Toward a Model Theory of Actions: How Agents do it in Branching Time

Munindar P. Singh
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
singh@ncsu.edu

July 10, 1996

Abstract

A clear understanding and formalization of actions is essential to computing, and especially so to reasoning about and constructing intelligent agents. Several approaches have been proposed over the years. However, most approaches concentrate on the causes and effects of actions, but do not give general characterizations of actions themselves. A useful formalization of actions would be based on a general, possibly nondiscrete, model of time that allows branching (to capture agents' choices). A desirable formalization would also allow actions to be of arbitrary duration and would permit multiple agents to act concurrently. We develop a branching-time framework that allows great flexibility in how time and action are modeled. We motivate and formalize several *coherence constraints* on our models, which capture some nice intuitions and validate some useful inferences relating actions with time.

Introduction

Over the years, actions and time have garnered much research attention in several areas of computing, and especially in artificial intelligence. Whereas much progress has been made in modeling time, corresponding progress has not been made in modeling actions. Indeed, temporal approaches run the gamut from discrete to continuous, point-based to interval-based, and linear to branching. By contrast, approaches to formalizing actions tend to be restricted to discrete models, typically linear and with additional assumptions such as that exactly one action happens at a time, and all actions have the same duration. Reasoning about actions traditionally focuses on the possible causes and effects of the actions, but not on their structure. This work is undoubtedly of value. However, we submit that its full potential can be realized only if actions themselves are formalized in a general framework.

Although the problems of representing and reasoning about actions arise in numerous subdisciplines related to intelligent agents and multiagent systems, they are especially important in planning, hypothetical reasoning about actions and plans, certain kinds of qualitative reasoning, natural language understanding, belief-desire-intention (BDI) architectures for agents, and communications and interactions among agents. A clear model theory of actions would have a bearing on these areas.

A general model of actions would provide the underpinnings for work on concepts such as intentions, ability, and know-how that supervene on actions. When actions are modeled restrictively, the technical results obtained on the above concepts end up with potentially superfluous or even pernicious restrictions, or rely on some irrelevant aspect of the underlying model of actions [Singh, 1992]. Even if our main interest is in formalizing the above concepts, we must model actions properly to succeed. The above concepts apply directly in understanding BDI architectures, and in characterizing communications and other interactions among agents in a multiagent system [Singh, 1994a; Singh, 1994b]. They, especially intentions, also apply in natural language understanding of sentences and discourses [Grosz & Sidner, 1986]. Intentions find further application in the related problems of formalizing plans and plan recognition [Pollack, 1991].

However, whether or not one is interested in the above high-level cognitive concepts, notions of time and actions are inherently crucial in representing and reasoning about plans and actions. This is especially so when the plans and actions involve continuous phenomena and must be integrated into a framework that combines discrete actions with continuous processes [Sandewall, 1989; Kuipers & Shults, 1994]. Hypothetical reasoning about potential eventualities is important to planning as well as to counterfactual reasoning in natural language [Harper *et al.*, 1981].

The present paper seeks to develop a general framework as a solid "infrastructure" through which recent advances in higher level concepts can be leveraged. More generally, it addresses the important question: *what are the properties that intuitively acceptable and technically feasible models of actions and time might support and how do those properties interact?* Our interest in this paper is conceptual and generic. We begin the task of systematically laying out the terrain of models involving actions and time. As this terrain becomes better mapped out, there will be a stronger basis for system implementation and experimentation. Although that is ultimately essential, it would presently be premature.

Section 1 presents a discussion of certain conceptual issues motivated by the applications of the proposed approach in reasoning about agents. Section 2 presents our formal language and formal model. It gives a formal semantics for our language, and discusses our key operators and primitives. Section 3 motivates and formalizes a number of *coherence constraints* on our models. These constraints formalize various useful properties. Section 4 uses these to prove some interesting results relating actions and time. Section 5 revisits some conceptual and historical issues and closes with a discussion of some open problems.

1 Conceptual Considerations

In conceptually understanding our contribution, it is useful to consider the three levels of the *adequacy* of a representation as defined by McCarthy & Hayes [1969, p. 434]. The three notions are as follows:

- A representation is *metaphysically* adequate "if the world could have that form without contradicting the facts of the aspects of reality that interest us."
- A representation is *epistemologically* adequate "if it can be used practically to express the facts that one actually has about the aspect of the world."
- A representation is *heuristically* adequate "if the reasoning processes actually gone through in solving a problem are expressible in the (representation) language."

We ought to seek representations that are adequate in the following order of importance: metaphysical before epistemological; epistemological before heuristic. Metaphysical adequacy is essential in order to obtain sufficient coverage of the domains and applications of interest for one's representational framework. No such framework can perhaps be adequate for all possible applications (unless one assumes that something like a physical representation of the entire universe would capture all possible properties of it—a philosophical position that we are not concerned with here). Our aim is to develop a framework for actions and time that covers many of the applications that involve representing and reasoning about actions and time.

An aspect of reality that particularly concerns applications involving sophisticated agents is that they have some kind of *choice* as to the actions they perform. Since we are interested in reasoning about actions that agents may or may not perform, a purely deterministic model is unacceptable. Appropriate models for intelligent agents must be constructed from the *intentional stance* [McCarthy, 1979] [Dennett, 1987, pp. 13–35], or at the *knowledge level* [Newell, 1982, p. 115]—we take these as equivalent for our purposes.

Time can variously be modeled as linear or branching. Allen presents an interval-based linear-time theory of actions in [Allen, 1984]. Turner [1984, p. 88] and Shoham [1988, ch. 2] show that Allen's theory is not conceptually clear, especially with regard to intervals. However, Shoham too restricts his models to be linear. Allen (p. 131) and Shoham (p. 36) both argue that branching time is unnecessary since the agents' ignorance can be modeled in other ways.

However, branching into the future is not a matter of ignorance, but of choice. That is why, ignorance apart, the past can be linear but the future must branch. Even assuming a perfectly deterministic universe, our models must embody choice to capture the designer's or *modeler's* lack of knowledge. The ignorance of agents, captured through their beliefs within the model, reflects the *agent's* lack of knowledge relative to the model—fundamentally quite different from choice.

Indeed, epistemological and heuristic improvements may sometimes be gained by treating even the past as branching: we allow this. Indeed, in many cases, it may be appropriate to reason with a representation of time that is cyclic! This happens if the states of the system are represented as vertices of a graph whose edges represent the next-state relation. The cyclicity reflects that the system being reasoned about visits the same state repeatedly, not that time is moving backwards. The corresponding model can be thought of as an infinite tree obtained by unraveling the cycles. In this tree, each original vertex may be captured as one or more moments.

In the present work, we propose a way to formally model actions. We begin with *basic* actions and extend our results to regular program. For our purposes, basic actions have two useful features. First, their internal structure is not modeled explicitly, the details being supplied by the specific application in which they arise. Second, basic actions are deemed to be performed by a single choice by their agent. We also assume that each basic action has exactly one agent performing it. Our formal model states when a basic action is performed and by who: we thus avoid much philosophical debate on the necessary and sufficient conditions under which an agent may be said to have performed a specific action.

Galton [1990] improves over Allen's approach in some respects but does not address constraints on actions *per se*. McDermott's approach, like ours, is point-based and involves branching-time [1982]. However, McDermott requires his models to be dense; also, clock values are essential to his semantics. McDermott notes, correctly, that an action cannot happen over overlapping intervals. But a lot more must be said that the previous approaches do not say. Related research includes [Thomason & Gupta, 1981; van Fraassen, 1981; Dean & Boddy, 1988; Haddawy, 1996], but it does not model the structure of actions as motivated here. We discuss some additional related literature in section 5. Briefly, our framework allows

- time to branch to model agents' choices,
- multiple agents to act simultaneously,
- actions to be of varying durations relative to one another, and
- time to be nondiscrete.

2 Technical Framework

The proposed formal model is based on a set of *moments* with a strict partial order, which denotes temporal precedence. The partiality of temporal precedence means that time may branch. In particular, time may branch into the future in most interesting applications. It may be taken as linear in the past, although nothing hinges upon this. Intuitively, *agents* are the active computational entities that live in the model and perform actions. Other aspects of agency are not relevant here. (For no particular reason, except perhaps habit, we use masculine pronouns to refer to agents.) The agents' ignorance about the past, as about anything else, is captured by the concepts of knowledge and belief [Moore, 1984], which we do not discuss here.

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. Each agent influences the future by acting, but the outcome also depends on other events. A *scenario* at a moment is a set of moments containing the given moment, and all moments in its future along some particular branch. Different scenarios thus correspond to different combinations of actions by the various agents.

Figure 1 shows a schematic picture of the formal model. In this example, the t_i are moments; the S_j are the scenarios at t_0; q and r are propositions or atomic conditions; and a, b, c, and d are basic actions. As can readily be verified, proposition r holds at t_0 and t_1 and proposition q holds at t_1 and t_2. Action a is performed from t_0 to t_1 or t_2; similarly, action c is performed from t_0 to t_1 or t_3. Figure 1 is labeled with the actions of two agents. The first agent can constrain the future to some extent by choosing to do action a or action b. If he does a, then the world progresses along one of the top two branches out of t_0; if he does b, then it progresses along one of the bottom two branches. However, the agent cannot control what exactly transpires.

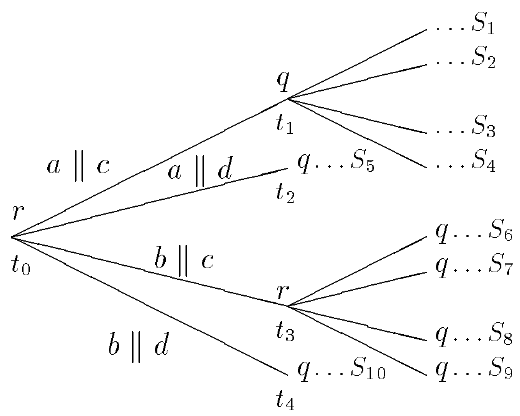


Figure 1: The Formal Model

For example, if he does action a, then whether t_1 or t_2 becomes the case depends on whether the second agent does c or d.

Intuitively, actions are package deals. They correspond to the granularity at which an agent can make his choices. An agent chooses to perform a whole action—with its concomitant range of consequences—or not at all. In Figure 1, in moment t_0 the first agent can perform action a or action b. Thus, he 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 cannot choose between t_1 and t_2, or between t_3 and t_4. Both *choice* and *limited control* are thus captured.

Since we seek to develop a generic, conceptual approach to actions and time, we do not explicitly restrict the actions that might be modeled. Philosophers, e.g., [Goldman, 1970], concern themselves with what is and is not a basic action. Our concerns are more modest, since computational applications come with some actions defined as within the universe of discourse. The application and the level at which we construct the model determine the basic actions. We allow composite actions to be structured from these using the well-known approach of regular programs. One of the goals of our approach is to facilitate multiple levels of representation or modeling—if we decide to "open up" an action into its pieces or simply to modify the underlying structure of time from discrete to continuous, which aspects of our model will not have to change and which will.

Example 1 The reader might consider the following possible actions in the examples and constraints that follow.

- A (human-like) agent raising his arm

- A (software) agent telling another agent a fact
- A (robotic) agent turning into a room
- A (robotic) agent moving a box from one place to another
- A (controller) agent adjusting the flow-rate of a valve

Some of these actions might be modeled as discrete or continuous, deterministic or nondeterministic, basic or composite, depending on one's needs. ∎

2.1 The Formal Language

We use a qualitative temporal language, $\mathcal{TAB}$, based on CTL* [Emerson, 1990] and extended with regular programs common in dynamic logic [Kozen & Tiurzyn, 1990]. (The name $\mathcal{TAB}$ is mnemonic for a temporal language with actions and branching.) Our language captures the essential properties of actions and time that are of interest. This paper focuses on the model-theoretic aspects of actions. As explained above, we would like to be able to study some of the interesting properties of actions in a branching-time framework where the agents act concurrently. We, of course, need a formal language in order to articulate the key properties of our framework. The language CTL* is one of the most well-known of the branching-time temporal logic languages. For many applications in traditional logics of programs, sublanguages of CTL* are used because they are computationally more tractable. The design of appropriate sublanguages is indeed very important, but cannot precede a clear understanding of the underlying semantics. Here we select CTL* because it is general and well-known, and therefore a good starting point. To simplify or language, we do not include past-directed operators, although they can be added if necessary.

Formally, $\mathcal{TAB}$ is the minimal set closed under the rules given below. $\mathcal{TAB}_s$ is the set of "scenario-formulae," which is used as an auxiliary definition. $\mathcal{PROG}$ is the set of regular programs, which gives the basic and the composite programs that agents can execute. Φ is a set of atomic propositional symbols, $\mathcal{X}$ is a set of agent symbols, and $\mathcal{B}$ is a set of basic action symbols. The atomic propositional symbols, or propositions for short, represent what may or may not hold in a given state. They are assigned truth values in each moment in the model. The agent symbols name the agents. The basic action symbols, or basic actions for short, name the actions that the agents can perform.

We specify the syntax of $\mathcal{TAB}$ through the following Backus-Naur Form (BNF) grammar. In the following, we use *slant* typeface for nonterminals; $\longrightarrow$ and $|$ are metasymbols of BNF specification; the remaining symbols are all terminals. The distinguished start symbol of this grammar is *Tab*: the strings of terminals that can be generated from it constitute the language $\mathcal{TAB}$. Similarly, the strings of terminals that can be generated from *TabS* constitute the language $\mathcal{TAB}_s$ and those that can be generated from *Prog* constitute $\mathcal{PROG}$. The words in $\ll$ and $\gg$ are comments and are included solely for expository purposes.

- *Tab* $\longrightarrow$ $\ll$formulae that are true or false on a moment$\gg$

  L1. *Prop* | $\ll$atomic propositions$\gg$

  L2. $\neg$ *Tab* | $\ll$negation$\gg$

  L3. *Tab* $\wedge$ *Tab* | $\ll$conjunction$\gg$

  L4. $(\bigvee B : \mathit{Tab})$ | $\ll$quantification over basic actions$\gg$

  L5. A *TabS* | $\ll$quantification over scenarios$\gg$

  L6. R *TabS* $\ll$reality$\gg$

- *TabS* $\longrightarrow$ $\ll$formulae that are true or false on a scenario$\gg$

  L7. *Tab* | $\ll$mutual recursion with *Tab*$\gg$

  L8. $\neg$ *TabS* |

  L9. *TabS* $\wedge$ *TabS* |

  L10. *TabS* U *TabS* | $\ll$until$\gg$

  L11. $(\bigvee B : \mathit{TabS})$ |

  L12. $X\ [\mathit{Prog}]\ \mathit{TabS}$ | $\ll$dynamic necessitation$\gg$

  L13. $X\ \langle \mathit{Prog} \rangle\ \mathit{TabS}$ | $\ll$dynamic possibility$\gg$

  L14. $X\ |\langle \mathit{Prog} \rangle|\ \mathit{TabS}$ $\ll$strong dynamic possibility$\gg$

- *Prog* $\longrightarrow$ $\ll$regular programs$\gg$

  L15. *TabS* ? | $\ll$test of a condition: ? is the operator$\gg$

  L16. B | $\ll$basic action$\gg$

  L17. $\mathit{Prog}\cdot\mathit{Prog}$ | $\ll$sequencing$\gg$

  L18. $\mathit{Prog}+\mathit{Prog}$ | $\ll$choice$\gg$

  L19. Prog^* $\ll$iteration$\gg$

L20. *Prop* $\longrightarrow$ a proposition symbol $\ll$member of $\Phi\gg$

L21. B $\longrightarrow$ a basic action symbol $\ll$member of $\mathcal{B}\gg$

L22. $X \longrightarrow$an agent symbol $\ll$member of $\mathcal{X}\gg$

The languages $\mathcal{TAB}$, $\mathcal{TAB}_s$, and $\mathcal{PROG}$ are thus defined through a mutual recursion, which bottoms out in atomic propositions, basic actions, and agent names. Syntax rule L7 defines a form of type coercion. As is customary in formal semantics, we are only concerned with abstract syntax here.

For notational simplicity, we use the following naming conventions in this paper.

- x, etc. refer to agents
- ψ, etc. are atomic propositions
- p, q, r, etc. are formulae in $\mathcal{TAB}$ or $\mathcal{TAB}_s$ (determined by the surrounding syntax—this is essential because of rule L7)
- a, b, etc. are basic actions
- π, ρ, etc. are regular programs
- t, etc. are moments
- S, etc. are scenarios.

2.2 The Formal Model

A model for $\mathcal{TAB}$ is a five-tuple, $M = (\mathbf{T}, <, \mathbf{X}, \mathbf{R}, [\![\,]\!])$. Here $\mathbf{T}$ is a set of possible moments ordered by $<$. $\mathbf{X}$ assigns agents to different moments; i.e., $\mathbf{X} : \mathbf{T} \mapsto \wp(\mathcal{X})$. For a moment t, $\mathbf{X}(t)$ gives the set of agents who can perform actions at t. $\mathbf{R}$ and $[\![\,]\!]$ are described below. The relation $<$ is a strict partial order:

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

2.3 Scenarios and Periods

A scenario at a moment t is any single branch of the relation $<$ that includes t and *all* moments in some future of t that is a linear subrelation of $<$. Different scenarios correspond to different ways in which the world may develop, as a result of the actions of agents and events in the environment.

Definition 1 Formally, a scenario at t is a set $S \subseteq \mathbf{T}$ that satisfies the following properties:

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

  If it is possible to extend S (in this formula, to t''), then it is extended, either to t'' (when $t''' = t''$), or along some other branch. In other words, a scenario cannot end arbitrarily when a future moment exists. However, this is consistent with having a last moment: time need not be eternal. All scenarios through a last moment end at that moment.

Definition 2 $\mathbf{S}_t$ 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, the following holds.

Observation 1 $t \neq t' \Rightarrow \mathbf{S}_t \cap \mathbf{S}_{t'} = \emptyset$ $\square$

Further, scenarios at a moment are included in scenarios at prior moments:

Observation 2 Let $t < t'$. Then for every scenario, $S' \in \mathbf{S}_{t'}$, there is a scenario, S, such that $S' \subset S$ and $S \in \mathbf{S}_t$. $\square$

Observation 3 Let $t < t'$. Given a scenario $S \in \mathbf{S}_t$, let $S' = \{t_0 : t_0 \in S$ and $t' \leq t_0\}$. Then $S' \in \mathbf{S}_{t'}$ or $S' = \emptyset$. $\square$

Definition 3 $[S; t, t'] = \{t'' : t'' \in S$ and $t \leq t'' \leq t'\}$. Thus, $[S; t, t']$ is the subset of S from t to t'. That is, $[S; t, t']$ denotes a period on scenario S from t to t', inclusive.

For notational simplicity, we require that $[S; t, t']$ presupposes $t, t' \in S$ and $t \leq t'$, because otherwise the set of moments is empty. A consequence of the definition of periods as sets is that if $[S_0; t, t'] \subseteq S_1$, then $[S_0; t, t'] = [S_1; t, t']$. Figure 2 illustrates this point.

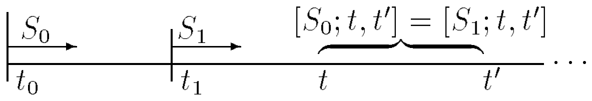


Figure 2: Periods in Overlapping Scenarios

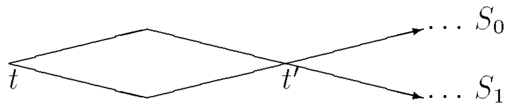


Figure 3: An Example of Branching Past

In Figure 2, the periods are identified by their beginning and ending moments. However, in general, we can have $[S_0; t, t'] \neq [S_1; t, t']$. This arises when there are multiple paths from t to t', i.e., when the model exhibits a *branching past* at t'. Figure 3 illustrates such a situation. In many situations, we wish to interpret the moments as snapshots of the modeled system. That is, the moments represent distinct states of the system. In such cases, the system may arrive at the same state through multiple paths. If the moments are identified by the corresponding states, this would lead to a moment having multiple possible pasts. We do not require this, but accommodate this in our notation by keeping the scenario explicit in the specification of a period.

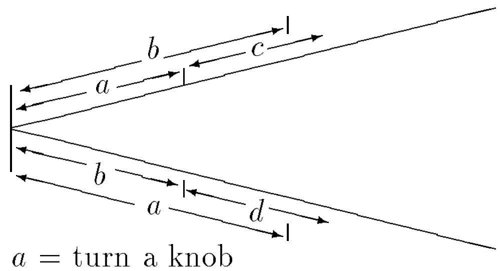
2.4 Basic Actions

Basic actions can be of arbitrary durations. Multiple agents may act simultaneously. The set of actions available to an agent can be different at different moments.

Example 2 The actions of moving a block may take more or less time than the action of turning a knob. This case is diagramed in Figure 4, which also shows that actions may begin and end arbitrarily. ▮

Definition 4 The intension, $[\![\,]\!]$, of an atomic proposition is the set of moments at which it is true. Thus $t \in [\![p]\!]$ means that p is true at moment t.

Example 3 Consider Figure 1. Restricting attention to moments $\{t_0, t_1, t_2, t_3, t_4\}$, we can see that $[\![q]\!] = \{t_1, t_2, t_4\}$. Similarly, $[\![r]\!] = \{t_0, t_3\}$. ▮



a = turn a knob
b = move a block

Figure 4: Actions: Nonsynchronized and of Varying Durations

Definition 5 The intension of a basic action symbol a is, for each agent symbol x, the set of periods in which an instance of a is performed by x. Thus $[S; t, t'] \in [\![a]\!]^x$ means that agent x is performing action a from moment t to moment t'.

Example 4 Consider Figure 1 again. Restrict attention to moments $\{t_0, t_1, t_2, t_3, t_4\}$ and assume that the first agent is called y and the second agent z. Then, we can easily check that $[\![a]\!]^y = \{[S_1; t_0, t_1], [S_5, t_0, t_2]\}$. Similarly, $[\![c]\!]^z = \{[S_1; t_0, t_1], [S_6, t_0, t_3]\}$. ▮

We also assume throughout that $[S; t, t'] \in [\![a]\!]^x$ entails $(\forall t'' : t \leq t'' \leq t' \Rightarrow x \in \mathbf{X}(t))$.

2.5 Regular Programs

The definition of intension is lifted to apply to regular programs in general, not only to basic actions. It still has the same intuitive interpretation—the intension of a program π is, for each agent symbol x, the set of periods in which π is performed by x. Thus $[S; t, t'] \in [\![\pi]\!]^x$ means that agent x is performing action π from moment t to moment t' (along scenario S). The presence of regular programs makes our language fairly expressive in terms of actions. (Further enhancements in terms of action types or schematically generated action symbols might of course be needed in some cases.) We now define the intension for regular programs.

Definition 6 The intension for a program $\pi \in \mathcal{PROG}$ is given by the following rules:

- $[\![p?]\!]^x = \{[S; t, t] : M \models_{S,t} p\}$, where $p \in \mathcal{TAB}_s$

  In other words, the test program terminates successfully and immediately when the tested condition holds; otherwise, it does not even begin.

- $[\![a]\!]^x$, where $a \in \mathcal{B}$, is as defined above.
- $[\![\pi + \rho]\!]^x = [\![\pi]\!]^x \cup [\![\rho]\!]^x$

  A program with choice is executed wherever one of the choices is executed.
- $[\![\pi \cdot \rho]\!]^x = \{[S; t_0, t_2] : (\exists t_1 : [S; t_0, t_1] \in [\![\pi]\!]^x \text{ and } [S; t_1, t_2] \in [\![\rho]\!]^x)\}$

  The sequence of π and ρ occurs wherever π occurs followed immediately by ρ.
- $[\![\pi^*]\!]^x = \{[S; t, t'] : (\exists t_0, \ldots, t_n : t = t_0 \text{ and } t' = t_n : (\forall i : 0 \leq i < n \Rightarrow [S; t_i, t_{i+1}] \in [\![\pi]\!]^x))\}$

  The iteration of π occurs wherever π occurs 0 or more times consecutively.

Example 5 Consider a program $\pi = (\psi? \cdot a + \neg\psi? \cdot b)$. This program is executed along a scenario S starting at a moment t if (1) ψ holds at t and a is performed on S starting at t or (2) ψ does not hold at t and b is performed on S starting at t. ▮

2.6 Reality

Definition 7 $\mathbf{R}$ assigns to each moment t one of the scenarios at t. Thus $\mathbf{R}(t) \in \mathbf{S}_t$. The assigned scenario is interpreted as being the *real* scenario at t.

Thus $\mathbf{R}$ enables us to address a potential difficulty of branching approaches to directly express what *will* happen, not just what *may* happen. The branching approaches capture the various possibilities of how the world may evolve, i.e., of the actions the agents may choose. Intuitively, given all these possibilities, $\mathbf{R}$ identifies the real scenario and the concomitant choices made by each agent.

Example 6 Consider Figure 5. The real scenarios are highlighted there. Thus, S_5 is the real scenario at t_0, S_3 at t_1, and S_6 at t_3. ▮

The constraints on what *will* happen are thus expressed as constraints on the properties of $\mathbf{R}(t)$, for appropriate t. Thus, to capture a constraint that an agent x will in fact perform action a at moment t, we can state that $(\exists t' : [\mathbf{R}(t); t, t'] \in [\![a]\!]^x)$. Several actions may be available to x, but a is the one he chooses. Of course, the most interesting constraints won't have a hardwired action symbol, but would apply to all actions that satisfy some other useful property. For example, we may state that if x knows of some action that guarantees success of his current intentions, then x will perform one such action. When x does not know such an action, then this constraint would not apply.

Notice that, unlike some traditional modal logic approaches, we do not require a unique reality in our model. The real scenario depends on the given moment. If a

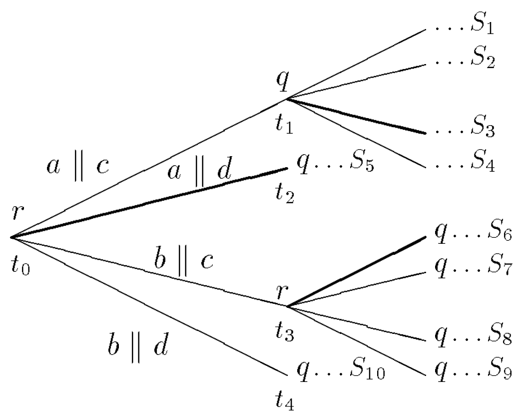


Figure 5: The Reality at Different Moments (Highlighted Scenarios)

unique real scenario could be identified for the entire model, we would not be able to reason about the moments that did not fall on this scenario. Constraints on reality would have no interesting meaning, since they would not apply on the moments outside of the chosen scenario. We will motivate a simple constraint on reality in section 3. However, independent of that constraint, it should be clear that we can talk about the reality relative to a moment t' even if t' is in the future of a moment t whose real scenario does not go through t'. This would happen, for instance, if the model had a unique initial moment. The model is still not necessarily linear—not everything that is possible has to be realized.

Our notion of reality makes it simple to express hypothetical claims, such as "if r held then x would have performed a." However, the semantics of such claims is a nontrivial matter [Harper *et al.*, 1981] and beyond the scope of this paper.

2.7 Temporal and Action Operators: Discussion

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 $\mathsf{trueU}p$. The formula $\mathsf{G}p$ means that p always holds in the future on the given scenario; it abbreviates $\neg\mathsf{F}\neg p$.

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 branching-time operator, R, denotes "in the *real* scenario at the present moment."

Example 7 In Figure 5, $\mathsf{EF}r$ and $\mathsf{AF}q$ hold at t_0, since r holds on some moment on some scenario at t_0 and q holds on some moment on each scenario. Similarly, $\mathsf{RF}q$ holds at t_3. ▮

$\mathcal{TAB}$, through $\mathcal{TAB}_s$, also contains operators on actions. Formally, these are operators on regular programs. They are based on operators in dynamic logic, but are given a linear rather than a branching semantics. (In our approach, the branching component is supplied by the operator A.) Although these operators are defined for regular programs, we recommend that the reader initially view them as applying to basic actions, which is why the following discussion is cast in terms of basic actions.

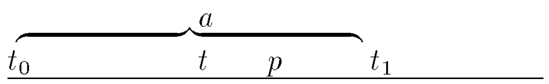


Figure 6: Dynamic Necessitation and Possibility

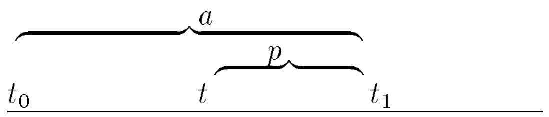


Figure 7: Dual of Dynamic Necessitation

For a basic 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 is performing a at t, then p holds at some moment after t while the given instance of a is being performed. Thus if x is not performing a at t, $x[a]p$ automatically holds. The formula $x\langle a\rangle p$ holds on a given scenario S and a moment t on it, iff, x is performing a at t and p holds at some moment after t while the given instance of a is being performed. Thus if x is not performing a at t, $x\langle a\rangle p$ automatically fails to hold. 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 requiring p to hold at the moment at which the given action completes. Figure 6 illustrates these operators: t is the current moment; a ranges from t_0 to t_1 and p holds at a moment between t and t_1. $[]$ is conditionalized on whether a occurs as shown; $\langle\,\rangle$ requires that a occurs as shown. Figure 7 shows the meaning of $x\neg[a]\neg p$, which is the formal dual of $x[a]p$.

Intuitively, $x\neg[a]\neg p$ means that a is performed on the given scenario and p holds throughout the part of a after t.

In the semantics of $x[a]p$ and $x\langle a\rangle p$, it is essential to allow the condition, p, 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 might 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 might end up north of the equator without ever being at it (according to the model). That would be quite unintuitive. For this reason, the present definitions are preferred although as a consequence, the operators $\langle\,\rangle$ and $[\,]$ are not formal duals of each other. But this is made up for by having a more intuitive set of definitions, which also enable the right relationship between the action operators and F, G, and U to be captured. Recall from above that $p\mathsf{U}q$ considers all moments between the given moment and the first occurrence of q, not just those at which different actions may end.

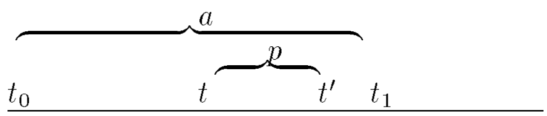


Figure 8: Strong Dynamic Possibility

Further, $x|\langle a\rangle|p$ holds on a scenario S and moment t if x performs action a at t and p holds in some initial subperiod of the remaining period over which a is performed. Figure 8 illustrates this operator: t is the current moment; a ranges from t_0 to t_1 and p holds at every moment between t and t_1. This operator is necessary to relate actions with time for the following reason. We allow actions to happen over periods which contain moments between their endpoints. Such cases can arise even in discrete models if all actions are not unit length. Consequently, if a is performed 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 4: this helps in giving the fixed point semantics of the temporal operators, which is a popular way of computing them [Burch *et al.*, 1990].

The above action modalities yield scenario-formulae, which can be combined with the branching-time operators A, E, and R. $\mathsf{A}x[a]p$ 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, $\mathsf{E}x\langle a\rangle p$ denotes that a is being performed 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, $\mathsf{A}x[a]p$ corresponds to the necessitation operator and $\mathsf{E}x\langle a\rangle p$ to the possibility operator in dynamic logic. R can be used similarly: $\mathsf{R}x\langle a\rangle p$ means that x performs a on the real scenario and p holds at its end.

Existential quantification over basic actions is a useful feature of our language. Of the several basic actions that an agent may do at a given moment, we often need to talk restrictively 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, e.g., one of those that ensure success. This is expressed as $(\bigvee a : p)$, which means that p can be made true by substituting some action symbol for a in it.

2.8 Semantics

Recall that a formal model for $\mathcal{TAB}$, M, is a five-tuple as defined in section 2.2. For $p \in \mathcal{TAB}$, $M \models_t p$ expresses "M satisfies p at t." For $p \in \mathcal{TAB}_s$, $M \models_{S,t} p$ expresses "M satisfies p at moment t on scenario S" (we require $t \in S$). We say p is *satisfiable* iff for some M and t, $M \models_t p$. The satisfaction conditions for the temporal operators are adapted from those in [Emerson, 1990]. It is assumed that each action symbol is quantified over at most once in any formula. Below, $p|_b^a$ is the formula resulting from the substitution of all occurrences of a in p by b. Two useful abbreviations are $\mathsf{false} \equiv (p \wedge \neg p)$, for any $p \in \Phi$, and $\mathsf{true} \equiv \neg\mathsf{false}$.

M1. $M \models_t \psi$ iff $t \in [\![\psi]\!]$, where $\psi \in \Phi$

M2. $M \models_t p \wedge q$ iff $M \models_t p$ and $M \models_t q$

M3. $M \models_t \neg p$ iff $M \not\models_t p$

M4. $M \models_t \mathsf{A}p$ iff $(\forall S : S \in \mathbf{S}_t \Rightarrow M \models_{S,t} p)$

M5. $M \models_t \mathsf{R}p$ iff $M \models_{\mathbf{R}(t),t} p$

M6. $M \models_t (\bigvee a : p)$ iff $(\exists b : b \in \mathcal{B}$ and $M \models_t p|_b^a)$, where $p \in \mathcal{TAB}$

M7. $M \models_{S,t} (\bigvee a : p)$ iff $(\exists b : b \in \mathcal{B}$ and $M \models_{S,t} p|_b^a)$, where $p \in (\mathcal{TAB}_s \setminus \mathcal{TAB})$

M8. $M \models_{S,t} p\mathsf{U}q$ iff $(\exists t' : t \leq t'$ and $M \models_{S,t'} q$ and $(\forall t'' : t \leq t'' \leq t' \Rightarrow M \models_{S,t''} p))$

M9. $M \models_{S,t} x[\pi]p$ iff $(\forall t_0, t_1 \in S : t_0 \leq t \leq t_1$ and $[S; t_0, t_1] \in [\![\pi]\!]^x \Rightarrow (\exists t' : t < t' \leq t_1$ and $M \models_{S,t'} p))$

M10. $M \models_{S,t} x\langle\pi\rangle p$ iff $(\exists t_0, t_1 \in S : t_0 \leq t \leq t_1$ and $[S; t_0, t_1] \in [\![\pi]\!]^x$ and $(\exists t' : t < t' \leq t_1$ and $M \models_{S,t'} p))$

M11. $M \models_{S,t} x|\!\langle\pi\rangle\!|p$ iff $(\exists t_0, t_1 \in S : t_0 \leq t \leq t_1$ and $[S; t_0, t_1] \in [\![\pi]\!]^x$ and $(\exists t' : t < t' \leq t_1$ and $(\forall t'' : t \leq t'' \leq t' \Rightarrow M \models_{S,t''} p)))$

M12. $M \models_{S,t} p \wedge q$ iff $M \models_{S,t} p$ and $M \models_{S,t} q$

M13. $M \models_{S,t} \neg p$ iff $M \not\models_{S,t} p$

M14. $M \models_{S,t} p$ iff $M \models_t p$, where $p \in \mathcal{TAB}$

3 Coherence Constraints

The above semantic definitions are fairly obvious. Our major modifications from traditional approaches have been

- the introduction of certain action operators: $[\,]$, $\langle\,\rangle$, $|\!\langle\,\rangle\!|$, and $\bigvee$.
- the generalization of the underlying model of time to be discrete or continuous, or any mixture of the two.

But in so generalizing the simpler approaches, have we given up too much? Have we lost certain important properties of actions that arose incidentally in the traditional approaches? We have indeed.

For the present approach to be coherent and useful, further technical constraints on models are required. These constraints, which can be thought of as expressing metaphysical necessities in the sense of McCarthy & Hayes, are motivated and formalized below. Coherence constraints are an important—and common—technical means by which logical semantic approaches can be made declarative and modular. Different constraints capture different properties of the subject of interest, ideally in a mutually independent manner. By presenting the constraints separately from the initial framework, we are able to highlight the trade-offs involved in capturing different properties of actions, as well as the interrelationships among them. Intuitively, this is the same idea as behind *correspondence theory* in modal logic, which inspires the present work [van Benthem, 1984].

It is an interesting sidelight that the development of the constraints below reflects the historical development of our approach: we started out with general definitions such as those above, but in trying to show that various intuitive properties held, we were forced to make additional constraints explicit. These constraints capture, what to us are, some of the key properties of the commonsense notion of actions. Section 3.1 discusses constraints pertaining to actions and section 3.2 those pertaining to time (and actions). Section 3.3 introduces branching determinism, which is a constraint essential to modeling physical computational systems in a branching-time framework.

3.1 Actions

Simply put, the framework of actions of section 2 defines an intension for each action as performed by an agent. The intensions of different actions are sets of periods, essentially the periods over which those actions occur. This is a simple and appealing way to characterize actions and is closely related to the intuitions behind the traditional approaches. However, the sets of periods that can legitimately be denotations of actions must be constrained in various ways so that certain desirable properties are obtained. We discuss different such properties and their corresponding constraints next. The constraints on actions are expressed using a set $\mathcal{A} \subseteq \mathcal{PROG}$. Initially, we assume that $\mathcal{A} = \mathcal{B}$, the set of basic actions. Later, in section 4, we allow it to be other subsets of $\mathcal{PROG}$, so as to test if the constraints apply to more general classes of actions.

C1. **Actions Take Time:** All actions in $\mathcal{A}$ take time. Formally,

$(\forall \alpha \in \mathcal{A}, x, S, t, t' : [S; t, t'] \in [\![\alpha]\!]^x \Rightarrow t < t')$

This makes it possible for actions in $\mathcal{A}$ to result in state changes, although it does not require that. (States are formalized later.)

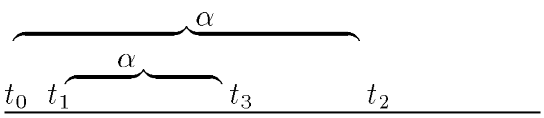


Figure 9: Case Disallowed by Uniqueness of Termination (1)

C2. **Uniqueness of Termination:** Starting at any given moment, each action in $\mathcal{A}$ can be performed in at most one way on any given scenario. Formally,

$(\forall \alpha \in \mathcal{A}, x, S, t_0, t_1, t_2, t_3 : [S; t_0, t_2] \in [\![\alpha]\!]^x$ and $[S; t_1, t_3] \in [\![\alpha]\!]^x \Rightarrow (t_0 \leq t_1 < t_2 \Rightarrow t_2 = t_3))$

This is needed to exclude ill-formed models in which an action does not have a unique moment of ending (see Figures 9 and 10). An agent cannot have

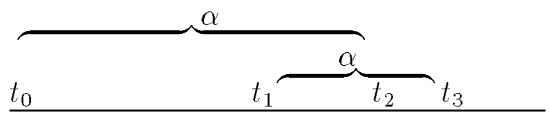


Figure 10: Case Disallowed by Uniqueness of Termination (2)

just performed the same action instance twice; the agent can of course perform different instances of the same type. As remarked in section 2.2, we concern ourselves with action instances only.

There may be uncertainty or lack of knowledge regarding the ending of an action, but that is captured like other kinds of ignorance or uncertainty. If an agent performs an action and then repeats it, the repetition counts as a separate instance, because it has a distinct starting moment. This constraint permits 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.

C3. **Uniqueness of Initiation:** Each instance of each action in $\mathcal{A}$ can be begun at exactly one moment. Formally,

$(\forall \alpha \in \mathcal{A}, x, S, t_0, t_1, t_2, t_3 : [S; t_0, t_2] \in [\![\alpha]\!]^x \text{ and } [S; t_1, t_3] \in [\![\alpha]\!]^x \Rightarrow (t_0 < t_3 \leq t_2 \Rightarrow t_0 = t_1))$

The motivation is similar to that for constraint C2. This constraint ensures that each action has a definite moment of starting. This interacts with constraint C4 and is revisited in section 4.1.

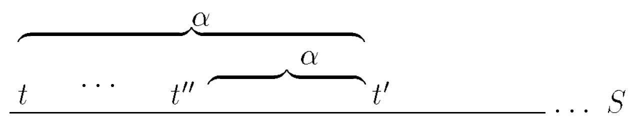


Figure 11: Actions in Progress

C4. **Actions in Progress:** The semantic definitions of our operators can be simplified if we require that an action in $\mathcal{A}$ is explicitly happening during the entire period of its occurrence. Formally,

$(\forall \alpha \in \mathcal{A}, x, S, t, t' : [S; t, t'] \in [\![\alpha]\!]^x \Rightarrow (\forall t'' : t \leq t'' < t' \Rightarrow [S; t'', t'] \in [\![\alpha]\!]^x))$

This allows us to talk of an agent's actions at any moment at which they are happening, not just where they begin. However, in accordance with constraint C2, actions begun at a moment still have a unique ending moment. This helps in relating our approach to previous approaches and is revisited in section 4.1. (Most of the simplification that results from this constraint is because the common operators are future-directed; an analogous constraint may be stated for past-directed operators as well, but we shall desist from doing so.)

Figure 11 shows how this constraint causes the intension of an action to be filled out by suffixes of the period over which it is performed. The period $[S; t', t']$ is not added to $[\![\alpha]\!]^x$, since that would cause ambiguity between an instance of α ending at t' and another instance of α beginning there. It would also lead to a violation of constraint C1.

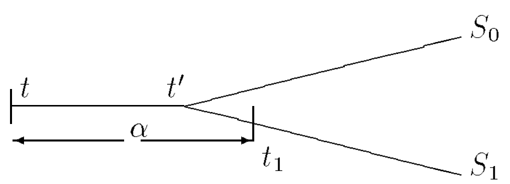


Figure 12: Actions Cannot be Partially Performed

C5. **Split Atomicity:** 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, performed with one atomic 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 possibly not be begun at all. Formally,

$(\forall \alpha \in \mathcal{A}, x, t, t', t_1 : t < t' < t_1 \Rightarrow (\forall S_0, S_1 \in \mathbf{S}_t : [S_1; t, t'] \subseteq S_0 \text{ and } [S_1; t, t_1] \in [\![\alpha]\!]^x \Rightarrow (\exists t_0 : [S_0; t, t_0] \in [\![\alpha]\!]^x)))$

Intuitively, $[S_1; t, t_1] \in [\![\alpha]\!]^x$ means that x is performing α from t to t_1. Therefore, he must be performing α in any subperiod of that period, including $[S_1; t, t']$, which is the same as $[S_0; t, t']$. Thus, α must be completed on S_0. Figure 12 illustrates this argument. Higher-level actions may not satisfy this. For example, Al may be crossing the street (on a given scenario) even if he did not cross it successfully on that scenario, e.g., by being run over by a bus. We revisit this point more abstractly in section 4.3.

In other words, the scenarios may split in the course of actions α by x for two reasons: (i) some other agent can exercise a choice at the moment of the split or (ii) x can decide whether to continue with α. In the former case, all scenario that split off must have α or neither may. In the latter case, the action α is really two actions by the same agent, since they are brought about by separate choices.

Example 8 In Figure 12, we can replace action α by actions b and c, where b lasts from t to t' and c from t' to t_1. ▮

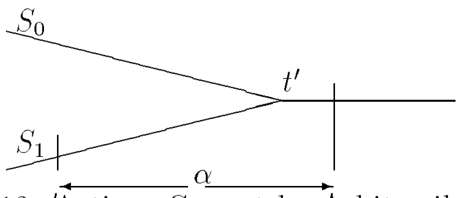


Figure 13: Actions Cannot be Arbitrarily Initiated

C6. **Merge Atomicity:** If an agent completes an action over a part of a scenario, then he must have initiated that action on that scenario. This makes sense since the actions in the model are basic actions, performed with one atomic choice by their agent. If the agent never made the appropriate choice along a scenario, he cannot be made to perform the action on it! This constraint is superficially similar to constraint C5, but it applies when scenarios merge, i.e., when the past is branching. As before, the motivation is that 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 possibly not be begun at all. Formally,

$(\forall \alpha \in \mathcal{A}, x, t, t', t_1 : t < t' < t_1 \Rightarrow (\forall S_0, S_1 \in \mathbf{S}_t : [S_1; t', t_1] \subseteq S_0$ and $[S_1; t, t_1] \in [\![\alpha]\!]^x \Rightarrow (\exists t_0 : [S_0; t_0, t_1] \in [\![\alpha]\!]^x)))$

Intuitively, $[S_1; t, t_1] \in [\![\alpha]\!]^x$ means that x is performing α from t to t_1. Therefore, he must be performing α in any subperiod of that period, including $[S_1; t', t_1]$, which is the same as $[S_0; t', t_1]$. Thus, α must be initiated on S_0. Figure 13 illustrates this argument.

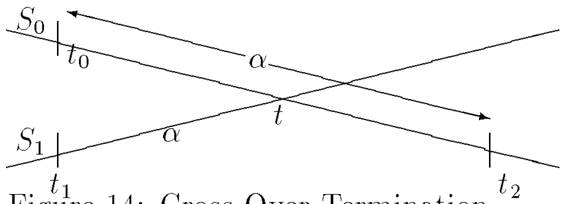


Figure 14: Cross-Over Termination

C7. **Cross-Over Termination Atomicity:** Suppose two scenarios intersect at a moment t. If on one scenario, an agent is performing an action α at t and on the other scenario, the agent terminates α at t, then the agent could just as well have terminated α at t on both scenarios. In other words, if t represents a valid termination point for α, then α can be terminated on it no matter where it came from, provided it is being performed anyway. Figure 14 illustrates this constraint. Formally,

$(\forall \alpha \in \mathcal{A}, x, t, t_0, t_1, t_2 : t_0 < t \leq t_2 \Rightarrow (\forall S_0, S_1 : ([S_1; t_1, t] \in [\![\alpha]\!]^x \text{ and } [S_0; t_0, t_2] \in [\![\alpha]\!]^x) \Rightarrow [S_0; t_0, t] \in [\![\alpha]\!]^x))$

Note that if we required α to end on S_0 just because it ended on S_1, we would have got an extremely strong constraint, which would imply that a given moment can result from the same set of actions.

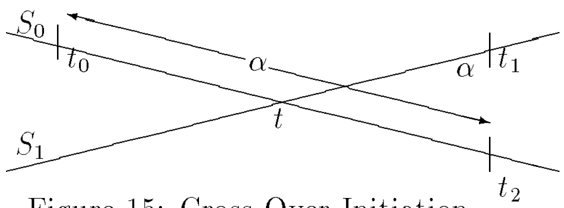


Figure 15: Cross-Over Initiation

C8. **Cross-Over Initiation Atomicity:** Suppose two scenarios intersect at a moment t. If on one scenario, an agent is performing an action α at t and on the other scenario, the agent initiates α at t, then the agent could just as well have initiated α at t on both scenarios. In other words, if t represents a valid initiation point for α, then α can be initiated on it no matter where it is going to, provided it is being performed there. Figure 15 illustrates this constraint. Formally,

$(\forall \alpha \in \mathcal{A}, x, t, t_0, t_1, t_2 : t_0 < t \leq t_2 \Rightarrow (\forall S_0, S_1 : ([S_1; t, t_1] \in [\![\alpha]\!]^x \text{ and } [S_0; t_0, t_2] \in [\![\alpha]\!]^x) \Rightarrow [S_0; t, t_2] \in [\![\alpha]\!]^x))$

Note that if we required α to begin on S_0 just because it began on S_1, we would trivialize the model, because every agent could do no more than one action at any moment! This would be intuitively worse than the analogous definition of constraint C7.

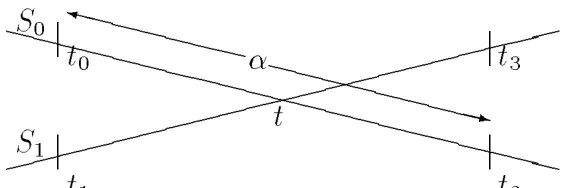


Figure 16: Cross-Over Progression

C9. **Cross-Over Progression Atomicity:** If α at t along one scenario, then it must be ongoing at t at every other scenario that passes through t. The rationale is that if t represents a valid point for α to proceed, then α should be ongoing at t independent of previous or later moments. Figure 16 illustrates this constraint. Formally,

$(\forall \alpha \in \mathcal{A}, x, t, t_0, t_2, S_0, S_1 : (t_0 < t < t_2 \text{ and } [S_0; t_0, t_2] \in [\![\alpha]\!]^x) \Rightarrow (\exists t_1, t_3 : (t_1 < t < t_3 \text{ and } [S_1; t_1, t_3] \in [\![\alpha]\!]^x)))$

This constraint is fairly strong, because it requires different intersecting scenarios to be quite similar—if α is ongoing on S_0, then α is ongoing on S_1 and therefore something else is prevented from happening at t on S_1.

3.2 Time

Now we present some coherence constraints that apply primarily to time.

C10. **Reality is Stable:** Given a real scenario, R, at all moments on R, the real scenario must be the appropriate suffix of R. Formally,

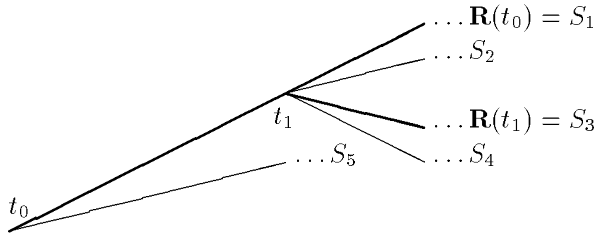


Figure 17: An Example of a Discontinuity in Reality

$(\forall t, t' : t' \in \mathbf{R}(t) \Rightarrow \mathbf{R}(t') \subseteq \mathbf{R}(t))$.

Figure 17 exhibits a model that violates this constraint. In that model, S_1 is the real scenario at t_0, but even as we proceed along that scenario, we come to a moment t_1 where the reality is no longer along S_1—clearly absurd with respect to our notion of reality.

C11. **Totality of Actions:** If an agent exists at a moment, than he performs some action at that moment. What is done may of course be some kind of a "dummy" action in the application of interest. However, it means that the assignment to actions is "total." For each moment (which is not a last moment) there is at least one period where the agent is assigned some action. Formally,

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

This assumption ensures that time does not just pass by itself, and is needed to make the appropriate connections between time and action. Otherwise, we would be left with moments in the model where no action is being performed.

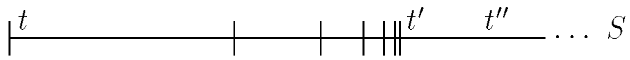


Figure 18: Limit Sequences Disallowed by Reachability of Moments

C12. **Reachability of Moments:** This constraint, when applicable, requires the following. For any scenario and two moments on it, there is a finite number of actions of each agent that, if performed on that scenario starting at the first moment, will lead to a moment in the future of the second moment. Formally,

$(\forall x, S : (\forall t, t' \in S : t < t' \Rightarrow (\exists t_0, t_1 : t_0 \leq t < t' \leq t_1 \text{ and } (\exists a_1, \ldots, a_n \text{ and } [S; t_0, t_1] \in [\![a_1 \cdot \ldots \cdot a_n]\!]^x))))$

This condition is intended to exclude models in which there are moments that would require infinitely long action sequences to reach. Such models, e.g., as in Figure 18, would allow a condition to be inevitable and yet unreachable through any finite sequence of actions.

C13. **Bounded Nondeterminism:** This constraint, when applicable, requires that the set of possible branches at a moment, t be finite. Formally,

$(\forall x, t : \bigcup_a \{t' : [S; t, t'] \in [\![a]\!]^x\} \text{ is finite})$

Example 9 Consider an agent who is controlling a valve. Under constraint C13, the agent has only finitely many valve settings and even in conjunction with the actions of all the other agents, only finitely many distinct resulting moments are possible. ∎

Constraints C12 and C13 are suggestive of a discrete control. In other words, time may be dense, but the actions impose a discrete view on top of it. Constraint C12 requires that the actions be "discrete" in the sense of being sufficiently long. Thus, the agents make their choices at discrete intervals. This constraint always holds in discrete models. Constraint C13 requires that the available choices should discriminate among a countable set of moments.

3.3 Branching Determinism

Our models represent physical systems, albeit nondeterministic ones. Section 1 gave some reasons why nondeterminism is essential to our purposes. However, since our target systems are physical, 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 agents' 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. We relate moments and states as follows. Define a relation $\sim$ to indicate the state-equivalence of moments and periods. The state at a moment is precisely characterized by the atomic propositions that hold at that moment.

Definition 8 For $t, t' \in \mathbf{T}$, $t \sim t'$ iff $\{\psi \in \Phi | t \in [\![\psi]\!]\} = \{\psi \in \Phi | t' \in [\![\psi]\!]\}$.

Observation 4 $\sim$ is an equivalence relation. $\Box$

Intuitively, $t \sim t'$ means that the same physical state occurs at moments t and t'.

Definition 9 States are the equivalence classes of $\sim$.

Definition 10 Given two sets U and V with an order $\prec$, a map f from U to V is an order-isomorphism iff (a) f is bijective and (b) $(\forall t, t_0 \in U : t \prec t_0$ iff $f(t) \prec f(t_0))$.

Definition 11 For sets of moments, U and V, we define $U \approx V$ in terms of an order-isomorphism, f. $U \approx V$ iff $(\exists f : f$ is an order-isomorphism and $(\forall t \in U \Rightarrow t \sim f(t)))$.

Observation 5 $\approx$ is an equivalence relation. $\Box$

$U \approx V$ means that the moments in U and V represent the same states occurring in the same temporal order. In other words, U and V represent the same trajectory in state space.

Definition 12 A scenario S' is a *suffix* of a scenario S iff $S' \subseteq S$ and $(\forall t \in S \setminus S', t' \in S' : t < t')$.

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]. We generalize these from discrete time, which is how they were originally developed and presented.

Definition 13 $\mathbf{S}$ is *suffix-closed* if the following property holds. If $S \in \mathbf{S}$, then all suffixes of S belong to $\mathbf{S}$.

Lemma 1 $(\bigcup_{t \in \mathbf{T}} \mathbf{S}_t)$ is suffix-closed. $\Box$

Definition 14 $\mathbf{S}$ is *limit-closed* if the following property holds. Let V be a set of states with an initial state. Let S_i be a sequence of scenarios such that $V \cap S_i \subset V \cap S_{i+1}$ and $(\forall v \in V : (\exists t \in v : (\exists i : t \in S_i)))$. Then, there exists a scenario $S \in \mathbf{S}$ such that $(\forall v \in V : (\exists t \in v : t \in S))$.

Lemma 2 $(\bigcup_{t \in \mathbf{T}} \mathbf{S}_t)$ is limit-closed. $\Box$

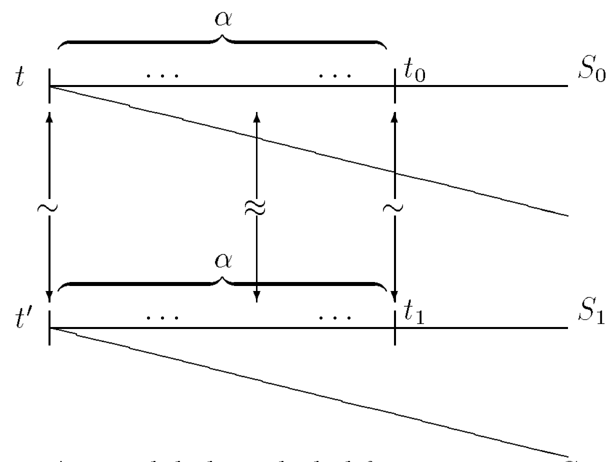
Definition 15 If $(\forall t \in U, t' \in V : t < t')$ then $U \circ V = U \cup V$, else $U \circ V$ is undefined. Intuitively, $\circ$ denotes concatenation of sets of moments. For simplicity, we assume $\circ$ associates left to right. We abbreviate $U \circ \{t\}$ as $U \circ t$.

Definition 16 **S** is *fusion-closed* if the following property holds. Let $S_0 = S_0^p \circ t_0 \circ S_0^f$ and $S_1 = S_1^p \circ t_1 \circ S_1^f$, such that $t_0 \sim t_1$, be two scenarios in **S**, which include the same *state*. Then, $(\exists t_2, t_3 : t_2 \sim t_0$ and $t_3 \sim t_1$ and $S_0^p \circ t_2 \circ S_1^f \in \mathbf{S}$ and $S_1^p \circ t_3 \circ S_0^f \in \mathbf{S})$. Intuitively, these scenarios are formed by concatenating the initial and later parts of S_0 and S_1. (Notice that S_i^p and S_i^f can be empty.)

The same state that occurs in the antecedent of the definition of fusion closure may arise at different (incomparable) moments. Thus, there may be no scenario in $(\bigcup_{t \in \mathbf{T}} \mathbf{S}_t)$ corresponding to $S_0^p \circ t \circ S_1^f$. Hence, we do not have a positive result, but we can add a constraint to achieve it. What this means is that we must weaken the definition of fusion closure to accommodate states.

Lemma 3 $(\bigcup_{t \in \mathbf{T}} \mathbf{S}_t)$ is not fusion-closed. $\Box$

Definition 17 **S** is *weak stative fusion-closed* if the following property holds ("stative" because it involves states). Let $S_0 = S_0^p \circ t_0 \circ S_0^f$ and $S_1 = S_1^p \circ t_1 \circ S_1^f$ be two scenarios in **S**, such that $t_0 \sim t_1$. Then there are scenarios $S_2 \approx S_0^p \circ t_2 \circ S_1^f$ and $S_3 \approx S_1^p \circ t_3 \circ S_0^f$, such that $t_2 \sim t_0$ and $t_3 \sim t_1$ and S_2 and S_3 belong to **S**. In other words, this is fusion closure up to isomorphism.



Action labels included for constraint C15

Figure 19: Branching Determinism

C14. **Weak Branching Determinism:** Consider two moments with the same state. Then the fragments of the model rooted at those moments must be isomorphic with respect to temporal precedence. Thus, we can define branching determinism as the following constraint.

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

In other words, if two moments satisfy exactly the same atomic propositions, then they lead to the same immediate possibilities, i.e., in periods immediately following the given moments. Figure 19 illustrates this constraint.

Theorem 4 Under constraint C14, with $\mathcal{A}$ set to $\mathcal{B}$, $(\bigcup_{t \in \mathbf{T}} \mathbf{S}_t)$ is weak stative fusion-closed. □

The above theorem takes care of the evolution of states. However, it does not apply to actions on the different scenarios. Since we have an interest in actions, we would like to ensure that not only the states but also the actions in the system are constrained as above. This motivates the following definition.

Definition 18 For sets of moments, U and V, $U \cong V$ iff $U \approx V$ and $(\forall x, a : (\forall U' \subseteq U : U' \in [\![a]\!]^x \Rightarrow (\exists V' \subseteq V \text{ and } U' \approx V' \text{ and } V' \in [\![a]\!]^x)))$.

Observation 6 $\cong$ is an equivalence relation. □

$U \cong V$ means that U and V represent the same trajectory in state space with the same actions.

Definition 19 $\mathbf{S}$ is *strong stative fusion-closed* if the following holds. Let $S_0 = S_0^p \circ t_0 \circ S_0^f$ and $S_1 = S_1^p \circ t_1 \circ S_1^f$ be two scenarios in $\mathbf{S}$, such that $t_0 \sim t_1$. Then there are scenarios $S_2 \cong S_0^p \circ t_2 \circ S_1^f$ and $S_3 \cong S_1^p \circ t_3 \circ S_0^f$, such that $t_2 \sim t_0$ and $t_3 \sim t_1$ and S_2 and S_3 belong to $\mathbf{S}$.

C15. **Strong Branching Determinism:** If two moments satisfy exactly the same atomic propositions, then any action that can be performed in one moment can be performed in the other moment, and for every period over which an action is performed at one moment, there is a corresponding period where the same action is performed at the other moment.

$(\forall x, \alpha \in \mathcal{A}, t, t', t_0, t_1 \in \mathbf{T}, S \in \mathbf{S}_t : (t \sim t' \text{ and } t_0 \leq t < t_1 \text{ and } [S; t_0, t_1] \in [\![\alpha]\!]^x) \Rightarrow (\exists t'_0, t'_1, S' \in \mathbf{S}_{t'} : t'_0 \leq t' < t'_1 \text{ and } [S'; t'_0, t'_1] \in [\![\alpha]\!]^x \text{ and } [S; t_0, t_1] \approx [S'; t'_0, t'_1]))$

In other words, the state at a moment determines the actions that are initiated or are ongoing at that moment. Figure 19 illustrates this constraint.

Theorem 5 Under constraint C15, with $\mathcal{A}$ set to $\mathcal{B}$, $(\bigcup_{t \in \mathbf{T}} \mathbf{S}_t)$ is strong stative fusion-closed. $\Box$

Branching determinism gives us the essential property of physical systems namely that their future behavior is completely characterized by their present state. By "completely characterized," we mean that the set of possibilities is determined by the state of the system, not that there is a unique possibility. The behavior that is realized, i.e., the one that corresponds to the real scenario, depends also on how the different agents choose their actions. We do not capture that within the formal model—additional primitives may be defined to capture them. By contrast, determinism would require a unique scenario at each state, thereby leading to a limited models, since at most one scenario would emerge from any moment. In such a model, it would not be possible reason about the choices of actions available to agents, nor of how proper choices might really be effected.

4 Results and Discussion

It is helpful in intuitively understanding formal definitions to attempt to prove some technical results that should follow from them. For this reason, we state and discuss some consequences of the above model and semantic definitions next. Section 4.1 discusses some interactions among the coherence constraints; section 4.2 relates the dynamic and temporal operators; section 4.3 how the constraints may extend to regular programs; and, section 4.4 states some natural models for the constraints.

4.1 Constraints

The above constraints are not all mutually satisfiable in all models. Uniqueness of termination (C2) entails that the actions have a definite moment of ending. We can convert a model that does not satisfy this constraint to one that satisfies it by replacing the periods in the intensions for actions by periods whose terminal moment was maximal. This can be done, however, only if the maximal terminal moment for each action instance is defined—i.e., it is definite. Uniqueness of initiation (C3) has the same effect for when the action instances begin. Together, these entail that the action instances do not overlap and have definite boundaries. McDermott [1982] noted that actions do not overlap, but was less clear about the existence of limits for the intensions of actions.

Interestingly, constraint C4, which fills in the intensions of actions with all suffixes of periods that are in the intension, is satisfiable with constraints C2 and C3. However, the three are satisfiable only if the model is discrete with unit length actions. Lemma 16 of section 4.2 shows this simplifies the operators.

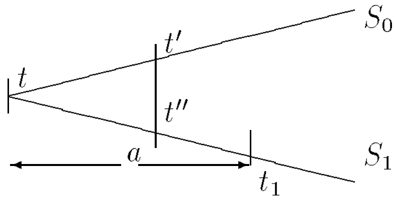


Figure 20: Impossibility of Splitting Scenarios

It is instructive to consider a possible simple approach to enforcing constraint C5. One might attempt to leave the actions alone (rather than expand them as in Example 8) and instead attempt to expand the set of available scenarios in the model. Figure 20 diagrams such a split. However, in the context of constraint C15, the splitting of scenarios is no help at all. Since the moments t' and t'' are state-equivalent, both or neither may have a continuation to the state corresponding to moment t_1. There is thus a tension between constraint C5 and constraint C12 on the reachability of moments, since constraint C5 pushes us toward ever smaller basic actions. We believe that the atomicity of actions is conceptually more fundamental than the reachability of moments. The relationship between choice and basic actions is crucial, but there can be applications where discrete control is not appropriate. Constraint C6 is similar. Whereas constraint C5 says that an agent's autonomous choice cannot be turned off, constraint C6 says that it cannot be turned on.

The cross-over atomicity constraints (C7, C8, C9) essentially require that the moment encode some information about what actions are being initiated or terminated there and what actions are ongoing. Intuitively, if this information is attached to the moment, then the above constraints are simple "consistency" requirements. In other words, if on one scenario a moment encodes that an action is ongoing, then it must do so on every scenario (because the moment is independent of the scenarios in which it features). Consequently, the action must be ongoing along every scenario through that moment. The argument for termination and initiation is similar, but must be tempered by the fact that the moment cannot be treated independently of the scenarios it is on. As explained in section 3.1, if we made moments absolutely independent of scenarios, we would eliminate choices (that agents can exercise) from the model. Intuitively, the cross-over constraints push us toward models in which scenarios intersect when actions are initiated or terminated.

One might wonder if there is a relationship between uniqueness of termination (constraint C2) and cross-over termination atomicity (constraint C7). One can construct a scenario S_2 from the initial part of S_1 and the latter part of S_0. It is easy to see that even if S_2 satisfies constraint C2, S_0 and S_1 may not satisfy constraint C7—the reason being that we did not state that $[S_2; t_1, t_2] \in [\![\alpha]\!]^x$. In the opposite direction, if we set $S = S_1 = S_2$ and apply constraint C7, we might still violate constraint C2, because nothing requires t to equal t_2. The arguments relating uniqueness of initiation and cross-over initiation atomicity can be similarly addressed.

4.2 Operators

We now give some results that are specific to our approach in that they mix time and actions as proposed above. Lemma 6 states that if a can be performed on a given scenario and p holds while a is being performed, then p holds somewhere on the given scenario. More interestingly, Lemma 7 states that if while performing a a state is achieved on the future of which on the given scenario, p will hold, then p will hold right from the start (before a) on the given scenario. The converse of these also holds. In other words, Lemma 8 states that if $\mathsf{F}p$ holds on a scenario, then whether p already holds or there is some action that can be performed on the given scenario (from its initial moment) and $\mathsf{F}p$ holds at some moment while that action is being performed. These lemmas hold because of constraints C11 and C12.

Lemma 6 $(x\langle a\rangle p) \rightarrow \mathsf{F}p$ □

Lemma 7 $(x\langle a\rangle \mathsf{F}p) \rightarrow \mathsf{F}p$ □

Lemma 8 $\mathsf{F}p \rightarrow p \vee (\bigvee a : x\langle a\rangle \mathsf{F}p)$ □

Lemmas 9, 10, and 11 use essentially the same argument as the above in order to develop a corresponding relationship between G and the action operators. We use the fact (exhibited in Figure 7) that $x\neg[a]\neg p$ means that a is performed on the given scenario and p holds throughout a.

Lemma 9 $\mathsf{G}p \rightarrow p$ □

Lemma 10 $\mathsf{G}p \rightarrow (\bigvee a : x\neg[a]\neg \mathsf{G}p)$ □

Lemma 11 $(p \wedge x\neg[a]\neg \mathsf{G}p) \rightarrow \mathsf{G}p$ □

Lemmas 12 through 15 establish a relationship between U and the action operators. Intuitively, we need the last disjunct of Lemma 15, because $p\mathsf{U}q$ may hold because q holds at a moment during some action a, but q may fail to hold at the endpoints of a.

Lemma 12 $(p \wedge q) \rightarrow p\mathsf{U}q$ □

Lemma 13 $(p \wedge x\neg[a]\neg(p\mathsf{U}q)) \rightarrow p\mathsf{U}q$ $\Box$

Lemma 14 $(p \wedge x|\langle a\rangle|(p\mathsf{U}q)) \rightarrow p\mathsf{U}q$ $\Box$

Lemma 15 $p\mathsf{U}q \rightarrow ((p \wedge q) \vee (p \wedge (\bigvee a : x\neg[a]\neg(p\mathsf{U}q))) \vee (p \wedge (\bigvee a : x|\langle a\rangle|(p\mathsf{U}q))))$ $\Box$

Lemma 16 In models that satisfy constraints C2, C3, and C4 (i.e., discrete models with unit length actions),

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

This is why traditionally, simpler semantic definitions of the dynamic operators are acceptable and why one action operator suffices in such models. $\Box$

4.3 Regular Programs

Constraints C1, C2, C3, C4, and C5 of section 3.1 described some properties of basic actions. (Recall that those constraints were parameterized with a class of programs $\mathcal{A}$, so they could be applied to more general actions than basic actions.) It is instructive to consider how these properties might apply to regular programs in general. It is interesting that several of them do apply. In this subsection, we assume that the basic action symbols in the statements of the above constraints have been replaced by π to denote general regular programs.

Definition 20 An (n-ary) regular program operator, $\otimes$, *inductively satisfies* a constraint, C, if the regular program $\otimes(\tau_1 \ldots \tau_n)$ satisfies C whenever each τ_i satisfies C. Otherwise, $\otimes$ *inductively violates* C.

Theorem 17 The regular program operator ? satisfies constraints C2, C3, C4, C5, C6, C7, C8, C9, C14, and C15. It violates constraint C1. $\Box$

Theorem 18 The regular program operator + inductively satisfies constraints C1, C4, C5, C6, C9, C14, and C15. It inductively violates constraints C2, C3, C7, and C8. $\Box$

Theorem 19 The regular program operator $\cdot$ inductively satisfies constraints C1, C2, C3, C14, and C15. It inductively violates constraints C4, C5, C6, C9, C7, and C8. $\Box$

Theorem 20 The regular program operator $*$ inductively satisfies constraints C4, C5, C6, C7, C8, C14, and C15. It inductively violates constraints C1, C2, C3, and C9. $\Box$

The above results show that if we generalize actions to include regular programs, each of the constraints would be violated by some or the other construct. The violation of constraints C5 and C6 suggests that the given operator goes beyond basic actions. This is hardly surprising. What is surprising about it, however, is that these constraints are not violated by all of the constructs! The usual culprit in preventing the extension of basic action results to regular programs is the $\cdot$ operator. Surprisingly, $*$ inductively violates constraint C9, but it satisfies it if we use a stronger inductive hypothesis:

Theorem 21 If π satisfies constraints C7, C8, and C9, then $\pi*$ satisfies the same constraints. $\Box$

The inductive violation of constraint C5 is significant. It is this violation that enables a situation where an agent is performing a complex action on a scenario but fails to complete it on that scenario. This is nothing but a form of the famous *imperfective paradox* of [Dowty, 1977], whose classic example is that one may be crossing a street, yet never actually cross it. The other constraints show how much richer the terrain of actions in branching time can be than usually assumed.

4.4 Models

We considered some of the relationships among the various constraints. These give us an idea of the models for the various constraint. However, for concreteness we now consider some specific models that satisfy various of our coherence constraints. By establishing that specific models exist for our constraints, we can show that (a) the framework does specialize in an intuitive manner to the cases of interest, and (b) different combinations of the constraints are indeed satisfiable. Briefly, the traditional linear, discrete model satisfies all the constraints automatically. For the other models, additional restrictions are necessary. Interestingly, a continuous model with unit length actions may not satisfy the cross-over constraints, because two scenarios may intersect at a moment where an action terminates on one of the scenarios but not the other.

Definition 21 A linear discrete model, $M_{L,D}$, is one in which (a) $\mathbf{T}$ is isomorphic (with respect to $<$) to the natural numbers, (b) the actions are unit length, (c) the sole scenario at each moment is the real scenario at that moment—let the scenario at 0 be called N, (d) each agent acts at every moment: $(\forall x, t : (\exists a : [N, t, t+1] \in$

$[\![a]\!]^x$)), and (e) the states determine actions and next states: $(\forall x, a, t, t'; (t \sim t'$ and $[N; t, t+1] \in [\![a]\!]^x) \Rightarrow ((t+1) \sim (t'+1)$ and $[N; t', t'+1] \in [\![a]\!]^x))$.

Theorem 22 With $\mathcal{A}$ set to $\mathcal{B}$ in each of the action constraints, model $M_{L,D}$ satisfies constraints C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, and C15. $\square$

Definition 22 A tree-like (i.e., branching-future and linear-past), discrete model, $M_{B,D}$, is one in which (a) each scenario is isomorphic (with respect to $<$) to the natural numbers, (b) the actions are unit length, (c) it is finitely branching, and (d) the real scenario at each moment is chosen in accordance with constraint C10.

Theorem 23 With $\mathcal{A}$ set to $\mathcal{B}$ in each of the action constraints, model $M_{B,D}$ satisfies constraints C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, and C13. $\square$

Definition 23 A linear continuous model, $M_{L,C}$, is one in which (a) $\mathbf{T}$ is isomorphic (with respect to $<$) to the nonnegative real numbers, (b) the actions are unit length, (c) the sole scenario at each moment is the reality at that moment—let the scenario at 0 be called R, (d) each agent acts at every moment: $(\forall x, t : (\exists a, t_0 : t_0 \leq t$ and $[R, t_0, t_0 + 1] \in [\![a]\!]^x))$, and (e) the states determine immediate future states: $(\forall x, a, t, t'; t \sim t' \Rightarrow [R; t, t+1] \approx [R; t', t'+1])$

Theorem 24 With $\mathcal{A}$ set to $\mathcal{B}$ in each of the action constraints, model $M_{L,C}$ satisfies constraints C1, C2, C3, C5, C6, C7, C8, C9, C10, C11, C12, C13, and C14. $\square$

Definition 24 A branching (i.e., branching-future and branching-past), continuous model, $M_{B,C}$, is one in which (a) each scenario is isomorphic (with respect to $<$) to the real numbers, (b) the actions are unit length with nonoverlapping periods in their intensions, (c) each agent can do a finite number of actions and each action is performed on a finite number of scenarios, and (d) the real scenario at each moment is chosen in accordance with constraint C10. Note that there is no initial moment in this model.

Definition 25 A branching (i.e., branching-future and branching-past), continuous model, $M_{B,C,V}$, is one in which (a) each scenario is isomorphic (with respect to $<$) to the real numbers, (b) the actions are variable nonzero length with nonoverlapping periods in their intensions, (c) each agent can do a finite number of actions and each action is performed on a finite number of scenarios, and (d) the real scenario at each moment is chosen in accordance with constraint C10.

Theorem 25 With $\mathcal{A}$ set to $\mathcal{B}$ in each of the action constraints, models $M_{B,C}$ and $M_{B,C,V}$ satisfy constraints C1, C2, C3, C10, C11, C12, and C13. $\square$

5 Conclusions and Open Problems

A mathematical understanding of actions and time is crucial to designing and analyzing intelligent agents. We sought to generalize the formalization of actions, so that several important properties are obtained. These include (a) the actions being of different durations, (b) the actions being performed concurrently by different agents, (c) the underlying notion of time being variously continuous or discrete, and (d) the underlying notion of time allowing branching into the future. We began with essentially uninterpreted moments and actions, defined the core notion of intension, and then stated coherence constraints that capture the intuitive properties of actions in different cases of interest. Our framework can thus serve as an underpinning for further research on notions such as intentions and know-how. Previous research on these concepts has been shackled by poor models of time and action, thereby leading to unnecessarily restricted or spurious results.

The logic CTL* was designed over a decade ago for reasoning about programs. It has increasingly found its way into the AI literature, e.g., [Bacchus & Kabanza, 1995; Rao & Georgeff, 1991; Singh & Asher, 1990; Wooldridge, 1995]. Therefore, it is a good choice for a formal language for a conceptual framework. CTL* has also been used in the specification of BDI systems. [Rao & Georgeff, 1991] develop a framework in which various properties of intentions can be stated and compared. They adopt the usual simplifying assumptions about discreteness and allowing only one action at a time. All too often the models of actions are secondary to the primary goal of specifying properties about concepts such as intentions. We believe that the present exercise will strengthen the hands of researchers exploring situations including richer models of actions. If the research community converges on richer models than the traditional ones, superior results could be more easily obtained and reported.

Usually, the models proposed for CTL* are discrete with either no explicit actions or unit length actions performed one at a time. We extended CTL* with new operators and gave our language a more general semantics that allows time to be discrete, dense, or continuous. One of our concerns was that our definitions specialize to each case properly. This is useful since our models must often function at multiple levels of abstraction. We also discovered that several constraints must be stated on models to capture the intuitive notion of basic actions. The sole traditional constraint of no overlap [McDermott, 1982] says too little.

Intuitively, acceptable models for the formal language must be branching to capture the branching-time operators and to enable us to represent agents' choices. Such models do not presuppose a tree structure in the technical presentation, because an equivalent presentation can be given using only sets of linear structures with no explicit branching structure. Our discussion in section 3.3 used such linear structures under the name of **S**. We find such linear structures acceptable—if inelegant—provided they meet the corresponding versions of the various constraints.

Recently, there has been a resurgence of interest in the situation calculus [McCarthy & Hayes, 1969], with a number of approaches being proposed that relax the requirements of the traditional models of the situation calculus, e.g., [Miller, 1996; Reiter, 1996]. These approaches are beginning to tackle some of the issues we discussed here, but they have not yet accommodated the views of branching time, and of the various coherence constraints that we motivated above. It would be fair to state that the model-theoretic aspects of actions studied here are of secondary importance in the other work. Additional predicates are being defined in the situation calculus community to handle the temporal and dynamic properties for which we used the standard operators. We have no doubt that the development of the present paper could be expressed in some enhanced version of the situation calculus with second-order quantification (or equivalent) to talk about scenarios. However, we believe that at this stage of the present research program of developing a model theory of actions, it is best to borrow the ideas and notations of classical computing and enhance them only where necessary.

A useful enhancement to our approach is through the attachment of probabilities to the various actions. We believe that sound probabilistic constraints can be stated in such a case, and the calculations given in [van Fraassen, 1981; Haddawy, 1996] can be carried out in the proposed approach with equal facility. We expect to develop the relevant constraints in future work.

Even though decision procedures are known for CTL*, no closed-form axiomatization is known [Emerson, 1994]. This is an important open problem, as is determining an axiomatization for our language, $\mathcal{TAB}$. Further research is required to determine the role of past-directed operators. Such operators are known to make formulae drastically compact in some cases, but they also raise the complexity of the required decision procedures significantly. Would it help to augment $\mathcal{TAB}$ with operators such as *since*? For models with exclusively unit length actions, one action operator is enough (instead of three). Are there other interesting classes of models for which $\mathcal{TAB}$ can be simplified?

We have focused here on representational issues. We have not explored the trade-offs between expressiveness and computational complexity. Clearly, efficiency can be gained by simplifying the formal language and model. One class of reasoning techniques that is likely to prove of much value is the one developed in the qualitative reasoning community, which routinely deals with continuous phenomena and looks for ways to express them in terms of significant transitions [Kuipers, 1986; Sandewall, 1989; Kuipers & Shults, 1994]. One approach is to define sublanguages of the formalism that have the expressive power needed for specific applications, and no more. For appropriately designed specializations of the model, such a sublanguage might lead to considerable improvements in complexity. For the purposes of reasoning, it is important to draw connections with the mu-calculus, which forms the basis for many systems for logics of programs [Burch *et al.*, 1990]. Some preliminary ideas are reported in [Singh, 1994a].

This paper is meta-theoretic, rather than experimental—it develops a general framework that can capture assumptions of a range of possible models. It would not necessarily be a good idea to implement such a framework directly. It would be better to implement specific systems optimized for the most desirable constraints. We hope that the research community will converge on sets of such constraints.

We showed that when actions are studied in a branching-time framework, which is desirable for many applications, a rich variety of properties of actions are immediately exposed. We also showed how different classes of models can be studied in a unified framework and properties of actions identified that apply uniformly across such models. This is significant for representing and reasoning intuitively about actions and time at different levels of abstraction and detail. In summary, we take important first steps in developing a general model theory of actions, but much work remains to be done.

6 Acknowledgments

A previous version of this paper appears as *Formalizing Actions in Branching Time: Model-Theoretic Considerations* in the Proceedings of the International Workshop on Temporal Representation and Reasoning (TIME), 1995. I am indebted to the TIME-95 reviewers and audience, especially Fahiem Bacchus, for comments. Over the years, I have benefited from several discussions on actions and time with Nicholas Asher and Allen Emerson.

References

[Allen, 1984] Allen, James F.; 1984. Towards a general theory of action and time. *Artificial Intelligence* 23(2):123–154.

[Bacchus & Kabanza, 1995] Bacchus, Fahiem and Kabanza, Froduald; 1995. Using temporal logic to control search in a forward chaining planner. In *Proceedings of the International Workshop on Temporal Representation and Reasoning.*

[Burch *et al.*, 1990] Burch, J. R.; Clarke, E. C.; McMillan, K. L.; Dill, D. L.; and Hwang, L. J.; 1990. Symbolic model checking: 10^{20} states and beyond. In *Proceedings of the International Symposium on Logic in Computer Science.*

[Buvac & Costello, 1996] Buvac, Sasa and Costello, Tom, editors. *Proceedings of Common Sense 96: Third Symposium on Logical Formalizations of Commonsense Reasoning*. `http://www-formal.Stanford.EDU/tjc/96FCS/Final-Papers/`.

[Dean & Boddy, 1988] Dean, Thomas and Boddy, Mark; 1988. Reasoning about partially ordered events. *Artificial Intelligence* 36:375–399.

[Dennett, 1987] Dennett, Daniel C.; 1987. *The Intentional Stance*. MIT Press, Cambridge, MA.

[Dowty, 1977] Dowty, David; 1977. Towards a semantic analysis of verb aspect and the English imperfective progressive. *Linguistics and Philosophy* 1:45–77.

[Emerson, 1990] Emerson, E. A.; 1990. Temporal and modal logic. In Leeuwen, J.van, editor, *Handbook of Theoretical Computer Science*, volume B. North-Holland Publishing Company, Amsterdam, The Netherlands.

[Emerson, 1994] Emerson, E. Allen; 1994. Personal Communication.

[Galton, 1990] Galton, Antony; 1990. A critical examination of Allen's theory of action and time. *Artificial Intelligence* 42:159–188.

[Goldman, 1970] Goldman, Alvin I.; 1970. *A Theory of Human Action*. Prentice-Hall, Inc., Englewood Cliffs, NJ.

[Grosz & Sidner, 1986] Grosz, Barbara and Sidner, Candace; 1986. Attentions, intentions, and discourse structure. *Computational Linguistics* 12(3):175–204.

[Haddawy, 1996] Haddawy, Peter; 1996. Believing change and changing belief. *IEEE Transactions on Systems, Man, and Cybernetics Special Issue on Higher-Order Uncertainty* 26(5).

[Harper *et al.*, 1981] Harper, William L.; Stalnaker, Robert; and Pearce, Glenn, editors. *IFS: Conditionals, Belief, Decision, Chance, and Time*. D. Reidel, Dordrecht, Netherlands.

[Kozen & Tiurzyn, 1990] Kozen, Dexter and Tiurzyn, Jerzy; 1990. Logics of program. In Leeuwen, J.van, editor, *Handbook of Theoretical Computer Science*. North-Holland Publishing Company, Amsterdam, The Netherlands.

[Kuipers & Shults, 1994] Kuipers, Benjamin J. and Shults, Benjamin; 1994. Reasoning in logic about continuous systems. In *Fourth Conference on Knowledge Representation and Reasoning*.

[Kuipers, 1986] Kuipers, Benjamin J.; 1986. Qualitative simulation. *Artificial Intelligence* 29:289–338.

[McCarthy & Hayes, 1969] McCarthy, John and Hayes, Patrick J.; 1969. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*. American Elsevier. Page numbers from the version reprinted in [Webber & Nilsson, 1981].

[McCarthy, 1979] McCarthy, John; 1979. Ascribing mental qualities to machines. In Ringle, Martin, editor, *Philosophical Perspectives in Artificial Intelligence*. Harvester Press. Page nos. from a revised version, issued as a report in 1987.

[McDermott, 1982] McDermott, Drew; 1982. A temporal logic for reasoning about processes and plans. *Cognitive Science* 6(2):101–155.

[Miller, 1996] Miller, Rob; 1996. A case study in reasoning about actions and continuous change. In *[Buvac & Costello, 1996]*.

[Moore, 1984] Moore, Robert C.; 1984. A formal theory of knowledge and action. In Hobbs, Jerry R. and Moore, Robert C., editors, *Formal Theories of the Commonsense World*. Ablex Publishing Company, Norwood, NJ. 319–358.

[Newell, 1982] Newell, Allen; 1982. The knowledge level. *Artificial Intelligence* 18(1):87–127.

[Pollack, 1991] Pollack, Martha E.; 1991. The uses of plans. Computers and Thought Award Lecture.

[Rao & Georgeff, 1991] Rao, Anand S. and Georgeff, Michael P.; 1991. Modeling rational agents within a BDI-architecture. In *Principles of Knowledge Representation and Reasoning*.

[Reiter, 1996] Reiter, Raymond; 1996. Natural actions, concurrency and continuous time in the situation calculus. In *[Buvac & Costello, 1996]*.

[Sandewall, 1989] Sandewall, Erik; 1989. Combining logic and differential equations for describing real-world systems. In *Principles of Knowledge Representation and Reasoning*.

[Shoham, 1988] Shoham, Yoav; 1988. *Reasoning About Change: Time and Causation from the Standpoint of AI*. MIT Press, Cambridge, MA.

[Singh & Asher, 1990] Singh, Munindar P. and Asher, Nicholas M.; 1990. Towards a formal theory of intentions. In *European Workshop on Logics in Artificial Intelligence.*

[Singh, 1992] Singh, Munindar P.; 1992. A critical examination of the Cohen-Levesque theory of intentions. In *10th European Conference on Artificial Intelligence.*

[Singh, 1994a] Singh, Munindar P.; 1994a. Maintenance and prevention: Formalization and fixpoint characterization. In *ECAI Workshop on Logic and Change.*

[Singh, 1994b] Singh, Munindar P.; 1994b. *Multiagent Systems: A Theoretical Framework for Intentions, Know-How, and Communications.* Springer Verlag, Heidelberg, Germany.

[Thomason & Gupta, 1981] Thomason, Richmond H. and Gupta, Anil; 1981. A theory of conditionals in the context of branching time. In *[Harper* et al.*, 1981].* 299–322.

[Turner, 1984] Turner, Raymond; 1984. *Logics for Artificial Intelligence.* Ellis Horwood Ltd., Chichester, UK.

[van Benthem, 1984] van Benthem, J. F. A. K.; 1984. Correspondence theory. In Gabbay, D. and Guenthner, F., editors, *Handbook of Philosophical Logic*, volume II. D. Reidel Publishing Company, Dordrecht, The Netherlands. 167–247.

[van Fraassen, 1981] van Fraassen, Bas C.; 1981. A temporal framework for conditionals and chance. In *[Harper* et al.*, 1981].* 323–340.

[Webber & Nilsson, 1981] Webber, Bonnie L. and Nilsson, Nils J., editors. *Readings in Artificial Intelligence.* Morgan Kaufmann.

[Wooldridge, 1995] Wooldridge, Michael; 1995. Time, knowledge, and choice. In *Intelligent Agents II: Proceedings of the IJCAI Workshop on Agent Theories, Architectures, and Languages.* Springer.