INTENTION IS COMMITMENT WITH EXPECTATION

A Thesis

by

JAMES SILAS CREEL

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2005

Major Subject: Computer Science

INTENTION IS COMMITMENT WITH EXPECTATION

A Thesis

by

JAMES SILAS CREEL

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Thomas Ioerger (Chair of Committee)	Donald Friesen (Member)
Christopher Menzel (Member)	Valerie Taylor (Head of Department)

May 2005

Major Subject: Computer Science

ABSTRACT

Intention Is Commitment with Expectation. (May 2005)

James Silas Creel, B.S., The University of Texas at Austin; B.A, The University of Texas at Austin

Chair of Advisory Committee: Dr. Thomas Ioerger

Modal logics with possible worlds semantics can be used to represent mental states such as belief, goal, and intention, allowing one to formally describe the rational behavior of agents. Agent's beliefs and goals are typically represented in these logics by primitive modal operators. However, the representation of agent's intentions varies greatly between theories. Some logics characterize intention as a primitive operator, while others define intention in terms of more primitive constructs. Taking the latter approach is a theory due to Philip Cohen and Hector Levesque, under which intentions are a special form of commitment or persistent goal. The theory has motivated theories of speech acts and joint intention and innovative applications in multiagent systems and industrial robotics. However, Munindar Singh shows the theory to have certain logical inconsistencies and permit certain absurd scenarios. This thesis presents a modification of the theory that preserves the desirable aspects of the original while addressing the criticism of Singh. This is achieved by the introduction of an additional operator describing the achievement of expectations, refined assumptions, and new definitions of intention. The modified theory gives a cogent account of the rational balance between agents' action and deliberation, and suggests the use of means-ends reasoning in agent implementations. A rule-based reasoner in Jess facilitates evaluation of the predictiveness and intuitiveness of the theory, and provides a prototypical agent based on the theory.

To Beth

ACKNOWLEDGMENTS

I would like to sincerly thank Dr. Thomas Ioerger, the chair of my advisory committee, for his formidable assistance and guidance throughout my graduate studies. This work would not have been possible without his inspiring AI and Multiagent systems classes and the excellent advice he gave me during the development of this thesis.

I would also like to thank my committee members, Dr. Donald Friesen and Dr. Christopher Menzel for their input and suggestions in this work. Dr. Friesen I thank especially for his munificent academic administration. Dr. Menzel I thank for his uncommonly elucidative commentary.

Finally, I thank Dr. Bart Childs and Dr. Nancy Amato for the assistance and support they provided me as graduate advisors during the course of my studies.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

A. Rational Agents

The concept of agents in software design has recently offered new benefits to programmers. Just as expert systems were applied to many surprising problems in the last century [31], agent based systems are now being applied to increasingly complex problems [1, 2, 38, 43, 26, 25, 19, 48, 54, 62]. In contrast with expert systems, which suit only classification problems, agent based systems can encompass solutions to a wide variety of AI problems involving human-computer interaction, information processing, planning, communication, and teamwork. For instance, countless agent based systems have been written to perform functions on the web involving processing and serving information [17, 20, 40].

When discussing agent systems, one often takes the *intentional stance*, under which agents are thought to have mental states such as beliefs, desires, wishes, etc. In answer to the question of whether such mental states should be ascribed to artificial agents, McCarthy [45] has argued that "To ascribe *beliefs, free will, intentions, consciousness, abilities,* or *wants* to a machine is legitimate when such an ascription expresses the same information about the machine that it expresses about a person. It is useful when the ascription helps us understand the structure of the machine, its past or future behaviour, or how to repair or improve it." Thus, the intentional stance is merely an abstraction tool for understanding and dealing with with complex systems.

Agents generally have a few characteristics that distinguish them from pro-

The journal model is *IEEE Transactions on Automatic Control.*

grams in general. Woolridge and Jennings [66] choose to define an agent as "a computer system that is *situated* in some environment, and that is capable of *autonomous action* in this environment in order to meet its design objectives." This definition is broad enough to extend even to thermostats and Unix demons, but one may investigate problems in these areas without the use of Agent-Oriented programming techniques [64, 57] or the intentional stance. As with Object-Oriented programming, the intentional stance is best applied when the abstractions intuitively fit the systems being described. Therefore, the the term *agent* is generally applied to programs that exhibit the following features (adapted from [66]): (1) autonomy: agents encapsulate a state which they must deliberate and act upon without direct outside intervention (2) reactivity[1]: agents are situated in an environment (real or simulated) that they must interact with in a timely fashion. (3) pro-activeness: in addition to responding to the environment, agents initiate goal directed behavior to affect their environment. Note that a purely reactive system cannot be pro-active. (4) social ability: agents interact with other agents.

Though purely reactive agents can be useful, proactive agents that exhibit goal directed behavior offer a greater promise of powerful applications, especially in the area of multiagent systems. These agents are most effective when endowed with learning capabilities [47], planning capabilities [3], or both [29].

Sometimes agents require learning capabilities to effectively interact with their environment. However, sometimes learning capabilities are undesirable:

[1] Note that this usage of the term *reactive* is but one of three usages of the term in AI, introduced by Kaelbing [37]. Pnueli's definition [50] extends to a larger class of systems and is useful outside AI. Connah and Wavish [12] define reactive agents as those agents which never reason explicitly about the environment. Reactive agents (in the Connah-Wavish sense) respond directly to stimulus without planning or deliberation, and can be termed *purely reactive*.

Georgeff, architect of the PRS agent system [26] gives the example of an air-traffic control system modifying its behaviour at run-time.

Planning capabilites are more generally applicable. The classical approach to planning problems typically involves languages similar to STRIPS [21], in which sets of propositional or first order literals represent environment states, goals are partially specified states, and actions are represented in terms of preconditions and postconditions. Goals are satisfied by consistent environment states. This formalization provides a means for agents to reason symbolically about interactions with environments, and about what they can achieve in the future. Planning of this sort is referred to as means-ends reasoning.

Rational, symbolic reasoning agents should at least have the ability to plan courses of events based on the preconditions and postconditions of actions. Indeed, one might consider many of our non-agent-based programs to have goals in this sense. But in open [32], multiagent environments, this ability alone seems inadequate. Practical reasoning requires both means-ends reasoning and deliberation [63] (pp 65–86). Programmers want to ascribe various mental states to processes that engage in complex reasoning and interaction, and logicians want to describe the mental states of such processes. In addition to goals and desires, sophisticated agents have other motivations that affect their decisions, including commitment and intention. These notions provide groundwork for notions of obligation and responsibility, making them integral to implementations of cooperation and teamwork. The philosopher Bratman [4, 5] notes that intentions influence an agent's behavior more strongly than goals or desires. Searle [55] argues that intention consists of *prior intention*, the premeditation of an intended act, and *intention in action*, the agent's self awareness in carrying out the intention.

Woolridge [63] shows that intentions influence practical reasoning in 4 ways.

(1) Intentions drive means-ends reasoning. Thus, upon forming intentions, agents should attempt to form plans to achieve those intentions, and revise those plans appropriately. (2) Intentions persist. That is, one should not give up an unachieved intention until it is believed impossible or the original reason for the intention is gone. (3) Intentions constrain future deliberation. An agent will not consider adopting intentions that conflict with its current intentions. (4) Intentions influence beliefs upon which future practical reasoning is based. In particular, one expects one's intentions to come true. Any theory of rational agency that models intention should fulfill at least these four requirements.

B. Related Logics and Languages of Agency

The importance of abstact concepts such as intention motivates the use of formal theories describing goal and intention directed behavior or rational agency. A popular approach to such formal theories is the use of modal logics with possible worlds semantics. Modal logics come in many flavors including epistemic logics of knowledge, doxastic logics of belief, conative logics of goal and desire, and deontic logics [33] of obligation and permissibility. Possible worlds semantics, originally used in epistemic logic by Hintikka [34], can provide meaning for any sort of normal modal logic under the framework devised by Kripke [28]. Researchers appreciate the power of these modal logics despite numerous problems including the logical omniscience of agents and a lack of architechtural grounding of the possible worlds, as described by Woolridge [63]. Possible worlds semantics also enjoy popularity due to the appeal of the associated correspondence theory [6].

Once one has settled on the idea of designing an artificial agent with intentional states, one has a choice of architectures. The classical approach to artificial intelligence is to apply logical deduction to formulae. Under this idealized notion of an agent as a theorem prover, an agent executes those actions that it can prove it should do on the basis of its state (data/knowledge base) and rules of deduction.

This approach suggests the use of logic programming languages such as Prolog [11] which perform backward chaining and resolution. A logic programming approach to multiagent legal systems is given by P. Quaresma and I. Rodrigues [51]. A logic programming language for multiagent systems is given by Consantini and Tocchio [13].

However, forward chaining systems (also known as production systems) like SOAR [42] have proven successful for agent architectures as well. Many such systems employ the rete match process [23] which allows excellent efficiency of execution. Languages that employ the rete match process include CLIPS [67], a forward chainer written in C which was developed for NASA, and its successor JESS (Java Expert System Shell) [53]. The efficiency of such systems permits them to respond to their environment reactively (as Kaelbing uses the term), which makes them well suited for such fast paced applications as the control of simulated fighter aircraft [36].

The formal logics that deal with rational agency must describe temporal aspects of the agent's world, including the passage of time and occurence of events. We thus find it necessary to integrate temporal logics into our modal logics. The two typical flavors of temporal logic are linear-time temporal logic and branching time temporal logic. The advantages of either system are analyzed at length by Emerson and Halpern [18]. A more recent survey of temporal logic in AI is given by Chittaro and Montanari [8]. Woolridge and Fisher developed a first-order branching time logic for multiagent systems [65].

It is unclear that either linear time or branching time temporal logic is superior for specification of agents. Generally, we find it necessary to use branching time temporal logics if the processes in question are nondeterministic and we must therefore explicitly represent multiple execution paths. Also, branching time structures have isomorphisms to game trees representing multiagent game theoretic interactions [41]. Woolridge and Pauly offer an application of modal logics and a type of branching time temporal logic known as ATL (Alternating-time Temporal Logic) which makes use of these game theoretic isomorphisms [49]. Linear time temporal logics, on the other hand, offer the benefit of simplicity.

To get an idea of the structure of a temporal reasoning agent consider the Concurrent MetateM language of temporal logic, developed by Michael Fisher[22]. It is based on direct execution of logical formulae of the form

$$\text{antecedent about the past} \quad \Rightarrow \quad \text{consequent about present and future}$$

Agents in a Concurrent MetateM system exist as concurrently executing processes that communicate via asynchronous broadcast message passing. Agent behaviour is based upon specification in temporal logic. This approach approximates the idealized notion of deductive agents as theorem provers. Agents in such a system can be termed *deductive reasoning agents.*

The Concurrent MetateM language deals only with temporal modalities. In the case of more complex intentional logics, it behooves us to examine the purely theoretical underpinnings of our representational schemes before attempting implementations, and to try to develop mathematically consistent and comprehensive models of agents' mental states. The ascription of intentional states to agents complicates the underlying logics and therefore obfuscates the details of implementation.

Integrated theories of rational agency include multiple modalities such as belief, knowledge, desire, wish, hope, goal, choice, commitment, intention, or obligation in various combinations and variations [9, 59, 15, 61, 52, 16, 39, 27, 14, 68]. A theory that includes any such modality or combination thereof can purport to model some aspect of rational agency. The success of such a model is determined by the mathematical and philosophical consistency of the logic itself and more importantly by its effectiveness in motivating innovative applications. Implementations of systems based on these logics are complicated by the fact that there is no architectural grounding for possible worlds semantics and by the fact that every integrated theory of rational agency may suggest one or more basic design approaches. Woolridge and van der Hoek provide a comparative survey of the primary approaches in the area of integrated logics of rational agency [60].

A famous approach introduced by Bratman and used by Rao and Georgeff in their logic of rational agency [52] is known as the Beliefs, Desires, and Intentions model (BDI). Under this framework, modal operators for belief, desire, and intention are treated as separate logical primitives. Georgeff and Lansky's Procedural Reasoning System (PRS) [26], which employs a BDI architecture, had as its first application domain fault detection on the NASA Space Shuttle.

C. The Cohen-Levesque Theory of Intentions

A slightly different approach from the logic of BDI is to introduce intention as a derived operator composed of other modalities, thus avoiding the introduction of a primitive modal operator for intention. Taking this latter approach is a well studied and venerable theory of agency entitled “Intention Is Choice with Commitment” [9] by Philip Cohen and Hector Levesque, henceforth C&L.

The logic is based on a linear time model, with modalities for goals and beliefs. Their approach, though rich in derived constructs is parsimonious with primitive constructs. Parsimony provides advantages in implementations, because modal operators are a source of great complication in the logic. Furthermore, C&L's adherence to formality combines with insight from philosophy of mind to produce what is in practice a robust and predictive theory, under which intentions exhibit certain desirable properties motivated by Bratman and avoid the side-effect problem (in which agents must intend all the forseen results of their intentions [24]). The theory has found several applications, such as Jennings industrial robotics application [35] and Tambe's multiagent system architecture, STEAM (Shell for TEAMwork)[58], which employs a rule-based system. The theoretical implications of this theory of intention extend beyond the model itself, since it has found use in the theory of joint intention [44] upon which Tambe's and Jennings' work is based, and in theories of speech acts [10]. C&L's theory of intention as a persistent goal was intended as a specification for the design of artificial agents (C&L p 257), and not a logic agents should use for reasoning about their or other agents' mental states. On this account it has been successful, as evidenced by the work of Tambe and Jennings. This stance confers upon us certain advantages in dealing with the logic, for we need not be conerned that it models intentional states in humans or other agents.

D. Problems with the Theory

The most well known of the criticisms of C&L is given by Singh in "A Critical Examination of the Cohen-Levesque Theory of Intentions" [56]. He proves some properties of the theory to be counterintuitive or incorrect. In particular, he shows that intentions permit certain absurd scenarios, that agents cannot maintain multiple intentions, and that agents cannot intend an action they have just completed. However, if the theory in its problematic form has provided the groundwork for interesting applications, a modified theory that avoids the problems while still meeting the desiderata shall prove a useful development. This thesis presents such a theory.

A fundamental construct in C&L's logic is the persistent goal (P-GOAL), which is a goal that an agent will not give up until it is believed achieved or forever unachieveable. Persistent goals are thus a sort of achievement goal, which is a goal to bring about something currently not the case. Persistent goals may be regarded as a form of *commitment*. Intention is modeled as a sort of persistent goal. C&L provide separate definitions for intention toward action and intention toward well formed formulas or states of the world. An intention toward an action entails a persistent goal that the action be done under certain circumstances.

C&L prove a theorem called *From persistence to eventualitites* stating that agents' P-GOALs will eventualy come about under certain conditions. Singh shows these conditions to be inadequate, and suggests that the theorem forces the agent's goals to come about even if the agent doesn't try to bring them about.

Singh's next criticisim is that that agents cannot intend the same action twice consecutively: the object of a persistent goal must be believed currently false, so if an agent has just successfully done an action, he will believe that action is done, and he cannot intend to presently do it again. The theory presented here allows agents with basic planning capabilities to overcome this restriction by intending their actions to have specific outcomes, or rationales.

Finally, Singh observes that it is impossible for agents to hold multiple simultaneous intentions when they are not sure which one they will finish first.

This demonstrates that agents should be able to adopt weak intentional states, where they have not yet settled on precise plans to bring about their intentions. This allows agents to maintain multiple intentions simultaneously. The theory presented here formalizes this notion.

E. Amendments to the Theory of Intention as a Persistent Goal

The theory presented here avoids the logical problems described, and does so without modification to the syntax of C&L's theory, or the low-level definitions. This requires modification of the assumptions of the theory. Then new definitions of intention which offer at least the advantages of the originals are presented. Finally, we consider an implementation of a minimal agent based on the new theory as a proof of concept.

Those who study intention (including Bratman and C&L) are concerned with the *rational balance* between an agent's deliberation and action. Rational balance addresses the problem that if agents act too hastily without planning things out, then they will reap bad results, whereas if agents constantly reconsider their actions, then they will accomplish nothing. A formal definition of intention should characterize where rational agents lie between these two extremes. In the theory presented, agents will intentionally act precisely when they can form plans to bring about their goals.

CHAPTER II

THE LOGIC OF “INTENTION IS CHOICE WITH COMMITMENT”

So that this document might be self contained, a primer on C&L’s theory is given, beginning with the syntax. The theory consists of a linear time temporal logic along with a conative and doxastic logic with possible worlds semantics.

A. Syntax

The formal syntax of C&L’s theory is given in table I. Some dynamic logic and other portions of the theory are omitted for reasons of space.

B. Semantics

Each sequence of events, a so called possible world, is represented by a function from the integers to primitive events. The temporal modalities HAPPENS and DONE are simply defined in terms of the linear sequence of events of each possible world. HAPPENS describes a sequence of events happening “next” after the current time. DONE describes a sequence of events happening as “just” having happened. The doxastic modalities BEL and SUSPECT are given in terms of the belief accessibility relation B among possible worlds, and the conative modalities GOAL and ACCEPT are defined in terms of the goal accessibility relation G.

The notation $[A \to B]$ is meant to denote the set of all functions from A to B. We define a model M as a structure $\langle U, Agt, T, B, G, \Phi \rangle$. Here, U, the universe of discourse is the union of three sets: Θ a set of things, P a set of agents, and E a set of primitive event types. $Agt \in [E \to P]$ specifies the single agent of an event. $T \subseteq [Z \to E]$ is a set of linear courses of events (intuitively,

Table I. Syntax

⟨ActionVariable⟩	::=	$a, a_1, a_2, \ldots, b, b_1, b_2, \ldots, e, e_1, e_2, \ldots$
⟨AgentVariable⟩	::=	$x, x_1, x_2, \ldots, y, y_1, y_2, \ldots$
⟨RegularVariable⟩	::=	$i, i_1, i_2, \ldots, j, j_1, j_2, \ldots$
⟨Variable⟩	::=	⟨AgentVariable⟩ $\mid$ ⟨ActionVariable⟩ $\mid$
		⟨RegularVariable⟩
⟨Numeral⟩	::=	$\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots$
⟨Predicate⟩	::=	(⟨PredicateSymbol⟩⟨Variable⟩$_1, \ldots,$
		⟨Variable⟩$_n$)
⟨PredicateSymbol⟩	::=	$Q, Q_1, Q_2, \ldots$
⟨Wff⟩	::=	⟨Predicate⟩ $\mid$
		$\neg$⟨Wff⟩ $\mid$
		⟨Wff⟩ $\vee$ ⟨Wff⟩ $\mid$
		$\exists$⟨Variable⟩⟨Wff⟩ $\mid$
		$\forall$⟨Variable⟩⟨Wff⟩ $\mid$
		⟨Variable⟩ = ⟨Variable⟩ $\mid$
		(HAPPENS⟨ActionExpression⟩) $\mid$
		(DONE⟨ActionExpression⟩) $\mid$
		(AGT⟨AgentVariable⟩⟨ActionVariable⟩) $\mid$
		(BEL⟨AgentVariable⟩⟨Wff⟩) $\mid$
		(SUSPECT⟨AgentVariable⟩⟨Wff⟩) $\mid$
		(GOAL⟨AgentVariable⟩⟨Wff⟩) $\mid$
		(ACCEPT⟨AgentVariable⟩⟨Wff⟩) $\mid$
		⟨TimeProposition⟩ $\mid$
		⟨ActionVariable⟩ $\leq$ ⟨ActionVariable⟩
⟨TimeProposition⟩	::=	⟨Numeral⟩
⟨ActionExpression⟩	::=	⟨ActionVariable⟩ $\mid$
		⟨ActionExpression⟩; ⟨ActionExpression⟩ $\mid$
		⟨Wff⟩?

possible worlds) specified as a function from the integers to elements of E. $B \subseteq T \times P \times Z \times T$ is the belief accessibility relation which is Euclidean, transitive and serial, yielding a KD45 doxastic logic. $G \subseteq T \times P \times Z \times T$ is the goal accessibility relation, and is serial, yielding a KD conative logic. Φ interprets predicates, that is $\Phi \subseteq [Predicate^k \times T \times Z \times D^k]$ where $D = \Theta \cup P \cup E^*$. Also, we define the relation $AGT \subseteq E^* \times P$, where $x \in AGT[e_1, \ldots, e_n] \Leftrightarrow \exists i, x = Agt(e_i)$. Thus, AGT, not to be confused with AGT, specifies the partial agents of a sequence of events. The operator AGT specifies the only agent of an event.

Semantics are given relative to a model M, an element of T σ, an integer n, and a set v. This v is a set of bindings of variables to objects in D such that if $v \in [Variable \rightarrow D]$, then v^d_x is that function which yields d for x and is the same as v elsewhere. If a model has a certain world σ that satisfies a Wff w at a given time under a certain binding, we write $M, \sigma, v, n \models w$. If all models under all bindings always satisfy a Wff w, that is w is valid, we write $\models w$.

The formal semantics are given in table II.

The test action α? occurs instantaneously if α is the case. An agent will believe a proposition iff it is true in all worlds given by the belief accessiblity relation B relative to σ. An agent suspects a proposition could be the case iff it is true in at least one world given by the belief accessiblity relation B relative to σ. An agent has a goal toward a proposition iff it is true in every world given by the goal accessibility relation G relative to σ. An agent accepts a proposition iff it is true in at least one world given by the goal accessibility relation G relative to σ. An agent's GOALs are the alternatives he implicitly chooses. What an agent SUSPECTs, the agent believes possible. Notice that BEL is the dual of SUSPECT in the sense that for any x and p, $\neg(\mathsf{BEL}\ x\ \neg p) \equiv (\mathsf{SUSPECT}\ x\ p)$, and vice versa, $\neg(\mathsf{SUSPECT}\ x\ \neg p) \equiv (\mathsf{BEL}\ x\ p)$. Likewise, GOAL is the dual of ACCEPT.

Table II. Semantics

1. $M, \sigma, v, n \models Q(x_1, \ldots, x_k)$	$\Leftrightarrow$	$\langle v(x_1) \ldots v(x_k) \rangle \in \Phi[Q, \sigma, n]$
2. $M, \sigma, v, n \models \neg\alpha$	$\Leftrightarrow$	$M, \sigma, v, n \not\models \alpha$
3. $M, \sigma, v, n \models (\alpha \vee \beta)$	$\Leftrightarrow$	$M, \sigma, v, n \models \alpha$ or $M, \sigma, v, n \models \beta$
4. $M, \sigma, v, n \models \exists x, \alpha$	$\Leftrightarrow$	$M, \sigma, v^x_d, n \models \alpha$ for some d in D
5. $M, \sigma, v, n \models \forall x, \alpha$	$\Leftrightarrow$	$M, \sigma, v^x_d, n \models \alpha$ for every d in D
6. $M, \sigma, v, n \models (x_1 = x_2)$	$\Leftrightarrow$	$v(x_1) = v(x_2)$
7. $M, \sigma, v, n, \models \langle \text{TimeProposition} \rangle$	$\Leftrightarrow$	$v(\langle \text{TimeProposition} \rangle) = n$
8. $M, \sigma, v, n, \models (e_1 \leq e_2)$	$\Leftrightarrow$	$v(e_1)$ is an initial subsequence of $v(e_2)$
9. $M, \sigma, v, n, \models (\mathsf{AGT}\ x\ e)$	$\Leftrightarrow$	$AGT[v(e)] = v(x)$
10. $M, \sigma, v, n, \models (\mathsf{HAPPENS}\ a)$	$\Leftrightarrow$	$\exists m, m \geq n,$ such that $M, \sigma, v, n \Vert a \Vert m$
11. $M, \sigma, v, n, \models (\mathsf{DONE}\ a)$	$\Leftrightarrow$	$\exists m, m \leq n,$ such that $M, \sigma, v, m \Vert a \Vert n$
12. $M, \sigma, v, n \Vert e \Vert n + m$	$\Leftrightarrow$	$v(e) = e_1 e_2 \ldots e_m$ and $\sigma(n + i) = e_i, 1 \leq i \leq m$
13. $M, \sigma, v, n \Vert a; b \Vert m$	$\Leftrightarrow$	$\exists k, n \leq k \leq m,$ such that $M, \sigma, v, n \Vert a \Vert k$ and $M, \sigma, v, k \Vert b \Vert m$
14. $M, \sigma, v, n \Vert \alpha? \Vert n$	$\Leftrightarrow$	$M, \sigma, v, n \models \alpha$
15. $M, \sigma, v, n \models (\mathsf{BEL}\ x\ \alpha)$	$\Leftrightarrow$	$\forall \sigma^*$ such that $\langle \sigma, n \rangle B[v(x)] \sigma^*$, $M, \sigma^*, v, n \models \alpha$
16. $M, \sigma, v, n \models (\mathsf{SUSPECT}\ x\ \alpha)$	$\Leftrightarrow$	$\exists \sigma^*$ such that $\langle \sigma, n \rangle B[v(x)] \sigma^*$, $M, \sigma^*, v, n \models \alpha$
17. $M, \sigma, v, n \models (\mathsf{GOAL}\ x\ \alpha)$	$\Leftrightarrow$	$\forall \sigma^*$ such that $\langle \sigma, n \rangle G[v(x)] \sigma^*$, $M, \sigma^*, v, n \models \alpha$
18. $M, \sigma, v, n \models (\mathsf{ACCEPT}\ x\ \alpha)$	$\Leftrightarrow$	$\exists \sigma^*$ such that $\langle \sigma, n \rangle G[v(x)] \sigma^*$, $M, \sigma^*, v, n \models \alpha$

C. Abbreviations, Assumptions, Constraints, and Definitions

The formulas below should be assumed to refer to all agents x, actions a, well formed formulas p, etc. unless stated otherwise.

1. Abbreviations

Several abbreviations will prove convenient in the logic. C&L define the empty sequence, $\mathsf{NIL} \equiv \forall x, (x = x)?$. Clearly, $\forall b, (\mathsf{NIL} \leq b)$, that is NIL is a subsequence of every event sequence. The abbreviation for the "singleton sequence" is $(\mathsf{SINGLE}\ e) \equiv (e \neq \mathsf{NIL}) \wedge (\forall x, (x \leq e) \rightarrow (x = e) \vee (x = \mathsf{NIL}))$. Also, these following versions of DONE and $\mathsf{HAPPENS}$ specify the agent: $(\mathsf{DONE}\ x\ a) \equiv (\mathsf{DONE}\ a) \wedge (\mathsf{AGT}\ x\ a)$ and $(\mathsf{HAPPENS}\ x\ a) \equiv (\mathsf{HAPPENS}\ a) \wedge (\mathsf{AGT}\ x\ a)$.

The symbol $\Diamond$ is an abbreviation for "eventally" as in $\Diamond\alpha \equiv \exists x(\mathsf{HAPPENS}\ x; \alpha?)$. The symbol $\Box$ is an abbreviation for "always" as in $\Box\alpha \equiv \neg\Diamond\neg\alpha$. The concept of "later" is defined as eventually but not currently. That is, $(\mathsf{LATER}\ p) \equiv \neg p \wedge \Diamond p$.

2. Assumptions

For their theory of intention, C&L adopt the assumption that agents are competent with respect to the primitive actions they have done: $\forall x, e\ (\mathsf{AGT}\ x\ e) \rightarrow [(\mathsf{DONE}\ e) \equiv (\mathsf{BEL}\ x\ (\mathsf{DONE}\ e))]$. Furthermore, they adopt the assumption that agents believe they shall realize the successful occurence of their actions. Specifically, C&L assume that "if an agent believes he is about to do e resulting in a world where α is true, then he also believes that after e, he will realize that α is true." (p 241) Formally,

$$\models \forall\ e, (\mathsf{BEL}\ x\ (\mathsf{HAPPENS}\ x\ e; \alpha?)) \rightarrow (\mathsf{BEL}\ x\ (\mathsf{HAPPENS}\ x\ e; (\mathsf{BEL}\ x\ \alpha?)).$$

Note that the assumption is silent on what an agent believes will happen if his action does not indeed occur.

Also, C&L make an assumption that appears tautological at first glance: "for each [primitive] event of which x is the agent, either he believes the next thing to happen is his causing the event, or he believes it is not the next thing to happen." (p 242) Formally,
$\models \forall e, (\mathsf{AGT}\ x\ e)\ \wedge\ (\mathsf{SINGLE}\ e)\ \rightarrow\ (\mathsf{OPINIONATED}\ \ x\ (\mathsf{HAPPENS}\ e))$.

C&L make the uncontentious claim that agents do not infinitely persist in trying to achieve their goals; neither do they infinitely procrastinate. C&L assume *No infinite persistence* to capture both these desiderata: $\models \Diamond\neg(\mathsf{GOAL}\ x\ (\mathsf{LATER}\ p))$. Singh argues that the assumption of *No infinite persistence* does not capture the notion of limited procrastination (*No infinite deferral*) and permits certain absurd scenarios.

3. Constraints

C&L place two reasonable constraints on the logic; the first of these is *Consistency* which states that B is Euclidean, transitive and serial, and G is serial. The second is *Realism*: $\forall \sigma, \sigma^*,\ $ if $\langle\sigma, n\rangle G[p]\sigma^*,$ then $\langle\sigma, n\rangle B[p]\sigma^*$. That is, $G \subseteq B$.

4. Definitions

Knowledge is naievely defined as true belief: $(\mathsf{KNOW}\ x\ p)\ \equiv\ p\ \wedge\ (\mathsf{BEL}\ x\ p)$. C&L define a notion of competency, which says that an agent's perception of a fact is correct, as $(\mathsf{COMPETENT}\ x\ p)\ \equiv\ (\mathsf{BEL}\ x\ p) \rightarrow (\mathsf{KNOW}\ x\ p)$. Also, an agent is opinionated toward a proposition if he believes it is true or false, as defined by $(\mathsf{OPINIONATED}\ x\ p)\ \equiv\ (\mathsf{BEL}\ x\ p)\ \vee\ (\mathsf{BEL}\ x\ \neg p)$.

To state that if a Wff q comes true, p comes true before it does, we may use the defininition $(\mathsf{BEFORE}\ p\ q) \equiv \forall c, (\mathsf{HAPPENS}\ c; q?) \rightarrow \exists a, (a \leq c) \wedge (\mathsf{HAPPENS}\ a; p?)$. We define an achievement goal as a goal not currently true, as distinguished from a maintenance goal. Formally,
$(\mathsf{A\text{-}GOAL}\ x\ p) \equiv (\mathsf{GOAL}\ x\ (\mathsf{LATER}\ p)) \wedge (\mathsf{BEL}\ x\ \neg p)$.

From the "atomic" pieces of this conative/doxastic/temporal logic, C&L build "molecular" constructs that describe rational agency. To capture the notion of commitment, C&L define a "persistent goal" as

$$\begin{aligned}(\mathsf{P\text{-}GOAL}\ x\ p) \equiv\ & (\mathsf{GOAL}\ x\ (\mathsf{LATER}\ p)) \wedge \\ & (\mathsf{BEL}\ x\ \neg p) \wedge \\ & [\mathsf{BEFORE}\ ((\mathsf{BEL}\ x\ p) \vee (\mathsf{BEL}\ x\ \Box\neg p)) \\ & \quad \neg(\mathsf{GOAL}\ x\ (\mathsf{LATER}\ p))]\end{aligned}$$

This definition forms the basis of C&L's definitions of intention, which are special kinds of commmitments. They define INTEND_1, intention toward an action, like so.

$$(\mathsf{INTEND}_1\ x\ a) \equiv (\mathsf{P\text{-}GOAL}\ x\ [\mathsf{DONE}\ x\ (\mathsf{BEL}\ x\ (\mathsf{HAPPENS}\ a))?; a])$$

where a is any action expression. That is, an intention toward an action is a commitment to have brought about that action immediately after having believed it was about to occur. They define INTEND_2, intention toward a proposition, like so:

$$\begin{aligned}& (\mathsf{INTEND}_2\ x\ p) \equiv \\ & \quad (\mathsf{P\text{-}GOAL}\ x\ \exists e, (\mathsf{DONE}\ x\ [(\mathsf{BEL}\ x\ \exists\ e', (\mathsf{HAPPENS}\ x\ e'; p?)) \wedge \\ & \quad \neg(\mathsf{GOAL}\ x\ \neg(\mathsf{HAPPENS}\ x\ e; p?))]?; e; p?))\end{aligned}$$

That is, an intention toward a proposition is a commitment that some plan e have brought about the proposition immediately after (1) having believed that there exists some event e' that shall bring about the proposition and (2) having

Table III. Axioms

1. $\models (\mathsf{HAPPENS}\ a; b) \equiv \mathsf{HAPPENS}\ a; (\mathsf{HAPPENS}\ b)?)$
2. $\models (\mathsf{HAPPENS}\ p?; q?) \equiv p \wedge q$
3. $\models \forall x, (\mathsf{BEL}\ x\ p) \wedge (\mathsf{BEL}\ x\ (p \rightarrow q)) \rightarrow (\mathsf{BEL}\ x\ q)$
4. $\models \forall x, (\mathsf{BEL}\ x\ p) \rightarrow (\mathsf{BEL}\ x\ (\mathsf{BEL}\ x\ p))$
5. $\models \forall x, \neg(\mathsf{BEL}\ x\ p) \rightarrow (\mathsf{BEL}\ x\ \neg(\mathsf{BEL}\ x\ p))$
6. $\models \forall x, (\mathsf{BEL}\ x\ p) \rightarrow \neg(\mathsf{BEL}\ x\ \neg p)$
7. *If* $\models \alpha,$ *then* $\models (\mathsf{BEL}\ x\ \alpha)$
8. $\models \forall x, (\mathsf{GOAL}\ x\ p) \wedge (\mathsf{GOAL}\ x\ (p \rightarrow q)) \rightarrow (\mathsf{GOAL}\ x\ q)$
9. $\models \forall x, (\mathsf{GOAL}\ x\ p) \rightarrow \neg(\mathsf{GOAL}\ x\ \neg p)$
10. *If* $\models \alpha,$ *then* $\models (\mathsf{GOAL}\ x\ \alpha)$

accepted that the particular plan e may bring about the proposition.

D. Axioms and Propositions

C&L adopt the axioms of table III for their formalism. Note that some of these can be derived from the constraints, but are given for clarity.

From the axioms, assumptions, and constraints C&L give proof for the propositions of table IV, except for proposition 12 which is true due to the *Realism* constraint.

E. Analysis of the P-GOAL

The definition of having a commitment, that is a P-GOAL, is not trivial for an agent to meet: Two requirements are placed on the agent's mental state and future mental states he may adopt. First, the agent has the achievement goal toward the object of commitment. Second, according to the BEFORE clause,

Table IV. Propositions

1. $\models (\mathsf{HAPPENS}\ a) \;\equiv\; (\mathsf{HAPPENS}\ a; (\mathsf{DONE}\ a)?)$

2. $\models (\mathsf{DONE}\ a) \;\equiv\; (\mathsf{DONE}\ \ (\mathsf{DONE}\ (\mathsf{HAPPENS}\ a)?; a))$

3. $\models p \equiv (\mathsf{DONE}\ p?)$

4. $\models p \rightarrow \Diamond p$

5. $\models \Diamond(p \vee q) \;\wedge\; \Box\neg q \;\rightarrow\; \Diamond p$

6. $\models \Box(p \rightarrow q) \;\wedge\; \Diamond p \;\rightarrow\; \Diamond q$

7. $\models \neg(\mathsf{LATER}\ \Diamond p)$

8. $\models \Diamond q \;\wedge\; (\mathsf{BEFORE}\ p\ q) \rightarrow \Diamond p$

9. $\models \neg p \rightarrow (\mathsf{BEFORE}\ (\exists e, (\mathsf{DONE}\ \neg p?; e; p?))\ p)$

10. $\mathit{If}\ \ \models \alpha, \mathit{then}\ \ \models (\mathsf{BEL}\ x\ \Box\alpha)$

11. $\models (\mathsf{BEL}\ x\ p) \rightarrow (\mathsf{GOAL}\ x\ p)$

12. $\models (\mathsf{ACCEPT}\ x\ p) \rightarrow (\mathsf{SUSPECT}\ x\ p)$

13. $\models\ \forall x, e\ (\mathsf{BEL}\ x\ (\mathsf{HAPPENS}\ x\ e)) \rightarrow (\mathsf{GOAL}\ x\ (\mathsf{HAPPENS}\ x\ e))$

14. $\models\ (\mathsf{GOAL}\ x\ p) \;\wedge\; (\mathsf{BEL}\ (p \rightarrow q)) \rightarrow (\mathsf{GOAL}\ x\ q)$

15. $\models (\mathsf{BEL}\ x\ \exists e \neq \mathsf{NIL}\ (\mathsf{HAPPENS}\ x\ e))$

$\rightarrow\ \exists\ e', (\mathsf{SINGLE}\ e') \wedge (\mathsf{BEL}\ x\ (\mathsf{HAPPENS}\ x\ e'))$

if the agent ever drops the achievement goal, he must first believe it achieved or impossible. The agent will obviously be competent about such achievement goals. However, the agent need not be aware of BEFORE clause. Therefore, the agent may misapprehend its own commitments. Theoretically, the commitments would still prove useful in motivating the agent's action, whether or not they were accurately represented in the agent's belief structure.

Though sometimes unwieldy, the P-GOAL offers important theoretical properties. It solves the side-effect problem in many cases, though not perfectly as C&L admit. Francesco [24] provides further analysis on the side-effect problem in C&L's theory.

The P-GOAL has appropriately weak logic, given in table V.

Table V. The Logic of P-GOAL

Conjunction, Disjunction, and negation

$$\not\models \ (\textsf{P-GOAL}\ x\ (p \wedge q)) \rightarrow (\textsf{P-GOAL}\ x\ p)\ \wedge\ (\textsf{P-GOAL}\ x\ q)$$

$$\not\models \ (\textsf{P-GOAL}\ x\ (p \wedge q)) \leftarrow (\textsf{P-GOAL}\ x\ p)\ \wedge\ (\textsf{P-GOAL}\ x\ q)$$

$$\not\models \ (\textsf{P-GOAL}\ x\ (p \vee q)) \rightarrow (\textsf{P-GOAL}\ x\ p)\ \vee\ (\textsf{P-GOAL}\ x\ q)$$

$$\not\models \ (\textsf{P-GOAL}\ x\ (p \vee q)) \leftarrow (\textsf{P-GOAL}\ x\ p)\ \vee\ (\textsf{P-GOAL}\ x\ q)$$

$$\models (\textsf{P-GOAL}\ x\ \neg p)\ \rightarrow\ \neg(\textsf{P-GOAL}\ x\ p)$$

No consequential closure of P-GOAL

$$\not\models \ ((\textsf{P-GOAL}\ x\ p)\ \wedge\ (p \rightarrow q))\ \rightarrow\ (\textsf{P-GOAL}\ x\ q)$$

$$\not\models \ [(\textsf{P-GOAL}\ x\ p)\ \wedge\ (\textsf{BEL}\ x\ (p \rightarrow q))]\ \rightarrow\ (\textsf{P-GOAL}\ x\ q)$$

$$\not\models \ [(\textsf{P-GOAL}\ x\ p)\ \wedge\ (\textsf{BEL}\ x\ \Box(p \rightarrow q))]\ \rightarrow\ (\textsf{P-GOAL}\ x\ q)$$

$$\not\models \ [(\textsf{P-GOAL}\ x\ p)\ \wedge\ \Box(\textsf{BEL}\ x\ \Box(p \rightarrow q))]\ \rightarrow\ (\textsf{P-GOAL}\ x\ q)$$

The entailment $\models\ (p \rightarrow q)$ *is compatible with*

$$(\textsf{P-GOAL}\ x\ p)\ \wedge\ \neg(\textsf{P-GOAL}\ x\ q)$$

If $\models\ (p \equiv q)$ *then* $\models\ (\textsf{P-GOAL}\ x\ p)\ \rightarrow\ (\textsf{P-GOAL}\ x\ q)$

CHAPTER III

THE CRITICISM DUE TO SINGH

A. Persistence Is Not Enough

Using their constructs, C&L prove a powerful theorem, called *From persistence to eventualities*, which states that "If someone has a persistent goal of bringing about p, p is within his area of competence, and, before dropping his goal, the agent will not believe p will never occur, then eventually p becomes true."(p 239) Formally,

$$\models ((\mathsf{P\text{-}GOAL}\ y\ p)\ \wedge\ \Box(\mathsf{COMPETENT}\ y\ p)\wedge$$
$$\neg[\mathsf{BEFORE}\ \ (\mathsf{BEL}\ y\ \Box\neg p)\ \neg(\mathsf{GOAL}\ y\ (\mathsf{LATER}\ p))]) \rightarrow \qquad \Diamond p$$

On the surface, this theorem seems pleasing. However, Singh describes a scenario that counters this intuition: "For example, let me be the agent and let *p* by my favorite implausible proposition: that Helmut Kohl is on top of Mt Everest. I can easily (1) have this P-GOAL, (2) for eternity not hold the belief that Herr Kohl will not ever make it to the top of Mt Everest, and (3) be always COMPETENT about *p*. Therefore, by the above theorem, Herr Kohl will get to the top of Mt Everest. He does not need to try; nor do I. He does not even need to know that his mountaineering feat had been my persistent goal."(sec. 3)

Therefore, according to Singh, the *From persistence to eventualities* theorem relates inadequate requirements on an agent to non-trivial requirements on the world. Singh indicates that the theory does not adequately address agents' ability to achieve their goals; he points out two aspects of the theory that fail in this respect. First he implicates the improper formalization of *No infinite persistence* as a culprit in making the theorem too powerful because it does not properly prohibit infinite deferral (procrastination). Second, Singh indicates that the theory includes no assumption of "fairness," an assumption whereby if an agent repeatedly attempts an action then it will eventually succeed. The modified theory drops the original assumption of *No infinite persistence* in favor of a more limited set of assumptions.

B. An Unexpected Property of INTEND_1 with Repeated Events

Singh's next argument (sec 4.1) is a counterexample to C&L's claim (p 247) that "an agent who intends $a;b$ also intends to do a." In this counterexample, the model's set of possible worlds contains σ and σ'. At time n, σ' is the only belief accessible world and the only goal accessible world from σ. Also, σ' is the only belief accessible world and the only goal accessible world from itself. This is compatible with having (DONE a) at time n in world σ', in which case the agent would be aware of having done a. Since having a $\mathsf{P\text{-}GOAL}$ toward a proposition means believing the propositon is currently false, the agent would be unable to have a $\mathsf{P\text{-}GOAL}$ toward having done a, and therefore would be unable to have an intention toward a.

The problem arises from the fact that intention involves a persistent goal which is a type of achievement goal (the object of which must be believed currently false). Yet we want an agent who intends $a;b$ to intend a as well, regardless of a's prior occurence. Why would an agent commit to bringing about what has just happened?

Suppose that our agent is a farmer, each time increment is a planting season, and as his action the farmer may elect to produce various crops, represented by actions including $a =$ *To raise alfalfa* and $b =$ *To raise beans.* In each season each crop would have an associated yield or utility which may not be known to the agent. In this scenario, it is clear why the agent could intend $a; b$ when a just took place: the agent would anticipate that alfalfa would produce the greatest utility, even though he planted it last season. The farmer does not engage in planting for its own sake, but rather for expected yields, from which he derives utility. So in a case like this, where actions are somewhat independent and can be reasonably conducted out of sequence, it is necessary that our definition of intention toward action should involve the outcome of the action.

Consider another case where actions are more closely related. Suppose an agent wishes to ascend a steep cliff which just barely within reach. In this simple story there are the primitive events of $a =$ *To jump with arms extended upward* and $b =$ *To grab ahold of the cliff.* To carry out b constitutes success in this story. One can concieve of many such stories; the point is that the actions must be performed in a specific sequence. At time $n - 1$, the agent who intends at this time to carry out $a; b$, performs a with the expectation that ($\mathsf{HAPPENS}$ $a; b$). On this account, the agent is incorrect, for his jump ends at time n with him standing once again on the ground rather than in midair in position to grasp the ledge. Whereupon he attempts the intended jump again at time n, with the original intention intact. Here, our definition of intention should address the issue of sequencing of actions. It may be the case that a was just done, but an achievement goal toward a is still possible if it stipulates that a bring about any condition that it did not bring about last time.

At this point, the incisive reader may wonder what kind of structure is enforced on E the set of primitive event types. In particular, agents would never intend what has already occured if the event types described actions occuring at specific points in time, i.e. if to perform an action at noon and 12:01 are two different event types. Fortunately, this is an unnecessary constraint, because E can be of arbitrary granularity. That is, E may include only one event type for each agent or arbitrarily many.

Since we allow for the repetition of events of the same event type in sequence, the only way characterize a commitment toward what has already been done with an intention (or any achievement goal) is to incorporate the reasons for the action, that is the expected outcome of the action. Otherwise, the requirement that the object of an achievement goal must be currently believed false will derail any repetitive intentions (recall that persistent goals and therefore intentions are a particular type of achievement goal). Of course, incorporating the reasons for an action into persistent goals means agents must have some conception of causality in order to be reasonable about what commitments they adopt. As this is a theory of intention, not a theory of everything, we must remain silent on the nature of causality. According to the semantics, there are no causal links between actions and predicates, only different interpretations of predicates on different time lines at different times. Nevertheless, a modeler creating a set of possible worlds and specifying Φ would intuitively consider the various effects of the events in E under different circumstances. Furthermore, any agent architecture capable of basic planning includes domain knowledge involving the results of actions. Therfore, it seems not unreasonable to stipulate that agents intend actions for a reason.

It may seem that we have sidestepped the issue of repeated actions only to introduce a problem with repeated outcomes. For our problem is not solved if we need agents to intend the same action and the same outcome consecutively. However, agents performing consecutive actions need not encounter this problem in practice. Indeed, to require that the successive Wffs are different encourages the creation of more robust agent architectures. Suppose an agent acts by means of conditionally executing recipes, i.e. it uses the most basic planning. Then in the context of a plan, an intended, successfully achieved Wff that results from an intended action that happens at time n can always be different from an intended, successfully achieved Wff that results from an intended action at time $n + 1$. Some examples illustrate this point. Consider the case of the two stories above. The farming agent tries to bring about some new condition that is not true yet when he iteratively (and intentionally) performs action a. The cliff ascending agent also tries to bring about something not yet the case as he performs action a twice in succession. The agents in these cases are not necessarily engaging in lengthy plans, and can intend things on a moment by moment basis. But more complex agents exhibit this feature. Consider the case of an agent iteratively chopping down a tree. One can imagine that in an implementation of the tree cutting agent, a top level intention to cut down the tree would, through whatever planning apparatus and domain knowledge, motivate individual intentions to chop the tree. Until his task is completed, in each timestep the agent intends to perform an action, say *chop*, with the outcome that the tree's girth be diminished, bringing it closer to falling than it was before. In such a situation, the Wff representing this outcome would upon the first chop at time n take a form like $WoodThickness_{n+1} = WoodThickness_n - x$, where the subscript indicates the time at which the variable is evaluated and x is the agent's impression of how much damage he can do with an axe. This indicates that the thickness of the tree should diminish with the chop. But in timestep $n + 1$, the outcome of the chop is subtly different, that $WoodThickness_{n+2} = WoodThickness_{n+1} - x$. Thus, we can see that repeated actions in the context of plans typically allow distinct, accumulative outcomes.

Now, it can be argued that if an agent's behavior is purely reactive (in the sense that it merely responds directly to the world and do not reason about the world), then the agent could reasonably intend to bring about exactly the same Wff repeatedly. Fortunately, we need not be concerned with this limitation because purely reactive agents have no use for logics of intention in their specifications.

C. An Unexpected Property of INTEND_1 and Multiple Intentions

Singh's next argument involves an agent having multiple intentions. Singh demonstrates that C&L's theory is restrictive in that it does not allow for agents to maintain multiple intentions simultaneously. This is because under C&L's definition, an agent is commited to bring about an intended action *immediately* after believing it was about to happen in its entirety. Singh demonstrates the problem in a brief story (sec 4.2). "As a natural example, imagine an agent who runs a cafeteria. He takes orders from his customers, forms the appropriate intentions, and acts on them. When asked to serve coffee, he forms an intention to do the following complex action: pick up a cup; pour coffee into it; take the cup to the table. When asked to serve tea, he forms an intention to do the corresponding action for tea. Suppose now that two orders are placed: one for tea and the other for coffee. The agent adopts two intentions as described above. The agent initially ought to pick up a cup; let us assume that this is the action he chooses, and the one he believes he is about to do. However, at the time the agent picks up a cup, he might not have decided what action he will do after that, i.e., whether he will pour coffee or pour tea into the cup. Indeed, whether he pours coffee or tea into the cup might depend on other factors, e.g., which of the two brews is prepared, or whether other agents are blocking the route to one of the pots."

He goes on to say that "While this is a fairly ordinary state of affairs and a natural way for an agent to operate, it is disallowed by the theory. This is because the theory requires beforehand that he is going to do the given action, no matter how complex it is (and then do it). In the present example this is not the case: the agent knows what he is doing before each subaction, but does not have a belief about a complex action before beginning to execute it. Also, for the agent to even *have* an intention, the theory requires that he have a P-GOAL to satisfy it in the above sense."

Here, let $a =$ *To pick up a cup*, $b_c =$ *To pour coffee in a cup*, $b_t =$ *To pour tea in a cup*, and $c =$ *To bring a cup to the table.* In Singh's example, prior to picking up a cup, the agent does not believe either complex action is about to happen. Formally, we have $\neg(\mathsf{BEL}\ x\ (\mathsf{HAPPENS}\ x\ a; b; c))$ where b is either b_c or b_t.

Therefore, as Singh argues, the agent x who first picks up the cup will not presently be able to succeed in

$[\mathsf{P\text{-}GOAL}\ x\ (\mathsf{DONE}\ x\ (\mathsf{BEL}\ x\ (\mathsf{HAPPENS}\ x\ a; b; c))?; a; b; c)]$

at time $n+3$ upon completion of the complex action of which picking up the cup is the first step, and he will not fulfill this P-GOAL as a result of not having held the above belief.

Thus, the agent will intentionally do neither the complex action $a; b_c; c$ nor $a; b_t; c$ since the object of his persistent goal will not have been satisfied at time $n + 3$ when, assuming all goes well, he finishes one of the complex actions by serving either coffee or tea.

However, Singh's story about the beverage serving agent does not preclude that the agent anticipate having *later* done the complex action right after believing it was about to happen. Formally, as the agent picks up the cup we could have

[$\mathsf{GOAL}\ x\ (\mathsf{LATER}\ \ (\mathsf{HAPPENS}\ x\ a;b;c))$]. Thus, given the definition of $\mathsf{P\text{-}GOAL}$, which stipulates that the agent choose that the proposition be brought about *later*, the agent could still be committed in the $\mathsf{P\text{-}GOAL}$ sense to performing these complex actions, and successfully fulfill such commitments.

Singh is right to say that when the agent recieves the two orders, he should form intentions toward both serving tea and serving coffee. These are not the strongest of intentions, however. In the above scenario, the agent clearly has not settled on precise plans to serve tea nor to serve coffee, as he inhabits an unpredictable environment that threatens to thwart either intention at every turn. The agent does however resolve to bring about a state of affairs for each intention, in which the beverage is dispensed and presented to the restaurant patron.

In the situation that Singh describes, the agent has at its disposal of a set of predefined sequence of events expected to bring about either intention. Such a sequence, which is known in advance but not currently planned to be conducted, could be called a recipe. This terminology corresponds to that in the SharedPlans framework of Grosz and Kraus [30]. If the agent had a specific plan we would definitely say in a strong sense that he intended to serve a beverage. But here, the agent has no plan, only recipes, and yet still should intend to serve both beverages. This demonstrates the ambiguity of different senses of the word “intention.” To capture these different senses a theory of intention should define a notion of a weak intention, whereby the agent is committed, but does not forsee the exact course of events that brings about his objective.

CHAPTER IV

NEW NOTIONS OF INTENTION

Three problems have been identified with the original theory. (1) The *From persistence to eventualities* theorem, which links persistent goals with eventual outcomes, is too powerful. To solve this problem most easily, one can drop the original assumption of *From Persistence to eventualities*, which is the crux of the proof of the undesirable theorem. (2) When an agent executes two actions of the same event type successfully two consecutive times, it is impossible that the second action be intentional. In regards to this problem, recall that INTEND_1, intention toward action, is a persistent goal that some action have been done under certain circumstances. Thus, an agent can not have the persistent goal (much less the intention) to have done what he has just done. One may solve this problem by requiring that an agent's intention toward action involve some propositional outcome of the action. In this way an agent may intentionally perform the same action twice consecutively as long as the intended outcome of the second action is different from the actual outcome of the first action. Intuitively, the agent should expect his action to cause some desired condition to become true. (3) Agents are unable to maintain multiple intentions simultaneously. The third problem makes clear the need for a definition of intention that corresponds to the sense of the word "intention" which does not have the connotation that the agent knows every detail of his plan to fulfill the intention. Only in the strongest, most bona-fide sense of intention (present directed intention) must the agent know in advance the action sequence supposed to bring about the intention.

New definitions of intention were therefore devised based on these criteria. The definitions were developed through gradual modification of the basic $\mathsf{P\text{-}GOAL}$

construct, progressively adding or removing operators and checking each new configuration against the desiderata for intention. We have here a presentation of the results themselves, not the thought processes that gave rise to them. Nevertheless, a basic account of the rationale for all these changes can be gleaned from the arguments of the previous section.

A. A Refined Notion of Commitment

The criticism regarding repeated actions suggests that intention toward action should entail commitment to a particular outcome. One could naively indicate this like so:

$$(\mathsf{WEAK\text{-}INTEND}_1\ x\ a)\ \equiv\ \exists p,$$
$$(\mathsf{P\text{-}GOAL}\ \ x\ (\mathsf{DONE}\ x\ a; p?))$$

However, the syntax prohibits quantification over formulas like p.

We may introduce another set of entites into the universe of discourse as surrogates for formulas. We define $J \subseteq \wp(T)$ corresponding to a pre-defined set of possible expected outcomes defined as sets of possible worlds. J, then, consists of sets of possible worlds (timelines). We introduce $\langle JustificationVariable \rangle$s whose denotations are elements of J. An element $j \in J$ may be used as T in constructing a model. Intuitively, J specifies sets of "desired" formulas, namely those compatible with its elements. We introduce semantics of a test action for expectations instead of formulas. The "¿" test action is analogous to the "?" test action, except it succeeds when the current world is an element of the "justification", or possible-worlds-set specification of the desired end state.

In the definition of a model, U therefore, is redefined to include J, the set of justifications. D, which is necessary for the interpretation of predicates by Φ, is redefined as $D = \Theta \cup P \cup E^* \cup J$.

Table VI. Revised Syntax

$\langle\textbf{JustificationVariable}\rangle$ ::= $s, s_1, s_2, \ldots,\ t, t_1, t_2, \ldots$

$\langle\textbf{Variable}\rangle$ ::= $\langle$JustificationVariable$\rangle$ | …

$\langle\textbf{Wff}\rangle$ ::= $\langle$JustificationVariable$\rangle$¿ | …

The syntax of C&L's theory needs no changes except the introduction of the "¿" test action, and variables whose denotations are elements of J. These changes are expressed in table VI. The semantics of the new operator are given by

$$M, \sigma, v, n \| s¿ \| n \;\Leftrightarrow\; v(s) = j \text{ and } \sigma \in j \in J \subseteq \wp(T).$$

The weak sense of intention is captured using personal commitments (persistent goals) that adhere the requirements discussed.

$$(\mathsf{WEAK\text{-}INTEND}_1\ x\ a) \;\equiv\; \exists s,$$
$$(\mathsf{P\text{-}GOAL}\ \ x\ (\mathsf{DONE}\ x\ a; s¿))$$

$$(\mathsf{WEAK\text{-}INTEND}_2\ x\ p) \;\equiv$$
$$(\mathsf{P\text{-}GOAL}\ \ x\ \exists a, (\mathsf{DONE}\ x\ a; p?))$$

Thus, an agent weakly intends that which he is committed to bringing about himself, regardless of having a plan to do so. However, in the case of intention toward action, the agent will indeed have a plan, this being the action itself. In the case of intention toward action, the agent also believes the action will have a particular result, to enable repetition of equivalent event types as discussed above. Under this definition, Singh's cafeteria-agent could say it was serving both coffee and tea intentionally.

B. Intention Is Commitment and Expectation

In the stronger sense of the word, one has a plan to carry out what one intends. This stronger notion of intention is applicable to both intention toward propositions INTEND_2 and intention toward actions INTEND_1. Therefore, the new definition of intention consists of a weak intention and a what the agent thinks is a sure-fire plan to presently bring it about.

$$\begin{aligned}
&(\mathsf{INTEND}'_1\ x\ a) \ \equiv\ \exists s,\\
&\quad(\mathsf{P\text{-}GOAL}\ \ x\ (\mathsf{DONE}\ x\ a; s\text{¿}))\\
&\quad\wedge\ (\mathsf{BEL}\ x\ (\mathsf{HAPPENS}\ x\ a; s\text{¿}))
\end{aligned}$$

$$\begin{aligned}
&(\mathsf{INTEND}'_2\ x\ p) \ \equiv\ \exists e,\\
&\quad(\mathsf{WEAK\text{-}INTEND}_2\ x\ p)\\
&\quad\wedge\ (\mathsf{BEL}\ x\ (\mathsf{HAPPENS}\ x\ e; p?))
\end{aligned}$$

In this very strong sense of intention, the agent does not forsee (believe in) any futures under which his intention does not come true. He believes that other outcomes are ruled out by the inevitability of his present action. However, the agent may freely change his mind about this fact as conditions develop (his belief is not restrained by any BEFORE clause), and thus readily update his intentions, which would maintain a degree of consistency due to the agent's commitment. Note that in the above story Singh's restaurant agent would likely form a strong intention toward the serving of a beverage at time $n + 1$, when he has to choose action b_t or b_c.

These definitions resemble Searle's analysis [55], under which intention entails a *prior intention* (here represented by commitment), and an *intention in action* (here a belief by the agent that it performs an action). In an agent architecture based on this theory, the *intention in action* would cause and be contemporaneous with the (current) action itself.

C. From Intention to Eventualities

As examined earlier, the *No infinite persistence* assumption of C&L produces undesirable results when combined with the definition of $\mathsf{P\text{-}GOAL}$. Since we retain the definition of $\mathsf{P\text{-}GOAL}$, we must drop the assumption. It could be replaced by an assumption on an agent's intentions. Therefore we redefine *No infinite persistence* as

$$\models \ \Diamond\neg(\mathsf{WEAK\text{-}INTEND}_2\ x\ p).$$

Notice that this redefinition alone may sanction theorems similar to *From persistence to eventualities.* However, the problem is eliminated by ensuring that agents to not perpetually procrastinate. Therefore, we also adopt the assumption of *No infinite deferral*, defined as

$$\begin{aligned} \models\ & (\mathsf{WEAK\text{-}INTEND}_2\ x\ p) \wedge \\ & \quad \neg[\mathsf{BEFORE}\ (\mathsf{BEL}\ x\ \Box\neg p)\ \neg(\mathsf{GOAL}\ x\ (\mathsf{LATER}\ p))] \\ & \to\ \Diamond(\mathsf{INTEND}'_2\ x\ p) \end{aligned}$$

This definition allows for the case where the agent realizes his weak intention has been derailed by uncontrollable events in the world, such as the intervention of other agents.

Under these assumptions, as long as an agent is competent with respect to his belief that he is carrying out the intention, then the intention must come true. That is, if the agent x is $(\mathsf{COMPETENT} x\ (\mathsf{HAPPENS}\ x\ e; p?))$ with e and p as in the definition of INTEND_2, then the strong intention's plan will succeed. One could define a notion of capability whereby the conditions are satisfied for *No infinite deferral*, and the agent is competent with respect to the belief component of the strong intention that comes about.

CHAPTER V

MEETING THE DESIDERATA FOR INTENTION

The revised assumptions and definitions complete the modifications to the theory needed to address all of the formal criticisms of Singh. One can show that the many desiderata given in section 7 of C&L still hold.

Intentions normally pose problems for the agent; the agent needs to determine a way to achieve them: In the case of intention toward action, the agent already plans to achieve its goals by performing the intended action. In the case of intention toward propositions, by the assumption of *No infinite deferral* and the new definition of INTEND_2 clearly an agent will try to come up with a plan once he is commits to doing something.

Intentions provide a "screen of admissibility" for adopting other intentions: If an agent intends b, as in (INTEND_1 b), and always believes that doing a forever prevents doing b, as in $\Box(\mathsf{BEL}\ (\mathsf{HAPPENS}\ a) \rightarrow \Box\neg(\mathsf{HAPPENS}\ b))$, then the agent cannot intend to do a before b in any sequence of actions. Suppose for contradiction that the agent did $(\mathsf{INTEND}'_1\ x\ a; b)$. Then the agent would believe a would occur, forever preventing b. But the agent would also believe b would occur, a contradiction. Therefore, we may formally say

$$\begin{aligned} \models\ & \forall x\ (\mathsf{INTEND}'_1\ x\ b) \\ & \quad \wedge \Box(\mathsf{BEL}\ x\ [(\mathsf{DONE}\ x\ a)\ \rightarrow\ \Box\neg(\mathsf{DONE}\ x\ b)])\ \rightarrow \\ & \quad \neg(\mathsf{INTEND}'_1\ x\ a; b) \end{aligned}$$

Agents "track" the success of their attempts to achieve intentions: Agents maintain their intentions after failure, contingent upon being able to come up with a plan. Suppose it is the case that

$$(\mathsf{DONE}\ x[(\mathsf{INTEND}_1'\ x\ a)\ \wedge\ (\mathsf{BEL}\ x\ (\mathsf{HAPPENS}\ x\ a;p?))]?;e;\neg p?),$$

that is the agent's intended action did not occur just now when he thought it would. Further suppose that

$$(\mathsf{BEL}\ x\ \neg(\mathsf{DONE}\ x\ a;p?))\ \wedge\ \neg(\mathsf{BEL}\ x\ \Box\neg(\mathsf{DONE}\ x\ a;p?)),$$

which means the agent is aware the intention did not succeed, but still believes it possible to succeed. By the BEFORE clause, its is obvious that the P-GOAL or weak intention component of the strong intention conjunct remains true. The agent at this point will be disposed to form a new belief about what action he should take; presumably, he would like to try again. In order to try again, he must resolve to do so; the agent would adopt $(\mathsf{BEL}\ x\ (\mathsf{HAPPENS}\ x\ e;p?))$. As we would expect, successful maintenance maintenance of strong intentions therefore depends on agents' ability to maintain plans. This case meets all the conditions for INTEND_1'.

If the agent intends action a, then the agent believes it is possible to do action a: The theory has no modal operator for possibility. However, by definition an agent cannot adopt intentions he considers impossible. Since the agent must adopt a goal, the agent will suspect the intention will succeed (see Axiom 12).

If the agent intends action a, sometimes the agent believes he will in fact do a: Under from the definition of $\mathsf{WEAK\text{-}INTEND}_1$, the agent has an persistent goal and therefore an achievement goal toward having done the action: $(\mathsf{A\text{-}GOAL}\ (\mathsf{DONE}\ a))$; hence the agent will have the goal that the intended act comes true later (by definition of A-GOAL) meaning $(\mathsf{GOAL}\ (\mathsf{LATER}\ (\mathsf{DONE}\ a)))$. Recalling that by definition goals entail acceptance, and applying axiom 12, we may conclude that (SUSPECT (LATER (DONE a))), meaning that the agent believes it possible that a occurs. This is not quite so strong as belief, but the agent has not made definitive plans to execute a.

On the other hand, in the case of strong intention under INTEND_1', quite simply the agent believes he is doing the intended act.

If the agent intends action a, the agent does not believe he will never do a: This follows from the *Realism* constraint.

Agents need not intend all the expected side-effects of their intentions: This follows from the lack of consequential closure of the P-GOAL, as discussed in the analysis of the P-GOAL.

Dropping futile intentions: The final theorem given by C&L, which appears at the end of section 7, states that "if an agent believes anyone else is truly going to achieve p, then either the agent does not intend to achieve p himself, or he does not believe p can be achieved only once. Contrapositively, if an agent intends to achieve p, and always believes p can be achieved only once, the agent cannot simultaneously believe someone else is going to achieve p"

$$\begin{aligned} \models\ \forall x,\ (y \neq x) \wedge (\mathsf{BEL}\ x\ \Diamond\exists e(\mathsf{DONE}\ y\ \neg p?; e; p?))\ \rightarrow \\ \neg(\mathsf{INTEND}_2\ x\ p) \wedge \\ \neg(\mathsf{BEL}\ x\ [\exists e\ (\mathsf{DONE}\ y\ \neg p?; e; p?)\ \rightarrow\ \Box\neg\exists e\ (\mathsf{DONE}\ x\ \neg p?; e; p?)]) \end{aligned}$$

This holds for the new definition as well. If the agent x were to believe y would achieve p, and intended to achieve p himself as well, then he would not believe that p could be achieved only once. If instead the agent x believed that y would achieve p and p could be achieved only once, then x could never adopt an intention to achieve p due to the fact that the requisite achievement goal could not hold: in all of x's belief accessible worlds, p is achieved only once, by y.

The Case Due to Chisholm: C&L rightly observe that their definition of INTEND_2 handles the following case, due to Chisholm [7], paraphrased by C&L (page 248): "An agent intends to kill his uncle. On the way to his uncle's house, this intention causes him to become so agitated that he loses control of his car, and runs over a pedestrian, who happens to be his uncle. Although the uncle is dead, we would surely say that the action that the agent did was not what was intended." Under the definition of $\mathsf{WEAK\text{-}INTEND}_2$, which we characterize as personal commitment, the agent would have fulfilled his intention in this case. Thus, agents are able to fulfill their commitments (weak intentions) accidentally. However, the accident will not occur in any of the agent's belief accessible worlds, so the agent could not have a strong intention (or even a goal) toward the occurrence.

Finally, note that the criticisims of Singh have been addressed. As discussed, agents' commitments will never be brought about inappropriately. Furthermore, agents may intend actions of the same type repeatedly by expecting their actions to have outcomes. Finally, agents may maintain multiple weak intentions and use these to form whatever strong intention is appropriate at a particular time, thus allowing agents to act opportunistically and interleave execution of different recipes.

CHAPTER VI

RAMIFICATIONS FOR SYSTEM ARCHITECTURES

This theory should be considered a high level specification to suggest design patterns for artificial agents. It should not be seen as a logic that agents should use for reasoning about mental states. It certainly should not be seen as a logic describing the mental states of human agents. In the case of artificial agents, however, it places a number of useful restrictions on architectures we may implement. Clearly, agents must have a particular mental state corresponding to GOAL and a particular mental state corresponding to BEL, with relationships as discussed. More importantly, agents should commit to certain goals, with the results that these goals persist through time. Also, Singh's criticism has revealed that agents should not intend actions for their own sake, but rather to have some effect. This should come as no surprise to those who study planning and means-ends reasoning.

SOAR [42] offers built-in belief maintenance capabilities for agents and other features that distinguish it from other rule based systems. STEAM essentially contains a SOAR implementation of joint-intentions, which we have not discussed here. Nevertheless, STEAM gives us precedent for implementing a C&L-style logic in a forward chaining system. However, in our logic, belief has a specific deontic meaning we want to encode, eschewing SOAR's automatic belief maintenance (STEAM involves no such concern: its rules involve inferences primarily about high level constructs like joint-intentions, not low level constructs like belief). Since any other forward chaining inference engine would do, Jess was chosen due to its ease of interface with Java and the efficiency of the rete match process. The agent developed serves as a proof of concept for agents based on the theory of intention as a persistent goal, and demonstrates the utility of principles embodied in the theory.

A. An Experimental Agent Implementation in *Jess*

In the Jess implementation presented, the agent and its environment are encoded by a set of rules that are used to infer facts from an initial fact base. These facts represent Wffs or action expressions in the logic. The rules represent axiomatic or theroem based inferences about the facts that affect the world or the agent. Integer cost values are attached to each fact to control the production system's infinite explosion of inferences. Thus, a crude solution to the logical omniscience problem is trivial in practice.

The basic data structures of the implementation include the unordered facts `actionExpression`, used to represent events, and the unordered fact `Wff` which represents formulas or propositions. A set of such facts can describe the states of the world and state transitions (events). For now, only one agent and one possible world are modeled. An `actionExpression` may represent a single primitive event, a sequence (denoted by the semicolon) which points to two other `actionExpressions`, or a test action (denoted by the question mark). A `Wff` may represent any formula expressible in the logic. Each fact encodes a single operator, and operators may be nested by having such facts refer to other facts as their subjects. The `Wffs` or `actionExpressions` that appear thusly "inside" another fact have the special time `abstract`. In contrast, an `actionExpression` or `Wff` that directly describes the world has an integer time value. The specifications for these unordered facts are given in figure 1.

To demonstrate the relation between planning and intention advocated in this paper, unordered facts representing "recipes" consisting of single or multiple actions enable means-ends reasoning on the part of the agent. These unordered facts are used to compose plans or courses of actions when motivated by a P-GOAL, and determine what event (action) the agent believes will happen next. As argued, believing one's own action is happening entails that one is attempting to indeed do it.

Code discussed by figure appears in Appendix A. The full code of the extended Jess scenarios and agent implementation appears in Appendix B. The complete code of the scenarios and the companion parser are availible as an online appendix at http://students.cs.tamu.edu/jsc6064/projects/appendix.

We now turn to an account of how persistent goals and recipes may motivate the formation of intentions. If the agent has a persistent goal, a recipe whose postcondition fulfills the goal, and whose preconditions are met, then unless the agent is involved in the execution of another plan, the agent will begin executing that recipe. The code to handle the simplest case applying single action recipes is given in figure 2. The multi-action case requires recipes to be constructed by chaining single actions together by matching preconditions to postconditions in a manner similar to STRIPS.

Scenarios suitable for initial encodings include McDermott's [46] story in which Little Nell is saved (or not) by Dudley (C&L p 219), the story about the beer delivering robot (C&L p 213), Chisholm's case, and Singh's restaurant agent. Let us consider the encoding of the world for the "Little Nell" story, given in figure 6.

The Little Nell story demonstrates a problem with naively designed planning systems. In this story, the heroine, called Nell, has been villainously tied to the railroad tracks with a train approaching. Believing that Nell will be mashed, Dudley accordingly plans to save Nell, with the result that he comes to believe that Nell will no longer be mashed. Since Nell will no longer be mashed, he drops the intention, but that means that Nell is going to be mashed again, causing him to again intend to save her, and so on. If one models intention as a persistent goal, this problem does not arise, since a persistent goal can be dropped only under certain circumstances (in particular, it will not be dropped just because it is expected to come true).

In the case of this story, the agent is able to form a strong intention toward saving Nell, which is motivated by his recipe to save Nell and his persistent goal to save Nell. In this instance, his efforts succeed. While Jess produces inferences based on the facts, we recieve word that the agent determines what to do: `Intention toward action save_Nell formed.` Typing `(facts)` at the `Jess>` prompt after this yields this fact (among many others):

```
(MAIN::Wff (mode INTEND1) (subj save_Nell) (subj2 nil)
    (time 0) (cost 10) (aux nil))
```

The fact of this intention, along with the persitent goal, allows for the inference of a variety of formulas concerning the agent's mental state, including the desired sets of goals and beliefs.

However, as these facts when printed out by the `(facts)` function refer to their component abstract facts by id, the facts are unreadable to humans. To alleviate this problem, a Java based parser was developed to produce human-readable versions of the facts. Thus,

```
f-344 (MAIN::Wff (mode Goal) (subj <Fact-332>) (subj2 nil)
    (time 0) (cost 10) (aux nil))
```

becomes after processing

`<Fact-344> At time 0 (Goal (Later (Neg (Pred NellInDanger ) ) ) ).` The human reader in this case avoids the three layers of indirection.

Chishom's story about the patrucidal [1] agent can play out in different ways. The agent could inadvertently run his uncle over on the way there, as in the original story. Alternatively, the agent could drive to his uncle's house and murder him. The version where the agent succeeds is encoded for demonstration purposes. The agent is motivated at the first timestep by his commitment that his uncle be dead to form an intention to kill his uncle by driving to his uncle's house and killing him. Therefore, at the first timestep, the agent intends to drive to his uncle's house. In the original story, he adopts no intention to kill his uncle at the next timestep, as his commitment has been accidentally satisfied. In the variant version encoded here, at the next timestep he actually does adopt the intention to kill his uncle.

The agent has a set of recipes encoding domain knowledge about killing his uncle. These single action recipes can be chained together to encode longer plans, by means of the modest planning rule in figure 3. According to this rule, an agent can produce a multiple action recipe consisting of two single action recipes if the postcondition of the first is the precondition of the second. Here, `?prequel` is the first recipe and `?r` is the second. The composite recipe has postcondition `?p`, the subject of a P-GOAL, this being the "ends" of our means ends reasoning. The recipes longer than two single actions are formed by means of the planning rule in figure 4.

Figure 5 completes the puzzle of how the first intention (toward driving to the uncle's house) is formed - it is recognized as part of a recipe that can presently

[1] Patrucide: The killing of one's paternal uncle.

bring about the agent's commitments. How then is the second intention (to kill the uncle) formed when no P-GOAL is explicitly declared for that timestep as required by the rule of figure 2? Simply because the P-GOAL persists, according to the BEFORE clause, from the first timestep to the second. These rules ensure that in each scenario presented, the agent adopts only those intentions that it theoretically should.

Finally, this implementation allows agents to act on the appropriate intention at the appropriate time, like Singh's restaurant agent. In an encoding of the restaurant story, the agent begins with persistent goals to serve coffee and tea, and at each timestep, adopts a strong intention toward an action that gets him closer to one of his goals (according to his recipes and the satisfaction of preconditions). The result is a series of six strong intentions: To pick up a cup, to pour tea into it, to serve tea, to pick up a cup, to pour coffee into it, and to serve coffee. The persistent goal to serve tea is dropped after serving tea, and the persistent goal to serve coffee is dropped after serving coffee.

This closed, single-agent system demonstrates the feasability of developing agents based on the theory of intention as a persistent goal. The agent in this implementation adopts only intentions toward actions, INTEND_1, as these suffice to produce the desired beliefs about action. For simple domains, rational agency in practice does not require the use of multiple definitions of intention. However, the underlying theory must include definitions to encompass nearly every natural language use of the word "intention" in order to counter problematic scenarios involving intention, including Chishom's, McDermott's, and Singh's examples.

This rule system provides an unwieldy means for encoding world states and domain knowledge, but it saves work in encoding agent states because arbitrarily complex sequences of intentions can be produced from a single initial commitment.

CHAPTER VII

CONCLUSION

Rational agents can be approached with a wide variety of logical theories. Open, multiagent environments require agents to proactively decide upon courses of action and reconsider their own activities. To specify the behavior of such an agent demands the use of a logic of intention. We have seen how one model with great potential for application but certain logical problems can be saved from absurdity without severely upsetting the underlying syntax or semantics.

C&L's theory of intention as a persistent goal has furnished a great foundation for new theory and application. Theories of joint intention and theories of speech acts can both be built from its constructs, and Tambe and Jennings have provided us with creditable implementations. However, Singh makes disturbing observations about the original theory. In particular, agents' commitments can be automatically (and inappropriately) brought about, agents can not intend the same action twice consecutively, and agents are unable to hold multiple intentions.

The original theory suffers from these problems, in addition to some vestige of the side-effect problem and the full fledged logical omniscience problem. Yet it has proved quite inspring in practical use. Since the theory is intended as a specification for the design of an agent, but the deduction of formulae in the logic is generally intractable, every agent based on it must embody an approximation of the theory. This approximation may include a subset of the logic with or without modifications. For example, the Jess implementation presented here can infer only a subset of true formulas, although these are adequate for the purposes of intention formation. Thus, designers are able to utilize the desirable features of the theory while ignoring or avoiding the undesirable aspects.

While the logical problems are not fatal to C&L's theory, there is insight to be had in solving some of these problems. The notion of intention as a persistent goal formalizes a partial solution to the problem of rational balance, making it clear that intentions should persist. The notion of weak intention as a form of commitment provides a framework to assist agents in constructing plans. Finally, plans may be executed by means of strong (present-directed) intentions, completing the formalization of rational balance.

By capturing these notions, and implicitly tying agents' beliefs to their actions, the theory presented here gives us certain insights into the rational agency. First of all, agents become more rational by being intentional, because they can persistently concentrate on goals and plans, and this facilitates means-ends reasoning. In addition, agents intend things to occur in the future with unspecified plans, but agents always know exactly what they are doing when they intentionally do it. These insights concur with certain conclusions of Searle and Bratman. The revised theory then, like the original, has philosophical appeal. On the implementation side of the problem, it seems feasible to design agents based on the revised theory of intention as a persistent goal, as demonstrated by the prototype agent presented. Future work involving the theory should involve applying the revisions to higher level constructs such as joint intentions, and developming more extensive experimental agent architectures, especially multiagent systems.

REFERENCES

[1] Joseph Bates, "The role of emotion in believable agents," Pittsburgh, Pennsylvania, Tech. Rep. CMU-CS-94-136, School of Computer Science, Carnegie Mellon University, April 1994.

[2] Russell Beale and Andrew Wood, "Agent-based interaction," in *People and Computers IX: Proceedings of HCI'94*, New York, 1994, pp. 239–245.

[3] Keith Biggers, "Automatic generation of communication and teamwork within multi-agent teams," M.S. thesis, Texas A&M University, College Station, 2001.

[4] M. Bratman, *Intentions, Plans, and Practical Reason*, Cambridge, Massachusetts: Harvard University Press, 1987.

[5] M. Bratman, "Planning and the stability of intention," *Minds and Machines*, vol. 2, no. 1, pp. 1–16, 1992.

[6] B. Chellas, *Modal Logic: An Introduction*, New York: Cambridge University Press, 1980.

[7] R.M. Chisholm, "Freedom and action," in *Freedom and Determinism.* New York: Random House, 1966, pp. 11–44.

[8] L. Chittaro and A. Montanari, "Temporal representation and reasoning in artificial intelligence: Issues and approaches," *Annals of Mathematics and Artificial Intelligence*, vol. 28, pp. 47–106, 2000.

[9] P.R. Cohen and H.J. Levesque, "Intention is choice with commitment," *Artificial Intelligence*, vol. 42, pp. 213–261, 1990.

[10] P.R. Cohen and H.J. Levesque, "Communicative actions for artificial agents," in *First International Conference on Multi-agent Systems*, San Francisco, California, 1995, pp. 65–72.

[11] A. Colmerauer, "The birth of prolog," in *The Second ACM-SIGPLAN History of Programming Languages Conference*, Cambridge, Massachusetts, March 1993, pp. 37–52.

[12] D. Connah and P. Wavish, "An experiment in cooperation," in *Decentralized AI - Proceedings of the 1st European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, y. Demazeau and J.-P. Müller, Eds., Amsterdam, The Netherlands, 1990, pp. 197–214.

[13] S. Consantini and A. Tocchio, "A logic programming language for multi-agent systems," in *Proceedings of the 8th European Conference on Logics in Artificial Intelligence*, Cosenza, Italy, September 2002, p. 113.

[14] N. Davies, "A first order theory of knowledge, belief and action," in *Proc. of the 10th ECAI*, Vienna, Austria, 1992, pp. 408–412.

[15] F. Dignum, J.-J. Ch. Meyer, R. Wieringa, and R. Kuiper, "A modal approach to intentions, commitments and obligations: Intention plus commitment yield obligation," in *Deontic Logic, Agency and Normative Systems.* Berlin, Germany: Springer 1996, pp. 80-97.

[16] J. Doyle, Y. Shoham, and M. Wellman, "A logic of relative desire," in *Methodologies for Intelligent Systems - Sixth International Symposium*, Palmero, Italy, 1991, pp. 16–31,

[17] Ralph Droms, "The knowbot information service," Reston, Virginia, FTP

Report, Corporation for National Research Initiatives (CNRI), December 1989.

[18] E.A. Emerson and J.Y. Halpern, "“Sometimes” and “Not Never” revisited: On branching versus linear time temporal logic," *Journal of the Association for Computing Machinery*, vol. 33, no. 1, pp. 151–178, January 1986.

[19] O. Etzioni, H.M. Levy, R.B. Segal, and C.A. Thekkath, "OS agents: Using AI techniques in the operating system environment," Seattle, Washington, Tech. Rep. UW-CSE-93-04-04, University of Washington, April 1993.

[20] O. Etzioni and D. Weld, "A softbot-based interface to the internet," *Communications of the ACM*, vol. 37, no. 7, pp. 72–76, July 1994.

[21] R. Fikes and N. Nillson, "STRIPS: A new approach to the application of theorem proving to problem solving," *Artificial Intelligence*, vol. 2, pp. 189-208, 1971.

[22] M. Fisher, "A survey of Concurrent MetateM- the language and its applications," in *Temporal Logic - 1st International Conference*, D.M. Gabbay and H. J. Ohlbach, Eds., Bonn, Germany, July 1994, pp. 480–505.

[23] C. Forgy, "A fast algorithm for the many patterns/many objects match problem," *Artificial Intelligence*, vol. 19, no. 1, pp. 17–37, 1982.

[24] O. Francesco, "Side-effects and Cohen's and Levesque's theory of intention as a persistent goal," *From the Logical Point of View*, vol. 3, no. 94, pp. 1–19, 1996.

[25] Michael R. Genesereth and Steven P. Ketchpel, "Software agents," *Communications of the ACM*, vol. 37, no. 7, pp. 49–53, July 1994.

[26] M.P. Georgeff and A. Lansky, "Reactive reasoning and planning," in *Proceedings of the 6th National Conference on Artificial Intelligence*, Seattle, Washington, July 1987, pp. 677–682.

[27] K. Ghedira and G. Verfaillie, "A multi-agent model for the resource allocation problem: A reactive approach," in *Proc. of the 10th ECAI*, Vienna, Austria, 1992, pp. 252–254.

[28] R. Goldblatt, "Mathematical modal logic: A view of its evolution," *Journal of Applied Logic*, vol. 1, pp. 309–392, 2003.

[29] Kreshna Gopal, "An adaptive planner based on learning of planning performance," M.S. thesis, Texas A&M University, College Station, 2000.

[30] B. Grosz and S. Kraus, "Collaborative plans for complex group action," *Artificial Intelligence*, vol. 86, pp. 269–357, 1996.

[31] F. Hayes-Roth, D. Waterman, and D. Lenat, Eds., *Building Expert Systems*, Redwood City, California: Addison-Wesley, 1983.

[32] C.E. Hewitt, "Offices are open systems," *ACM Transactions on Office Information Systems*, vol. 4, no. 3, pp. 271–287, 1986.

[33] R. Hilpinen, *Deontic Logic: Introductory and Systematic Readings*, Dordrecht, Germany: Reidel, 1971.

[34] J. Hintikka, *Knowledge and Belief*, Ithaca, New York: Cornell University Press, 1962.

[35] N.R. Jennings, "Controlling cooperative problem solving in industrial multi-agent systems using joint intentions," *Artificial Intelligence*, vol. 75, no. 2, pp. 195–240, June 1995,

[36] R. Jones, J.E. Laird, and P.E. Nielsen, "Automated intelligent pilots for combat flight simulation," in *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, Wisconsin, 1998, pp. 1047–1054.

[37] L.P. Kaelbing, "An architecture for intelligent reactive systems," in *Reasoning About Actions and Plans - Proceedings of the 1986 Workshop*, M.P. Georgeff and A.L. Lansky, Eds., Los Altos, California, 1986, pp. 395–410.

[38] S. Kannapan, (1993) Have your agent call my agent, Cornell University, Ithaca, New York, [online]
http://www.tc.cornell.edu/er/sci93/dis14agent/dis14agent.html

[39] G. Kiss and H. Reichgelt, "Towards a semantics of desires," in *Decentralized AI 3 - Proc. 3rd European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, E. Werner and Y. Demazeau, Eds., Kaiserslautern, Germany, 1992, pp. 115–128.

[40] M. Koster, (2005) "The web robots faq," The Web Robots Pages, Greenhills, England, [online] http://www.robotstxt.org/wc/faq.html

[41] R. Ladner and J. Reif, "The logic of distributed protocols: Preliminary report," in *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*. Monterey, California, 1986, pp. 207–222.

[42] John E. Laird, Allen Newell, and Paul S. Rosenbloom, "SOAR: An architecture for general intelligence," *Artificial Intelligence*, vol. 33, no. 1, pp. 1–64, 1987.

[43] Yezdi Lashkari, Max Metral, and Pattie Maes, "Collaborative interface agents," in *Proceedings of AAAI'94*, Seattle, Washington, August 1994,

pp. 715–720.

[44] H.J. Levesque, P.R. Cohen, and José H. T. Nunes, "On acting together," in *Proc. of the 8th National Conference on Artificial Intelligence*, Boston, Massachusetts, 1990, pp. 94–99.

[45] J. McCarthy, "Ascribing mental qualities to machines," Stanford, California Tech. Rep., AI Lab, Stanford University, 1978, http://www-formal.stanford.edu/jmc/ascribing/ascribing.html

[46] D. McDermott, "A temporal logic for reasoning about processes and plans," *Cognitive Science*, vol. 6, pp. 101–155, 1986.

[47] Max Metral, "Design of a generic learning interface agent," B.Sc. dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1993,

[48] Clifford Nass, J. Steuer, and E.R. Tauber, "Computers are social actors," in *CHI'94 Conference Proceedings*, Boston, Massachusetts, April 1994, pp. 72–78.

[49] M. Pauly and M. Woolridge, (2003) Logic for mechanism design - a manifesto, University of Liverpool, Liverpool, England, [online] http://citeseer.csail.mit.edu/577560.html

[50] A. Pnueli, "Specification and development of reactive systems," in *Information Processing*, H.-J. Kugler, Ed., Amsterdam, The Netherlands: Elsevier Science Publishers, Ltd., 1986, pp. 845–858.

[51] P. Quaresma and I. Rodrigues, "Using logic programming to model multi-agent web legal systems - an application report," in *Proc. of the Inter-*

national Conference on Artificial Intelligence and Law, St. Louis, Missouri, 2001, p. 10.

[52] A. Rao and M.P. Georgeff, "Modeling rational agents within a BDI-architecture," in *Proceedings of Knowledge Representation and Reasoning*, Cambridge, Massachusetts, April 1991, pp. 473–484.

[53] Sandia National Labs, (1997), Java Expert System Shell, Sandia National Laboratories, Livermore, Canada, [online] http://herzberg.ca.sandia.gov/jess

[54] J.C. Schlimmer and L.A. Hermens, "Software agents: Completing patterns and constructing user interfaces," *Journal of Artificial Intelligence Research*, vol. 1, pp. 61–89, November 1993.

[55] J. Searle, *Intentionality: An Essay in the Philosophy of Mind*, New York: Cambridge University Press, 1983.

[56] M. Singh, "A critical examination of the Cohen-Levesque theory of intentions," in *Proc. of the 10th European Conference on Artificial Intelligence*, Vienna, Austria, August 1992, pp. 364–368.

[57] Y. Shoham, "Agent-oriented programming," *Artificial Intelligence*, vol. 60, no. 1, pp. 51–92, 1993.

[58] M. Tambe, "Towards flexible teamwork," *Journal of Artificial Intelligence Research*, vol. 7, pp. 83–124, 1997.

[59] W. van der Hoek, B. van Linder, and J.-J. Ch. Meyer, "An integrated modal approach to rational agents," in *Foundations of Rational Agency, volume 14*

of Applied Logic Series, M. Woolridge and A. Rao, Eds., Dordrecht, Germany: Kluwer Academic Publishers, 1997, pp. 133–168, .

[60] W. van der Hoek and M. Woolridge, "Towards a logic of rational agency," *Logic Journal of the IGPL*, vol. 11, no. 2, pp. 133–157, March 2003.

[61] J. Wainer, "Yet another semantics of goals and goal priorities," in *Proceedings of the 11th European Conference on Artificial Intelligence*, Amsterdam, The Netherlands, August 1994, pp. 269–273.

[62] Andrew Wood, "Agent-based interaction," PhD progress report PR-94-4, University of Birmingham, Edgbaston, England, May 1994.

[63] M. Woolridge, *An Introduction to Multiagent Systems*, New York: John Wiley and Sons, 2002.

[64] M. Woolridge and P. Ciancarini, "Agent-oriented software engineering: The state of the art," *Lecture Notes in Computer Science*, vol. 1957, 2001.

[65] M. Woolridge and M. Fisher, "A first-order branching time logic of multiagent systems," in *Proc. of the 10th European Conference on Artificial Intelligence*, Vienna, Austria, 1992, pp. 234–238.

[66] M. Woolridge and N.R. Jennings, "Intelligent agents: Theory and practice," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.

[67] R.M. Wygant, "CLIPS - a powerful development and delivery expert system tool," *Computers and Industrial Engineering*, vol. 17, pp. 546–549, 1989.

[68] G. Zaverucha, "Logical foundations of a modal defeasible relevant logic of belief," in *Proc. of the 10th ECAI*, Vienna, Austria, 1992, pp. 615–619.

APPENDIX A

CODE SELECTIONS

```
(deftemplate actionExpression

 "an action expression is an action var, two
 actionExpressions separated by ;, or a Wff?."
 (slot mode (default Actionvar))
 (slot subj (default e))
 (slot subj2 (default nil))
 (slot length (default 1))
 (slot time (default abstract))
 (slot cost (default 10))
 )

(deftemplate Wff

 "a well formed formula "
 (slot mode (default Pred))
 (slot subj (default nil))
 (slot subj2 (default nil))
 (slot time (default abstract))
 (slot cost (default 10))
 (slot aux (default nil))
 )

(deftemplate Single-Action-Recipe

 (slot precondition)
 (slot action)
 (slot postcondition)
)

(deftemplate Multi-Action-Recipe

 (slot precondition)
 (slot first)
 (slot rest)
 (slot postcondition)
 )
```

Fig. 1. Data Structures of the Implementation

```
(defrule handleSingleActionExecution

 (Wff (mode PGOAL) (subj ?p) (time ?t))
 ?pre <- (Wff (time abstract) (mode ?m1) (subj ?s1) (subj2
      ?s2))
 ?pre_met <- (Wff (time ?t&~abstract) (mode ?m1)
       (subj ?s1)
      (subj2 ?s2))
 ?r <- (Single-Action-Recipe (precondition ?pre)
      (action ?e) (postcondition ?p))
 =>
 (assert (Wff (mode INTEND1) (subj ?e) (time ?t)))
 (printout t "Intention toward action " ?e " formed." crlf)
 )
```

Fig. 2. Intending the Immediate

```
(defrule handleSingleActionReasoning

 (Wff (mode PGOAL) (subj ?p) (time ?t))
 ?pre <- (Wff (time abstract) (mode ?m1) (subj ?s1)
      (subj2 ?s2))
 ?r <- (Single-Action-Recipe (precondition ?pre)
      (action?e) (postcondition ?p))
 (not (Wff (time ?t&~abstract) (mode ?m1) (subj ?s1)
      (subj2 ?s2)))
 ?prequel <- (Single-Action-Recipe (precondition ?prepre)
      (action ?e2) (postcondition ?pre))

 (not (Multi-Action-Recipe (precondition ?prepre)
 (postcondition ?p)))
 =>
 (assert (Multi-Action-Recipe (precondition ?prepre)
 (first ?prequel)
      (rest ?r) (postcondition ?p)))
 (printout t "2-Action-Recipe formed from " ?prequel " and
      " ?r crlf)
 )
```

Fig. 3. Planning Two-Action Sequences of Events

```
(defrule handleMultiActionReasoning
 (Wff (mode PGOAL) (subj ?p) (time ?t))
 ?pre <- (Wff (time abstract) (mode ?m1) (subj ?s1)
      (subj2 ?s2))
 ?r <- (Multi-Action-Recipe (precondition ?pre) (first ?e1)
      (rest ?e2)
      (postcondition ?p))
 ; precondition of our recipe is not met:
  (not (Wff (time ?t&~abstract) (mode ?m1) (subj ?s1)
      (subj2 ?s2)))
 ; but there exists a recipe to meet it:
  ?prequel <- (Single-Action-Recipe (precondition ?prepre)
      (action ?e)
      (postcondition ?pre))
 (not (Multi-Action-Recipe (precondition ?prepre)
 (postcondition ?p)))
 =>
 (assert (Wff (mode PGOAL) (subj ?pre) (time ?t)))
 (assert (Multi-Action-Recipe (precondition ?prepre)
      (first ?prequel)
      (rest ?r) (postcondition ?p)))
 (printout t "Multi-Action-Recipe formed from " ?prequel "
      and " ?rcrlf)
 )
```

Fig. 4. Planning Multi-Action Sequences of Events

```
( defrule handleMultiActionExecution
 (Wff (mode PGOAL) (subj ?p) (time ?t))
 ?pre <- (Wff (time abstract) (mode ?m1) (subj ?s1)
      (subj2 ?s2))
 ; actual preconditions met:
  ?pre_met <- (Wff (time ?t&~abstract) (mode ?m1)
      (subj ?s1) (subj2 ?s2))
 ?r <- (Multi-Action-Recipe (precondition ?pre) (first ?r1)
      (rest ?rs) (postcondition ?p))
 (not (Wff (mode INTEND1) (time ?t)))
 =>
 (assert (Wff (mode INTEND1) (subj (fact-slot-value ?r1
      action))
      (time ?t)))
 (printout t "Intention toward " (fact-slot-value ?r1
      action) ", first
      action of " ?r " formed at time " ?t " because "
      ?pre_met crlf)
 )
```

Fig. 5. Intending Progress on Long Action Sequences

```
(deffacts just-the-facts
 ;First, we need some abstract propositions to work with
 ;fact 1:  Little Nell in Danger
 (Wff (subj NellInDanger) (time abstract) (cost 0))
 ;fact 2:  Little Nell saved - not in danger
 (Wff (subj (fact-id 1)) (mode Neg) (time abstract)
      (cost 0)(aux Special))
 ;Next, we need a plan library for the agent
 (Single-Action-Recipe (precondition (fact-id 1))
      (action save_Nell) (postcondition (fact-id 2)))
 ;Now we can describe the world at each timestep
 ; agent commits to save nell;
 (Wff (subj (fact-id 2)) (mode PGOAL) (time 0)
 (cost 0))
 (Wff (subj NellInDanger) (time 0) (cost 0))
 (Wff (subj (fact-id 1)) (mode Bel) (time 0) (cost 0))
 (actionExpression (subj save_Nell) (time 0) (cost 0))
 (Wff (subj (fact-id 1)) (mode Neg) (time 1) (cost 0))
 (Wff (subj (fact-id 2)) (mode Bel) (time 1) (cost 0))
 )
```

Fig. 6. Encoding of McDermott’s Little Nell Story

APPENDIX B

UNABBREVIATED AGENT AND SCENARIO CODE

A. Chisholm's Patrucidal Agent

```
(deffacts chisholm-facts

 ;First, we need some abstract propositions to work with
 ; fact 1:  At Agent's House
 (Wff (subj AtHome) (time abstract) (cost 0))
 ;fact 2:  At Uncle's
 (Wff (subj AtUncles) (time abstract) (cost 0)
      (aux Special))
 ;fact 3:  Uncle alive
 (Wff (subj UncleAlive) (time abstract) (cost 0))
 ;fact 4:  Not at home
 (Wff (subj (fact-id 1)) (mode Neg) (time abstract)
      (cost 0)
 (aux Special))
 ;fact 5:  Not at Uncle's
 (Wff (subj (fact-id 2)) (mode Neg) (time abstract)
      (cost 0))
 ;fact 6:  Uncle dead
 (Wff (subj (fact-id 3)) (mode Neg) (time abstract)
      (cost 0) (aux Special))
 ;fact 7:  To drive to the Uncle's house
 (actionExpression (subj drive_to_uncles) (time abstract)
      (cost 0))
 ;fact 8:  To kill the Uncle
 (actionExpression (subj kill_uncle) (time abstract)
      (cost 0))
 ;fact 9:  agent at uncle's and uncle is living
 (Wff (subj (fact-id 2)) (subj2 (fact-id 3)) (cost 1)
      (mode And))
 ;fact 10:  agent at home and his uncle is living
 (Wff (subj (fact-id 1)) (subj2 (fact-id 3)) (cost 1)
      (mode And))
 ;Next, we need a plan library for the agent
 ;recipe 1:  driving to the uncle's will get us there
 ;with him alive!
  (Single-Action-Recipe (precondition (fact-id 10)) (action
      (fact-id 7))
      (postcondition (fact-id 9)))
 ;recipe 2:  if one is at the uncle's house and the uncle
 ; is alive, then one can kill the uncle
 (Single-Action-Recipe (precondition (fact-id 9)) (action
      (fact-id 8))
      (postcondition (fact-id 6)))

; time 0:  agent commits to uncle being dead

 (Wff (subj (fact-id 6)) (mode PGOAL) (time 0)
      (cost 1))
 ; at home
 (Wff (subj AtHome) (time 0) (cost 1) (aux Conj))
 ; not at uncle's
 (Wff (subj (fact-id 2)) (mode Neg) (time 0) (cost 1))
 ; uncle alive
 (Wff (subj UncleAlive) (time 0) (cost 1) (aux Conj))
 ; believe at home
 (Wff (subj (fact-id 1)) (mode Bel) (time 0) (cost 1))
 ; believe uncle alive
 (Wff (subj (fact-id 3)) (mode Bel) (time 0) (cost 1))
 (actionExpression (subj drive_to_uncles) (time 0)
      (cost 1))

;time 1:  agent drove to uncle's house

 ; not at home
 (Wff (subj (fact-id 1)) (mode Neg) (time 1) (cost 1))
 ; at uncle's
 (Wff (subj AtUncles) (time 1) (cost 1) (aux Conj))
 ; uncle alive
 (Wff (subj UncleAlive) (time 1) (cost 1) (aux Conj))
 ; believe at uncle's
 (Wff (subj (fact-id 2)) (mode Bel) (time 1) (cost 1))
 ; and believe uncle alive
 (Wff (subj (fact-id 3)) (mode Bel) (time 1) (cost 1))
 (actionExpression (subj kill_uncle) (time 1) (cost 1))

;time 2:  agent succeeded in heinous crime

 ; not at home
 (Wff (subj (fact-id 1)) (mode Neg) (time 2) (cost 1))
 ; at uncle's
 (Wff (subj AtUncles) (time 2) (cost 1))
 ; uncle dead
 (Wff (subj (fact-id 3)) (mode Neg) (time 2) (cost 1))
 ; believe at uncle's
 (Wff (subj (fact-id 2)) (mode Bel) (time 2) (cost 1))
 ; and believe uncle dead
 (Wff (subj (fact-id 6)) (mode Bel) (time 2) (cost 1))
 )
```

B. Singh's Restaurant Agent

```
(deffacts singh-restaurant

 "An agent intends to serve tea and coffee; he knows the recipes to
 do so.  To serve tea, he may place a filled cup of coffee on the
 counter.  To serve coffee, he may place a filled cup of coffee on
 the counter.  To fill a cup with tea, he may pour from the pitcher
 into an empty cup he holds.  To fill a cup with coffee, he may
 pour from the coffee pot into an empty cup he holds.  To hold a
 cup, he must pick up a cup."
 ;First, we need some abstract propositions to work with
 ;fact 1:  HoldingCupTea
 (Wff (subj HoldingCupTea) (time abstract) (cost 0)
      (aux Special))
```

```
;fact 2:  HoldingCupCoffee
(Wff (subj HoldingCupCoffee) (time abstract) (cost 0)
     (aux Special))
;fact 3:  HoldingEmptyCup
(Wff (subj HoldingEmptyCup) (time abstract) (cost 0)
     (aux Special))
;fact 4:  CoffeePotAvilible
(Wff (subj CoffeePotAvailible) (time abstract) (cost 0))
;fact 5:  TeaPitcherAvailible
(Wff (subj TeaPitcherAvailible) (time abstract) (cost 0)
     (aux Special))
;fact 6:  EmptyCupAvailible
(Wff (subj EmptyCupAvailible) (time abstract) (cost 0))
;fact 7:  TeaServed
(Wff (subj TeaServed) (time abstract) (cost 0)
     (aux Special))
;fact 8:  CoffeeServed
(Wff (subj CoffeeServed) (time abstract) (cost 0)
     aux Special)
;fact 9:  Not holding a cup of tea
(Wff (subj (fact-id 1)) (mode Neg) (time abstract)
     (cost 0) )
;fact 10:  Not holding a cup of coffee
(Wff (subj (fact-id 2)) (mode Neg) (time abstract)
     (cost 0))
;fact 11:  Not holding an empty cup
(Wff (subj (fact-id 3)) (mode Neg) (time abstract)
     (cost 0) )
;fact 12:  Coffee Pot is not availible
(Wff (subj (fact-id 4)) (mode Neg) (time abstract)
     (cost 0) )
;fact 13:  Tea Pitcher is not avialible
(Wff (subj (fact-id 5)) (mode Neg) (time abstract)
     (cost 0))
;fact 14:  An Empty Cup is availible
(Wff (subj (fact-id 6)) (mode Neg) (time abstract)
      (cost 0) )
 ;fact 15:  Tea Not Served
 (Wff (subj (fact-id 7)) (mode Neg) (time abstract)
      (cost 0))
 ;fact 16:  Coffee Not Served
 (Wff (subj (fact-id 8)) (mode Neg) (time abstract)
      (cost 0) )
 ;fact 17:  To serve tea
 (actionExpression (subj serve_tea) (time abstract)
      (cost 0))
 ;fact 18:  To serve coffee
 (actionExpression (subj serve_coffee) (time abstract)
      (cost 0))
 ;fact 19:  To pour tea
(actionExpression (subj pour_tea) (time abstract)
      (cost 0))
 ;fact 20:  To pour coffee
 (actionExpression (subj pour_coffee) (time abstract)
      (cost 0))
 ;fact 21:  To pick up a cup
 (actionExpression (subj pick_up_cup) (time abstract)
      (cost 0))
 ;fact 22:  Tea Pitcher availible and holding empty cup
 (Wff (subj (fact-id 3)) (subj2 (fact-id 5)) (cost 1)
      (mode And) (aux Special))
 ;fact 23:  Coffee Pot availible and holding empty cup
 (Wff (subj (fact-id 3)) (subj2 (fact-id 4)) (cost 1)
      (mode And) (aux Special))
 ;Next, we need a plan library for the agent
 ;recipe 1:  to serve tea, place a filled cup of tea on
 ;the counter
 (Single-Action-Recipe (precondition (fact-id 1))
      (action (fact-id 17)) (postcondition (fact-id 7)))
 ;recipe 2:  to serve coffee, place a filled cup of coffee
 ;on the counter
 (Single-Action-Recipe (precondition (fact-id 2))
      (action (fact-id 18)) (postcondition (fact-id 8)))
 ;recipe 3:  to fill tea, pour pitcher into cup you are
 ;holding
 (Single-Action-Recipe (precondition (fact-id 22))
      (action (fact-id 19)) (postcondition (fact-id 1)))
 ;recipe 4:  to fill coffee, pour pot into cup you are
 ;holding
 (Single-Action-Recipe (precondition (fact-id 23))
      (action (fact-id 20)) (postcondition (fact-id 2)))
 ;recipe 5:  to pick up cup, pick up an availible cup
 (Single-Action-Recipe (precondition (fact-id 6))
      (action (fact-id 21)) (postcondition (fact-id 22)))
 ;recipe 6:  to pick up cup, pick up an availible cup
 (Single-Action-Recipe (precondition (fact-id 6))
      (action (fact-id 21)) (postcondition (fact-id 23)))
 ;Now we can describe the world at each timestep
 ;time 0:  agent commits to serve tea and coffee
 ;the agent should pick up cup
 ;the empty cup is availible.
  ; agent commits to serve tea
 (Wff (subj (fact-id 7)) (mode PGOAL) (time 0) (cost 1))
 ; agent commits to serve coffee
 (Wff (subj (fact-id 8)) (mode PGOAL) (time 0) (cost 1))
 ; empty cup availible
 (Wff (subj EmptyCupAvailible) (time 0) (cost 1))
 ;time 1:  the coffee pot is not availible.  The
 ;tea pitcher is availible.  (the agent should pour tea)
 ; Tea Pitcher availible and holding empty cup
 (Wff (subj (fact-id 3)) (subj2 (fact-id 5)) (mode And)
      (time 1) (cost 1))
 ;time 2:  (It happens that the agent should
      ;serve tea)
 ; holding cup of tea.
  (Wff (subj HoldingCupTea) (time 2) (cost 1))
 ;time 3:  An empty cup is availible.  (the agent
      ;should pick up the cup)
 (Wff (subj EmptyCupAvailible) (time 3) (cost 1))
 (Wff (subj TeaServed) (time 3) (cost 1))
 ;time 4:  The coffee pot is availible.  (the
      ;agent should pour coffee)
 (Wff (subj (fact-id 3)) (subj2 (fact-id 4)) (mode And)
      (time 4) (cost 1))
 (Wff (subj HoldingEmptyCup) (time 4) (cost 1))
 ;time 5:  (It happens that the agent should
      ;serve coffee.)
  (Wff (subj HoldingCupCoffee) (time 5) (cost 1))
 ;time 6:  coffee is served
 (Wff (subj CoffeeServed) (time 6) (cost 1))
 )
```

C. Agent and World State Code

```
;******************************************************************
;General helper rules - generate abstract propositions to work with
;******************************************************************


;eleminate negations
(defrule eliminateNegations
?p <- (Wff (mode ?m) (subj ?s) (subj2 ?s2) (cost ?c)
(time abstract))
?neg <- (Wff (mode Neg) (subj ?p) (time abstract))
?neg2 <- (Wff (mode Neg) (subj ?neg) (time abstract))
?prop <- (Wff (mode ?what) (subj ?neg2) (subj2 nil) (time ?t))
(not (Wff (mode ?what) (subj ?p) (subj2 nil) (time ?t)))
=>
(assert (Wff (mode ?what) (subj ?p) (cost (+ ?c 1)) (time ?t)))
```

```
;(printout t "Elminated a double negation in " ?prop crlf)
)

;error - logical contradiction
(defrule contradiction
?a <- (Wff (time ?t&~abstract))
?b <- (Wff (mode Neg) (subj ?a) (time ?t&~abstract))
=>
(printout t " ***** ERROR: Logical contradiction between " ?a "
and "
?b crlf)
)

(defrule abstractBelief
?p <- (Wff (cost ?c) (time abstract))
(not (Wff (mode Bel) (subj ?p)))
(test (< ?c 2))
=>
(assert (Wff (mode Bel) (subj ?p)))
)

;Abstract Propositions
(defrule abstractProps
(Wff (mode ?m) (time ?t&~abstract) (subj ?x) (subj2 ?y)
(cost ?c))
(not (Wff (mode ?m) (time abstract) (subj ?x) (subj2 ?y)))
```

```
(test (< ?c 2))
=>
(assert (Wff (mode ?m) (time abstract) (subj ?x) (subj2 ?y)
(cost (+ 1 ?c))))
)

;Abstract negations of propositions
(defrule abstractNegs
(Wff (mode ?m) (time ?t&˜abstract) (subj ?x) (subj2 ?y)
(cost ?c))
?pabs <- (Wff (mode ?m) (time abstract) (subj ?x) (subj2 ?y)
(cost ?c2))
(not (Wff (mode Neg) (time abstract) (subj ?pabs)))
(test (< ?c2 2))
=>
(assert (Wff (mode Neg) (time abstract) (subj ?pabs)
(cost ?c2)))
)

;NOT NOT
(defrule doublenegL
?pabs <- (Wff (time abstract) (cost ?c) (mode ?m) (subj ?x)
(subj2 ?y))
?p <- (Wff (time ?t&˜abstract) (cost ?c2) (mode ?m )
(subj ?x)
(subj2 ?y))
```

```
(test (< ?c2 2))
?pnegabs <- (Wff (time abstract) (cost ?c3) (mode Neg)
(subj ?pabs))
(not (Wff (time ?t) (mode Neg) (subj ?pnegabs)))
=>
(assert (Wff (time ?t) (cost (+ 1 ?c)) (mode Neg)
(subj ?pnegabs)))
)

;Abstract time events - useless if events abstractly
;declared in facts
(defrule abstractEvents
(actionExpression (mode ?m) (time ?t&~abstract)
(length ?l)
(subj ?ev) (subj2 ?ev2) (cost ?c))
(not (actionExpression (mode ?m) (time abstract)
(subj ?ev) (subj2 ?ev2)
(length ?l)))
(test (< ?c 2))
=>
(assert (actionExpression (time abstract) (cost (+ 1 ?c))
(subj ?ev)
(subj2 ?ev2) (length ?l) (mode ?m)))
)

;*******************************************************************
```

```
; Definitions - Produce elaborations on world state description
;*******************************************************************
;Occurrence of sequences (defrule SequenceOccurs
?e1 <- (actionExpression (mode ?typ1) (subj ?s)
(subj2 ?s2)
(time ?t1&~abstract) (length ?l) (cost ?c1))
?e1abs <- (actionExpression (mode ?typ1) (subj ?s)
(subj2 ?s2)
(time abstract) (length ?l))
?e2 <- (actionExpression (mode ?typ2) (subj ?s3)
(subj2 ?s4)
(time ?t2&~abstract) (length ?l2) (cost ?c2))
?e2abs <- (actionExpression (mode ?typ2) (subj ?s3)
(subj2 ?s4)
(time abstract) (length ?l2))
(test (< (+ ?c1 ?c2) 1))
(test (eq (+ ?t1 ?l) ?t2))
(not (actionExpression (mode Semicolon) (time ?t1)
(subj ?e1abs)
(subj2 ?e2abs)))
=>
(assert (actionExpression (mode Semicolon) (time ?t1)
(length (+ ?l ?l2))
(cost ( + 1 (+ ?c1 ?c2))) (subj ?e1abs ) (subj2 ?e2abs )))
;(printout t "Sequence " ?e1 ";" ?e2 " occurs at time "
?t1 " to " ?t2 ",
cost " (+ ?c1 ?c2) crlf)
)

;Test Action
(defrule TestActionOccurs
(Wff (subj ?s) (subj2 ?s2) (mode ?m) (time ?ti&~abstract)
(cost ?c))
?wabs <- (Wff (subj ?s) (subj2 ?s2) (mode ?m) (time abstract)
(cost ?c1))
(test (< ?c 2))
(not (actionExpression (mode Testaction) (subj ?wabs)
(time ?ti)))
=>
(assert (actionExpression (mode Testaction) (subj ?wabs)
(time ?ti)
(length 0) (cost (+ ?c 1))))
;(printout t "Test Action on " ?wabs " occurs at time
" ?ti ", cost "
?c crlf)
)

;EVENTUALLY
(defrule whenEventually
?trash <- (Wff (time ?t&~abstract) (cost ?c1) )
?pabs <- (Wff (time abstract) (mode ?m) (subj ?s) (subj2 ?s2)
(cost ?c3))
```

```
?p <- (Wff (time ?t2&~abstract)(mode ?m) (subj ?s)
(subj2 ?s2)
(cost ?c2))
(not (Wff (time ?t) (mode Eventually) (subj ?pabs)))
(test (<= ?t ?t2))
(test (< ?c2 3))
=>
(assert (Wff (time ?t) (mode Eventually) (subj ?pabs)
(cost (+ 1 ?c2))))
;(printout t "Eventually " ?pabs " occurs at time "
?t ", cost "
(+ ?c1 ?c2) crlf)
)


;LATER (defrule whenLater
;If ?p is not the case but is the case eventually,
;then Later ?p.
?pneg <- (Wff (time ?t&~abstract) (mode Neg)
(subj ?p) (cost ?c1))
?peventually <- (Wff (time ?t) (mode Eventually)
(subj ?p) (cost ?c2))
(test (< (+ ?c1 ?c2) 4))
(not (Wff (time ?t) (subj ?p) (mode Later)))
=>
(assert (Wff (time ?t) (cost (+ ?c2 ?c1)) (mode Later)
(subj ?p) ))
```

```
;(printout t "Later " ?p " occurs at time " ?t ", cost "
(+ 1 (+ ?c2 ?c1))
crlf)
)

;HAPPENS
(defrule ActionsHappen
?act <- (actionExpression (mode ?typ) (subj ?s) (subj2 ?s2)
(time ?t&~abstract) (length ?l) (cost ?c))
?actabs <- (actionExpression (mode ?typ) (subj ?s) (subj2 ?s2)
(time abstract) (length ?l) )
(test (< ?c 3))
(not (Wff (mode Hap) (subj ?actabs) (time ?t) ))
=>
(assert (Wff (mode Hap) (subj ?actabs) (time ?t)
(cost (+ ?c 1))))
;(printout t "Happens Action " ?actabs " at time " ?t ", cost "
?c crlf)
)

;DONE
(defrule ActionsDone
?act <- (actionExpression (mode ?typ) (subj ?e) (subj2 ?e2)
(time ?t&~abstract) (length ?l) (cost ?c))
?actabs <- (actionExpression (mode ?typ) (subj ?e) (subj2 ?e2)
(time abstract) (length ?l) )
```

```
(test (< ?c 3))
(not (Wff (mode Done) (subj ?actabs) (time ?t)))
=>
(assert (Wff (mode Done) (subj ?actabs) (time (+ ?t ?l))
(cost (+ ?c 1))))
;(printout t "Done Action " ?actabs " at time " (+ ?t ?l) ",
cost " ?c
crlf)
)

;BEFORE
;(BEFORE p q)
; if q occurs in a sequence of events, p occurs before it does.
(defrule whenBefore
(Wff (time ?t&~abstract)) ?q <- (Wff (time ?tq&~abstract) (subj ?x)
(subj2 ?x2)
(mode ?m)
(cost ?cq))
?qabs <- (Wff (time abstract) (subj ?x) (subj2 ?x2) (mode ?m))
?p <- (Wff (time ?tp&~abstract) (subj ?y) (subj2 ?y2)
(mode ?m2)
(cost ?cp))
?pabs <- (Wff (time abstract) (subj ?y) (subj2 ?y2) (mode ?m2))
(test ( <= ?t ?tp ))
(test ( <= ?tp ?tq))
(test ( < (+ ?cp ?cq) 1))
```

```
(not (Wff (mode Before) (subj ?pabs) (subj2 ?qabs)))
=>
(assert (Wff (mode Before) (subj ?pabs) (subj2 ?qabs) (cost (+ 1
(+ ?cp ?cq))) (time ?t)))
;(printout t "Before " ?pabs " " ?qabs " at time " ?tp " and "
?tq ", cost " (+ (+ 1 ?cp) ?cq) crlf)
)

;AND
(defrule whenAnd
?alpha <- (Wff (time abstract) (subj ?x1) (subj2 ?y1) (mode ?m))
?beta <- (Wff (time abstract) (subj ?x2) (subj2 ?y2) (mode ?m2))
(and (Wff (time ?t&~abstract) (subj ?x1) (subj2 ?y1) (mode ?m)
(cost ?c1) (aux Conj)) (Wff (time ?t&~abstract) (subj ?x2)
(subj2 ?y2) (mode ?m2) (cost ?c2) (aux Conj)))
(not (Wff (time ?t) (subj ?alpha) (subj ?beta) (mode And)))
(test (< (+ ?c1 ?c2) 3))
=>
(assert (Wff (time ?t) (mode And) (subj ?alpha) (subj2 ?beta)
(cost ( + (+ ?c1 ?c2) 1 ))))
;(printout t ?alpha " AND " ?beta " occurs at time " ?t " cost "
?c1 " +"
?c2 " = " (+ ?c1 ?c2) crlf)
)

;AND2
```

```
(defrule whenAnd2
?alpha <- (Wff (time abstract) (subj ?x1) (subj2 ?y1) (mode ?m))
?beta <- (Wff (time abstract) (subj ?x2) (subj2 ?y2) (mode ?m2))
(and (Wff (time ?t&~abstract) (subj ?x1) (subj2 ?y1) (mode ?m)
(cost ?c1) (aux Conj)) (Wff (time ?t&~abstract) (subj ?x2)
(subj2 ?y2) (mode ?m2) (cost ?c2) (aux Conj)))
(not (Wff (time ?t) (subj ?beta) (subj ?alpha) (mode And)))
(test (< (+ ?c1 ?c2) 3))
=>
(assert (Wff (time ?t) (mode And) (subj ?beta) (subj2 ?alpha)
(cost ( + (+ ?c1 ?c2) 1 ))))
;(printout t ?beta " AND " ?alpha " occurs at time " ?t " cost "
?c1 " +"
?c2 " = " (+ ?c1 ?c2) crlf)
)

;OR
(defrule whenOr
?alpha <- (Wff (time abstract) (subj ?x1) (subj2 ?y1) (mode ?m))
?beta <- (Wff (time abstract) (subj ?x2) (subj2 ?y2) (mode ?m2))
?trash <- (Wff (time ?t&~abstract) (subj ?x1|?x2)
(subj2 ?y1|?y2) (mode ?m|?m2) (cost ?c1|?c2))
(or (Wff (time ?t&~abstract) (subj ?x1) (subj2 ?y1) (mode ?m)
(cost ?c1)) (Wff (time ?t&~abstract) (subj ?x2) (subj2 ?y2)
(mode ?m2) (cost ?c2)))
(not (Wff (time ?t) (subj ?alpha) (subj ?beta) (mode Or)))
(test (< (+ ?c1 ?c2) 2))
=>
(assert (Wff (time ?t) (mode Or) (subj ?alpha) (subj2 ?beta)
(cost ( + (+ ?c1 ?c2) 1 ))))
;(printout t ?alpha " OR " ?beta " occurs at time " ?t " cost "
(+ ?c1 ?c2) crlf)
)

(defrule whenOr2
?alpha <- (Wff (time abstract) (subj ?x1) (subj2 ?y1) (mode ?m))
?beta <- (Wff (time abstract) (subj ?x2) (subj2 ?y2) (mode ?m2))
?trash <- (Wff (time ?t&~abstract) (subj ?x1|?x2)
(subj2 ?y1|?y2) (mode ?m|?m2) (cost ?c1|?c2))
(or (Wff (time ?t&~abstract) (subj ?x1) (subj2 ?y1) (mode ?m)
(cost ?c1)) (Wff (time ?t&~abstract) (subj ?x2) (subj2 ?y2)
(mode ?m2) (cost ?c2)))
(not (Wff (time ?t) (subj ?beta) (subj ?alpha) (mode Or)))
(test (< (+ ?c1 ?c2) 2))
=>
(assert (Wff (time ?t) (mode Or) (subj ?beta) (subj2 ?alpha)
(cost ( + (+ ?c1 ?c2) 1 ))))
;(printout t ?beta " OR " ?alpha " occurs at time " ?t " cost "
(+ ?c1 ?c2) crlf)
)

;******************************************************************
```

```
; REST OF IMPLEMENTATION IS AGENT CODE
;******************************************************************
;******************************************************************
; Rules for planning
;******************************************************************
;What we should do about single actions if no current plan but
preconditions met
(defrule handleSingleActionExecution
(Wff (mode PGOAL) (subj ?p) (time ?t))
?pre <- (Wff (time abstract) (mode ?m1) (subj ?s1) (subj2 ?s2))
?pre_met <- (Wff (time ?t&~abstract) (mode ?m1) (subj ?s1)
(subj2 ?s2))
(not (Wff (mode INTEND1) (time ?t)))
?r <- (Single-Action-Recipe (precondition ?pre) (action ?e)
(postcondition ?p))
=>
(assert (Wff (mode INTEND1) (subj ?e) (time ?t)))
(printout t "Intention toward action " ?e " formed at time " ?t
crlf)
)


;What we should do about single actions if no current plan
; and no preconditions met
(defrule handleSingleActionReasoning
(Wff (mode PGOAL) (subj ?p) (time ?t))
?pre <- (Wff (time abstract) (mode ?m1) (subj ?s1) (subj2 ?s2))
```

```
?r <- (Single-Action-Recipe (precondition ?pre) (action ?e)
(postcondition ?p))
; precondition of our recipe is not met...
(not (Wff (time ?t&~abstract) (mode ?m1) (subj ?s1)
(subj2 ?s2)))
?prequel <- (Single-Action-Recipe (precondition ?prepre)
(action ?e2)
(postcondition ?pre)) ; but there exists a recipe to meet it!
(not (Multi-Action-Recipe (precondition ?prepre)
(postcondition ?p)))
=>
(assert (Multi-Action-Recipe (precondition ?prepre)
(first ?prequel)
(rest ?r) (postcondition ?p)))
(printout t "2-Action-Recipe formed from " ?prequel " and " ?r
" (whose
precondition is " (fact-slot-value ?r precondition) ") with
precondition "
?prepre " and postcondition " ?p crlf)
)

(defrule handleMultiActionReasoning
(Wff (mode PGOAL) (subj ?p) (time ?t))
?pre <- (Wff (time abstract) (mode ?m1) (subj ?s1) (subj2 ?s2))
?r <- (Multi-Action-Recipe (precondition ?pre) (first ?e1)
(rest ?e2) (postcondition ?p))
```

```
; precondition of our recipe is not met:
(not (Wff (time ?t&~abstract) (mode ?m1) (subj ?s1)
(subj2 ?s2)))
; but there exists a recipe to meet it:
?prequel <- (Single-Action-Recipe (precondition ?prepre)
(action ?e) (postcondition ?pre))
(not (Multi-Action-Recipe (precondition ?prepre)
(postcondition ?p)))
=>
(assert (Wff (mode PGOAL) (subj ?pre) (time ?t)))
(assert (Multi-Action-Recipe (precondition ?prepre)
(first ?prequel)
(rest ?r) (postcondition ?p)))
(printout t "Multi-Action-Recipe formed from " ?prequel " and "
?r crlf)
)

(defrule handleMultiActionExecution
(Wff (mode PGOAL) (subj ?p) (time ?t))
?pre <- (Wff (time abstract) (mode ?m1) (subj ?s1) (subj2 ?s2))
; actual preconditions met:
?pre_met <- (Wff (time ?t&~abstract) (mode ?m1) (subj ?s1)
(subj2 ?s2))
?r <- (Multi-Action-Recipe (precondition ?pre) (first ?r1)
(rest ?rs) (postcondition ?p)) (not (Wff (mode INTEND1) (time ?t)))
=>
```

```
; then intend the action
(assert (Wff (mode INTEND1) (subj (fact-slot-value ?r1 action))
(time ?t)))
(printout t "Intention toward " (fact-slot-value ?r1 action) ",
first action of " ?r " formed at time " ?t " because "
?pre_met crlf)
)
;******************************************************************
; Agency helper rules
;******************************************************************

(defrule PersistenceOfPGOAL
?p <- (Wff (time abstract) (mode ?m) (subj ?s1) (subj2 ?s2))
?notp <- (Wff (time abstract) (mode Neg) (subj ?p))
?alwaysnotp <- (Wff (time abstract) (mode Always) (subj ?notp))
(Wff (mode PGOAL) (subj ?p) (time ?t&~abstract) (cost ?c))
; our P-GOAL does not come true next time
(not (Wff (time ?u&:(eq ?u (+ ?t 1))) (mode ?m) (subj ?s1)
(subj2 ?s2)))
; and we don't believe always not p
(not (Wff (time ?u&:(eq ?u (+ ?t 1))) (mode Bel)
(subj ?alwaysnotp)))
(not (Wff (time ?u&:(eq ?u (+ ?t 1))) (mode PGOAL) (subj ?p)))
=>
(printout t "P-GOAL of " ?p " persists from time " ?t " to "
(+ ?t 1)
crlf)
(assert (Wff (time (+ ?t 1)) (mode PGOAL) (subj ?p)))
)

;rule to generate abstract always not p
(defrule PGOAL_helper_gen_always_not_p
?p <- (Wff (time abstract) (aux Special))
?neg <- (Wff (mode Neg) (time abstract) (subj ?p) (cost ?c))
(not (Wff (mode Always) (subj ?neg)))
=>
(assert (Wff (mode Always) (subj ?neg) (cost (+ ?c 1))))
(printout t "P-GOAL helper Always Not " ?p crlf)
)

;rule go generate Bel of always not p
(defrule PGOAL_helper_gen_bel_always_not_p
?p <- (Wff (time abstract) (aux Special))
?neg <- (Wff (mode Neg) (time abstract) (subj ?p))
?always <- (Wff (mode Always) (subj ?neg) (cost ?c))
(not (Wff (mode Bel) (time abstract) (subj ?always)))
=>
(assert (Wff (mode Bel) (subj ?always) (cost (+ ?c 1))))
(printout t "P-GOAL helper Bel Always Not " ?p crlf)
)

;rule to generate disjunction of bel and bel always not p
```

```
(defrule PGOAL_helper_gen_disj

?p <- (Wff (time abstract)(aux Special))

?neg <- (Wff (time abstract) (mode Neg) (subj ?p))

?always <- (Wff (mode Always) (subj ?neg))

?bel2 <- (Wff (time abstract) (mode Bel) (subj ?always))

?bel1 <- (Wff (time abstract) (mode Bel) (subj ?p))

(not (Wff (time abstract) (mode Or) (subj ?bel1) (subj2 ?bel2)))

=>

(assert (Wff (time abstract) (mode Or) (subj ?bel1)

(subj2 ?bel2)))

(printout t "P-GOAL helper disj " ?bel1 ?bel2 crlf)

)


;rule to generate abstract later of p

(defrule PGOAL_helper_gen_later_p

?p <- (Wff (time abstract) (cost ?c)(aux Special))

(not (Wff (mode Later) (time abstract) (subj ?p)))

(test (< ?c 2))

=>

(assert (Wff (mode Later) (subj ?p) (cost (+ ?c 1))))

(printout t "P-GOAL helper later " ?p crlf)

)


;rule to generate abstract of Not p

(defrule PGOAL_helper_gen_not_p

?p <- (Wff (time abstract) (cost ?c)(aux Special))
```

```
(not (Wff (mode Neg) (time abstract) (subj ?p)))
(test (< ?c 2))
=>
(assert (Wff (mode Neg) (subj ?p) (cost (+ ?c 1))))
(printout t "P-GOAL helper Not " ?p crlf)
)

;rule to generate abstract goal of later of p
(defrule PGOAL_helper_gen_goal_later_p
?p <- (Wff (time abstract)(aux Special))
?laterp <- (Wff (time abstract) (subj ?p) (mode Later)
(cost ?c))
(not (Wff (mode Goal) (subj ?laterp)))
(test (< ?c 3))
=>
(assert (Wff (mode Goal) (subj ?laterp) (cost (+ ?c 1))))
(printout t "P-GOAL helper goal later " ?p crlf)
)

;rule to generate abstract neg goal later p
(defrule PGOAL_helper_gen_neg_goal_later_p
?p <- (Wff (time abstract)(aux Special))
?laterp <- (Wff (time abstract) (subj ?p) (mode Later))
?goallaterp <- (Wff (mode Goal) (time abstract) (subj ?laterp)
(cost ?c))
(not (Wff (mode Neg) (subj ?goallaterp)))
```

```
(test (< ?c 4))
=>
(assert (Wff (mode Neg) (subj ?goallaterp) (cost (+ ?c 1))))
(printout t "P-GOAL helper neg goal later " ?p crlf)
)

(defrule handleINTEND1_1
(Wff (mode INTEND1) (subj ?s) (time ?t) )
(not (Wff (mode Happens) (subj ?s) (time abstract)))
=>
(assert (Wff (mode Happens) (subj ?s) (time abstract)))
)

(defrule handleINTEND1_2
(Wff (mode INTEND1) (subj ?s) (time ?t) )
?h <- (Wff (mode Happens) (subj ?s) (time abstract))
(not (Wff (mode Bel) (subj ?h) (time ?t)))
=>
(assert (Wff (mode Bel) (subj ?h) (time ?t)))
)

;*******************************************************************
; PGOAL definition
;*******************************************************************
;infers existence of a P-GOAL
;P-GOAL
```

```
(defrule whenPGOAL
?p <- (Wff (time abstract) (mode ?m) (subj ?s) (subj2 ?s2)
(cost ?c))
; *** Abstract elements of the P-GOAL ***
;abstract of Later p
?later <- (Wff (time abstract) (mode Later) (subj ?p))
;abstract Neg of p
?not <- (Wff (time abstract) (mode Neg) (subj ?later))
;abstract of Always Not p
?always_not <- (Wff (time abstract) (mode Always)
(subj ?not))
;abstract of Bel Always Not p
?bel_always_not <- (Wff (time abstract) (mode Bel)
(subj ?always_not))
;abstract of Bel p
?bel <- (Wff (time abstract) (mode Bel) (subj ?p))
;abstract disjunction ?bel
;or ?bel_always_not
?disj <- (Wff (time abstract) (mode Or) (subj ?bel)
(subj2 ?bel_always_not))
;abstract goal of ?later
?goal_later <- (Wff (time abstract) (mode Goal)
(subj ?later))
;abstract Neg of ?goal_later
?neg_goal_later <- (Wff (time abstract) (mode Neg)
(subj ?goal_later))
```

```
; top Level composition of the P-GOAL in current rule
;starts here
;goal of ?later, time t
(Wff (time ?t&~abstract) (mode Goal) (subj ?later))
;bel of ?not, time t
(Wff (time ?t) (mode Bel) (subj ?not))
;before of ?disj ?neg_goal_later, time t
(Wff (time ?t) (mode Before) (subj ?disj)
(subj2 ?neg_goal_later))
;P-GOAL not asserted yet
(not (Wff (time ?t) (mode PGOAL) (subj ?p)))
=>
(assert (Wff (time ?t) (mode PGOAL) (subj ?p)
(cost (+ 1 ?c))))
(printout t "P-GOAL of " ?p " at time " ?t ", cost "
?c crlf)
)

;*****************************************************************
; results of a PGOAL
;*****************************************************************
;what results from a PGOAL
(defrule whatPGOAL
(Wff (mode PGOAL) (subj ?p) (time ?t&~abstract))
;abstract of Later p - has helper
?later <- (Wff (time abstract) (mode Later) (subj ?p))
```

```
(not (Wff (time ?t) (mode Goal) (subj ?later)))
=>
(assert (Wff (time ?t) (mode Goal) (subj ?later)))
(printout t "Asserted the Goal conjunct of a P-GOAL of " ?p crlf)
)

(defrule whatPGOAL2
(Wff (mode PGOAL) (subj ?p) (time ?t&~abstract))
(Wff (mode PGOAL) (subj ?p) (time ?t&~abstract))
;abstract Neg of p
?not <- (Wff (time abstract) (mode Neg) (subj ?p))
(not (Wff (time ?t) (mode Bel) (subj ?not)))
=>
;bel of ?neg, time t
(assert (Wff (time ?t) (mode Bel) (subj ?not)))
(printout t "Asserted the Bel conjunct of a P-GOAL of " ?p crlf)
)

(defrule whatPGOAL3
(Wff (mode PGOAL) (subj ?p) (time ?t&~abstract))
;abstract neg of p
?not <- (Wff (time abstract) (mode Neg) (subj ?p))
;abstract of Later p - has helper
?later <- (Wff (time abstract) (mode Later) (subj ?p))
;abstract of Bel p
?bel <- (Wff (time abstract) (mode Bel) (subj ?p))
```

```
;abstract of Always Not p - has helper
?always_not <- (Wff (time abstract) (mode Always) (subj ?not))
;abstract of Bel Always Not p
?bel_always_not <- (Wff (time abstract) (mode Bel)
(subj ?always_not))
;abstract disjunction of ?bel and ?bel_always_not
?disj <- (Wff (time abstract) (mode Or) (subj ?bel)
(subj2 ?bel_always_not))
;abstract goal of ?later
?goal_later <- (Wff (time abstract) (mode Goal) (subj ?later))
;abstract Neg of ?goal_later
?neg_goal_later <- (Wff (time abstract) (mode Neg)
(subj ?goal_later))
(not (Wff (time ?t) (mode Before) (subj ?disj)
(subj2 ?neg_goal_later)))
=>
;before of ?disj ?neg_goal_later, time t
(assert (Wff (time ?t) (mode Before) (subj ?disj)
(subj2 ?neg_goal_later)))
(printout t "Asserted the Before conjunct of a P-GOAL of " ?p crlf)
)

;*****************************************************************
; results of an INTEND1 - the agent belives the act occurs.
;*****************************************************************
```

```
;what results from an intention toward action

(defrule whatINT1

(Wff (mode INTEND1) (subj ?e) (time ?t&~abstract))

?happening <- (Wff (mode Happens) (subj ?e) (time abstract))

(not (Wff (time ?t) (mode Bel) (subj ?happening)))

=>

(assert (Wff (time ?t) (mode Bel) (subj ?happening)))

(printout t "Asserted the Bel conjunct of an INTEND1 of " ?e crlf)

)
```

VITA

James Silas Creel was born on July 10, 1980. He attended The University of Texas at Austin, receiving a B.A. degree in economics and a B.S. degree in computer science in 2003. As an undergraduate, he worked under Dr. Bruce Porter in the area of knowledge representation. He went on to attend Texas A&M University, working under Dr. Thomas Ioerger in the area of formal logics for multiagent systems. His interests include intelligent agents, knowledge representation, and philosophy of mind.

James Silas Creel
Department of Computer Science
Texas A&M University
301 Harvey R. Bright Building
College Station, TX 77843-3112
EMAIL: james.creel@gmail.com

The typist for this thesis was the author.