mathbf{T} : (t' < t''' < t'') \Rightarrow t''' \in S)$
- *Relative Maximality:* $(\forall t' \in S, t'' \in \mathbf{T} : (t' < t'') \Rightarrow (\exists t''' \in S : (t' < t''') \wedge (t''' \not< t'')))$

  Intuitively, maximality means that if it is possible to extend the scenario S (here to t''), then it is extended, either to t'' (when $t''' = t''$), or along some other branch. Note that this assumption by itself does not entail that time be eternal. That is assumed separately in coherence constraint Coh-2 below.

$\mathbf{S}_t$ is the set of all scenarios at moment t. Since each scenario at a moment is rooted at that moment, the sets of scenarios at different moments are disjoint, that is, $t \neq t' \Rightarrow \mathbf{S}_t \cap \mathbf{S}_{t'} = \emptyset$. If t' is such that $t < t'$, then for every scenario, $S' \in \mathbf{S}_{t'}$, there is a scenario, S, such that $S' \subset S$ and $S \in \mathbf{S}_t$. Conversely, for every scenario $S \in \mathbf{S}_t$, for each moment $t' \in S$, there is a scenario $S' \in \mathbf{S}_{t'}$, such that $S' \subseteq S$.

$[S; t, t']$ denotes a period on scenario S from t to t', inclusive, i.e., the subset of S from t to t'. Thus, if $[S_0; t, t'] \subseteq S_1$, then $[S_0; t, t'] = [S_1; t, t']$ (because they are both the same set of moments). However, in general, $[S_0; t, t'] \neq [S_1; t, t']$. For notational simplicity, $[S; t, t']$ presupposes $t, t' \in S$ and $t \leq t'$.

The notion of basic actions needed for a general theory of intentions and know-how is different from the one that is traditionally assumed. Traditionally, only actions of unit length are considered and only one agent is

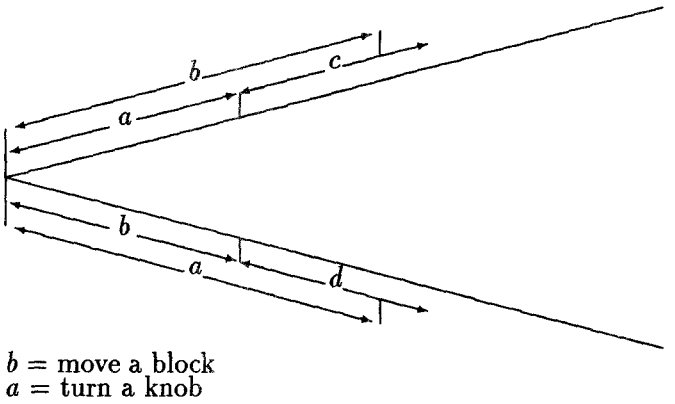

Figure 2.2: Actions: Nonsynchronized and of Varying Durations

assumed to act at a time (e.g., [Cohen & Levesque, 1990]). These assumptions are both somewhat restrictive. I submit that the key intuition behind basic actions is that they are done by the agent with a single *choice*, irrespective of the duration for which they last. More than one agent may act simultaneously. The set of actions available to an agent can be different at different moments. Basic actions may have different durations relative to one another in different scenarios, including those scenarios that begin at the same moment. For example, the actions of moving a block may take more or less time than the action of turning a knob, depending on how another agent obstructs one of these actions. Such a case is diagramed in Figure 2.2, which also shows that actions may begin and end at different moments.

The intension, $[\![\,]\!]$, gives the semantic content of some of the symbols of the formal language. Intensions, to be distinguished from intentions, are well-known from modal logic. The intension of an atomic proposition is the set of moments at which it is true. The intension of an action symbol a is, for each agent symbol x, the set of periods in the model in which an instance of a is done by x. Formally, $[\![\,]\!]$ is the union of two functions of types $\Phi \mapsto \wp(\mathbf{T})$ and $\mathcal{A} \times \mathcal{B} \mapsto \wp(\wp(\mathbf{T}) \times \mathbf{T} \times \mathbf{T})$, respectively. Thus $t \in [\![p]\!]$ means that p is true at moment t; and, $[S; t, t'] \in [\![a]\!]^x$ means that agent x is performing action a on

S from moment t to moment t'. As explained in constraints COH-1 and COH-3 of section 2.3, when $[S;t,t'] \in [\![a]\!]^x$, t' corresponds to the ending of a, but t does not correspond to the initiation of a. This is because a may already be in progress before t. All basic actions take time. That is, if $[S;t,t'] \in [\![a]\!]^x$, then $t < t'$. The superscript denoting the agent is elided when it can be understood from the context.

It is useful for some of the definitions that follow to extend the definition of intension of an action in the following way. Let $s = a_0, \ldots, a_{m-1}$ be a sequence of actions of x. Then $[\![s]\!] = \{[S;t,t'] | (\exists t_0 \leq \ldots \leq t_m : t = t_0 \wedge t' = t_m \wedge (\forall j : j \in [1 \ldots m] \Rightarrow [S;t_{j-1},t_j] \in [\![a_{j-1}]\!]^x))\}$. That is, $[\![s]\!]$ is the set of periods over which sequence s is done. In other words, $[S;t,t'] \in [\![s]\!]$ means that s begins at t and ends at t'.

Finally, the component $\mathbf{R}$ of the model simply assigns a scenario to each moment. Therefore, it is of the type $\mathbf{T} \mapsto \wp(\wp(\mathbf{T}))$. In particular, $\mathbf{R}(t) \in \mathbf{S}_t$; in other words, reality is possible.

Restrictions on the intension, $[\![\,]\!]$, can be used to express the limitations of agents as well as how the actions of agents may depend on certain conditions holding at the moments at which they are begun and how they may interfere with the actions of others. For example, the proposition that Bob cannot pick up three blocks at once can be modeled by making the intension of his picking up three blocks empty. Similarly, the constraint that at most one person can enter the elevator at a time can be modeled by requiring that the intersection of the intensions of the actions of two persons entering it be empty. Each of these restrictions may be made contingent upon other conditions. Relations between beliefs and actions will be considered in subsequent chapters.

Note that, intuitively, if an agent is deemed to be performing an action at a moment, he must be alive then. Thus the births and deaths of agents can be accounted for in the formal model: all the non-wait actions performed by an agent occur between the moments of his birth and death; however, this observation is not used in any part of the formal theory here. If we wished to incorporate this in the theory, we would have to restrict different conditions, e.g., constraint COH-5 below, to apply only to live agents.

2.1.3 Semantics

The semantics of sentences, i.e., formulae, in the formal language is given relative to a model, as defined above, and a moment in that model. $M \models_t p$ expresses "M satisfies p at t." This is the main notion of satisfaction here.

For formulae in $\mathcal{L}_s$, it is useful to define an auxiliary notion of satisfaction, $M \models_{S,t} p$, which expresses "M satisfies p at moment t on scenario S." For notational simplicity, $M \models_{S,t} p$ is taken to entail that $t \in S$. We say p is *satisfiable* iff for some M and t, $M \models_t p$; we say p is *valid* in M iff it is satisfied at all moments in M. The satisfaction conditions for the temporal operators are adapted from those in [Emerson, 1990]. It is assumed that each action symbol is quantified over at most once in any formula. Below, $p|^a_b$ is the formula resulting from the substitution of all occurrences of a in p by b. Formally, we have the following definitions:

SEM-1. $M \models_t \psi$ iff $t \in [\![\psi]\!]$, where $\psi \in \Phi$

SEM-2. $M \models_t p \wedge q$ iff $M \models_t p$ and $M \models_t q$

SEM-3. $M \models_t \neg p$ iff $M \not\models_t p$

SEM-4. $M \models_t \mathsf{A}p$ iff $(\forall S : S \in \mathbf{S}_t \Rightarrow M \models_{S,t} p)$

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

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

SEM-7. $M \models_t (\bigvee a : p)$ iff $(\exists b : b \in \mathcal{B}$ and $M \models_t p|^a_b)$, where $p \in \mathcal{L}$

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

SEM-9. $M \models_{S,t} p\mathsf{U}q$ iff $(\exists t' : t \leq t'$ and $M \models_{S,t'} q$ and $(\forall t'' : t \leq t'' \leq t' \Rightarrow M \models_{S,t''} p))$

SEM-10. $M \models_{S,t} x[a]p$ iff $(\exists t' \in S : [S; t, t'] \in [\![a]\!]^x) \Rightarrow (\exists t' \in S : [S; t, t'] \in [\![a]\!]$ and $(\exists t'' : t < t'' \leq t'$ and $M \models_{S,t''} p))$

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

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

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

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

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

Two useful abbreviations are **false** $\equiv (p \wedge \neg p)$, for any $p \in \Phi$, and **true** $\equiv \neg$**false**. The definition of $\mathcal{L}$ as given so far is by no means complete; additions to it are defined in subsequent sections of this chapter and in the succeeding chapters after further intuitive motivation.

It should be clear from the above that in this work, the term *model* is used in the standard sense of logic. Statements of fact, including statements of what a given agent intends or believes, are evaluated with respect to a model that consists of different possible moments. For example, whether the statement "it is raining" is true in the model or not depends only on the moment relative to which this statement is evaluated, *not* on the beliefs of any agent. Similarly, whether an agent intends something is to be differentiated from the question of whether he (or someone else) believes that he intends something. This point is obvious and standard, but can cause grave misconceptions if not kept in mind.

2.2 Temporal and Action Operators: Discussion

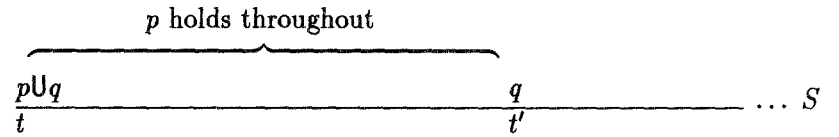

Figure 2.3: Temporal Operators: $p\mathsf{U}q$

Figure 2.4: Temporal Operators: $\mathsf{F}p$

The formula $p\mathsf{U}q$ is true at a moment t on a scenario, iff q holds at a future moment on the given scenario and p holds on all moments between t and the selected occurrence of q. The formula $\mathsf{F}p$ means that p holds sometimes in the future on the given scenario and abbreviates **true**Up. The formula $\mathsf{G}p$ means

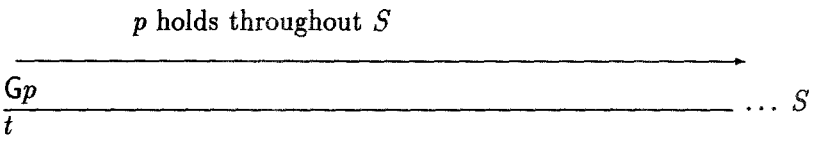

Figure 2.5: Temporal Operators: Gp

that p always holds in the future on the given scenario; it abbreviates $\neg \mathsf{F} \neg p$. These definitions are illustrated in Figures 2.3, 2.4, and 2.5. The formula $\mathsf{P}p$ denotes p held at some moment in the past. The boolean or propositional logic operators, $\wedge$ and $\neg$, are used to compose formulae in the usual manner. Implications ($p \rightarrow q$) and disjunctions ($p \vee q$) of formulae are defined as the usual abbreviations.

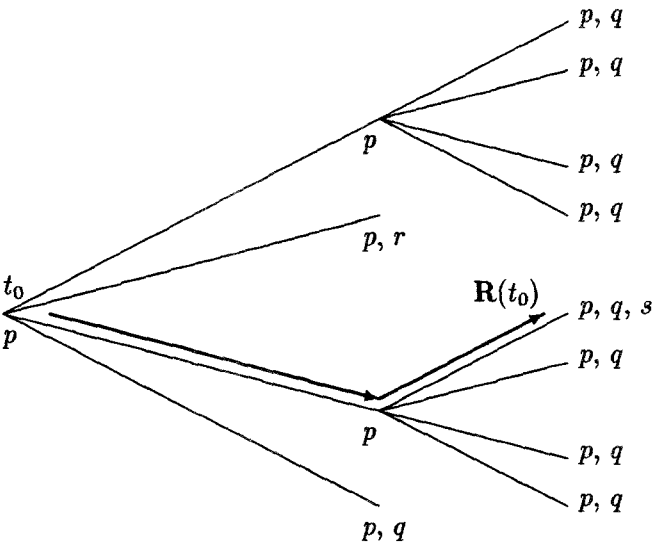

Figure 2.6: Temporal Operators: A, E, R

The branching-time operator, A, denotes "in *all* scenarios at the present moment." Here "the present moment" refers to the moment at which a given formula is evaluated. A useful abbreviation is E, which denotes "in *some* scenario at the present moment." In other words, $\mathsf{E}p \equiv \neg \mathsf{A} \neg p$. The reality

operator, R, captures the notion of what will really be the case. It denotes "in the *real* scenario at the given moment." Now consider Figure 2.6. In that figure, assume that p holds at all moments in the future of those shown. Then at t_0, AGp holds, because p holds at all moments on all scenarios beginning at that moment. Similarly, EFr, AFq, and EGp also hold at that moment. The arrow marks the real scenario at t_0. Therefore, RFs also holds at t_0.

I introduce two new modalities for actions. The proposed definitions loosely follow standard dynamic logic [Kozen & Tiurzyn, 1990], but differ in several important respects. For an action symbol a, an agent symbol x, and a formula p, $x[a]p$ holds on a given scenario S and a moment t on it, iff, if x performs a on S starting at t, then p holds at some moment while a is being performed. The formula $x\langle a\rangle p$ holds on a given scenario S and a moment t on it, iff, x performs a on S starting at t and p holds at some moment while a is being performed. The agent symbol is elided when it is obvious from the context. These definitions require p to hold at any moment in the (left-open and right-closed) period in which the given action is being performed. Thus they are weaker than possible definitions that require p to hold at the moment at which the given action completes.

In assigning meanings to $x[a]p$ and $x\langle a\rangle p$, it is essential to allow the condition to hold at any moment in the period over which the action is performed. This is because we are not assuming that time is discrete or that all actions are of equal durations and synchronized to begin and end together. Intuitively, if we insisted that the relevant condition hold at the end of the action, then an agent could effectively leap over a condition. In that case, even if a condition occurs while an action is performed, we may not have $x\langle a\rangle p$. For example, if p is "the agent is at the equator," and the agent performs the action of hopping northwards from just south of the equator, he may end up north of the equator without ever (officially) being at it. That would be quite unintuitive. For this reason, the present definitions are preferred although as a consequence of them, the operators $\langle\,\rangle$ and $[\,]$ are not formal duals of each other. But this is made up for not only by having a more intuitive set of definitions, but also by the natural axiomatization for know-how in section 4.2 that the chosen definitions facilitate. Further, the present definitions enable the right relationship between $\langle\,\rangle$ and U to be captured. Recall from above that pUq considers all moments between the given moment and the first occurrence of q, not just those at which different actions may end.

Further, $x\langle\!\langle a\rangle\!\rangle p$ holds on a scenario S and moment t if x performs action a starting at t and p holds in some initial subperiod of the period over which a is done. This operator is necessary to relate actions with time for the following reason. In dense models, actions happen over periods which

contain moments between their endpoints. Even in discrete models whose actions are not all of unit length, actions can happen over nonempty periods. Consequently, if s is done at t and q holds at an internal moment of a and p holds throughout, then $p\mathsf{U}q$ holds at t. But absent the $\|\langle\rangle\|$ operator, we cannot characterize $p\mathsf{U}q$ recursively in terms of actions. One useful characterization is given in section 2.4: this helps in giving the fixed point semantics of the temporal operators, which is essential to computing them efficiently.

The above dynamic modalities yield scenario-formulae, which can be combined with the branching-time operators, A, E, and R. Thus $\mathsf{A}[a]p$ denotes that on all scenarios S at the present moment, if a is performed on S, then p holds at some moment on S between the present moment and the moment at which a is completed. Similarly, $\mathsf{E}\langle a\rangle p$ denotes that a is being done on some scenario at the present moment and that on this scenario p holds at some moment between the present moment and the moment at which a is completed. In other words, $\mathsf{A}[a]p$ corresponds to the necessitation operator and $\mathsf{E}\langle a\rangle p$ to the possibility operator in dynamic logic.

One difference with the definition of the operators in dynamic logic is that here the relevant condition can hold at any moment *during* the course of the action. This is in accordance with time not being constrained to be discrete. In dynamic logic, actions are modeled as pairs of states, which usually are discrete snapshots of the world. Therefore, in that context, it does not make sense to talk of intermediate states. But in the model here, moments are defined independently of specific actions. Of course, the present definitions specialize correctly to discrete models of time in which all actions are of equal duration and synchronized. Furthermore, the operators [] and $\langle\rangle$ are evaluated on scenarios. The advantage of doing so is that it simplifies the connection with branching-time logic. It also allows us to express conditions like $\mathsf{A}x\langle a\rangle p$, which have no correlate in dynamic logic. In effect, $\mathsf{A}x\langle a\rangle p$ means that a is the only action (of the agent x) that can be performed at the given moment (i.e., the moment where this formula is evaluated) and p is the condition that results from doing a.

Existential quantification over basic actions is a useful feature for our purposes. Of the several basic actions that an agent may do at a given moment, we would often like to restrictively talk of the subset of actions that have some interesting property. Indeed, we need something like this to formally express the idea of *choice:* an agent may be able to do several actions, but would, in fact, choose to do one. For each action that an agent may choose to do, there is a set of scenarios over which those actions are attempted and done. Usually, this set of scenarios is not a singleton, because the actions of other agents and environmental events would contribute to determining which scenario is finally

selected. The agent constrains the scenario that would be realized by doing some action, but the one that is, in fact, realized also depends on events beyond his direct control. It sometimes helps to use the dual of this operator, universal quantification over actions, as well.

2.3 Coherence Constraints

For the models introduced above to be coherent and useful as models of actions and time for reasoning about multiagent systems, they must satisfy a number of technical constraints. Many of these are motivated and formalized below.

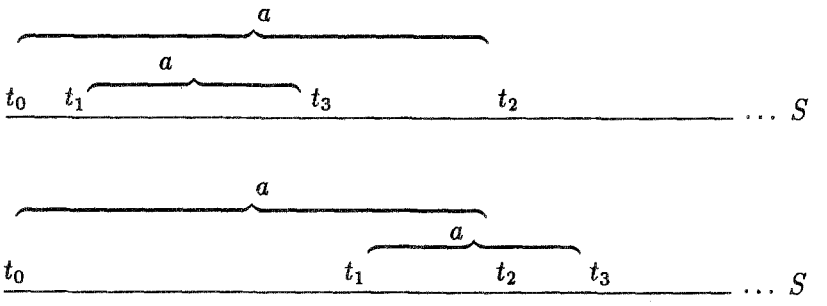

Figure 2.7: Cases Disallowed by Action Uniqueness

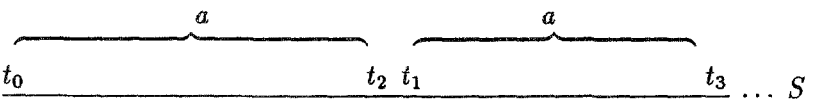

Figure 2.8: Case Allowed by Uniqueness of Termination of Actions

COH-1. **Uniqueness of Termination of Actions:** Starting at any given moment, each action can be performed in at most one way on any given scenario. In other words, for any action a, scenario S, and moments t_0, t_1, t_2, t_3 in S, we have that $[S; t_0, t_2] \in [\![a]\!]$ and $[S; t_1, t_3] \in [\![a]\!]$ implies that, if $t_0 \leq t_1 < t_2$, then $t_2 = t_3$. This might seem too elementary to be mentioned explicitly. However, it is needed to

exclude ill-formed models in which an action does not have a unique moment of ending. The two main classes of such ill-formed models are diagramed in Figure 2.7. If an agent performs an action and then repeats it, the repetition counts as a separate instance, because it has a distinct starting moment. Such a case is shown in Figure 2.8; this constraint allows $t_1 = t_2$ in that figure. Note that the present constraint only states that each action has a unique endpoint. It permits several different actions with possibly distinct endpoints to happen simultaneously. In discrete models with unit length actions, both endpoints are necessarily unique; here only the termination point is assumed to be unique.

COH-2. **Eternity:** At each moment, there is a future moment available in the model. Or, time never comes to an end. Formally, $(\forall t : (\exists t' : t < t'))$. In conjunction with the maximality property of scenarios, this is equivalent to the statement that there is always a scenario available along which the world may evolve. This statement is intuitively helpful to remember, even though its formalization is more complex than the above version: $(\forall t : (\exists S : S \in \mathbf{S}_t \text{ and } (\exists t' : t' \in S \text{ and } t < t')))$.

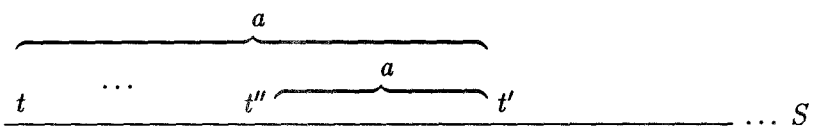

Figure 2.9: Actions in Progress

COH-3. **Actions in Progress:** It is also useful in relating moments with actions to impose the following condition on the models: $[S; t, t'] \in [\![a]\!] \Rightarrow (\forall t'' : t \leq t'' < t' \Rightarrow [S; t'', t'] \in [\![a]\!])$. This constraint allows us to talk of an agent's actions at any moment at which they are happening, not just where they begin. Of course, in discrete models with unit length actions, there is no moment properly between t and t', so our constraint holds vacuously of such models. However, note that in accordance with condition COH-1, actions begun at a moment still have a unique ending moment. As a result of this constraint, the operators [] and $\langle\rangle$ on actions and propositions, which were defined informally in section 2.1.1, behave properly at all scenarios and moments in the model. For example, if an agent can achieve a condition

by performing some action, then he can also achieve it while in the process of performing that action.

The "real" choice is exercised by the agent when he begins a particular action; the present constraint may be understood as stating that until an initiated action completes, the agent implicitly reaffirms his choice. Figure 2.9 shows how this constraint causes the intension of an action to be filled out by suffixes of the period over which it is performed. Note that the period $[S; t', t']$ is not added to $[\![a]\!]$, since that would lead to a violation of our assumption that $[S; t, t'] \in [\![a]\!]$ implies that $t < t'$. This would cause ambiguity between an action instance ending at t' and another beginning there. In any case, there is no additional information in $[S; t', t']$ and our definitions are simpler when it is kept out of $[\![a]\!]$.

COH-4. **Passage of Time:** For any scenario at a given moment, there is an action that is done on that scenario. That is, something must be *done* by each agent along each scenario in the model, even if it is some kind of a dummy action. In other words, even waiting is an action. This assumption ensures that time does not just pass by itself, and is needed to make the appropriate connections between time and action. And, assuming that every agent acts helps simplify some technical definitions later on. Formally, $(\forall t \in \mathbf{T}, x \in \mathbf{A}(t), S \in \mathbf{S}_t \Rightarrow ((\exists t' \in S) \Rightarrow (\exists t' \in S, a : [S; t, t'] \in [\![a]\!]^x)))$.

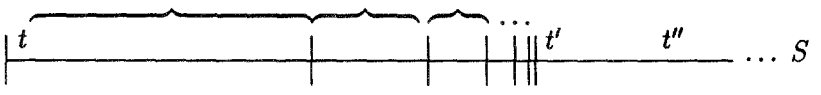

Figure 2.10: Limit Sequences Disallowed by Reachability of Moments

COH-5. **Reachability of Moments:** For any scenario and two moments on it, there is a finite number of actions of each agent that, if done on that scenario starting at the first moment, will lead to a moment in the future of the second moment. Formally, $(\forall S : (\forall t, t' \in S : t < t' \Rightarrow (\exists t'' : t' \leq t''$ and $(\exists a_1, \ldots, a_n$ and $[S; t, t''] \in [\![a_1, \ldots, a_n]\!])))))$. This condition is intended to exclude models in which there are moments that would require infinitely long action sequences to reach.

Such models, an example of which is diagramed in Figure 2.10, would allow a condition to be inevitable and yet unreachable though any finite sequence of actions. Since each action corresponds to a choice on the part of the agent, it is important that this not be the case for inevitability to relate properly with know-how. Infinite sequences of the kind excluded by this constraint cannot arise in discrete models, since there are only a finite number of moments between any two moments and each action consumes at least one.

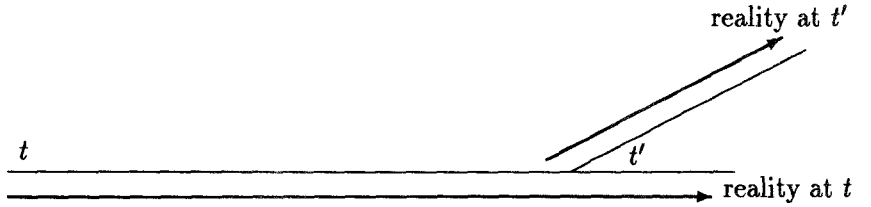

Figure 2.11: Illegal Discontinuity in Reality

COH-6. **Reality does not Change:** The model component **R** assigns to each moment the real scenario at that moment. If a scenario is determined to be the real scenario at some moment, then at any moment on that scenario, the appropriate suffix of that scenario should be the real scenario. The absence of this requirement would mean that reality may change arbitrarily. This would be strange: after all, we are considering reality *per se*, not beliefs about it. The required constraint can be captured as follows: $(\forall t, t' : t' \in \mathbf{R}(t) \Rightarrow \mathbf{R}(t') \subseteq \mathbf{R}(t))$. Figure 2.11 gives an example of the discontinuity in reality that is forbidden by this constraint.

COH-7. **Atomicity of Basic Actions:** If an agent is performing an action over a part of a scenario, then he completes that action on that scenario. This makes sense since the actions in the model are basic actions, done with one choice by their agent. If an action in some domain can in fact be chopped into a prefix and suffix such that the suffix is optional, then it should be modeled as two separate basic actions, the first of which completes entirely and the second of which may not be begun at all.

Formally, let $t, t', t_1 \in \mathbf{T}$, such that $t < t' < t_1$. Let $S_0, S_1 \in \mathbf{S}_t$, such that $[S_1; t, t'] \in S_0$. Then $[S_1; t, t_1] \in [\![a]\!]^x$ implies that $(\exists t_0 \in S_0 : [S_0; t, t_0] \in [\![a]\!]^x)$.

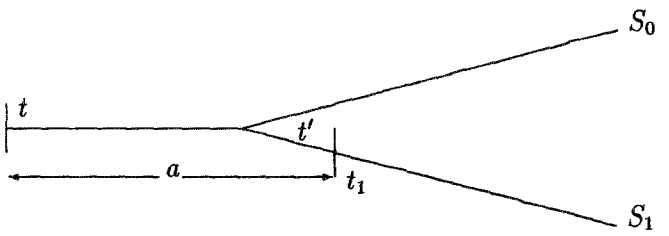

Figure 2.12: Actions Cannot be Partially Performed on any Scenario

> Intuitively, $[S_1; t, t_1] \in [\![a]\!]^x$ means that x is performing a from t to t_1. Therefore, he must be performing a in any subperiod of that, including $[S_1; t, t']$, which is the same as $[S_0; t, t']$. Thus, a must be completed on S_0. By contrast, higher-level actions may not satisfy this. For example, Al may be crossing the street (on a scenario) even if he did not cross it successfully on that scenario, e.g., by being run over by a bus.

The basic formal model is now in place to reason about actions and time. However, some further assumptions are required in order to capture some important properties of the concepts to be formalized and to enable their formalization in a sufficiently general and intuitively appealing manner. One of these properties is that the intentions of an agent do not entail his know-how, and vice versa. That is, an agent who intends p may not know how to achieve it, and one who knows how to achieve p may not intend it. Another property is that intentions constrain the actions an agent may choose, roughly, to be among those that would lead to his intentions being fulfilled. Still another property is that intentions coupled with know-how can, if acted upon, lead to success. These and other such properties are studied in detail in later chapters. However, we must enforce certain additional constraints on our model to facilitate their expression in the present framework. Some of these constraints are motivated and introduced later. However, one that is particularly important and general is described next.

Our models represent physical systems, albeit nondeterministic ones. The actions available to the agents and the conditions that hold on different scenarios leading from a given state are determined by that state itself. Constraints on agent's choices, abilities, or intentions can thus be flexibly modeled.

A well-known alternative characterization of models of time is by the set of all scenarios at all states. Let $\mathbf{S} = \bigcup_{t \in \mathbf{T}} \mathbf{S}_t$. For a model to represent a physical system and be specifiable by a transition relation among different states, the corresponding set of scenarios, $\mathbf{S}$, must satisfy the following closure properties [Emerson, 1990, p. 1014]. I generalize these from discrete time.

- *Suffix closure:* If $S \in \mathbf{S}$, then all suffixes of S belong to $\mathbf{S}$.
- *Limit closure:* If for an ordered set of states $T = \{t_0 \ldots t_n \ldots\}$, scenarios containing each initial fragment $t_0 \ldots t_n$, for $n \geq 0$ are in $\mathbf{S}$, then a scenario S such that $T \subseteq S$ is also in $\mathbf{S}$.
- *Fusion closure:* If $S_0 = S_0^p \cdot t \cdot S_0^f$ and $S_1 = S_1^p \cdot t \cdot S_1^f$ in $\mathbf{S}$ include the same *state* t, then the scenarios $S_0^p \cdot t \cdot S_1^f$ and $S_1^p \cdot t \cdot S_0^f$ formed by concatenating the initial and later parts of S_0 and S_1 with respect to t also belong to $\mathbf{S}$ (here $\cdot$ indicates concatenation). Fusion closure means that the futures available at a state depend only on the state itself, not on the history by which it may be attained.

Lemma 2.1 By construction, $\mathbf{S}$ derived from the proposed model satisfies suffix and limit closures. □

However, fusion closure is not satisfied in general. I show next how to satisfy it by imposing an additional constraint on the proposed model. This constraint relies on a notion of *state*. However, the components of the proposed model are moments and periods. Therefore, I first formalize states in the proposed model. For this, I define a relation, $\sim$, which indicates the state-equivalence of moments and periods. States are the equivalence classes of moments under $\sim$.

For moments, t and t', define $t \sim t'$ iff they satisfy the same atomic propositions. For sets of moments, L and L', define $L \sim L'$ in terms of an *order-isomorphism*, f.

AUX-1. Given two sets L and L' with an order $<$, a map f from L to L' is an order-isomorphism iff

- f is onto,
- $(t \in L$ iff $f(t) \in L')$, and
- $(\forall t, t_0 \in L : t < t_0$ iff $f(t) < f(t_0))$

AUX-2. $t \sim t'$ iff $\{\psi \in \Phi | t \in [\![\psi]\!]\} = \{\psi \in \Phi | t' \in [\![\psi]\!]\}$

AUX-3. $L \sim L'$ iff $(\exists f : f$ is an order-isomorphism and $(\forall t \in L \Rightarrow t \sim f(t)))$

Observation 2.2 $\sim$ is an equivalence relation □

Thus, $t \sim t'$ means that the same physical state occurs at moments t and t'. In other words, states are the equivalence classes of $\sim$ on moments. Similarly, $L \sim L'$ means that the moments in L and L' represent the same states occurring in the same temporal order. In other words, L and L' represent the same trajectory in state space.

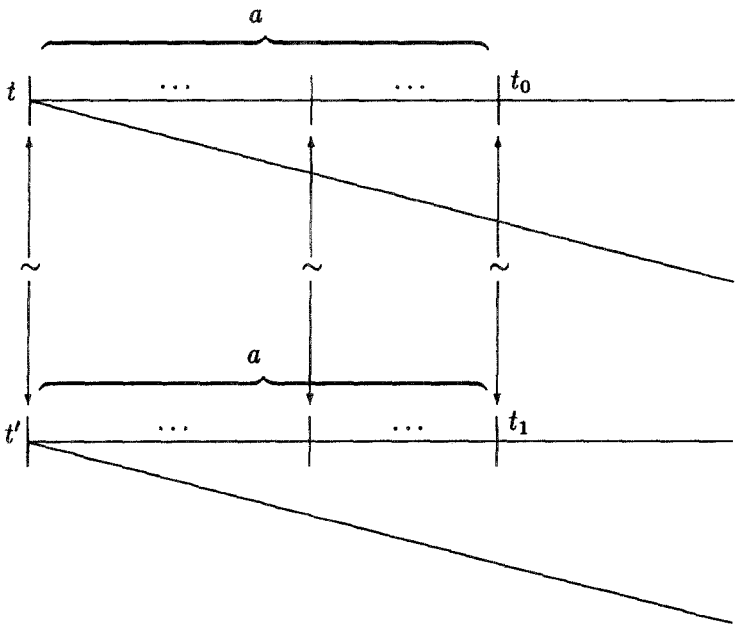

Figure 2.13: Weak Determinism

COH-8. **Weak Determinism:** If two moments satisfy precisely the same atomic propositions, then the fragments of the model rooted at those moments must be isomorphic with respect to the temporal precedence relation and the atomic propositions in the formal language. Thus, we can define weak determinism as the following constraint.

$(\forall x \in \mathcal{A}, a \in \mathcal{B}, t, t', t_0 \in \mathbf{T}, S_0 : t \sim t' \Rightarrow ([S_0; t, t_0] \in [\![a]\!]^x \Rightarrow (\exists S_1 \in \mathbf{S}_{t'}, t_1 : [S_1; t', t_1] \in [\![a]\!]^x$ and $[S_0; t, t_0] \sim [S_1; t', t_1]))))$

Lemma 2.3 Under weak determinism, **S** derived from the proposed model satisfies fusion closure. □

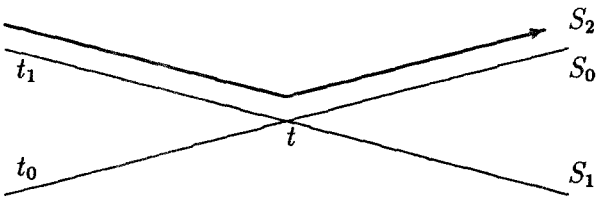

Figure 2.14: Weak Determinism as Fusion Closure in State Space

Figure 2.14 shows an example of fusion closure and how it is satisfied by weak determinism. Note that this figure shows the state space. In other words, the relation $\sim$ is replaced by identity in this figure. The figure shows that, if S_0 and S_1 are scenarios, then S_2 is a scenario at t_1. This holds by the following reasoning. Let t'_0 and t'_1 be the moments in S_0 and S_1, respectively, that have the state t. Then $t'_0 \sim t'_1$. Therefore, by weak determinism, for all moments in S_0 after t'_0, there are state-equivalent moments in **T** that follow t'_1 in the same order. In Figure 2.14, these moments represent the state space trajectory of S_0 after t.

The key intuition behind the present approach is that agents and their environments are physical systems with respect to which we take the intentional stance. This means that all relevant information about how a multiagent system *might* physically evolve is captured by the state in the formal model. The actual evolution or behavior of a system is determined by the actions that the agents may perform and the events that may occur in the environment. In other words, the real scenario at a moment depends on the agents' intentions and beliefs, but the set of possible scenarios is independent of the agents' intentions and beliefs. That is, the actions that are available to the agents and the conditions that hold on different scenarios leading from a given state are determined by that state itself. The state at a moment is precisely characterized by the atomic propositions that hold at that moment.

A purely physical stance is one in which we take the agents' intentions and beliefs to be determined by the physical state of the system. Under such a stance, the actions that are physically not possible at a moment would be considered on par with the actions that are physically available, but happen not to be chosen by the agents (given the agents' intentions and beliefs). Such

a stance would lead to a model in which the only source of nondeterminism is quantum-mechanical, i.e., physical, nondeterminism. However, as argued in Chapter 1, a purely physical stance is not scientifically helpful for the study of multiagent systems. Indeed, the whole point of taking the intentional stance is to facilitate abstract, i.e., non-physical, descriptions of intelligent agents.

For the same reason, even if the underlying model were deterministic, we would prefer that the model be nondeterministic, so that the choices that agents make and a variety of potential constraints on those choices can be explicitly captured. Classical distributed computing models of temporal logic also allow this kind of nondeterminism (through branching or multiple possible computations). However, a key difference is that the notion of state in the present work is of physical state, which explicitly excludes aspects like intentions and beliefs. In classical temporal logic models, there is no notion of intentions or beliefs and the state by itself characterizes the potential behaviors of the system.

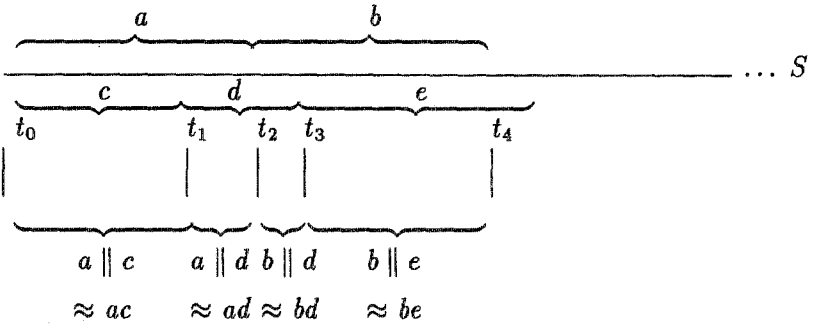

Figure 2.15: Each Agent Performs one Action at a Time

AUX-4. **Additional assumption to simplify notation.** *Each agent performs one action at a time:* It is technically convenient to limit each agent to do exactly one basic action at a moment. This does not restrict potential models, since we can consider the set of basic actions of an agent to contain actions that are the combinations of the actions he would otherwise have been said to be performing simultaneously.

For example, an agent who can walk and whistle at the same time can be thought of as having a basic action, called "walk-whistle," that has exactly the same effects in each given state that the actions of walking and whistling would have when done simultaneously. Figure 2.15 shows how we may convert any model in which an agent can perform multiple actions simultaneously to

one in which he performs exactly one action at a time. The original actions can be of arbitrarily different durations. The moments at which any action is begun or completed are especially important. These are shown as moments t_i in the figure. Between any two successive such moments, the agent simultaneously performs a well-defined set of actions. We can generate a set of action symbols, one for each set of actions that the agent performs simultaneously. The agent's choices can be seen in the original model as selecting a set of actions; in the revised model, the choices pertain to the extend set of basic actions. In each model, the choices of an agent are constrained by the durations of the actions that he has already selected.

The above is not a claim about how actions should be represented in a reasoning system; it is merely an assumption designed to simplify the quantifications over actions that many of the later definitions involve. If this assumption is made, those definitions become significantly more readable. I should reiterate that nothing is lost of our ability to model different applications: we can transform any model into one in which this assumption is satisfied. This can be accomplished by changing the set of basic actions and the intensions of basic actions appropriately. As a result, while each agent does one action at a moment, different agents can act simultaneously. In the worst case, the number of basic actions in the transformed model can be exponential in the number of basic actions in the original model. However, that is not a problem since, if all combinations of actions were possible and if we wanted to reason about them, we could not have done any better anyway. For, even then, we would have to consider how an agent might select an appropriate subset of his basic actions.

Observation 2.4 Under assumption AUX-4, $(x\langle a\rangle\mathsf{true} \wedge x\langle b\rangle\mathsf{true}) \rightarrow a = b$ □

The formal model described above is motivated from general intuitions about actions and time. The frame component of it will remain unchanged throughout this work, although the interpretation component will need to be extended. The key intuitions behind the model are that the world is seen to be in different states at different moments. The agents can act in different ways, each combination of their choices leading to different scenarios being realized. The definitions of intentions and know-how depend on the relations between the agents' possible actions and the conditions that result from them.

2.4 Results on Time and Actions

It is helpful in intuitively understanding formal definitions to attempt to prove some technical results that should follow from them. For this reason, I state and discuss some consequences of the above model and semantic definitions next.

It appears that constraint COH-1 is what McDermott sought to achieve by requiring that actions do not overlap. Unfortunately, that also eliminates COH-3, which is essential, e.g., so that $\mathsf{F}p$ can be concluded at all moments which precede p (Observation 2.18). Constraints COH-4 and COH-5 are required for Observation 2.18 and related results about G and U.

Observation 2.5 $\neg(p\mathsf{U}\neg p)$ □

Observation 2.6 $p\rightarrow \mathsf{F}p$ □

Observation 2.7 $\mathsf{G}p\rightarrow \mathsf{F}p$ □

Observation 2.8 $\mathsf{F}p \equiv \mathsf{F}\mathsf{F}p$ □

Observation 2.9 $\mathsf{G}p \equiv \mathsf{G}\mathsf{G}p$ □

Observation 2.10 $\mathsf{F}\mathsf{G}p\rightarrow \mathsf{G}\mathsf{F}p$ □

Observation 2.11 $\mathsf{G}\mathsf{F}p \not\rightarrow \mathsf{F}\mathsf{G}p$ □

Observation 2.12 $p \wedge q\rightarrow p\mathsf{U}q$ □

Observation 2.13 $\mathsf{G}p \not\rightarrow p\mathsf{U}q$ □

Observation 2.14 $(\mathsf{G}p \wedge \mathsf{F}q)\rightarrow p\mathsf{U}q$ □

Observation 2.15 $(p\mathsf{U}p) \equiv p$ □

Observation 2.16 $(x\langle a\rangle p)\rightarrow \mathsf{F}p$ □

Observation 2.17 $(x\langle a\rangle \mathsf{F}p)\rightarrow \mathsf{F}p$ □

Observation 2.18 $\mathsf{F}p \rightarrow p \vee (\bigvee a : x\langle a\rangle \mathsf{F}p)$ □

Observation 2.19 $\mathsf{G}p \rightarrow (\bigvee a : x\neg[a]\neg\mathsf{G}p)$ □

Observation 2.20 $(p \wedge x\neg[a]\neg\mathsf{G}p) \rightarrow \mathsf{G}p$ □

Observation 2.21 $(p \wedge q) \rightarrow p\mathsf{U}q$ □

Observation 2.22 $(p \wedge x\neg[a]\neg(p\mathsf{U}q)) \rightarrow p\mathsf{U}q$ □

Observation 2.23 $(p \wedge x\langle\!\langle a\rangle\!\rangle(p\mathsf{U}q)) \rightarrow p\mathsf{U}q$ □

Observation 2.24 $p\mathsf{U}q \rightarrow ((p \wedge q) \vee (p \wedge (\bigvee a : x\neg[a]\neg(p\mathsf{U}q))) \vee (p \wedge (\bigvee a : x\langle\!\langle a\rangle\!\rangle(p\mathsf{U}q))))$ □

The following shows that one action operator suffices in discrete models with unit length actions.

Observation 2.25 In models with unit length actions, $x\langle a\rangle p \equiv x\neg[a]\neg p$ and $x\langle a\rangle p \equiv x\langle\!\langle a\rangle\!\rangle p$ □

In the presence of constraint Coh-1, we can simplify the semantic condition for $x[a]p$ as follows. I will freely use this version in the sequel.

Observation 2.26 $M \models_{S,t} x[a]p$ iff $(\forall t' \in S : [S;t,t'] \in [\![a]\!]^x$ implies that $(\exists t'' : t < t'' \leq t'$ and $M \models_{S,t''} p))$ □

The following observation highlights that $x\neg[a]\neg p$ means that a is performed and p holds throughout a.

Observation 2.27 $M \models_{S,t} \neg[a]\neg p$ iff $(\exists t' \in S : [S;t,t'] \in [\![a]\!]^x$ and $(\forall t'' : t < t'' \leq t'$ implies that $M \models_{S,t''} p))$ □

2.5 Strategies

It is useful to think of intelligent systems as having a *reactive* component. This is the component that takes care of the actions that agents do at the greatest level of detail. Typically, these are actions that cannot be planned in advance, because of uncertainty about the state of the world and the rapidly changing nature of the relevant parameters. These actions are selected by an agent on the fly on the basis of the state of the environment he finds himself facing. For example, a detailed plan of how a robot should walk down a hall would in general not be feasible. For, even if the locations of all the objects in the hall were known precisely initially, the exact path taken would depend on the paths taken by other agents and objects in the hall while the robot was in it.

To the extent that it has been developed in the preceding sections, the proposed formal model can accommodate the reactive component of intelligent systems most naturally. This is because it can model actions fairly generally and nothing more is required. It will, however, help to be able to define useful abstractions over the behaviors of agents. These abstractions make it simple for us to understand, specify, and implement intelligent agents.

These abstract descriptions of behavior, I call *strategies*. The idea of using strategies such as these for describing intelligent agents can be traced back to [Miller *et al.*, 1960, p. 17], who credit [Kochen & Galanter, 1958] (p. 47). To my knowledge, the first computer science usage of this term in a related sense is in [McCarthy & Hayes, 1969]. Strategies here are taken simply to characterize an agent's behavior, possibly in quite coarse terms. This is in greater agreement with the definition of [Miller *et al.*, 1960] than of [McCarthy & Hayes, 1969]. Also, there is no commitment here to strategies being implemented as symbolic structures or as programs. They could just be the compact descriptions of a particular architecture, i.e., realized in the hardware. How strategies are realized is clearly of great importance to the implementor. However, from a logical point of view, we can fruitfully study them independent of the form in which they ultimately may be realized.

The formal definition of strategies here is derived from regular programs in dynamic logic, which are a standard notation for describing programs and computations in theoretical computer science [Fischer & Ladner, 1979; Kozen & Tiurzyn, 1990]. I define the set of strategies, $\mathcal{L}_y$, recursively as below. This set includes the empty strategy and the abstract strategies of achieving different conditions. It is closed under sequencing, conditionalization, and iteration. An important feature of this language is that it is deterministic. That is, all choices concerning what substrategy to execute next have guard conditions. A particular option is selected only if its guard is satisfied or, sometimes, only

if its guard is known to be satisfied.

STRAT-1. **skip** $\in \mathcal{L}_y$

STRAT-2. $q \in \mathcal{L}$ implies that $\mathbf{do}(q) \in \mathcal{L}_y$

STRAT-3. $Y_1, Y_2 \in \mathcal{L}_y$ implies that $Y_1; Y_2 \in \mathcal{L}_y$

STRAT-4. $q \in \mathcal{L}$ and $Y_1, Y_2 \in \mathcal{L}_y$ implies that **if** q **then** Y_1 **else** $Y_2 \in \mathcal{L}_y$

STRAT-5. $q \in \mathcal{L}$ and $Y_1 \in \mathcal{L}_y$ implies that **while** q **do** $Y_1 \in \mathcal{L}_y$

Thus the main difference between strategies and deterministic regular programs is that the former are composed of abstract strategies for achieving different conditions, while the latter are composed from a finite alphabet of basic action symbols. Intuitively, the strategy $\mathbf{do}(q)$ denotes an abstract action, namely, the action of achieving q. It could be realized by any sequence of basic actions that yields q.

Thus, in architectural terms, strategies can serve as macro-operators over the reactively realized behaviors of agents. For instance, we might implement the following agent. This agent would have a simple sensory system through which it would be assigned one of a limited repertoire of household tasks. These tasks could include making dinner, getting a newspaper, or checking the mail. It is natural to think of different strategies being associated with these tasks. While the strategies for the different tasks would have to be distinct, they could share significant components. For example, the strategies of getting a newspaper and checking the mail, respectively, share the components of getting to the front door, opening and closing it, going down and up the porch steps, and so on. These subtasks are not trivial. However, they do call for reactive solutions. This is because the movements of other agents, the changes in the location of the furniture, the intensity of the breeze, and the wetness of the porch are unpredictable factors that determine the exact actions that the agent must perform to succeed with the relevant subtasks. On the other hand, if we have designed the agent to be able to perform these subtasks successfully, then we can simply invoke them as higher-order primitives from the other strategies.

Strategies do not add any special capability to the agents. They simply help us, designers and analyzers, better organize the skills and capabilities that agents have anyway. Hierarchical or partial plans of agents, thus, turn out to be good examples of strategies. Considering strategies explicitly as a part of the formal language allows us to model agents who have plans, but who are also capable of acting reactively and must usually do so. Such

agents are important in current research into intelligent systems [Mitchell, 1990; Spector & Hendler, 1991]. Furthermore, we can use strategies to describe computational entities that do not explicitly have plans, but simply execute programs. Thus we can consider agents who may not explicitly symbolically represent and manipulate their action descriptions. This is in concordance with the spirit of the intentional stance, which I have adopted here: this stance can apply to all interesting systems, not just those that are, for independent reasons, known to be intelligent. In this way, strategies help put the different schools of thought about intelligent systems in a unifying perspective.

The component $\mathbf{Y}$ of the model was defined in section 2.1.2 as a function that assigns a strategy to each agent at each moment. Now we can formalize its type as $\mathcal{A} \times \mathbf{T} \mapsto \mathcal{L}_y$. Intuitively, the strategy assigned to an agent is the one that the agent is currently following or attempting to follow. Of course, there is no guarantee that the agent will succeed with it. I return to these points in section 3.1.

It is common in the AI literature to consider *goals* as primitives that determine what an agent seeks to achieve [Georgeff, 1987]. Goals are just seen to be descriptions of states that may be passed as inputs to a planning program to determine a sequence of actions for an agent. In some cases, goals are considered as possible descriptions of states that an agent may decide to achieve; then, adopted goals correspond to intentions. The notion of goals can be easily accommodated in the present approach. Indeed, one can associate a goal for a condition q as the simple strategy $\mathbf{do}(q)$. The definition of strategies given here allows more complex specifications of goals; however, goals as given traditionally can be captured here. Just as in the traditional approaches, it is possible to consider goals independently of whether they have actually been adopted by an agent. However, details of how one might plan an agent's actions are not focused on here.

Strategies are also powerful enough to capture many classes of behavior that may seem to be, and may be presented as being, non-teleological. I submit that many varieties of such behavior must be expressible in standard programming languages, such as Pascal. Since strategies are deterministic regular programs, albeit with abstraction (as in $\mathbf{do}(q)$), they can directly capture just about any kind of terminating behavior that can be expressed in Pascal and other imperative programming languages. Indeed, deterministic regular programs have been found interesting in theoretical computer science precisely because of their similarity with standard programming languages. Once a strategy can be specified, it can be used to assign intentions to agents. In this way, we can take the intentional stance even towards systems that are initially given as not engaging in goal-directed behavior.

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

Table 2.1: Definition of $\downarrow$ of Strategies

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

Table 2.2: Definition of $\uparrow$ of Strategies

It is useful to define two metalanguage functions, $\downarrow$ and $\uparrow$, on strategies. These functions depend on the moment at which they are evaluated; the relevant moment is notated as a subscript. Let Y be a strategy. $\downarrow_t Y$ denotes the part of Y up for execution at moment t, and $\uparrow_t Y$ the part of Y that would remain after $\downarrow_t Y$ has been done. It is convenient to assume that strategies are normalized with respect to the following constraints, although it is not technically essential to do so.

- **skip;**$Y = Y$, for all Y, and
- Y**;skip** $= Y$, for all Y.

Both the $\downarrow_t$ and $\uparrow_t$ of a strategy depend on the moment t. For example, the $\downarrow$ of a conditional strategy depends on whether the relevant condition is true or false at t. It should be easy to see from the above that for any strategy, Y, $\downarrow_t Y =$ **skip** or is of the form **do**(q), for some q. And if $Y \neq$ **skip**, then $\downarrow_t Y$ is necessarily of the latter form. Tables 2.1 and 2.2 give the definitions of $\downarrow$ and $\uparrow$. A consequence of those definitions is Lemma 2.28 below.

Lemma 2.28 $\downarrow_t Y =$ **skip** entails that $\uparrow_t Y =$ **skip**

Proof. By inspection of the conditions in Tables 2.1 and 2.2, for each form of a strategy. □

This lemma simplifies the statement of certain conditions later on, especially, the condition of persistence discussed in Chapters 3 and 5. It is not needed for most of the other definitions, however.

In succeeding chapters, I will use the set $\mathcal{L}_y$ and the metalanguage functions defined on it to give the semantics of the operators for intentions and know-how.

2.6 Belief and Knowledge

Two main kinds of formal definitions of knowledge (or belief) are known in the literature. The *sentential* approach states that an agent knows every proposition that is stored in his knowledge base [Konolige, 1986]. The *possible-worlds* approach states that an agent knows every proposition that is true in all the worlds (or moments, in the present terminology) that he "considers" possible [Hintikka, 1962]. Since typically these worlds are not characterized separately, but only through the agent's knowledge, the agent may be said to know every proposition that is true in all the worlds that are compatible with what

he knows. Each approach has its trade-offs. The sentential approach does not consider models of the world and, thus, does not assign semantic content to knowledge. The possible-worlds approach gives a perspicuous semantics, but at the cost of validating inferences such as: an agent knows all logical consequences of his knowledge. This is in direct conflict with the fact that agents are not perfect reasoners and, in general, do not know what the consequences of their knowledge might be. Alternative approaches exist [Asher, 1986; Fagin & Halpern, 1988; Singh & Asher, 1993] that seek to avoid both these problems, but they are technically more complex than either of the approaches mentioned above.

The definition given below is a possible-worlds definition and thus imperfect in the ways mentioned above. However, it relates quite naturally to our model of actions and time; it is, therefore, a reasonable first approximation. The interpretation function, $\mathbf{B}$, defined in section 2.1.2 assigns a set of moments to each agent at each moment. At a given moment, the set of moments assigned to an agent by $\mathbf{B}$ denotes the states of affairs that the agent considers as possible (at the given moment). Thus, what the agent really believes are the propositions that hold in each of the moments he considers possible. This motivates the following semantic definition for $x\mathsf{B}p$:

SEM-16. $M \models_t x\mathsf{B}p$ iff $(\forall t' : (t,t') \in \mathbf{B}(x)$ implies $M \models_{t'} p)$

An important special case occurs when for all moments, t, $(t,t) \in \mathbf{B}(x)$. In that case, $x\mathsf{B}p \rightarrow p$. That is, all of x's beliefs are true. Following standard practice, true beliefs are identified with knowledge. In that case, it is mnemonically helpful to use the formula $x\mathsf{K_t}p$ instead of $x\mathsf{B}p$, where $\mathsf{K_t}$ stands for *know-that.*

It is customary to assume that each of the relations $\mathbf{B}(x)$ has the following properties [Moore, 1984].

1. *Reflexivity:* $(\forall t : (t,t) \in \mathbf{B}(x))$
2. *Transitivity:* $(\forall t,t',t'' : (t,t'),(t',t'') \in \mathbf{B}(x) \Rightarrow (t,t'') \in \mathbf{B}(x))$

In that case, the B operator defined above can be replaced by $\mathsf{K_t}$. The following axioms hold of it.

AX-BEL-1. $x\mathsf{K_t}p \rightarrow p$

AX-BEL-2. $x\mathsf{K_t}p \rightarrow x\mathsf{K_t}x\mathsf{K_t}p$

AX-BEL-3. $x\mathsf{K}_t\mathsf{true}$

AX-BEL-4. $x\mathsf{K}_t(p \rightarrow q) \rightarrow (x\mathsf{K}_t p \rightarrow x\mathsf{K}_t q)$

Theorem 2.29 Axioms AX-BEL-1 through AX-BEL-4 constitute a sound and complete axiomatization for the operator K_t.

This theorem is due to Kripke. A proof is available in [Chellas, 1980, pp. 177–178]. □

The primary relationship between knowledge and actions is that, given a particular strategy, the actions an agent chooses are determined by his knowledge. This connection is studied in detail in Chapters 4 and 5. One constraint between actions and knowledge that is often applicable is the following.

COH-9. **Knowledge of Choices:** This states that an agent knows what actions he can perform. In other words, if an agent can perform an action at a given moment, then he can perform it at all belief-alternative moments. Formally,

$(\forall t : (\exists S, t_0 : [S; t, t_0] \in [\![a]\!]^x) \Rightarrow (\forall t' : (t, t') \in \mathbf{B}(x) \Rightarrow (\exists S', t_1 : [S'; t', t_1] \in [\![a]\!]^x)))$

Lemma 2.30 If a model satisfies constraint COH-9, then it validates the following formula: $\mathsf{E}x\langle a\rangle\mathsf{true} \rightarrow \mathsf{K}_t\mathsf{E}x\langle a\rangle\mathsf{true}$ □

2.7 More on Actions and Other Events

The formal framework described above includes actions and other events and relates them to time. Only actions, which are events due to an agent, are included in the formal language because actions is all we need for our purposes. But it is easy to augment the formal language to refer to non-action events as well, if that is needed.

Although the formal framework allows several actions and events to happen concurrently and asynchronously, it presents only a bare-bones view of them. Actions and events have been intensively studied in linguistics [Vendler, 1967; Link, 1987; Krifka, 1989], philosophy [Davidson, 1980; Goldman, 1970; Thomason & Gupta, 1981; Asher, 1992], and AI [McDermott, 1982; Allen, 1984; Shoham, 1988; Bacchus *et al.*, 1989]. The most sophisticated of these studies

have been the ones in linguistics and philosophy. The former is especially useful, since most of our intuitions about events are derived from how they are referred to in natural language. I shall, therefore, concentrate on linguistic and philosophical theories and consider AI approaches only at the end.

2.7.1 Events in Natural Language

The classification of events proposed in [Vendler, 1967, chapter 4] is based on data from natural languages. It captures many of our commonsense intuitions about events and is fundamental to much of the other work on events. At the top level, events are distinguished from states. The major categories of events are *telic* and *atelic*. Telic events are those that have a well-defined endpoint; atelic events are those that do not. Examples of telic event types are "build a house" or "eat an apple," which have a set moment of ending. Examples of atelic event types are "push a cart" or "walk in the park." There are subtle relationships between the event category denoted by a natural language sentence and certain properties of the different parts of speech in that sentence [Krifka, 1989]. These shall not concern us here.

The important observation from our point of view is that these theories are, for the most part, not about the nature of events *per se*, but rather about descriptions of those events. The description of an event typically refers to the entities involved in it, its result state, its structure (whether it is telic, whether it iterates, and so on), and the manner in which it happens. These can be taken care of in the proposed framework, provided we extend the formal language to make it sufficiently expressive: the model itself need not be augmented.

The introduction of strategies in section 2.5 serves to extend the formal language to describe actions based on their resulting states. The strategy $\mathbf{do}(q)$ denotes the action of achieving a state in which q holds. This is common to many natural language descriptions of actions, for instance, "He shut the door" and other telic sentences. Although the model does not admit instantaneous basic actions and events, strategies can be instantaneously satisfied: this happens when the relevant condition holds already. As a result, instantaneous events can be said to have occurred wherever the given condition holds. This might seem problematic, since events such as "Al woke up" cannot be said to have happened in every state in which Al is awake. However, the status of zero-duration events is suspect, given our knowledge of Physics. Therefore, we can avoid modeling events as instantaneous, even though natural languages allow some of them to be treated as if they were.

When events are described in terms of states, they usually involve entering or exiting a certain state. This change of state can be modeled by strategies of the form $\mathbf{do}(\neg q);\mathbf{do}(q)$. Such strategies require that an appropriate condition, q, come to hold after its negation has held. Thus they are satisfied only if we can find two moments, the earlier of which satisfies $\neg q$ and the later of which satisfies q. Such strategies cannot be begun and satisfied at a moment at which q holds; in fact, they always take time. Such strategies may be used to model events such as reaching a mountain peak, which happen when one is not initially on the peak. However, this proposal permits someone on the peak to reach it by leaving it and then returning to it. This might seem counterintuitive since, in natural language, reaching a place *again* is different from reaching it the first time. This distinction too can be captured by explicitly using the past operator, P, to state whether the given condition was achieved for the first time or not. Ultimately, however, the present qualitative formal language is too weak to directly represent all natural language phenomena, e.g., the anaphoric nature of temporal reference [Partee, 1973]. An indirect approach, which suffices for most purposes of specifying multiagent systems, is discussed in section 2.8.

However, atelic events cannot naturally be expressed as involving changes of state. Such events are, therefore, not easily captured in the proposed approach. Fortunately, though, such events do not arise in the specifications of artificial systems. For example, one never requires that an agent take a walk, but rather that an agent take a walk for a certain duration or walk until he arrives at some destination or achieves some other condition.

Link's and Krifka's theories allow composite events to be formed by joining atomic events [Link, 1987; Krifka, 1989]. Their approach is abstract in that no constraints are stated on how and when two events may be composed. Their main aim is to be able to derive certain properties of natural language sentences and, thereby, to explain certain linguistic phenomena. The obvious connection to the proposed framework is that only events that are on the same scenario may be composed. Single events distinguish scenarios on which they occur from those on which they do not. Similarly, compositions of events can distinguish scenarios too. The properties of events studied by these researchers include telicity and others that depend on event descriptions. These are not of interest here.

2.7.2 Trying to Act

Basic actions were defined as the choices that an agent can make. Agents thus automatically succeed with the basic actions they try. But, as described above,

it is often useful to be able to identify actions by their effects. However, usually, the effects of actions are far from certain. This observation is captured in the model by allowing each action begun at a moment to be performed on several different scenarios, possibly leading to different states on each. As a result, when actions are described by their effects, there is a profound distinction between trying to perform an action and actually performing it.

Indeed, this is one reason why the study of know-how and intentions is interesting: know-how and intentions are means of talking about abstract actions that are defined by their effects. When we are interested in such abstract actions, the present framework allows us to distinguish between successful performances of them and unsuccessful attempts to perform them. I shall formalize a constraint later that states that, if agents can, they act in ways to best achieve their strategies; such actions constitute attempts at achieving the given strategies and at satisfying the associated intentions. Those attempts can be guaranteed to be successful only in the presence of know-how.

It is possible in natural language to distinguish between an action and an attempt to perform it, for instance, by using an explicit indicator like the verb "try." Quite often, the same verb is used for both purposes. For example, we can use the verb "push" in the sense of "trying to push" to felicitously say "He pushed the box, but it did not move." The same verb can also be used in its normal sense, as in "He pushed the box to the left wall." This usage specifies the resulting state and it is impossible for the following sentences, which refer to the same box, to both be true at once: "John is pushing the box to the left wall" and "Al is pushing the box to the right wall." However, attempts of the described actions can occur simultaneously, because an attempt to perform an abstract action may occur, even though that action does not.

Telic events, when they occur simultaneously, necessarily have the effects that define them individually. Their joint ramifications could, of course, vary significantly from their individual ramifications. For example, John's and Al's pushing different boxes to different sides of a ship cabin may individually cause the ship to tilt, but jointly may not. The ramifications of atelic events vary similarly. Although atelic events are not defined in terms of any specific terminal effects as such, they can be associated with some effects on some salient objects. For example, one takes a walk only for so long as one actually walks. And, one pushes a block only so long as one keeps it moving. It is worth considering the example that Allen gives of the actions of pushing a block one unit to the left, and one unit to the right [1984, p. 125]. He says that performing both actions simultaneously does not cause the block to move. But, it seems that he is using the verb "push" in the sense of "tried to push." The action of pushing a block one unit to the left could not possibly have occurred if the

block did not move: it could at most have been unsuccessfully attempted. The point of this is to show that, in formalizing commonsense domains, one must carefully distinguish actions from attempts to perform them.

The philosopher Goldman proposed the theory of *generation* [1970]. An action *generates* another action if it is the means of performing the second action. In other words, a generates b iff the given agent performs b by doing a. Goldman defines generation as applying between instances of actions that are spatiotemporally identical, but are in some other way distinct. Most of the time, the only distinction possible between these actions is their descriptions. For our purposes, the more relevant component of Goldman's theory is the relation of conditional generation between action types, which presupposes the truth of some salient conditions under which an instance of the first action will generate an instance of the second action. In the present framework, when abstract actions are identified with strategies, the basic actions associated with those strategies can be seen as generating the abstract actions with which they are associated. This is also related to the notion of trying, since an agent may perform a sequence of basic actions, but not succeed in generating the corresponding abstract action: in that case, in the presence of appropriate intentions, the agent may be said to have tried to perform that abstract action.

2.7.3 Actions and Events in Artificial intelligence

Actions and time have drawn much attention in AI. However, most extant approaches are shallow. They do not formalize the properties that coherent models of actions should support and focus instead on the language aspects, e.g., whether predicates like *holds* should be used or not. In other words, they are metalinguistic and not model-theoretic [Turner, 1984, p. 88]. Further, though these theories are advanced in some respects, e.g., in allowing continuous time, they validate too few natural inferences to facilitate formalization of concepts that build on actions, e.g., intentions and ability. Thus most work on those concepts assumes that time is discrete and actions are performed one at a time (e.g., [Rao & Georgeff, 1991a]).

Much of the AI work on actions and events has been concerned with either specifying the time intervals over which they occur, or with their normal preconditions and effects. The internal structure of actions and events, as discussed in the preceding subsections, has drawn much less attention, although some AI researchers have borrowed heavily from the linguistic and philosophical literatures. Most of such contributions have been in the subarea of natural language processing. Events are studied in other parts of computer science, notably in frameworks for semantics of distributed computation. The internal

structure of events has not been intensively studied there either. Thus most of mainstream computer science and AI work on events has been closer in focus to the present approach than linguistic or philosophical work. McDermott's approach bears the greatest similarities to the present framework [1982].

Logics and models of time fall into two major categories: branching-time and linear-time, respectively. The former category variously considers basic temporal structures as branching into the past or the future or both; the latter category requires them to be scenarios. There has been much debate in theoretical computer science about the relative merits of the above two approaches with respect to the specification and verification of classical distributed systems. For our purposes, branching-time approaches yield a natural framework for describing the behavior of multiagent systems. This is because multiagent systems are composed of intelligent agents who have limited control on the future of the world and exercise their choices independently of each other. Our models must incorporate the different choices available to agents explicitly, if we are to represent and reason about those choices and their optimality in our framework. Indeed, any formal framework that is sufficiently powerful for this purpose must involve at least some notion of branching time, implicit or explicit.

Allen presents an interval-based linear-time theory of actions in [1984]. Turner [1984, p. 88] and Shoham [1988, ch. 2] show that Allen's theory is not clear, especially with regard to intervals. Allen objects to branching-time approaches on grounds that branching times are required only for hypothetical reasoning, due to incomplete knowledge about the future [1984, p. 131]. Shoham agrees with this view; he too restricts his models to be linear (p. 36). Allen, who discusses this subject in greater detail, argues that hypothetical reasoning about the future is essentially the same as hypothetical reasoning about anything else, including the past or the present. This remark embodies a fundamental confusion between models and representations. What the branching futures at a moment capture is not the incompleteness of some agent's knowledge, but rather the fact that there are several different ways in which agents may act and the world may evolve. We need to capture different branches of time into the future in order to explicitly consider the choices that agents can make. Since there are no choices to be made about the past, we can allow it to be linear. (Sometimes, efficiency may be gained by treating even the past as branching: I allow this.) The incompleteness of the agents' knowledge, be it about the past, present, or future, is captured by the alternativeness relations that are assigned to each of them. A similar point is made by McDermott [1982, p. 108].

McDermott's temporal models are in some ways similar to the ones

developed here. McDermott, however, requires his models to be dense; no such requirement is imposed here, though density is permitted. Scenarios as defined above are related to the *fullpaths* of Emerson [1990, p. 1014] and the *chronicles* of McDermott [1982, p. 106]. The key differences are that fullpaths are defined over discrete models and are necessarily discrete; and, chronicles are necessarily dense. By contrast, scenarios are maximally dense relative to the temporal precedence relation. That is, they derive their structure from $<$. McDermott does not impose any of the coherence constraints described here; it is not clear if he makes use of any of them implicitly.

2.8 Rationale for Qualitative Temporal Logic

The qualitative temporal logic approach adopted here captures the essential aspects of the concepts being formalized, beginning with basic actions and going on to intentions and know-how. One can always move to a quantitative framework or to one in which times are explicit in the formalization. For the former, one may assign dates or clock values to each moment, such that dates are totally ordered and are shared by moments along different scenarios. The present framework can easily accommodate dates. For the latter, one may base the language not on propositions, but on predicates with an explicit temporal argument. Alternatively, one may modify operators, such as U and P, to have quantitative arguments and interpretations. Doing so would allow us to reason about real time within the logic.

Extensions of notation would, of course, be needed to formalize certain kinds of applications. For example, the prohibitive "do not assign runway B" would apply only for a salient interval, say till 2:10 pm, not forever. It can be formalized as if it were the following prohibitive: "do not assign runway B when the time is prior to 2:10 pm." Once the time becomes 2:10 pm, this prohibitive can no longer be violated. Thus we can capture this aspect of temporal specifications simply by enriching the sublanguage from which the atomic propositions of our formal language are drawn. The syntax and semantics of the formal language remain unchanged.

Interestingly enough, the qualitative temporal logic CTL*, on which the present formal language is based, is as expressive as the monadic second-order theory of two successors with set quantification restricted to infinite paths (i.e., scenarios), over infinite models. A similar result holds for the linear fragment of CTL* with respect to the first order language of linear order [Emerson, 1990, pp. 1021–1026]. Thus, in a qualitative framework, expressiveness is not a concern.

I see the steps of augmenting the framework with explicit dates or moving to a language in which moments are explicit as adding notational complexity. However, neither of these steps significantly aids our understanding of the concepts of intentions, know-how, and communications, which are what I primarily focus on here. They are needed solely to make the language more expressive for the lower-level details of a specification.

Another, methodological, reason for proceeding with a qualitative temporal (and dynamic) logic framework is to draw as many similarities as possible with classical distributed computing. Ideally, only the conceptually significant distinctions would be apparent. An important goal of the present approach is to develop a semantics for multiagent systems that is closely related to the semantics for classical systems, thereby making the implementations of such systems on standard platforms with close to standard techniques more obvious.

As remarked above, the underlying language from which the atomic propositions are drawn would need to be extended for most practical languages. Such extensions may be specialized to different applications. Considering only an abstract language makes the framework simpler to understand. However, it leaves two shortcomings. One, we are unable to reason about quantitative durations, since they are packaged inside the atomic propositions. Two, we are unable, without metarules, to refer to times relativized to some salient moment that would be determined during execution. Such relativized times are required in specifications such as "the controller responds within 1 minute of receiving a request for permission to land." On the other hand, languages that are expressive enough to admit such specifications have high computational complexity for problems such as validity checking.