Active Logics: A Unified Formal Approach to Episodic Reasoning*

Jennifer Elgot-Drapkin[2]
Sarit Kraus[1,3]
Michael Miller[4]
Madhura Nirkhe[5]
Donald Perlis[1,6]
[1]Institute for Advanced Computer Studies, University of Maryland

[2]Arizona State University, Tempe, AZ, drapkin@asu.edu
[3]Bar Ilan University, Israel, sarit@bimacs.cs.biu.ac.il
[4]Intelligent Automation, Inc., Rockville, MD, mm@i-a-i.com
[5]Microsoft Corporation, Redmond, WA, madhuran@microsoft.com
[6]University of Maryland, College Park, MD, perlis@cs.umd.edu

First-order logic doesn't have
a good concept of time.
Jerry Feldman, circa 1980

...robots with a life of their own...
Nils Nilsson, 1983

Past is Prologue
Shakespeare

Abstract

Artificial intelligence research falls roughly into two categories: formal and implementational. This division is not completely firm: there are implementational studies based on (formal or informal) theories (e.g., CYC, SOAR, OSCAR), and there are theories framed with an eye toward implementability (e.g., predicate circumscription). Nevertheless, formal/theoretical work tends to focus on very narrow problems (and even on very special cases of very narrow problems) while trying to get them "right" in a very strict sense, while implementational work tends to aim at fairly broad ranges of behavior but often at the expense of any kind of overall conceptually unifying framework that informs understanding. It is sometimes urged that this gap is intrinsic to the topic: intelligence is not a unitary thing for which there will be a unifying theory, but rather a "society" of subintelligences whose overall behavior cannot be reduced to useful characterizing and predictive principles.

Here we describe a formal architecture that is more closely tied to implementational constraints than is usual for formalisms, and which has been used to solve a number of commonsense problems in a unified manner. In particular, we address the issue of formal, integrated, and longitudinal reasoning: inferentially-modeled behavior that incorporates a fairly wide variety of types of commonsense reasoning within the context of a single extended episode of activity requiring keeping track of ongoing progress, and altering plans and beliefs accordingly. Instead of aiming at optimal solutions to isolated, well-specified and temporally narrow problems, we focus on satisficing solutions to under-specified and temporally-extended problems, much closer to real-world needs. We believe that such a focus is required for AI to arrive at truly intelligent mechanisms with the ability to behave effectively over considerably longer time periods and range of circumstances than is common in AI today. While this will surely lead to less elegant formalisms, it also surely is requisite if AI is to get fully out of the blocks-world and into the real world.

*This research was supported in part by Army Research Office Award DAAH049510628, National Science Foundation Award IRI-9210906, and National Science Foundation Award IRI-9311988.

Contents

1 Introduction

Artificial intelligence research falls roughly into two categories: formal and implementational. This division is not completely firm: there are implementational studies based on (formal or informal) theories (e.g., CYC [70, 71, 45], SOAR [68, 123], OSCAR [117, 115]), and there are theories framed with an eye toward implementability (e.g., predicate circumscription [82]). Nevertheless, formal/theoretical work tends to focus on very narrow problems (and even on very special cases of very narrow problems) while trying to get them "right" in a very strict sense, while implementational work tends to aim at fairly broad and/or efficient modes of behavior but often at the expense of any kind of overall conceptually unifying framework that informs understanding. It is sometimes urged that this gap is intrinsic to the topic: intelligence is not a unitary thing for which there will be a unifying theory, but rather a "society" of subintelligences whose overall behavior cannot be reduced to useful characterizing and predictive principles (e.g., Minsky [95]).

Here we describe a formal architecture—known as *active logic*—that is more closely tied to implementational constraints than is usual for formalisms, and which has been used to solve a number of commonsense problems in a unified manner. In particular, we address the issue of formal, integrated, and longitudinal reasoning: inferentially-modeled behavior that incorporates a fairly wide variety of types of commonsense reasoning within the context of a single extended episode of activity requiring keeping track of ongoing progress, and altering plans and beliefs accordingly. Instead of aiming at optimal solutions to isolated, well-specified and temporally narrow problems, we focus on satisficing solutions to under-specified and temporally-extended problems, much closer to real-world needs. We believe that such a focus is required for AI to arrive at truly intelligent mechanisms with the ability to behave effectively over considerably longer time periods and range of circumstances than is common in AI today. While this will surely lead to less elegant formalisms, it also surely is requisite if AI is to get fully out of the blocks-world and into the real world.

1.1 Motivation

Active logics (originally known as "step logics") were introduced in order to achieve a number of related goals; over time we discovered yet further advantages, changing the name to "active logic". Chief among the original goals were more realistic representation and reasoning about time, inconsistency, and the reasoning process itself. These goals remain central in work we report here; in addition we examine issues involving language, multi-agent cooperation, and others.

We start by presenting a fairly abstract definition of an active logic. Such a logic is closely linked to an inference engine; the notion of theorem is woven together with the notion of the current (evolving) time production of the theorem.

An active logic consists of a formal language (typically first-order) and inference rules, such that the application of a rule depends not only on what formulas have (or have not) been proven so far (this is also true of static logics; see below) but also on what formulas are in the "current" belief set. Not every previously proven formula need be current; in general the current beliefs are only a subset of all formulas proven so far: each is believed when first proven but some may subsequently have been rejected.

We will examine this definition in considerable detail later. Related work will be discussed at length in a separate section; here we simply call attention to work of Russell and Subramanian [127], who discuss how one might incorporate computational limits into machine rationality concepts, and to Wellman [139], who notes that any formalization of a rational agent must necessarily idealize away the computational process itself. Active logic, however, is an attempt to bring together these two conflicting goals.

1.1.1 A formal distinction

Most formal logics or theories are what we call "static;" they consist of prescriptions defining the set of theorems (or entailments), but do not provide for that set to evolve over time. To be slightly more precise, a static logic is one whose theorem-set is fixed, independent of any actual manner of producing those theorems, or even of whether or not any theorems are in fact "produced." In particular the theorem-set of a static logic cannot in any meaningful way be said to change; it has one and only one set of theorems. Even when a static logic does provide proof-theoretic inference rules for producing theorems in a step-like process (a proof tree), nothing in the syntax or theorem set reflects a changing state as that process occurs. Indeed, concerns for the process of producing theorems is generally regarded as a task for an "inference engine" rather than for a logic *per se*. There is a division of labor, in the traditional view, between theoremhood or entailment (labors of logic) and theorem-production or model-building (labors of an engine). An engine is an implementational matter with implementational concerns: Some are slow, some fast; some use resolution, and some do not, etc. Indeed, many highly distinct engines may implement the same (static) logic.

As we shall argue below, in commonsense reasoning, this division of labor is inappropriate—many commonsense "theorems" depend crucially on their manner (e.g., time) of production (in order to be theorems). As a case in point, the assertion "I've been working on this chore for 15 minutes now, and still have not finished it (maybe I should give up)" can reasonably be concluded only in the context of an immediately preceding period of relevant effort. Or: "From the two previous observations the proposition P follows." Or: "Here I am looking up Matt's phone number and Matt is coming in the door (so I can put away the phone book)." In general, explanations or commentaries on one's activities often require temporal embedding of and *within* the reasoning process. Moreover, such explanations or commentaries are not mere icing on the cake—much depends on our ability to explicitly express facts about what we are currently doing. This paper aims to (i) present the thesis that such reasoning requires more than traditional meta- and temporal-reasoning, (ii) draw conclusions about underlying mechanisms relevant to the enterprise of formalizing such activity, (iii) illustrate an array of technical formal logics that we have developed toward such an enterprise, and (iv) bring together a number of distinct application areas within the context of a single unifying framework.

As Ginsberg points out [40], Doyle's seminal paper on truth maintenance [14] presented reasoning as a process of revision as well as inference, and yet the revision feature was largely ignored in later formal work, while nonmonotonicity became almost exclusively attached to the idea of theory-comparison. More recently Gärdenfors and others (e.g. [34]) initiated a new formal look at belief revision, which however treats this as distinct from the inference process. Namely, that treatment assumes that a reasoner has the logical closure of its beliefs in hand, and which it may revise in light of new observation. But we argue that inference itself provides new information; indeed, the inference that Q, from P and P implies Q, can be regarded as the observation that Q, not from perceptual organs but rather from inferential organs. In Gärdenfors' terminology, this might be a case of theory expansion, where the theory however is partial, its theorems being gradually built up over time, much as perceptual observations come in over time. See [113, 112] for a more detailed discussion.

If we follow this idea, that inference itself is a form of belief revision (or expansion or contraction), then a number of new behaviors become possible. This paper is an account of these behaviors as we have studied them to date. One interesting feature, also clearly a part of human reasoning, is that what is plausible at one moment may no longer be so given more time to reason. Pollock [116] calls this behavior *diachronically defeasible*, i.e."a proposition can be justified at one stage of reasoning and unjustified at a later stage, without any additional perceptual input." Active logic allows for diachronically defeasible reasoning, as in the Brother Problem (see Section 3) and also the projection problem (see Section 8). Indeed, since an active logic always spreads its conclusions out in time (the time taken to prove them) then diachronic defeasibility is just one aspect of the larger fact of diachronic belief-change. This is the antithesis of logical omniscience: in real life, as in active logics, reasoning takes time, reasons and results come and go, often due to nothing more than the passage of time and mental effort.[1]

The result of allowing diachronic defeasibility, and more generally of uniting inference and observation in one formal framework, is to bring together again two themes of Doyle: NMR and belief-revision become restored to the role of working together rather as separate modules. Indeed, we have brought them even closer together than Doyle, since we do not view them as passing messages back and forth, so much as going on together hand-in-hand.

This paper is both a plea for highly integrated and general purpose (computational) intelligence, and an extended illustration of what such an intelligence might look like, or at least what one such effort-in-progress currently looks like. It is our hope that such integrated and general purpose architectures will lend themselves to both theoretical and empirical studies, providing a much missed link that many, including ourselves, worry is holding our science back.

We envision a highly integrated reasoning-acting system, based on a formal logic with a clear semantics, but also equal to the task of actually getting around in the world. It will of course have many procedural aspects, as it needs to *do* things, in addition to manipulating symbols. It also will not simply perform a stipulated task and grind to a halt; it will have a lifetime of its own, and need to recall and reason about its own past, modify its goals as unforeseen circumstances dictate, and correct errors. We think that *active logics* may provide a suitable framework for such a system.

In order to give a sharper motivation and focus to this long-term research project, we present a scenario about an imaginary idealized commonsense reasoning agent, Agenta, designed to illustrate the kind of behavior we think essential to the AI dream of truly intelligent systems, and to which we think active logics are well suited. To date active logics have

[1] A better phrase than "omniscience" for our purposes, might be "prescience": over-idealized omniscient agents have prior knowledge of all the future conclusions that a real agent might draw. We have in previous work also referred to such reasoning as "final-tray" reasoning, in which there is an actual limiting state of reasoning that contains all the beliefs implicit in given axioms. The agents we study have no such end-state: they are always in a state of reasoning, with time ahead for more.

been successfully applied to most of the technical issues in this scenario, but not yet fully integrated into a single system of axioms, rules of inference, and procedures. Such integration is in itself a major undertaking, but we think that many of the essential knowledge representation issues have been solved.

We wish to motivate formal work toward such integrated systems and to make a plea for "lifetime" studies rather than isolated problem-solving or planning investigations. We think that lifetime-studies (i) are essential to the long-range success of AI, (ii) present new issues not addressed in one-shot studies, and (iii) also provide more robust solutions to some one-shot problems due to the broader availability of information over the course of repeated efforts. In particular, we propose *case-based learning*, in the form of trial-and-error and advice-taking,[2] as a suitable and essential aspect of CSR.

Specific capabilities illustrated in the scenario which follows, and at least partially solved in isolation by existing active-logic formalisms, are:[3]

- keep temporal indexicals up to date during reasoning (35)
- re-establish communication using a focal algorithm (19–20)
- reason in real time about others' evolving knowledge and reasoning (15–19)
- perform deadline-coupled real-time planning (18–19,33–36)
- reason effectively in the presence of contradictions (7–8,13–14)
- find sensible explanations for past events (27–33)
- change of terminology (23–25)
- enlargement of language (31–32)
- learning (7–8,12–13)

1.1.2 One day in the life of Agenta

In [94] Millgram argues for a design of an agent that recovers gracefully when faced with unexpected situations—old goals must be abandoned in favor of new ones which allow the agent to reformulate a plan of action appropriate for the newly arisen situation. We claim that active logic serves well to model such a robust, highly adaptive agent. Active logic not only permits inconsistencies (which are bound to arise when things don't go quite as planned), but allows the course of reasoning to change as new information becomes available (whether through direct observation or through inference). We demonstrate this flow of reasoning in the following scenario.

The scenario (adapted from [113, 112]) is an example of what we consider an *episode* of commonsense reasoning. It is extended in time and involves a single overall theme but allows many changes and on-the-fly aspects as plans and actions unfold. A single conclusion in not sufficient to "solve" the problem. What is required is an initial plan that is updated as time goes on. In order not to be too overambitious, we keep irrelevant distractions to a minimum, while nevertheless staying close to real world complexities that directly involve reasoning. It is distinguished from a "burst" of reasoning described in Section 1.1.3 below, which appears to be the standard model of formal commonsense reasoning. Unlike what we are proposing, these bursts are performed in a highly idealized setting without benefit (or detriment) of real-world interactions that may require rethinking of those idealizations.

`Agenta and her friend Sue have the task of painting a barn today. They make the following plan: Agenta will buy the paint and brushes and Sue will buy the ladder. Sue will go to Main Street Hardware, which has a bargain on ladders, even though it is farther away than the Ellis Street paint store which also carries ladders. They expect to meet at the barn at noon and begin painting. They set off in opposite directions. Agenta heads off to the paint store to make her purchase. At a bridge she intends to cross, she notices a sign saying the bridge`

[2] Long advocated by McCarthy [80] but paid little heed; see also the review by Giunchiglia [41]

[3] Numbers in parentheses refer to the line numbers of the scenario in Section 1.1.2 in which that capability is illustrated; we chose only some of the most obvious cases for each.

is under repair and uncrossable. This forces her to back up a considerable distance and take an alternate and much less direct route. As she goes, she realizes they will need two ladders and a plank, and that she should get a second ladder and a plank at the paint store.

Agenta then sees another sign saying that all Main Street stores are closed for the day due to a water main outage, and she assumes Sue will now try to go to the more expensive paint store to get a ladder. She also assumes Sue does not know about the bridge being out, since else she surely would have warned her. Agenta reasons that it will take Sue at least three extra hours to get to the bridge, notice the sign, back up to the alternate route to get to the paint store, buy the ladder, and bring it to the barn. This will make it too late to paint the barn today. She decides to purchase the paint and then wait for Sue at the barn anyway, as the only obvious place to meet up with her and replan for tomorrow; but to forgo getting a second ladder and a plank until they can meet to work out a new plan, perhaps purchasing both ladders and plank tomorrow at Main Street Hardware.

Agenta arrives at the paint store but the street sign reads ``Ellis Avenue'' rather than ``Ellis Street''---she decides it must be the right place since the paint store is right there and ``Avenue'' and ``Street'' are easy to switch, and in any case she can get paint and brushes there.

Agenta arrives back at the barn at 12:20, and finds a message from Sue, marked 11:45, saying that she has placed the ladder in the barn, and will return by 12:30 to start painting. She is puzzled by the apparent contradiction, then reasons that she must have found the Main Street hardware store open after all. Then she sees that the ladder has a tag on it reading ``Main Steet Hardware II, Harwood Lane location.'' She does not know how Sue found out about the second location of the store, but evidently she did. Since Harwood Lane is very close, she decides to make a quick trip there for a second ladder and plank. Then she reasons that she should wait instead, since it is by now 12:25 and Sue will be here at any moment.

1.1.3 Agenta, Active logics, and Non-monotonic Reasoning

Let us clarify one underlying assumption at the outset: we are interested in logics whose theorems or entailments are considered to be *beliefs of an idealized commonsense reasoner* (whom we will continue to call Agenta). We will call this matching of theorems and beliefs the "directness" hypothesis; it serves to eliminate from consideration meta-theories whose theorems encode assertions *about* Agenta's beliefs (see [62]) but are not themselves candidates for those beliefs. Thus we are critiquing the *direct* use of static logics for commonsense reasoning.

While Agenta may differ in important ways from a real-world reasoner, nevertheless she (or the logic) is expected to have as theorems the "desired" formulas that encode a "solution" to whatever commonsense problem is being analyzed, and is not to have the negations of those formulas. All this we think is well within the standard tradition in formal AI research.

Now clearly a real reasoner must have, in addition perhaps to some sort of logic, an engine to actually produce theorems over time. Thus the actual belief-set of a real reasoner changes over time (it grows, and also may shrink if former beliefs are rejected). The traditional idealization for a reasoner like Agenta leaves out these details: all that is found is the "final" belief set after all the reasoning has been carried out.

Curiously, research in nonmonotonic reasoning (NMR) is often presented as aimed at characterizing just such changes in belief-sets, yet it does no such thing.[4] Nearly all investigations in NMR utilize static logics. What is characterized is a relation between two static theories, T_1 and T_2, where we may suppose T_1 to have been Agenta's theory initially and T_2 a

[4] Reiter[119] and McDermott and Doyle [88] give such a characterization, for example. More recently in [1, 34] attention has swung to belief-set change, but still not to the *process* of such change, i.e. this is still a study of relationships between static theories. See Section 1.2.5 for more discussion on this.

subsequent theory held by Agenta. Just how it is that Agenta gives up T_1 and adopts T_2 is left up in the air, except to say that something changed her beliefs. We are asked to suppose that at least one new belief is handed Agenta, and this apparently causes the rest of the change. Thus Agenta undergoes "bursts" of reasoning, each characterized by a beginning and an end. First Agenta has beliefs (corresponding to) T_1, then something happens, and eventually Agenta ends up with T_2.

Now it certainly is worthwhile to have such a characterization of a beginning and an endpoint for a burst of commonsense reasoning. For one thing, it gives us a clearer sense of what sort of behavior we are looking for in an intelligent system: at the very least it gives us some "boundary conditions" on Agenta's behavior. But once we have this, we then need to see how such a burst can come about, i.e., what the underlying behavior is. This is what has usually been thought of as an implementation issue, and what we shall argue is anything but that.

The first active logics studied were step-logics (see [22, 27]).[5] While one goal was to make formal commonsense reasoning more realistic (e.g., respecting temporal limitations) this was by no means the only goal; certain kinds of problems do not appear to admit of static representation, let alone static solution.

In most active logics studied to date, there is a formal notion of "Now" that determines what is current, and that in turn is determined by a "clock" rule that between times t and $t+1$ changes the "current" theorem $Now(t)$ into $Now(t+1)$. Thus the clock (rule) takes one unit of time to fire, and this fact itself is recorded syntactically as a change in the theorem (belief) set: the "old" belief $Now(t)$ is erased and the "new" belief $Now(t+1)$ replaces it. In effect, new information regularly comes into the logic, in the form of clock-readings (among other possibilities).

Thus in effect an active logic has time-sensitive inference rules and consequently time-sensitive inferences. This can be neatly illustrated in the following inference rule:

```
  Now (i)
-----------
 Now (i+1)
```

In an English gloss, this reads: If the time is now i, then now it is i+1.[6] A more dramatic example is that of reasoning that it is now time for some event to happen; e.g., when it is noon, eat lunch.

```
Now(noon) implies do(eat).
```

This will produce the conclusion do(eat) only when it has been inferred that Now(noon), which will not occur until it is noon, i.e., until, say, Now(noon) has been proven, which takes, nicely enough, noon = (12 - 8) * 60 * 60 steps from when it was Now(0).[7]

1.2 Related work

In comparing active logics to other work, we have had to decide which aspects of active logics are to be viewed as basic design features and aims, and which are better seen as applications. This is so, because active logics are intended as a general-purpose framework for commonsense reasoning, and as such are potentially applicable to most AI problems. However, this is not to say that the active logic framework on its own does the best job at any given application; this will depend on details of how the logic is tailored, what knowledge is provided, and so on. So, instead of comparing active logics to, say, planning systems, we view planning as an application, described in another section (with brief mention there of other work on planning and published comparisons with active logics).

[5] Since their introduction, step-logics have been extended and renamed as *active logics* to allow several new features, including limited short-term memory (see Section 10), and the introduction of new expressions into the language over time (see Section 6).

[6] There is implicit here an assumption of an elementary unit of time corresponding to this elementary inference, perhaps a bit like a just-noticeable-difference in time, or the 100 milliseconds often associated with simple cognitive processes. Nothing essential turns on this assumption; it is made solely for formal simplicity. Indeed, we assume that all inference rules in the active logics studied to date take the same time to fire, again for simplicity. Future work includes the relaxation of this requirement.

[7] If we assume the robot wakes up at 8 am, and that seconds are its primitive time units. Here "do" has both a truth-value semantics (it is time-to-do) and an action semantics (event commands are being issued). We do not currently have a formal treatment of action semantics; we assume the appropriate events take place.

Here, we simply address what we regard as the basic features of active logics and how they differ from other approaches to reasoning. Chief among these are: limited reasoning, temporal reasoning, meta-reasoning, real-time planning, reasoning engines, indexicality, and belief revision.

1.2.1 Limited Reasoning

Doyle [17] discusses the complementary roles that the economics theory of rationality and mathematical logic play in the development of rational automated agents. In particular Doyle explores some major limitations of agents that influence their rationality. Active logic was developed to overcome such limitations of automated agents. The two main types of limitations that Doyle describes are the following:

Information limitations: Doyle indicates that the information available to a reasoner may be incomplete, inconsistent, intermidate or difficult to change. The reasoner's information may also be wrong. Active logic provides a formal framework to cope with such limitations. In active logic, at each time step, the reasoning agent has a finite set of beliefs, and this set can increase, either through observations or by inferences. As we describe in Sections 3 and 5, active logic can recover from inconsistencies, and the relevant beliefs of the agent may change over time. An agent may drop a belief, if it comes to the conclusion that it is wrong, and it may even change its language (see Section 6). The issue of context sensitivity that is mentioned by Doyle, is demonstrated in Section 8 where we discuss deadline planning in active logic. There we show the way an agent can reason in the context of different plans it develops simultaneously.

Resource limitation: Doyle mentions the most obvious limitations on resources which are time and memory availability. Active logics was developed to take into consideration the time that passes during the reasoning. The time of the reasoning can be used in the reasoning process itself.

The purpose of the development of mechanisms for short term memory and long term memory for active logics (Section 10) was to handle the limited memory problem. Further work is needed on these issues.

The literature contains a number of approaches to limited (non-omniscient) reasoning, apparently with similar motivation as our own. However, with very little exception, the idealization of a "final" state of reasoning is maintained, and the limitation amounts to a reduced set of consequences rather than an ever-changing set of tentative conclusions.

Konolige [61] suggests a deduction model that is deductively closed under a set of rules, though not necessarily *consequentially* closed, since the rules may not be logically complete. The assumption of deductive closure greatly simplifies the technical problems by disregarding any particular control strategy. It is suggested that systems that are not deductively closed can be modeled by employing a *low cost bound* on derivations, thereby deriving only those proof trees whose depth is bounded; this extension is not part of the model. The model ignores the time element present in the inferencing, assuming that the agent can perform the necessary computations in a time interval which is relatively short with respect to its ability to act.

In contrast, we are concerned with limited rationality of a time-situated nature. There may or may not be a limit to the amount of time the agent can spend on reasoning, but at any given time the agent has obviously had only up to that time to reason. It is what the agent has actually been able to conclude at any particular time, that we seek to model.

Active logic gives an account of one's explicit set of beliefs. Most of the formalizations of reasoning in the literature deal only with one's implicit set of beliefs. Levesque [74], however, does distinguish these two, giving an intuitively plausible semantic account of both *implicit* and *explicit* beliefs. An agent's implicit beliefs include all valid formulas, his explicit beliefs, and the logical consequences of his explicit beliefs. His explicit beliefs, on the other hand, are closed under a much weaker set of conditions. An agent does not necessarily explicitly believe all valid formulas, nor does it necessarily explicitly believe β, simply because it explicitly believes α and $\alpha \rightarrow \beta$. Using the set of explicit beliefs, Levesque is able to describe an agent who is not logically omniscient.

Levesque's logic, however, does not allow meta-reasoning about one's own beliefs or reasoning about other agents' beliefs. These abilities are needed in many situations, including planning and goal-directed behavior, where one may have to reason about the knowledge that one has as well as the knowledge that others may possess. Fagin and Halpern ([28]) extend Levesque's notion of implicit and explicit beliefs to allow for multiple agents and beliefs of beliefs.

In another related approach, Lakemeyer [69] presents a logic to model the beliefs of a limited reasoner. His model combines features from both possible-world semantics and relevance semantics. The main difference between active logic and Lakemeyer's logic is that his logic is *not* intended to be used by the limited agent itself; instead the logic provides an external view which specifies the reasoning ability of a deductively limited agent. Active logic, on the other hand, is intended to be used by the limited agent itself.

A very recent treatment of omniscience is given by Fagin, Halpern, and Vardi [30]. They use a possible worlds approach using a nonstandard propositional logic and show that this characterizes the reasoning studied in the above works of Lakemeyer and of Levesque.

Unlike active logic, in all these cases, the steps of reasoning are not made explicit, and the limitations are not necessarily ones of actual computational restrictions (time or space).

Fagin and Halpern [28] propose a logic of awareness (again with explicit and implicit beliefs) which is based on the idea that one cannot have beliefs about something of which one has no knowledge. Intuitively, given a primitive proposition p, if the agent explicitly believes $p \vee \neg p$ then the agent is aware of p. As in [74]'s logic, an agent's implicit beliefs include all valid formulas and all the logical consequences of his implicit beliefs. The explicit beliefs, on the other hand, are generated by awareness of primitive propositions. As in [74], the explicit beliefs do not necessarily include all valid formulas but, unlike [74], *are* closed under implication. Our own approach provides a rather different notion of awareness, where the agent is aware of all closed sub-formulas of its beliefs; hence the awareness set changes over time. Goodwin [43] comes a little closer to meeting our desiderata but still maintains a largely final-tray-like perspective.

In the same paper, Fagin and Halpern extend their logic of awareness to include awareness of arbitrary formulas (not just primitive propositions). In addition to the operators for implicit and explicit belief (L_i and B_i, respectively), an operator for awareness, A_i, is introduced. An agent explicitly believes a formula α if he implicitly believes α and he is aware of α (that is, $B_i\alpha \longleftrightarrow L_i\alpha \wedge A_i\alpha$). As in the previous logic, an agent does not explicitly believe all valid formulas; however, unlike the previous logic, an agent's explicit beliefs are not necessarily closed under implication. Thus it is possible for an agent to explicitly believe p and $p \rightarrow q$ without explicitly believing q. The intuitive explanation given for this is that the agent is not aware of q.[8] It is interesting to note that L_i acts like the classical belief operator, so that, for instance, if one assumes that the agents are aware of all formulas, the logic reduces to the classical logic of belief, weak S5 (see [9]).

Finally, Fagin and Halpern [28] also present a logic of local reasoning which allows agents to hold inconsistent beliefs. It is based on the fact that humans don't focus on all issues simultaneously. Thus one can view a reasoning agent as a society of minds, each with its own set of beliefs. Unlike the previous logics that [28] propose, in this logic there is not necessarily only one set of states that an agent thinks possible, but rather many sets, each one corresponding to a different set of beliefs. That is, each set represents the "worlds" the agents thinks are possible in a given frame of mind, when focusing on a certain set of issues. It is then possible for conclusions that are drawn in one world to be inconsistent with conclusions drawn in another.

Another way to view limited reasoning is in terms of what conclusions the reasoner has come to at any moment when the reasoning may be stopped, say by an interrupt. This is close in spirit to active logics, where however we do not envision interruption but rather simply ongoing reasoning for the lifetime of the reasoner. Nevertheless, it is instructive to make comparison with the notion of anytime computation,[9] introduced by Frisch and Haddawy in [51]; in [33] they present the idea of anytime deduction procedures that can return partial information even before a complete proof is found, by exploiting the capacity of any multi-valued logic to express intermediate results. Before the ultimate value, or value interval, of the target sentence is computed, it is possible to have an intermediate result stating that some truth values have been eliminated. Active logic is a two valued logic. There is no intermediate step regardless of the truth of a specific formula. However, the beliefs in a given time step are only partial. When the agent spends more time on its reasoning, its beliefs become more accurate. These partial sets of beliefs may be used for deciding how to act, as demonstrated in Section 8.

Although these approaches all model *limited* reasoning, the process is still in terms of the standard mold of *static* reasoning. We do indeed have a restricted view of what counts as a theorem, but the logic is still final-tray-like. Although the final tray is smaller than in the conventional *omniscient* approach (it is catching less, if you will), it is still only the *final* set of consequences that are evident. In Fagin and Halpern's logic of general awareness ([28]), for example, $\alpha, \alpha \rightarrow \beta, \alpha \rightarrow \gamma$ and γ may all appear in the tray without β, given that the agent is unaware of β. Although the tray is catching less here, the over-simplification of a "final" state of reasoning is nonetheless maintained. All the conclusions are still drawn instantaneously. The effort involved in actually performing the deductions is not taken into consideration.

Given that time plays such an important role in this discussion, we next examine related work in temporal logic.

[8] It seems a little odd to say that we can be aware of $p \rightarrow q$ (which the agent must, since he explicitly believes $p \rightarrow q$) without being aware of q. The notion of awareness which is built into active logic does not allow this peculiarity.

[9] See also our discussion below of anytime algorithms applied to planning.

1.2.2 Temporal Logic

Formal reasoning about time and action is not new. A great deal of research has been devoted to this field. Perhaps the two most influential temporal formalisms are those of Allen ([2]) and McDermott ([87]).

In [2], a logic which permits reasoning about time is developed. In it time intervals are the principal objects in the domain. Three basic entities are associated with time: properties, events, and processes. $HOLD(p, i)$, where p is a property type (e.g. red) and i is an interval of time, is used to denote the fact that property p holds for the interval i. A property is true for an interval iff it holds for every subinterval. The fact that an event e occurred over an interval i is denoted by $OCCUR(e, i)$. Finally, $OCCURRING(p, i)$ denotes the fact that process p occurred over interval i. A process is said to occur over an interval i iff it occurs over some subinterval of i. Having set up a way to handle temporal information, [2] then proceeds to handle actions, causation, intentions, and plans.

McDermott [87] constructs his theory using *fact types* and *event types*. Unlike Allen, McDermott uses time *points* as primitive. $T(t, p)$ denotes the fact that fact type p holds at time t. $OCC(t_1, t_2, e)$ denotes the fact that event type e occurred over the interval $< t_1, t_2 >$. [87] then uses these primitives to reason about temporal information and events.

Many others have contributed formalisms for reasoning about time and action including [50, 52, 79, 89, 98, 76, 131], and most recently [44]. In contrast to these theories, the focus of active logic is not primarily to be able to reason *about* time, but rather to be able to reason *in* time. That is, active logic provides a time-situated view of reasoning, where the fact that the reasoning process itself takes time is part of that very same reasoning.

This means, in particular, that active logics reason about their own reasoning, e.g., about the time being taken by their own reasoning. This is then meta-reasoning; however, it differs from other treatments of meta-reasoning, in that there is not a second level of representation for meta-beliefs about beliefs. We now turn to a review of related work on meta-reasoning, and to one of the key applications of active logics: deadline-situated planning.

1.2.3 Meta-reasoning and Real-time Planning

Some of the motivations behind active logics, that take the time of reasoning into considerations, is to enable the agent to reason and act in real-time. For example, an agent may need to finish to perform some actions till some deadlines. An excellent survey of related research in deliberative real-time AI is available in [35]. They categorize real-time systems into *purely reactive* (those that hard-wire reactions completely), *combined response* systems (those that have distinct asynchronous components that handle deliberation and reaction) and *integrated* systems (those that have a single architecture that is capable of a wide range of timely responses depending upon the time criticality requirements).

Those in the last category put the time that is available in the best use. These approaches have been collectively characterized by terms such as *flexible computation* [57], *deliberation scheduling* [5], and *anytime algorithms* [12, 141]. They spend the resources available to the agent in deciding whether to act, how to act, and when to act. The main differences between our approach and these is the following:

(1) they do not account for the time-cost of the *deliberation scheduling algorithms* themselves, only for the cost of deliberation that they consider; while our mechanism is completely situated in time;

(2) they require prior complex (meta) knowledge about their reasoning algorithms or procedures themselves, and their characteristics with respect to time; they also require a great deal of knowledge about the domain in the form of probabilities of events and expected utilities of actions that the agent must be aware of;

(3) they usually attempt to solve an *optimization* problem in a specific domain, whereas our approach is to come up with a formalism that accounts for all the time spent by the agent on its reasoning.

Thus, we note that these approaches are not *alternatives* to our, but rather that they are suited for a different range of more informed problem solving.

Meta-reasoning has become almost synonymous with nonmonotonic reasoning. This is largely due to the fact that nonmonotonic formalisms always rely on some assessment of what a reasoner does not know (or cannot prove), this state of ignorance itself being a piece of meta-information about the reasoner. Sometimes this ignorance is explicit (as in McDermott and Doyle's approach) and sometimes implicit (as in Reiter's and McCarthy's approaches). However, in general, meta-reasoning is a broader notion, involving introspective beliefs in general, whether defeasible or not. For instance, positive introspection (if I know α then I also know that I know: $Know(I, \alpha)$ is in my knowledge base, or at least in the logical closure of my knowledge base).

Active logic explicitly represents such an introspection predicate, $Know$, which however, has a time parameter (and which suppresses the "I" as understood). This is, among other things, the means by which active logic endows its reasoner with a history. If the belief α occurs at step (time) t, then at later steps the belief $Know(t, \alpha)$ can be present, recording the fact that α had been a belief at step t. We say "can be present" because this is not required as part of the definition of active logic; we have found it (or its negative counterpart: infer $\neg Know(I, \alpha)$ when α is not a step-t belief) useful in many cases, but it also leads to an overabundance of unnecessary introspective conclusions.[10] To offset this, we are exploring limited-memory versions of active logic (see Section 10); see also our discussion of the RABIT system in below.

This issue brings up squarely the fact that an active logic is a logic engine as well as a formalism; it is both at once, since the formalism has an indexical (Now) whose meaning is tied to actual inferences being performed over time. We thus take a look at implemented reasoning systems.

1.2.4 Implemented Reasoning Systems

An active logic is both a formalism and an inference engine. This is not to say that we use the phrase "active" logic in two different ways, however. Rather, as an *embedded* or *time-situated* formalism, an active logic has a proof-generation algorithm specified within the formalism itself. Thus an active logic contains its own semantics, at least as far as time goes (or, at least as far as the *current*—and changing—time goes).

Thus it is appropriate to compare active logic both to (traditional) formalisms (which in general are not also inference engines, although sometimes inference engines can be or have been implemented that do produce the theorems of those logics) and also to implemented reasoning systems. In regard to the latter, we will look at the systems known as CYC [71], OSCAR [117, 115], SOAR [68], and RABIT [36, 37].

CYC is a program under construction, that has been undertaken in an attempt to capture a vast amount of commonsense information, on the order of millions of facts, and yet reason commonsensically with this as well. This includes contextual and defeasible data. As such it falls within the broad intentions of active logic as well; however, to the best of our knowledge, CYC is not time-situated, and for instance cannot reason to the conclusion that now, after an hour of working on a problem, it is time to draw a tentative conclusion (as would be needed, for example, in problem-solving with a deadline).

OSCAR is an interest-driven nonmonotonic inference engine, and as is the case with active logics, it can also be viewed as a formalism. Again as with CYC, however, OSCAR is not time-situated. On the other hand, OSCAR allows for diachronic defeasibility, as does active logic.

SOAR is largely a learning algorithm, with strong human/cognitive modeling aspects. It has in common with active logic the maintenance of a history, specifically used by SOAR for inductive learning. SOAR does not, however, reformulate past beliefs based on new information, as does active logic (see the Rosalie's Car problem in Section 6). See [48] for a more-detailed comparison of SOAR and active logic.

RABIT is an implemented reasoning system based loosely on the active logic formalism. Similar to the active logic enhancements to deal with issues of limited space (discussed in Section 10), RABIT's two main memory modules are STM (Short Term Memory), representing the current focus of attention, and LTM (Long Term Memory), representing the full set of beliefs. Recent enhancements to RABIT have included a re-design of LTM into a network of concept and belief nodes, to allow for a spreading activation (marker passing) approach to associative retrieval of beliefs from LTM to STM. With the marker passing mechanism, RABIT is able to deal effectively with large[11], potentially incomplete and inconsistent knowledge bases.

1.2.5 Other related work

Chapter 10 of [29] discusses their generalization of active logics, which allows for the possibility of fairly arbitrary relationships between "stages" of reasoning. The generality is such that very little can be said about specific inferences without making further assumptions leading to a more concrete version (active logic or other), however. In particular, an indexical time-passage aspect is not required.

[10] That is, the belief set remains finite, which is good, but may grow exponentially, which is bad.

[11] on the order of tens of thousands of nodes

Doyle's truth maintenance notion was the first serious effort at belief revision; curiously, however, belief revision was largely ignored in the flurry of work on nonmonotonicity, until the recent work by Alchourron, Gardenfors, and Makinson (often referred to as the AGM approach) [1, 34] and especially showing a strong connection between NMR and belief revision. The AGM treatment focuses on theory change rather than simply belief change. That is, a theory is viewed as having a fixed set of beliefs (axioms, theorems); this is of course the traditional notion of theory in mathematical logic. Beliefs change by means of a change in axioms, notably via new incoming information that christens a new theory; some old axioms (and theorems) may be lost in the reshuffle, whence the nonmonotonicity. But the process of adjudicating between new and old, and of reasoning in general, is not addressed, and there is no associated time-situated aspect.

Active logics, by contrast, are constantly changing as part of their very design, and this change can be thought of as simple theorem-proving or as altering old beliefs in the face of new information—it is the same uniform mechanism of deciding based an all available information (which itself is in flux) what to believe *now*. There is no notion of replacing one theory with another, so much as a single constantly evolving theory. One consequence is that incoming information need not be given special status: a new belief can become rejected based on older ones; this is not allowed in the AGM treatment. More to the point, there is no information in an active logic that is beyond question. In active logics, belief revision is not an occasional activity done to reset things to normal so inference can proceed again; rather belief revision *is* inference. Meta-reasoning and "ordinary" reasoning are not distinguished in active logics; beliefs are both used and mentioned, and when mentioned they are simply objects like any other. Thus active logics are a bit like set theory where everything is a set. The mechanism for treating beliefs as objects is quotation, which forms from the belief α at step i, the new belief at i+1: $Know(i, "\alpha")$, where $"\alpha"$ is treated like any other object.

In [72] Lespérance and Levesque present a logical theory of knowledge and action that distinguishes indexical and objective knowledge. They formalize the notions of indexical knowledge, time, action, and ability, and provide a formal semantics for the knowledge operator. Although the logic has a notion of "now", this notion does not evolve as the reasoning progresses, making the logic incapable of addressing a key feature of active logic, namely, reasoning about the *process* of reasoning. Another aspect of Lespérance and Levesque's logic left unaddressed is the problem of logical omniscience.

In [122] Roos discusses reasoning with inconsistent knowledge, and presents a particular logic for this. His logic has in common with active logic (and in contrast with paraconsistent logics) that the establishment of a contradiction is taken seriously, in that certain premises are then retracted. However, the temporal element in Roos' logic does not play a central formal role as in active logic; in particular, there is no notion of an evolving Now. Also the basis for retraction in his logic is a reliability relation; whereas in active logic it is simply the contradiction itself, causing both contradictands to be no longer inherited. Which—if either—is later reinstated is not specified in the active logic framework itself but rather left to the particular application (a reliability notion is one we have used). So, the reasoner may never reach a reinstatement decision once the contradiction is noted, for example in the Nixon diamond. This we think is not inappropriate (depending on formal details and just how much world knowledge is represented).

It is worth noting that the unifying theme of the recent text by Russell and Norvig [126], is the concept of an intelligent agent. They view the AI problem as that of describing and building intelligent agents that can process inputs from the environment and perform actions accordingly. They take the position that "...different agent designs are appropriate depending on the nature of the task and environment." In this paper we present an architecture suitable for an intelligent agent that must reach satisfactory solutions to problems (goals) encountered in an incompletely specified, but temporally extended environment.

1.3 Preview

Having discussed background, motivation and related work we now move on to the specifics of active logics. In particular, Section 2 describes the preliminary active logic concepts, including formal definitions, examples of active logics, and a semantics. The subsequent sections then describe particular types of problems we have been able to solve using this formal approach. Section 3 describes the ease with which we can use active logics to do non-monotonic reasoning. In Section 4 we present an example of using active logics to reason about others' reasoning. Section 5 describes the ease with which active logics can reason in the midst of contradictions. In Section 6 we present several scenarios involving the ability of an active logic to deal with changes in the underlying language. Section 7 illustrates how active logics have been used for multi-agent coordination without communication through the use of focal points. In Section 8 we discuss the use of active logics to plan in situations involving deadlines. Section 9 demonstrates how an active logic can be used to solve the Yale Shooting Problem. Section 10 briefly describes our techniques for handling limited resources of time, space, and computation. We conclude in Section 11 with future directions for our research in active logics.

2 Preliminary concepts

$$
\begin{array}{llcl}
0: & & \emptyset & \\
\vdots & & & \\
i: & \ldots & \alpha & \ldots \\
i+1: & \ldots & \alpha \rightarrow \beta, \beta \rightarrow \gamma & \ldots \\
i+2: & \ldots & \beta & \ldots \\
i+3: & \ldots & \gamma & \ldots \\
\vdots & & &
\end{array}
$$

Figure 1: Active logic studies

An active logic is characterized by a language, observations and inference rules. A *step* is defined as a fundamental unit of inference time. Beliefs are parameterized by the time taken for their inference, and these time parameters can themselves play a role in the specification of the inference rules and axioms. The most obvious way time parameters can enter is via the expression $Now(i)$, indicating the time is now i. Observations are inputs from the external world, and may arise at any step i. In many of our examples, these observations take the form of domain axioms. When an observation appears, it is considered a belief in the same time-step. Each step of reasoning advances i by 1. At each new step i, the only information available to the agent upon which to base his further reasoning is a snap-shot of his deduction process completed up to and including step $i-1$.

AL, our commonsense reasoning agent, stores his world knowledge in the form of a database of beliefs. These contain domain specific axioms. A number of inference rules constitute the inference engine. Among them may be rules such as *Modus Ponens* and rules to incorporate new observations into the knowledge base as well as rules specific to, for example, deadline-coupled planning, such as checking the feasibility of a partial plan or refining a partial plan. Figure 1, adapted from [22] illustrates four steps in an active logic with *Modus Ponens* ($\frac{i:\alpha, \alpha\rightarrow\beta}{i+1:\beta}$) as one of its inference rules. Note that once α and $\alpha \rightarrow \beta$ are among the beliefs, *modus ponens* is used to derive β. γ cannot be derived until the next step when *modus ponens* is used on β and $\beta \rightarrow \gamma$.

The following features of this framework relate and contrast it to conventional commonsense reasoning systems:[12]

Thinking takes time: Reasoning actions occur concurrently with AL's other physical actions and with the ticking of a clock. AL can not only keep track of the approaching deadline as he enacts his plan, but can treat other facets of planning (including plan formulation and its simultaneous or subsequent execution and feasibility analysis) as deadline-coupled. Related to this feature of active logics is the fact that there is no longer a final theorem set (no "final tray" of conclusions). Rather, theorems (beliefs) are proven (believed) at certain times and sometimes no longer believed at later times. Provability is time-relative and best thought of in terms of AL's ongoing lifetime of changing views of the world. This leads to the issue of contradictions below. Instead of being prescient, knowing in advance all their conclusions, our agents learn of them only by deriving them, and this does not happen all at once.

Lack of Omniscience: AL is not omniscient, i.e., his conclusions are not the logical closure of his knowledge at any instant, but rather only those consequences that he has been actually able to draw.[13]

Handling contradictions: Consider Fermat's Last Theorem (FLT). Suppose AL believes FLT is true (after reading so in the New York Times). But (let us suppose) in fact FLT is false; then AL has contradictory beliefs, even though he is unaware of this. He has among his beliefs all the usual ones about elementary arithmetic, sufficient to disprove FLT,

[12] This description is necessarily very brief; for details see the various papers by Elgot-Drapkin et al.

[13] Konolige [62], Levesque [75] and Fagin and Halpern [28] proposed systems in which the agents are not omniscient. However, the inference time is not explicitly captured in their systems; and despite being non-omniscient, these systems are still *prescient*: whatever knowledge is derived from axioms is treated as simply there all at once, before it can have been actually derived.

even though he does not have the skills, inclination, or time to do so. Yet the (implicit) contradiction causes him no difficulties at all!

Since commonsense agents have a multitude of defeasible beliefs, they often encounter contradictions as more knowledge is obtained and default assumptions have to be withdrawn. While a contradiction completely throws an omniscient agent off track[14], the active-logic reasoner is not so affected. The agent only has a finite set of conclusions from his past computation, hence contradictions may be detected and resolved in the course of further reasoning.

Nonmonotonicity: Active logics are inherently nonmonotonic, in that further reasoning always leads to retraction of some prior beliefs. The most obvious one is $Now(i)$, which is believed at step i but not at step $i+1$. The nonmonotonic behavior enables the frame-default reasoning of which the commonsense agent must be capable [84].

There are two distinct types of formalisms in which we are interested: the meta-theory SL^n *about* an agent, and the agent-theory SL_n itself. Here n is simply an index serving to distinguish different versions of active logics. It is the latter, SL_n, that is to be step-like; the former, SL^n, is simply our assurance that we have been honest in describing what we mean by a particular agent's reasoning. Thus the meta-theory is to be a scientific theory subject to the usual strictures such as consistency and completeness. The agent theory, on the other hand, may be inconsistent and incomplete; indeed if the agent is an ordinary fallible reasoner it *will* be so. The two theories together form an active logic *pair*.

A notion of completeness for the meta-theory was defined in [22, 27] and is repeated here:

Definition 2.1 *A meta-theory SL^n is* analytically complete*, if for every positive integer i, and every constant α naming an agent wff of the corresponding agent-theory, either $SL^n \vdash K(i,\alpha)$ or $SL^n \vdash \neg K(i,\alpha)$.*[15]

In [18] we showed that our SL^0 formalism is in fact analytically complete. But what kind of completeness might be wanted for an *agent* theory? In SL_0, it is desirable that every tautology be (eventually) provable. This is the case, since every tautology has a proof in propositional logic and, for a sufficiently large value of i, all axioms (i.e., the "observations") in such a proof will have appeared (by design of SL_0) by step i. Thus SL_0 is complete with respect to the intended domain, namely, tautologies. However, for other active logics the case is not so simple, for the intended domain, namely, the commonsense world, has no well-understood precise definition. Nevertheless, we can isolate special cases in which certain meta-theorems are possible. In particular, if no non-logical axioms (beliefs) are given to an agent at step 0 (or any later time), then it is reasonable to expect the agent to remain consistent. This we are able to establish for all our agent logics in which the logical axioms do not contain the predicate symbol "Now". See Section 2.3.

2.1 Definitions

We repeat now certain key definitions from our formal development in [22, 27]. Most of the definitions are analogous to standard definitions from first-order logic; consequently certain results follow trivially from their first-order counterparts.

Intuitively, we view an agent as an inference mechanism that may be given external inputs or observations. Inferred wffs are called beliefs; these may include certain observations.

Let $\mathcal{L}$ be a first-order language, and let $\mathcal{W}$ be the set of wffs of $\mathcal{L}$.

Definition 2.2 *An* observation-function *is a function $OBS : \mathbf{N} \to \mathcal{P}(\mathcal{W})$, where $\mathcal{P}(\mathcal{W})$ is the powerset of $\mathcal{W}$, and where for each $i \in \mathbf{N}$, the set $OBS(i)$ is finite. If $\alpha \in OBS(i)$, then α is called an i-*observation.

Definition 2.3 *A* history *is a finite tuple of pairs of finite subsets of $\mathcal{W}$. $\mathcal{H}$ is the class of all histories.*

Definition 2.4 *An* inference-function *is a function $INF : \mathcal{H} \to \mathcal{P}(W)$, where for each $h \in \mathcal{H}$, $INF(h)$ is finite.*

Intuitively, a history is a conceivable temporal sequence of belief-set/observation-set pairs. The history is a *finite* tuple; it represents the temporal sequence up to a certain point in time. H consists of all conceivable histories, not merely those that

[14] We call this the *swamping problem*—namely that from a contradiction all wffs are concluded.

[15] K then has two roles: in SL^n as used here, and in SL_n. The context will make the role clear.

occur in some actual course of reasoning. The inference-function extends the temporal sequence of belief sets by one more step beyond the history. Figure 2 illustrates one such observation-function and inference-function. We can see that INF depends both on OBS and the histories, and that any given history depends both on OBS and INF. We have illustrated one such history: the history of the first five steps.[16] Definitions 2.5 and 2.6 formalize these concepts in terms of a step-logic SL.

Let

- $OBS(i) = \begin{cases} \{bird(x) \rightarrow flies(x)\} & \text{if } i = 1 \\ \{bird(tweety)\} & \text{if } i = 3 \\ \emptyset & \text{otherwise} \end{cases}$
- $Thm_i \subseteq \mathcal{W},\ 0 \leq i < n\,;\, Thm_0 = \emptyset$;
- $INF(<< Thm_0, OBS(1) >, \ldots, < Thm_{n-1}, OBS(n) >>) =$ $Thm_{n-1} \cup OBS(n) \cup \{\alpha(t) \mid (\exists\beta)(\beta(t), \beta(x) \rightarrow \alpha(x) \in (Thm_{n-1} \cup OBS(n)))\}$.

The history h of the first five steps then would be:

$$\begin{array}{rlcl} h = & << \quad \emptyset & ,\{bird(x) \rightarrow flies(x)\} & >, \\ & < \{bird(x) \rightarrow flies(x)\} & , \quad \emptyset & >, \\ & < \{bird(x) \rightarrow flies(x)\} & , \{bird(tweety)\} & >, \\ & < \{bird(x) \rightarrow flies(x), bird(tweety), flies(tweety)\}, & \emptyset & >, \\ & < \{bird(x) \rightarrow flies(x), bird(tweety), flies(tweety)\}, & \emptyset & >> \end{array}$$

Figure 2: Example of a particular OBS and INF

Definition 2.5 *An SL-*theory *over a language $\mathcal{L}$ is a triple, $< \mathcal{L}, OBS, INF >$, where $\mathcal{L}$ is a first-order language, OBS is an observation-function, and INF is an inference-function. We use the notation, $SL(OBS, INF)$, for such a theory (the language $\mathcal{L}$ is implicit in the definitions of OBS and INF). If we wish to consider a fixed INF but varied OBS, we write $SL(\cdot, INF)$.*

Let $SL(OBS, INF)$ be an SL-theory over $\mathcal{L}$.

Definition 2.6 *Let the set of 0-theorems, denoted Thm_0, be empty. For $i > 0$, let the set of i-theorems, denoted Thm_i, be $INF(< < Thm_0, OBS(1) >, < Thm_1, OBS(2) >, \ldots, < Thm_{i-1}, OBS(i) > >)$. We write $SL(OBS, INF) \vdash_i \alpha$ to mean α is an i-theorem of $SL(OBS, INF)$.*[17]

Definition 2.7 *Given a theory $SL(OBS, INF)$, a corresponding SL^n-*theory*, written $SL^n(OBS, INF)$, is a first-order theory having binary predicate symbol K,[18] numerals, and names for the wffs in $\mathcal{L}$, such that $SL^n(OBS, INF) \vdash K(i, \alpha)$ iff $SL(OBS, INF) \vdash_i \alpha$.*

Thus in $SL^n(OBS, INF)$, $K(i, \alpha)$ is intended to express that α is an i-theorem of $SL(OBS, INF)$.[19]

See Section 2.3 for the formal details of a semantics for active logic.

2.2 Example Active Logics

In this section we describe two examples of active logics: SL_0 and SL_7.

[16] This example serves to illustrate how these three concepts are inter-related. There are many possibilities for defining the functions OBS and INF; hence, many different histories are possible.

[17] Note the non-standard use of the turnstile here.

[18] We see that the predicate letter K has two roles: in SL^n and in SL. The context will make the role clear.

[19] In [20, 19] we used ${}^i\alpha$ for $K(i, \alpha)$.

2.2.1 SL_0

SL_0 has none of the three mechanisms of time, self-knowledge, and retraction. The language of the agent theory, SL_0, is propositional, with propositional letters $P_0, P_1, P_2, \ldots$. The meta-theory SL^0 is a first-order theory as described in Definition 2.7. SL_0 corresponds to the reasoning of a very simple agent that can deduce only tautologies. The agent is "fed" beliefs (its "observations") consisting of special tautologies, from which it is to draw others. In [18] we formalized the meta-theory SL^0 for describing the steps taken by such an agent.[20]

To have the agent deduce all tautologies, it is necessary to provide sufficiently many axioms. The usual approaches involve schemata encoding an infinite number of axioms (see [90]), yet we wish the agent to have only a finite number of beliefs at each step. To achieve this, we "feed in" first-order logical axioms little by little (according to increasing bounds on their lengths (i.e. the number of connectives) and ranges of symbols used) through the observation-function. That is, an instance α of an axiom schema is an i-observation iff the length of α and the highest index j of any propositional letter P_j in α are both less than i. For example, $P_0 \rightarrow (P_0 \rightarrow P_0)$ is a 3-theorem, but is not a 0-,1-, or 2-theorem. Although the highest index of this wff is zero, it has a length of two, and is therefore not "fed in" until step 3.

Of interest is the following theorem.

Theorem 2.1 *SL^0 is analytically complete.*

The proof is a long series of lemmas involving induction on the length of formulas. See [18] for the complete proof.

SL^0 was studied to gain an understanding of the underlying idea of active logics, and to gain some practical experience.[21] Although SL^0 was studied in some detail, SL_0 is not an appropriate active logic for commonsense reasoning: not only is the propositional language too weak, but an arbitrarily large number of tautologies are fed in at each step. A commonsense reasoner should have only a relatively small number of active beliefs with which to work at each step.[22]

2.2.2 SL_7

An SL_7 logic has all three of the mechanisms of time, self-knowledge, and retraction. SL_7[23], as stated earlier, is *not* intended in general to be consistent. If supplied *only* with logically valid wffs that do not contain the predicate Now, then indeed SL_7 will remain consistent over time: there will be no step i at which the conclusion set is inconsistent, for its rules of inference are sound (see Theorem 2.4 on page 20). However, virtually all the interesting applications of SL_7 involve providing the agent with some non-logical and potentially false axioms, thus opening the way to derivation of contradictions. This behavior is what we are interested in studying, in a way that avoids the swamping problem. The controlled growth of deductions in active logic provides a convenient tool for this, as we will see.

The language of SL_7 is first-order, having unary predicate symbol, Now, binary predicate symbol, K, and ternary predicate symbol, $Contra$, for time, knowledge, and contradiction, respectively. We write $Now(i)$ to mean the time is now i, and $K(i, \alpha)$ to mean α is known [24] at step i. $Contra(\{\alpha, \beta\}, i)$[25] is intended to mean that α and β are in direct contradiction (one is the negation of the other) and both are i-theorems.

The formulas that the agent has at step i (the i-theorems) are precisely all those that can be deduced from step $i-1$ using the applicable rules of inference. As previously stated, the agent is to have only a finite number of theorems (conclusions, beliefs, or simply wffs) at any given step. We write:

$$\begin{array}{rl} i: & \ldots, \alpha \\ i+1: & \ldots, \beta \end{array}$$

to mean that α is an i-theorem, and β is an $i+1$-theorem. There is no implicit assumption that α (or any other wff other than β) is present (or not present) at step $i+1$. The ellipsis simply indicates that there might be other wffs present. Wffs are

[20] Although there we did not yet use the notational distinction of SL_0 and SL^0.
[21] An implementation of SL^0 has been written in PROLOG, and was run on an IBM PC-AT.
[22] This failing of SL_0 can be seen in our implementation, where at a very early step so many theorems have accumulated that their computation on an IBM PC-AT is severely hindered.
[23] The notation SL_7 represents any of a family of active logics whose OBS and INF involve the predicates Now and K and contain a retraction mechanism. Choosing OBS and INF fixes the theory within the family.
[24] We do not distinguish between knowledge, belief, and theoremhood.
[25] Note this was written as $Contra(i, \alpha, \beta)$ in [22]. We change the notation for convenience.

not assumed to be inherited or retained in passing from one step to the next, unless explicitly stated in an inference rule. In Figure 3 below, we illustrate one possible inference function, denoted INF_B, involving a rule for special types of inheritance; see Rule 7.

The inference rules given here correspond to an inference-function, INF_B. For any given history, INF_B returns the set of all immediate consequences of Rules 1–7 applied to the last step in that history. Note that Rule 5 is the only default rule.

Rule 1(CLOCK)	$\dfrac{\mathbf{i}:}{\mathbf{i+1}:Now(i+1)}$	Agent notes the time
Rule 2(OBS)	$\dfrac{\mathbf{i}:}{\mathbf{i+1}:\alpha}$	If $\alpha \in OBS(i+1)$
Rule 3(MP)	$\dfrac{\mathbf{i}:\alpha, \alpha \rightarrow \beta}{\mathbf{i+1}:\beta}$	Modus ponens
Rule 4(XMP)	$\dfrac{\mathbf{i}:P_1\overline{a}, \ldots, P_n\overline{a}, (\forall\overline{x})[(P_1\overline{x} \wedge \ldots \wedge P_n\overline{x}) \rightarrow Q\overline{x}]}{\mathbf{i+1}:Q\overline{a}}$	Extended modus ponens
Rule 5(INTRO)	$\dfrac{\mathbf{i}:}{\mathbf{i+1}:\neg K(i,\beta)}$	Negative introspection[a]
Rule 6(CONTRA)	$\dfrac{\mathbf{i}:\alpha, \neg\alpha}{\mathbf{i+1}:Contra(\{\alpha,\neg\alpha\},i)}$	Presence of (direct) contradiction
Rule 7(INH)	$\dfrac{\mathbf{i}:\alpha}{\mathbf{i+1}:\alpha}$	Inheritance[b]

Figure 3: Rules of inference corresponding to INF_B

---

[a]where β is not a theorem at step i, but is a closed sub-formula at step i.

[b]where nothing of the form $Contra(\{\alpha,\beta\}, i-1)$ nor $Contra(\{\beta,\alpha\}, i-1)$ is an i-theorem, nor where α is of the form $Now(\beta)$. That is, contradictions and time-notes are not inherited. The intuitive reason a time-note is not inherited is that time changes at each step. The intuitive reason contradictory wffs α and β are not inherited is that not both can be true, and so the agent should, for that reason, be unwilling to simply assume either to be the case without further justification. This does not mean, however, that neither will appear at the next step, for either or both may appear for other reasons, as will be seen. Note also that the wff $Contra(\{\alpha,\neg\alpha\}, i)$ *will* be inherited, since it is not itself either a time-note or a contradiction, and (intuitively) it expresses a fact (that there was a contradiction at step i) that remains true.

For *time*, we envision a clock which is ticking as the agent is reasoning. At each step in its reasoning, the agent looks at this clock to obtain the time.[26] The wff $Now(i)$ is an i-theorem. $Now(i)$ corresponds intuitively to the statement "The time is now i."

Self-knowledge involves the predicate K, and (in INF_B) a new rule of inference, namely a rule of (negative) introspection; see Rule 5 in Figure 3. This rule is intended to have the following effect. $\neg K(i,\alpha)$ is to be deduced at step $i+1$ if α is not an i-theorem, but does appear as a closed sub-formula at step i.[27] We regard the closed sub-formulas at step i as approximating the wffs that the agent is "aware of" at i.[28] Thus the idea is that the agent can tell at $i+1$ that a given wff it is *aware* of at step i is not one of those it has as a *conclusion* at i. See [28] for another treatment of awareness. We will use the K concept to allow the agent to negatively introspect, i.e., to reason at step $i+1$ that it did not know β at step i. Thus, using INF_B, if α and $\alpha \rightarrow \beta$ are i-theorems, then β and $\neg K(i,\beta)$ will be $i+1$-theorems (concluded via Rules 3 and 5, respectively). Currently we do not employ positive introspection (i.e., from α at i infer $K(i,\alpha)$ at $i+1$), although it can be recaptured from axioms if needed.

[26]Richard Weyhrauch analyzed this idea in a rather different way in his talk at the Sardinia Workshop on Meta-Architectures and Reflection, 1986; see [140].

[27]A sub-formula of a wff is any consecutive portion of the wff that itself is a wff. Note that there are only finitely many such sub-formulas at any given step. Rule 5 formalizes the introspective time-delay discussed in [22, 27].

[28]"You can't know you don't know something, if you never heard of it." Thus from beliefs $Bird(x) \rightarrow Flies(x)$ and $Bird(tweety)$ at step i, $Bird(tweety) \rightarrow Flies(tweety)$ may follow at step $i+1$. Then at step $i+1$, $Flies(tweety)$ would become something the agent is aware of. (In INF_B this will certainly be the case, and in fact $Flies(tweety)$ will even be a theorem.)

Retractions are used to facilitate removal of certain conflicting data. Handling contradictions in a system of this sort can be quite tricky. In this active logic we handle contradictions by simply not inheriting the formulas directly involved (see Rule 6 in Figure 3). Unlike SL_0 which is monotonic (that is, if α is an i-theorem, then α is also an $i+1$-theorem), SL_7 is non-monotonic. In $SL_7(\cdot, INF_B)$[29], a conclusion in a given step, i, is inherited to step $i+1$ if it is not contradicted at step i and it is not the predicate $Now(j)$, for some j; see Rule 7 in Figure 3. Section 5 describes a more sophisticated method of handling contradictions.

$SL_7(\cdot, INF_B)$ was formulated with applications such as the *Brother problem* (see Section 3) in mind. This led to the rules of inference listed in Figure 3. Rule 3 states, for instance, that if α and $\alpha \rightarrow \beta$ are i-theorems, then β will be an $i+1$-theorem. Rule 3 makes no claim about whether or not α and/or $\alpha \rightarrow \beta$ are $i+1$-theorems. The axioms (i.e., the "observations") are those listed in Section 3.

Central to the approach is the idea that, for at least some conclusions that our agent is to make, the time the conclusion is drawn is important. The issue of deciding which conclusions to time-stamp, however, is a complex one. Time-stamping *all* conclusions seems neither intuitively correct nor formally feasible (a complex inheritance scheme would be required). On the other hand, it is important the agent be able to distinguish, for example, the belief that it didn't know B, from the belief that it currently knows B. That is, the reasoning agent may conclude, say at time 5, that wff B is unknown. Later, it may come to know B. This latter event, however, should not force the agent to forget the (still true) fact that *at time 5*, B was unknown. It thus seems important to time-stamp all introspections. Currently we are also time-stamping contradictions and "clock look-ups".

Note the use of the predicate *Contra* in Rule 6. It is used to note a *direct* contradiction only; indirect contradictions, for example, $\neg\alpha, \neg\beta$, and $\alpha \vee \beta$, may co-exist within the agent's current set of beliefs (for example, they may all be i-theorems). It is only when a direct contradiction occurs that the agent is forced to do something about it. Suppose, for example, the agent later deduces, say at time $i+1$, β. Step $i+1$ would then contain a direct contradiction. This would then be noted (via Rule 6) at step $i+2$ with the wff $Contra(\{\beta, \neg\beta\}, i+1)$. Then (by Rule 7) neither β nor $\neg\beta$ would be inherited to step $i+3$. See Section 5 for more on how we handle contradictions.

Note that what is not inherited is context-dependent: if a slightly different line of reasoning had led from the same wffs at step i to a different contradiction at step $i+1$, different wffs would fail to be inherited. Thus it is the actual time-trace of past reasoning that is reflected in the decision as to what wffs to distrust. Also note that if the extra wff that allowed the implicit contradiction to become direct had not been present, the implicit contradiction might have remained indefinitely. This behavior we regard as within the spirit of the reasoning we wish to study, since it follows real-time vagaries of what is actually done, rather than an externally proscribed notion of validity.

2.3 Semantics

What is semantics for? Classically, there are two rather distinct purposes. On the one hand, semantics simply is an accounting for meanings attached to certain syntactic strings; these meanings allow in principle a determination of which strings are true and which are false and which are neither. Once such a determination is provided (and it need not be computable) then the language has a meaning. This is the *primary* notion of semantics, not only in everyday usage but also in formal studies. First and foremost, we need to be able to say what it is for a formula to be true (or satisfied) in a given structure, if we are to have any useful intuitions (let alone meta-theorems) concerning the language.

Upon this primary semantics rests a key definition: given a set of formulas, a structure satisfying them all is a *model* of those formulas; in particular, a theory consists of a language and a set of so-called axioms, whose models are the *models of that theory.* We also from this derive the central notion of consequence: a formula F follows from a set S of formulas, if F is true in all models of S. Thus meaning, truth and consequence are the essence of the first or primary notion of semantics.

Also upon this primary semantics rests one of the most important theorems in logic: the completeness theorem of first-order logic. This meta-theorem provides a semi-decision procedure to determine those formulas that are valid (true in all structures) or that are entailments (consequences of given axioms). This procedure is simply the recursive application of first-order inference rules, beginning with axioms.

Upon the completeness theorem (and the soundness theorem) rests a *secondary* notion of semantics: a characterization of inference (provability from axioms and rules) solely in terms of entailment from axioms. This is the basis for many

[29] We use the notation $SL_7(OBS, INF_B)$ to describe an SL_7 theory defined by the observation function OBS and the inference function INF. When we have fixed the inference function, but have not fixed the observation function, we use the notation $SL_7(\cdot, INF_B)$.

applications of the completeness theorem: instead of going to the trouble of finding an actual chain of inferences constituting a proof of a formula of interest, one might instead be able to show that all models satisfy the formula, which establishes the existence of a proof. This however not only can be a useful way to circumvent proof-construction, it also serves [130] to give insights into the structure of the theorem-set. In the case of Agenta, a completeness theorem allows us another way to think about and assess his beliefs.

Such is very useful, for instance in NMR, where comparison between T_1 and T_2 is often made semantically, i.e., their theorem-sets are compared by looking at their models rather than at their inference rules. However, in such cases we often do not consider *all* conceivable structures in which the primary semantics may satisfy axioms; rather we tend to look only at preferred models that match our goals for what we think an ideal agent should believe. This in fact already occurs in mathematical logic, such as in set theory where only "standard" or "natural" models may be of interest, or in second-order logic (SOL) where "full" models are usually the preferred ones.

It is further noteworthy that some so-called logics came into being without even a primary semantics (e.g., first-order logic, modal logic, and Reiter's default logic) and some others without inference rules (e.g., Moore's autoepistemic logic); later research aimed to fill these gaps. Still other logics were defined with both inference rules and primary semantics at the outset (e.g., circumscription) and only later was a secondary semantics (completeness) established (or refuted), showing theoremhood (inference rules) and entailment (satisfaction) to match (or not to match). The only feature that seems present in all logics is a precise language (or notion of formula); either proof-theory or primary semantics may be lacking, let alone a completeness theorem. Indeed, for some logics, such as (preferred-model semantics) second-order logic (SOL), there can be no effective proof-theory that is complete.

Thus without further amplification, it is far from clear what is wanted in asking for a semantics for a logic. Nevertheless, a primary semantics is very easy to supply for active logics (or at least for those that have been studied to date). Namely, we use ordinary first-order Tarskian semantics for all predicates except Now. And for Now we use clock semantics: $Now(t)$ is true if and only if t is the current time. Thus the notion of structure must be tailored to include a "clock"; the ones we have investigated so far have "natural-number" clocks that correspond to the non-negative integers. However, alternatives (such as continuum or interval clocks) are under consideration as well.

2.3.1 Logic and AI

The use of logic in AI is frequently that of supplying a characterization of the beliefs that an artificial agent holds (or would come to hold given infinite inferential resources), based on certain given starting beliefs or axioms. We have two big complaints about this. One is that some rather important aspects of commonsense reasoning are not *persistent*: an inferred belief may be given up later *even without any new information being taken in.* Our simplest example is time itself: Having noted (as a starting "axiom") that it is now noon, we do not reason on and on in the belief that it is still noon. To the contrary, we reason that it is now a little later, and now even later. This is then not ordinary nonmonotonicity; some beliefs once accepted are then rejected even without even any new data being received—unless one considers the knowledge that one's inferences are taking place as time passes a kind of new data.

Our second objection is closely related to the first: the notion of axiom is largely inappropriate to the commonsense world. Little if any information is sacrosanct; indeed, "given" P one might then come to believe not-P on the basis of reasoning alone (and other "axioms" at one's disposal). Indeed, from a background belief set B and a new datum P, one might come in time to reject P, or even to reject parts of B. Beliefs form a collection that is constantly undergoing change, and little if any of it persists permanently.

The main point with regard to semantics here is that reason is a process of inference, it is not a royal road to truth. Traditional semantics of the sort we have called primary simply defines what truth *is*, it does not find out *what* is true. Secondary semantics throws a net around truth, but in so doing it distances itself from reasoning which is a more tentative and always defeasible thing, a groping toward truth rather than a doctrinaire seizing of it. Now to be sure, much research in NMR has aimed to "throw a net around" (plausible) truth; but it has done so within the strictures of axioms and persistence: what is given is permanent. This means, first, that a proper accounting for time-passage during reasoning cannot be provided, and second that the automated reasoner is hobbled by not being able to consider that its "axioms" may be suspect.

Consider for instance being told "The foxes will try to eat the chickens, and the chickens will try to eat the grain; we should not leave either the chickens and the grain together or the chickens and the wolves together." It is reasonable to suppose that "wolves" was a mis-statement and should have been "foxes". Whether or not one agrees that there are sufficient grounds given for that conclusion, traditional formalisms do not even allow for the considering that the initial assertion may

	number of languages	number of extensions	semantics	inference	completeness
FOL	1	1	yes	yes	yes
SOL	1	1	yes	yes	no
DL	1	unlimited	yes	yes	no
CIRC	1	1	yes	yes	no
AEL	1	1	yes	no	-
AL	unlimited	unlimited	yes	yes	no

Figure 4: Classifications of Various Logics

contain an error. Note that there is in fact no contradiction in that assertion; any contradiction that may exist would likely be between that assertion and certain default assumptions about the intentions of the speaker in making that assertion. Traditional formalisms always respect bare "axioms" and defeat only default assumptions; thus the assumption that the speaker means the second part of the sentence to be inferred from the first will be defeated, rather than the error-freeness of the utterance itself.

Thus a model of reasoning is (or should be) a model of inference processes, not of logical (semantic) consequence.

2.3.2 What sort of logic is an active logic?

An active logic is an inference-based logic for which at least some inference rules are time-sensitive; in which proofs are relative to models, to the clock in a model. So semantics (models) and inference are linked in the very definition of an active logic. The logic "acts", it is defined to be (realized as) an engine running in time.

This does not mean it is simply an implementation, though: it is an abstraction, it requires only very particular "concrete" elements, most notably a clock. But it is not so concrete as to be a coded "system", although we do have implementations as well. It is defined abstractly, formally, and can be studied as a formal system, meta-theorems proven, etc. In a very simple case (propositional) we even have a completeness theorem. In fact, in one nontrivial sense we do get full first-order (FOL) completeness in AL, where Now is interpreted by the model-clock and Bel by an agent self-model within the model. However, when the belief-set is inconsistent, there is no model; this does not clash with completeness (any more than it does in FOL) but it does lessen its interest.

We are working on various alternate semantics that may shed more light on the inconsistent case [106, 42]. One intuitively appealing one is a limit-semantics where the agent has no new observations after a given step; it is of note that this kind of constraint is the basis for an active logic contradiction-recovery theorem (see Section 5).

We can attempt a very general definition of logic, to include all familiar cases: we will need a collection L of languages, to allow for language change as Agenta learns new expressions; and a collection Th of theorem-sets, to allow not only for changes in Agenta's beliefs but also for cautious reasoning (sanctioned alternative beliefs). We do not require inference rules; nor models. Each theorem-set has a corresponding language; but this can be a many-one correspondence. Given the above, now we can classify various logics, using not only the collections L and Th, but also primary semantics (Sem), a notion of inference (inf), and a completeness theorem (Comp). Below we show the classification for first-order logic (FOL), second-order logic (SOL), default logic (DL), circumscription (CIRC), autoepistemic logic (AEL) and active logic (AL):

The table in Figure 4 shows, for instance, that a first-order logic has a fixed language and theorem-set, theorems can be characterized either by semantics or by inference, and that these coincide due to the appropriate completeness result. In SOL, inferential and semantic notions of theorem do not coincide: there is no appropriate completeness theorem. A default logic has one language but, via the notion of extension, has in general many distinct possible theorem-sets, based on non-deterministic inference rules; the meaning of wffs is the same as for FOL, but this does not provide an alternate route to defining theorems. A circumscriptive logic is essentially a first-order (or second-order) logic with a special notion of preferred model providing only partial completeness. An autoepistemic logic also has a single language and theorems are semantically defined. An active logic has a potential infinity of languages as time passes, since new expressions may enter the reasoning process, as well as new meanings for old expressions; the theorem-set also may grow and shrink over time, so that at distinct times there are distinct theorem-sets; and theorem-hood is defined by inference, not models, even though the meaning of wffs is traditional just as with DL.

2.3.3 Contradictions

In a classical formal system—and even in temporal logics—a contradiction nullifies any usefulness of the logic, since all formulas in the language are inferred. No information is present as to the time at which a given formula is inferred: the logic does not model the ongoing process of reasoning but rather only the infinite "ideal, omniscient" limit of reasoning, as if the robot using the logic would have the luxury of thinking forever before acting. However, in an active logic, the ongoing process of inference is captured via the formal introduction of a shifting indexical predicate expression $Now(t)$ which has the intuitive meaning that the time is now (currently) t. As reasoning proceeds, $Now(10)$ will become true and then false as the next inference is drawn and $Now(11)$ becomes true, and so on. Thus active logics keep track of time taken by inference, thereby allowing the robot to also reason about the nearing of a deadline as it plans a course of action.

The very same $Now(t)$ mechanism is what allows active logics to deal safely with contradictions. If a direct contradiction, P and $\neg P$, is inferred at time t, then even though all formulas may be inferable from this, it will in general take an infinite amount of time: active logic rules of inference produce only finitely- many inferences in each time step. Indeed, at time $t+1$, a special contradiction-rule produces the inference $Contra(P, \neg P)$

2.3.4 A truth-oriented semantics

Below are some formal details of a traditional truth-oriented semantics for active logics first defined in [22, 27].

Let $\mathcal{L}'$ be the language having the symbols of $\mathcal{L}$ and the (possibly additional) predicate symbols K and Now. Thus $\mathcal{L}'$ may be $\mathcal{L}$ itself.

Definition 2.8 *A* step-interpretation *for $\mathcal{L}'$ is a sequence $M = < M_0, M_1, \ldots, M_i, \ldots >$, where*

1. *Each M_i is an ordinary first-order interpretation of $\mathcal{L}'$.*
2. *$M_i \models Now(i)$.*

Definition 2.9 *A* step-model *for $SL_n(OBS, INF)$ is a step-interpretation M satisfying*

1. *$M_i \models K(j, \alpha)$ iff $SL_n(OBS, INF) \vdash_j \alpha$.*
2. *$M_i \models \alpha$ whenever $SL_n(OBS, INF) \vdash_i \alpha$.*

Condition 1 insures that a chronological record of the j-theorems exists in each M_i; and Condition 2 insures that the i-theorems are in fact true. M should not be thought of as the real external world, corresponding to an agent's beliefs. Rather, M is just a reflection of those beliefs and may or may not correspond to external matters. In particular, a wff B can be true in M_i and false in M_{i+1} simply because the agent has changed its mind.

Definition 2.10 *A wff α is* i-true *in a step-model M (written $M \models_i \alpha$) if $M_i \models \alpha$.*

Definition 2.11 *$SL_n(OBS, INF)$ is* step-wise consistent *if for each $i \in \mathbf{N}$, the set of i-theorems is consistent (classically, i.e., the set has a first-order model).*

Definition 2.12 *$SL_n(OBS, INF)$ is* eventually consistent *if $\exists i$ such that $\forall j > i$, the set of j-theorems is consistent.*

Definition 2.13 *An observation-function OBS is* finite *if* $\exists i$ *such that* $\forall j > i$, $OBS(j) = \emptyset$.

Definition 2.14 $SL_n(\cdot, INF)$ *is* self-stabilizing *if for every finite OBS,* $SL_n(OBS, INF)$ *is eventually consistent.*

Remark 1:

1. Even if $SL_n(OBS, INF)$ is step-wise consistent, it can have conflicting wffs at *different* steps, e.g., $SL_n(OBS, INF) \vdash_{10} Now(10)$ and $SL_n(OBS, INF) \vdash_{11} \neg Now(10)$.
2. Any step-wise consistent theory is eventually consistent.
3. Intuitively a self-stabilizing theory $SL_n(\cdot, INF)$ corresponds to a fixed agent that can regain and retain consistency after being given arbitrarily (but finitely) many contradictory initial beliefs.

Theorem 2.2 *If* $SL_n(OBS, INF)$ *has a step-model, then it is step-wise consistent.*[30]

Theorem 2.3 (Soundness) *Every step-logic* $SL_n(OBS, INF)$ *is sound with respect to step-models. That is, every i-theorem* α *of* $SL_n(OBS, INF)$ *is i-true in every step-model* M *of* $SL_n(OBS, INF)$*, i.e., if* $SL_n(OBS, INF) \vdash_i \alpha$ *then* $M \models_i \alpha$.

Definition 2.15 *A wff is said to be* P-free *if it does not contain the predicate letter* P.

Definition 2.16 *An observation-function OBS is said to be* P-free *if* $\forall i \forall \alpha (\alpha \in OBS(i) \rightarrow \alpha$ *is P-free* $)$.

Definition 2.17 *An observation-function OBS is said to be* valid *if* $\forall i \forall \alpha (\alpha \in OBS(i) \rightarrow \alpha$ *is logically valid*$)$.

Theorem 2.4 $SL_7(OBS, INF_B)$ *is step-wise consistent if OBS is both valid and Now-free.*

3 Nonmonotonicity

Active logics provide at least two major means for modeling nonmonotonic reasoning. The main advantage of Active logic is its ability One is, simply, to always allow the default inference initially (e.g., that birds fly) and then if it is found to contradict other information, retract the conclusion. Since active logics have the luxury of formally represented time-passage, this does not present the usual problem of contradictions, since a contradiction now may exist only briefly and then become part of hte reasoner's history accessible to recall via an past-introspection predicate.[31]

Another means for carrying out nonmonotonic reasoning in active logics, which is a bit closer in spirit to standard treatments, is to represent the reasoner's *current* beliefs (via the Now predicate) and sanction a default only if its negation is not among those beliefs. (Since in active logics this is always a finite set of beliefs, there is little computational expense here.) Thus we might have:

$$\frac{P \wedge -Know(-Q, t) \wedge Now(t)}{Q}$$

where the default principle is: infer Q given P, if possible.

Although these two approaches may seem very different, the difference is rather small in the active-logic framework. This is so because, even if a default does not contradict current knowledge, it may contradict knowledge that comes to be believed later (e.g., due to observations). Thus a contradiction may occur and require a retraction at a later time.

[30] This result is useful in showing certain active logics are consistent; however, by the same token, since many interesting active logics are *inconsistent* (and in fact derive much of their interest from their inconsistency), step-models are not sufficiently general as defined.

[31] Many systems that exhibit nonmonotonic behavior have been described and studied in the literature, e.g., [10, 82, 88, 119, 120]. However, they are static systems, and don't model the on-going reasoning of the agent. Thus, they don't provide a mechanism to recover when a default conclusion turned out to be wrong.

The projection problem (deciding what to infer about future states and predicates true in the present) can be treated similarly. If it is not known that a predicate will change its truth, we infer it will remain true, and later change our mind if need be. This mechanism, in rough outline, is used in our applications of active logics to deadline-based planning,(see section 8) and for instance in our treatment of YSP citenirkhe/kraus:imagination. However, there we utilize a more complex KR mechanism in order to complex contextual information relevant to planning and acting.

We have used active logics for the famous case of birds (see jj-diss), among others. Here we present a brief illustration using the brother problem.

We now turn to a particular problem for which active logics seem well-suited: default reasoning. We use Moore's *Brother problem* (see [97]) to provide examples of an SL_7 at work. In Moore's *Brother problem* one reasons, "Since I don't know I have a brother, I must not." This problem can be broken down into two: the first requires that the reasoner be able to decide he doesn't know he has a brother; the second that, on that basis, he, in fact, does not have a brother (from *modus ponens* and the assumption that "If I had a brother, I'd know it.") The first of these seems to lend itself readily to active logic, in that the negative reflection problem (determining when something is not known) reduces to a simple look-up.[32]

In the following three sub-sections we present synopses of computer-generated results for three different scenarios where the reasoning agent must determine whether or not a brother exists. We use $SL_7(\cdot, INF_B)$ (defined in Figure 3 on page 15) for this example. Each scenario has its own unique set of axioms (observation function). Let B be a 0-argument predicate letter representing the proposition that a brother exists. Let P be a 0-argument predicate letter (other than B) that represents a proposition that implies that a brother exists.[33] In each case, at some step i the agent has the axiom $P \rightarrow B$, and also the following autoepistemic axiom which represents the belief that not knowing B "now" implies $\neg B$.

Axiom 1 $(\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B]$[34]

The following three distinct behaviors are illustrated:

- If B is among the wffs of which the agent is aware at step i, but not one that is believed at step i, then the agent will come to know this fact ($\neg K(i, B)$, that it was not believed at step i) at step $i+1$. As a consequence of this, other information may be deduced. In this case, the agent concludes $\neg B$ from the autoepistemic axiom (Axiom 1). Clearly the Now predicate plays a critical role. Section 3.1 below illustrates this case.
- The agent must refrain from such negative introspection when in fact B is already known; see Section 3.2.
- A conflict may occur if something is coming to be known while negative introspection is simultaneously leading to its negation. The third illustration (see Section 3.3 below) shows this being resolved in an intuitive manner (though not one that will generalize as much as we would like; see Section 5 for a far more sophisticated method of handling contradictions).

3.1 Simple negative introspection succeeds

In this example (see Figure 5) the agent is not able to deduce the proposition B, that he has a brother, and hence is able to deduce $\neg B$, that he does *not* have a brother. Here, and in example scenarios in the remainder of the paper, for ease of reading we underline in each step those wffs which are new (i.e., which appear through other than inheritance). For the purposes of illustration, let i be arbitrary and let our axioms be:

$$OBS_{B_1}(j) = \begin{cases} \{P \rightarrow B, (\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B]\} & \text{if } j = i \\ \emptyset & \text{otherwise} \end{cases}$$

Since B is not an i-observation (and thus in this case is not an i-theorem), the agent uses Rule 5, the negative introspection rule, to conclude $\neg K(i, B)$ at step $i+1$. At step $i+2$ the agent concludes $\neg B$ from the given autoepistemic knowledge and the use of the alternate version of modus ponens, Rule 4.

[32] Remember, all active logics ensure only a finite number of beliefs at any given step.

[33] P might be something like "My parents have two sons," together with appropriate axioms.

[34] No real arithmetic is involved here; simple syntactic devices can obviate any genuine subtraction. We can replace, for instance, $K(i-1, \alpha)$ by $J(i, \alpha)$ with the intuitive meaning that α was known "just a moment ago", i.e., at i. Alternatively, we can use successor notation for natural numbers.

$$i: \quad \underline{Now(i)}, \underline{P \rightarrow B}, \underline{(\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B]}$$

$$i+1: \quad \underline{Now(i+1)}, P \rightarrow B, (\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B], \underline{\neg K(i, B)}, \underline{\neg K(i, \neg B)}, \underline{\neg K(i, P)}$$

$$i+2: \quad \underline{Now(i+2)}, P \rightarrow B, (\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B], \neg K(i, B), \neg K(i, \neg B), \neg K(i, P), \underline{\neg B}, \underline{\neg K(i+1, B)}, \underline{\neg K(i+1, \neg B)}, \underline{\neg K(i+1, P)}$$

Figure 5: Negative introspection succeeds

3.2 Simple negative introspection fails (appropriately)

In this example, let our axioms be:

$$OBS_{B_2}(j) = \begin{cases} \{P \rightarrow B, (\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B], B\} & \text{if } j = i \\ \emptyset & \text{otherwise} \end{cases}$$

Thus the agent has B at step i, and is blocked (appropriately for this example) from deducing at step $i+1$ the wffs $\neg K(i, B)$ and $\neg B$. See Figure 6.

$$i: \quad \underline{Now(i)}, \underline{P \rightarrow B}, \underline{(\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B]}, \underline{B}$$

$$i+1: \quad \underline{Now(i+1)}, P \rightarrow B, (\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B], B, \underline{\neg K(i, \neg B)}, \underline{\neg K(i, P)}$$

Figure 6: Negative introspection fails appropriately

Note that a traditional final-tray-like approach (see page 11) could produce quite similar behavior to that seen in Figures 5 and 6 if it is endowed with a suitable introspection device, although it would not have the real-time step-like character we are trying to achieve.

3.3 Introspection contradicts other deduction

It is in this third example that a traditional final-tray-like approach would encounter difficulties, because of the introduction of a contradiction in step $i+2$. The final tray for a tray-like model of a reasoning agent would simply be filled with all wffs in the language—and no basis for a resolution would be possible *within* such a logic. In an active logic, however, the contradiction poses no threat—the contradiction is noted, then steps (pun intended!) are taken to resolve the contradiction. In this case the contradiction resolves quite naturally: once the contradiction is noted, neither belief is inherited; one of the beliefs is then re-deduced (due to its existing justification in other beliefs), and the other is not (it was originally deduced based on negatively introspecting, yet the set of beliefs has changed and this introspection no longer produces the same belief).

In this example, let our axioms be:

$$OBS_{B_3}(j) = \begin{cases} \{P \rightarrow B, (\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B], P\} & \text{if } j = i \\ \emptyset & \text{otherwise} \end{cases}$$

In Figure 7 we see then that the agent does not have B at step i, but is able to *deduce* B at step $i+1$ from $P \rightarrow B$ and P at step i. Since the agent is *aware* (in our sense) of B at step i, and yet does not have B as a *conclusion* at i, it will deduce $\neg K(i, B)$ at step $i+1$. Thus both B and $\neg K(i, B)$ are concluded at step $i+1$. At step $i+2$ Axiom 1 (the autoepistemic axiom), together with $Now(i+1)$ and $\neg K(i, B)$ and Rule 4, will produce $\neg B$. A conflict results, which is noted at step $i+3$. This then inhibits inheritance of both B and $\neg B$ at step $i+4$. Although neither B nor $\neg B$ is *inherited* to step $i+4$, B is *re-deduced* at step $i+4$ via modus ponens from step $i+3$. Thus B "wins out" over $\neg B$ due to its existing justification in other wffs, while $\neg B$'s justification is "too old": $\neg K(i+2, B)$, rather than $\neg K(i, B)$, would be needed. We see then that the conflict resolves due to the special nature of the time-bound "now" feature of introspection. This case works out somewhat fortuitously. In general, resolving contradictions is difficult; see section 5 for more on this.

$$\begin{array}{rl}
i: & \underline{Now(i)}, \underline{P \rightarrow B}, \underline{(\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B]}, \underline{P} \\[6pt]
i+1: & \underline{Now(i+1)}, P \rightarrow B, (\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B], P, \underline{B}, \underline{\neg K(i, B)}, \underline{\neg K(i, \neg B)} \\[6pt]
i+2: & \underline{Now(i+2)}, P \rightarrow B, (\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B], P, B, \neg K(i, B), \neg K(i, \neg B), \\
& \underline{\neg B}, \underline{\neg K(i+1, \neg B)} \\[6pt]
i+3: & \underline{Now(i+3)}, P \rightarrow B, (\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B], P, B, \neg K(i, B), \neg K(i, \neg B), \\
& \neg B, \neg K(i+1, \neg B), \underline{Contra(\{B, \neg B\}, i+2)} \\[6pt]
i+4: & \underline{Now(i+4)}, P \rightarrow B, (\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B], P, \neg K(i, B), \neg K(i, \neg B) \\
& \neg K(i+1, \neg B), Contra(\{B, \neg B\}, i+2), \underline{B}, \underline{Contra(\{B, \neg B\}, i+3)}
\end{array}$$

Figure 7: Introspection conflicts with other deduction and resolves

Remark 2: The following are true about the consistency of each of the SL_7 theories given in the brother examples:

1. $SL_7(OBS_{B_1}, INF_B)$ is step-wise consistent.
2. $SL_7(OBS_{B_2}, INF_B)$ is step-wise consistent.
3. $SL_7(OBS_{B_3}, INF_B)$ is eventually consistent (but not step-wise consistent[35]).

See [27] for the proof.

4 Reasoning about others' reasoning

In this section we present a version of the classic *Three Wisemen Problem* which was first introduced to the AI literature by McCarthy in [81]. This version best illustrates a type of reasoning that is often characteristic of commonsense reasoners: the ability to reason about others' reasoning. We shall see that active logic provides an intuitive solution to this problem. We repeat from [27] our solution to this problem. For additional details, see [25].

> *A king wishes to know whether his three advisors are as wise as they claim to be. Three chairs are lined up in a column, all facing the same direction, one behind the other. The wisemen are instructed to sit down. The wiseman in the back (wiseman #3) can see the backs of the other two men. The man in the middle (wiseman #2) can only see the one wiseman in front of him (wiseman #1); and the wiseman in front (wiseman #1) can see neither wiseman #3 nor wiseman #2. The king informs the wisemen that he has three cards, all of which are either black or white, at least one of which is white. He places one card,*

[35] This is why a traditional final-tray-like approach would encounter difficulties with this example.

face up, behind each of the three wisemen. Each wiseman must determine the color of his own card and announce what it is as soon as he knows. The first to correctly announce the color of his own card will be aptly rewarded. All know that this will happen. The room is silent; then, after several minutes, wiseman #1 says "My card is white!".

We assume the following: the wisemen do not lie; the wisemen all have the same reasoning capabilities; and the wisemen can all think at the same speed. We can then postulate that the following reasoning took place.

> Each wiseman knows there is at least one white card. If the cards of wiseman #2 and wiseman #1 were black, then wiseman #3 would have been able to announce immediately that his card was white. They all realize this (they are all truly wise). Since wiseman #3 kept silent, either wiseman #2's card is white, or wiseman #1's is. At this point wiseman #2 would be able to determine, if wiseman #1's were black, that his card was white. They all realize this. Since wiseman #2 also remains silent, wiseman #1 knows his card must be white.

It is clear that a versatile commonsense reasoning agent must be able to reason "If such and such were true at *that* time, then so and so would have realized it by *this* time." So, for instance, if wiseman #2 is able to determine that wiseman #3 would have already been able to figure out that wiseman #3's card is white, and wiseman #2 has heard nothing, then wiseman #2 knows that wiseman #3 does *not* know the color of his card. Active logic is particularly well-suited to this type of deduction since it focuses on the actual individual deductive steps. Others have studied this problem (e.g. see [62, 63, 4]) from the perspective of logically-closed reasoning, in which each agent already knows all the logical consequences of his beliefs, and thus are not able to address this temporal aspect of the problem: assessing what others have been able to conclude *so far*.

In [29] Fagin, Halpern, Moses, and Vardi discuss the Three Wisemen problem and others like it from a very general perspective and give an analysis in terms of modal logic and common or mutual knowledge. However, that treatment too (because of built-in features of possible-world modal semantics) involves the unrealistic assumption that each agent knows at any time all logical consequences of his beliefs. (Elsewhere in the same work (chapter 10) the authors discuss alternatives closer in spirit to our approach.) In [106] we describe an alternative modal semantics that comes closer to real-time (evolving) belief, but still does not fully escape the possible-world difficulties of many beliefs "before their time" (modeled as being believed before a reasoner could possibly come to believe them).

4.1 Formulation

The active logic used to model the *Three-wise-men problem* is defined in Figures 8 and 9. The problem is modeled from wiseman #1's point of view. The observation-function contains all the axioms that wiseman #1 needs to solve the problem, and the inference-function provides the allowable rules of inference.

We use an SL_5 theory. An SL_5 theory gives the reasoner knowledge of its own beliefs as well as knowledge of the passage of time.[36] The language of SL_5 is first-order, having binary predicate symbols K_j and U, and function symbol s. $K_j(i, `\alpha')$ expresses the fact that "$\alpha$ is known by agent $j$ at step $i$". Note that this gives the agent the expressive power to introspect on his own beliefs as well as the beliefs of others. $U(i, `x')$ expresses the fact that an utterance of x is made at step i.[37] $s(i)$ is the successor function (where $s^k(0)$ is used as an abbreviation for $\underbrace{s(s(\cdots(s}_{k}(0))\cdots))$). W_i and B_i express the facts that i's card is white, and i's card is black, respectively.

Recall that in an active logic, wffs are not assumed to be inherited or retained in passing from one step to the next, unless explicitly stated in an inference rule. Note that Rule 8 in Figure 9, does provide an unrestricted form of inheritance.[38]

We note several points about the axioms which wiseman #1 requires. (Refer to Figure 8.) wiseman #1 knows the following:

[36] For more details on SL_n theories, see [22].
[37] For simplicity, in the remainder of the paper we drop the quotes around the second argument of predicates U and K_j.
[38] Although many commonsense reasoning problems require former conclusions to be withdrawn (based on new evidence), as did the formulation of the *Brother Problem*, this particular formulation of the *Three-wise-men Problem* does not require any conclusions to be retracted. We can thus use an unrestricted form of inheritance.

OBS_{W_3} is defined as follows.

$$OBS_{W_3}(i) = \begin{cases} \left\{ \begin{array}{l} (\forall j)K_2(j,(\forall i)(\forall x)(\forall y)[K_3(i,x \rightarrow y) \rightarrow \\ \qquad\qquad\qquad (K_3(i,x) \rightarrow K_3(s(i),y))]) \\ (\forall j)K_2(j,K_3(s(0),(B_1 \wedge B_2) \rightarrow W_3)) \\ (\forall j)K_2(j,(B_1 \wedge B_2) \rightarrow K_3(s(0),B_1 \wedge B_2)) \\ (\forall j)K_2(j,\neg(B_1 \wedge B_2) \rightarrow (B_1 \rightarrow W_2)) \\ (\forall j)K_2(j,(\forall i)[\neg U(s(i),W_3) \rightarrow \neg K_3(i,W_3)]) \\ (\forall i)(\forall x)[\neg K_1(s(i),U(i,x)) \rightarrow \neg U(i,x)] \\ (\forall i)[\neg U(i,W_3) \rightarrow K_2(s(i),\neg U(i,W_3))] \\ (\forall i)(\forall x)(\forall y)[K_2(i,x \rightarrow y) \rightarrow (K_2(i,x) \rightarrow K_2(s(i),y))] \\ (\forall i)(\forall x)(\forall x')(\forall y)(\forall y') \\ \quad [(K_2(i,\neg(x \wedge x') \rightarrow (y \wedge y')) \wedge K_2(i,\neg(x \wedge x'))) \rightarrow \\ \quad\; K_2(s(i),y \wedge y')] \\ (\forall j)(\forall k)(\forall z)(\forall z')(\forall w) \\ \quad [(K_2(j,(\forall i)(\forall x)(\forall y)[K_3(i,x \rightarrow y) \rightarrow \\ \qquad\qquad\qquad (K_3(i,x) \rightarrow K_3(s(i),y))])\wedge \\ \quad\; K_2(j,K_3(k,(z \wedge z') \rightarrow w))) \rightarrow \\ \quad K_2(s(j),K_3(k,z \wedge z') \rightarrow K_3(s(k),w))] \\ (\forall j)(\forall k) \\ \quad [(K_2(j,(\forall i)[\neg U(s(i),W_3) \rightarrow \neg K_3(i,W_3)])\wedge \\ \quad\; K_2(j,\neg U(s(k),W_3))) \rightarrow \\ \quad K_2(s(j),\neg K_3(k,W_3))] \\ (\forall i)(\forall x)(\forall y)[(K_2(i,x \rightarrow y) \wedge K_2(i,\neg y)) \rightarrow K_2(s(i),\neg x)] \\ (\forall i)(\forall x)(\forall x')(\forall y) \\ \quad [(K_2(i,(x \wedge x') \rightarrow y) \wedge K_2(i,\neg y)) \rightarrow K_2(s(i),\neg(x \wedge x'))] \\ (\forall i)[B_1 \rightarrow K_2(i,B_1)] \\ (\neg B_1 \rightarrow W_1) \\ (\forall i)[\neg U(s(i),W_2) \rightarrow \neg K_2(i,W_2)] \end{array} \right\} & \text{if } i = 1 \\ \emptyset & \text{otherwise} \end{cases}$$

Figure 8: OBS_{W_3} for the Three-wise-men Problem

1. wiseman #2 knows (at every step) that wiseman #3 uses the rule of *modus ponens*.
2. wiseman #2 uses the rules of *modus ponens* and *modus tolens*.
3. wiseman #2 knows (at every step) that if both my card and his card are black, then wiseman #3 would know this fact at step 1.
4. wiseman #2 knows (at every step) that if it's not the case that both my card and his are black, then if mine is black, then his is white.[39]
5. wiseman #2 knows (at every step) that if there's no utterance of W_3 at a given step, then wiseman #3 did not know W_3 at the previous step. (wiseman #2 knows (at every step) that there will be an utterance of W_3 the step after wiseman #3 has proven that his card is white.)
6. If I don't know about a given utterance, then it has not been made at the previous step.
7. If there's no utterance of W_3 at a given step, then wiseman #2 will know this at the next step.[40]
8. If my card is black, then wiseman #2 knows this (at every step).

[39] In other words, if wiseman #2 knows that at least one of our cards is white, then my card being black would mean that his is white. Indeed, this axiom gives wiseman #2 quite a bit of information, perhaps too much. (He should be able to deduce some of this himself.) This is discussed in more detail in [22, 25].

[40] Interestingly, it is not necessary for wiseman #1 to know there was no utterance; wiseman #1 only needs to know that wiseman #2 will know there was no utterance.

The inference rules given here correspond to an inference-function, INF_{W_3}. For any given history, INF_{W_3} returns the set of all immediate consequences of Rules 1–8 applied to the last step in that history.

Rule 1(OBS)	$\dfrac{i : \ldots}{i+1 : \alpha}$	if $\alpha \in OBS(i+1)$
Rule 2(MP)	$\dfrac{i : \ldots, \alpha, (\alpha \rightarrow \beta)}{i+1 : \ldots, \beta}$	Modus ponens
Rule 3(XMP)	$\dfrac{i : P_1\overline{a}, \ldots, P_n\overline{a}, (\forall\overline{x})[(P_1\overline{x} \wedge \ldots \wedge P_n\overline{x}) \rightarrow Q\overline{x}]}{i+1 : Q\overline{a}}$	Extended modus ponens
Rule 4	$\dfrac{i : \ldots, \neg\beta, (\alpha \rightarrow \beta)}{i+1 : \ldots, \neg\alpha}$	Modus tolens
Rule 5	$\dfrac{i : \neg Q\overline{a}, (\forall\overline{x})(P\overline{x} \rightarrow Q\overline{x})}{i+1 : \neg P\overline{a}}$	Extended modus tolens
Rule 6	$\dfrac{i : \ldots}{i+1 : \ldots, \neg K_1(s^i(0), U(s^{i-1}(0), W_j))}$	if $U(s^{i-1}(0), W_j) \notin \vdash_i$, $j = 2, 3, \ i > 1$
Rule 7	$\dfrac{i : (\forall j)K_2(j, \alpha)}{i+1 : \ldots, K_2(s^i(0), \alpha)}$	Instantiation
Rule 8	$\dfrac{i : \ldots, \alpha}{i+1 : \ldots, \alpha}$	Inheritance

Figure 9: INF_{W_3} for the Three-wise-men Problem

9. If there is no utterance of W_2 at a given step, then wise man #2 doesn't know at the previous step that his card is white. (There would be an utterance of W_2 the step after wiseman #2 knows his card is white.)

Note the following concerning the inference rules:

1. Rule 6 is a rule of introspection. wiseman #1 can introspect on what utterances have been made.[41]
2. The rule for extended *modus ponens*, Rule 3, allows an arbitrary number of variables.
3. Rule 7 is a rule of instantiation. If wiseman #1 knows that wiseman #2 knows α at *each* step then, in particular, wiseman #1 will know at step $i+1$ that wiseman #2 knew α at step i.
4. The rule of inheritance, Rule 8, is quite general: *everything* is inherited from one step to the next.[42]

4.2 Solution

The solution to the problem is given in Figure 10. The step number is listed on the left. The reason (inference rule used) for each deduction is listed on the right. To allow for ease of reading, only the wffs in which we are interested are shown at each step. In addition, none of the inherited wffs are shown. This means that a rule appears to be operating on a step other than the previous one; the wffs involved have, in fact, actually been inherited to the appropriate step.

[41] We limit the number of wffs on which the agent can introspect in order to keep the set of beliefs at any given step finite.
[42] For other commonsense reasoning problems, a far more restrictive version of inheritance is necessary.

0:		$\emptyset$	
1:	(a)–(p)	All wffs in $OBS_{W_3}(1)$	(R1)
2:		(no new deductions of interest)	
3:		(no new deductions of interest)	
4:		(no new deductions of interest)	
5:	(a)	$\neg K_1(s^4(0), U(s^3(0), W_3))$	(R6)
	(b)	$K_2(s^4(0), (\forall i)(\forall x)(\forall y)$ $[K_3(i, x \to y) \to (K_3(i, x) \to K_3(s(i), y))])$	(R7,1a)
	(c)	$K_2(s^4(0), K_3(s(0), (B_1 \wedge B_2) \to W_3))$	(R7,1b)
	(d)	$K_2(s^4(0), (\forall i)[\neg U(s(i), W_3) \to \neg K_3(i, W_3)])$	(R7,1e)
6:	(a)	$\neg U(s^3(0), W_3)$	(R3,5a,1f)
	(b)	$K_2(s^5(0), K_3(s(0), B_1 \wedge B_2) \to K_3(s^2(0), W_3))$	(R3,5b,5c,1j)
7:	(a)	$K_2(s^4(0), \neg U(s^3(0), W_3))$	(R3,6a,1g)
	(b)	$K_2(s^6(0), (B_1 \wedge B_2) \to K_3(s(0), B_1 \wedge B_2))$	(R7,1c)
8:	(a)	$K_2(s^5(0), \neg K_3(s^2(0), W_3))$	(R3,7a,5d,1k)
	(b)	$K_2(s^7(0), \neg(B_1 \wedge B_2) \to (B_1 \to W_2))$	(R7,1d)
9:		$K_2(s^6(0), \neg K_3(s(0), B_1 \wedge B_2))$	(R3,8a,6b,1l)
10:		$K_2(s^7(0), \neg(B_1 \wedge B_2))$	(R3,9,7b,1m)
11:		$K_2(s^8(0), B_1 \to W_2)$	(R3,10,8b,1i)
12:	(a)	$(K_2(s^8(0), B_1) \to K_2(s^9(0), W_2))$	(R3,11,1h)
	(b)	$\neg K_1(s^{11}(0), U(s^{10}(0), W_2))$	(R6)
13:		$\neg U(s^{10}(0), W_2)$	(R3,12b,1f)
14:		$\neg K_2(s^9(0), W_2)$	(R3,13,1p)
15:		$\neg K_2(s^8(0), B_1)$	(R4,14,12a)
16:		$\neg B_1$	(R5,15,1n)
17:		W_1	(R2,16,1o)

Figure 10: Solution to the Three-wise-men Problem

In step 1 all the initial axioms ($OBS_{W_3}(1)$) have been inferred through the use of Rule 1.[43] Nothing of interest is inferred in steps 2 through 4. In step 5, wiseman #1 is able to negatively introspect and determine that no utterance of W_3 was made at step 3. Note the time delay: wiseman #1 is able to prove *at step 5* that he did not know *at step 4* of an utterance made *at step 3*.[44] The remaining wffs shown in step 5 were all inferred through the use of Rule 7, the rule of instantiation. wiseman #1 needs to know that wiseman #2 knows these particular facts at step 4.[45] The reasoning continues from step to step. Note that at step 11, wiseman #1 has been able to deduce that wiseman #2 knows that if wiseman #1's card is black, then his is white. From this step on, we essentially have the *Two-wise-men problem*. (See [23].) In step 17 wiseman #1 is finally able to deduce that his card is white.

We see that active logic is a useful vehicle for formulating and solving a problem of this kind in which the time that something occurs is important. wiseman #1 does indeed determine "if wiseman #2 or wiseman #3 knew the color of his card, he would have announced it by now." wiseman #1 then reasons backwards from here to determine that his card must not be black, and hence must be white.

Many formulations of the *Three-wise-men problem* have involved the use of a meta-language that describes the reasoning of all three wisemen, rather than an object language that serves directly the reasoning needs of one of the agents. The latter is more in the spirit of active logics, where the idea is to allow the reasoner itself enough power (with no outside "oracle" intervention) to solve the problem. Thus we model the agent directly, rather than using a meta-theory. For more details on the use of active logic to model this problem, see [24].

The kind of reasoning considered above is easily seen to be nonmonotonic, for it depends on the reasoner recognizing that it does *not* have certain beliefs, e.g., that another wiseman has uttered something, or that an utterance has been heard. Had further information been supplied, in the form of observations or utterances, then some conclusions would not have been arrived at.

[43] To save space we have not repeated them in the figure. See Figure 8 for the individual axioms.
[44] For a detailed description of this phenomenon, see [22].
[45] Note that Rule 7 is producing wffs *at each step*, but for ease of exposition, we show only those of interest.

We have so far discussed nonmonotonicity in general, and in terms of reasoning about others reasoning. Next we turn to a consideration of reasoning in the face of contradictions. This also turns out to be related to nonmonotonicity.

5 Reasoning in the face of contradictions

Contradiction and conflict play a key mediating role in the commonsense reasoning we often wish to formalize. The intuition here is that commonsense reasoners at times come to hold conflicting beliefs (temporarily) which can serve to signal that the reasoner's past beliefs must be re-assessed and revised. In most formal AI treatments, contradictions are anathema since most logics become useless in their presence. However human reasoning is not usually thrown into such disarray by contradictions. Thus we have sought formal ways to be more accommodating of contradictions. Little more than lip-service has been paid to the treatment of contradictory information in commonsense reasoning. Probably this is due to the customary reliance on standard logics having the "ex contradictione quodlibet" feature: from a contradiction all is entailed. We refer to this as the "swamping" problem. There are non-standard logics, the paraconsistent logics, that do allow contradiction without swamping; however, in commonsense reasoning one wants not only to avoid swamping but also to somehow undo or at least cease believing the contradiction. Early active logic work had a way to *ignore* contradictions (or more precisely to *note* and then *disinherit* direct, simultaneously occurring contradictions; some α and $\neg\alpha$ appearing together as theorems at some step i). But more is needed. Not only must we adjudicate between contradictands, we must also prevent earlier mistaken beliefs (revealed by contradiction) from infecting *future* reasoning. Conflicting beliefs, mistaken beliefs, and their consequences must be controlled, so as not to infect other beliefs indefinitely into the future.

Recovering from contradiction was broached in [22], but only in an ad hoc way. There a conjecture was formulated, to the effect that, under (unspecified) circumstances, an active logic should be able to regain consistency from an initially inconsistent set of beliefs. In this section we discuss some inroads we have made. In particular we describe the first non-trivial class of active logics which we have developed that, under suitable, yet reasonable, conditions "recover" from *direct contradictions* (our *dc-recovery* theorem). In short this means that antecedent theorems which have led to direct contradictions, consequential theorems derived from direct contradictands, and the direct contradictands themselves are all rendered harmless while other theorems persist.

The technique described here amounts to importing much of a truth-maintenance, or belief revision, system[46] *into* the logic, which then – unlike a usual belief revision system – operates *during and as part of* the ordinary reasoning of the logic. This means that world knowledge can be brought to bear on the truth-maintenance (belief update) process, and other reasoning need not be halted while the belief updating is occurring.

5.1 The lingering consequences and causes of contradictions

Early active logics rely heavily on the rules OBS, MP, and INH (See rules 2, 3, and 7 of Figure 3 on page 15).[47]

Suppose we apply these rules to the observation function OBS_1:

$$OBS_1(j) = \begin{cases} P, P \rightarrow Q & \text{if } j = k \\ \neg P & \text{if } j = k + n \\ \emptyset & \text{otherwise} \end{cases}$$

for fixed $k, n > 0$. Notice what happens (see Figure 11): P and $P \rightarrow Q$ will be (the only) k-theorems and so by MP, Q will become a $k+1$-theorem. Then (at step $k+n$) $\neg P$ is "observed", causing a direct contradiction and the disinheritance of both P and $\neg P$ (see the stipulation on the rule INH). But Q persists, though its only "derivation" is *questionable* as it relies on P, which is itself now unreliable since it conflicts with later observation of $\neg P$.

Here Q, a *consequence* of a theorem (belief) which is not "trustworthy" lingers beyond the step marking the disinheritance of its justification (P). Moreover, in this case Q will be inherited, and hence appear as a theorem, at *every* step $i > k+1$. Intuitively, at least in some cases, this behavior is undesirable; once P is "disbelieved", so too should be Q.

An even more pathological, though related, difficulty arises if we instead consider OBS_2:

[46] a la Doyle and deKleer (see [15, 13])

[47] This discussion will not consider the rules of *extended MP* and *negative introspection* which also appear in Figure 3. Those rules are important; however, they do not seem to help alleviate the pathological behavior we discuss here.

$$\begin{array}{rl}
\mathbf{k}: & \underline{P, P \rightarrow Q} \\
\mathbf{k+1}: & P, P \rightarrow Q, \underline{Q} \\
\vdots & \vdots \\
\mathbf{k+n}: & P, P \rightarrow Q, Q, \underline{\neg P} \\
\mathbf{k+n+1}: & P \rightarrow Q, Q, \underline{Contra(\{P, \neg P\}, k+n)}
\end{array}$$

Figure 11: A belief (Q) based on a questionable former belief (P) persists.

$$OBS_2(j) = \begin{cases} Q, Q \rightarrow R, Q \rightarrow \neg R & \text{if } j = k \\ \emptyset & \text{otherwise} \end{cases}$$

Here, each of the wffs Q, $Q \rightarrow R$ and $Q \rightarrow \neg R$, will persist as theorems indefinitely. The rule MP then will be used at each step to produce as theorems at the next step the (direct) contradiction R and $\neg R$ (see Figure 12).[48]

$$\begin{array}{rl}
\mathbf{k}: & \underline{Q, Q \rightarrow R, Q \rightarrow \neg R} \\
\mathbf{k+1}: & Q, Q \rightarrow R, Q \rightarrow \neg R, \underline{R, \neg R} \\
\mathbf{k+2}: & Q, Q \rightarrow R, Q \rightarrow \neg R, \underline{Contra(\{R, \neg R\}, k+1), R, \neg R} \\
\mathbf{k+3}: & Q, Q \rightarrow R, Q \rightarrow \neg R, Contra(\{R, \neg R\}, k+1), \underline{Contra(\{R, \neg R\}, k+2), R \neg R}
\end{array}$$

Figure 12: The contradiction $\{R, \neg R\}$ is reproven at each step.

We might try to alleviate these problems by restricting the application of MP and INH. For instance: (i) if both α and its direct contradiction appear at some step i then INH should not apply to the contradictands, causing them to be disinherited at step $i+1$, and (ii) if α and $\alpha \rightarrow \beta$ are both i-theorems and so too is the direct contradictand of either, then MP should not apply to produce β as an $i+1$ theorem. The idea here is to (i) prohibit direct contradictands from being inherited, and (ii) restrict the use of MP to antecedent wffs whose contradiction(s) is(are) not "current" theorems.

Unfortunately these restrictions are insufficient to prevent the continual re-emergence of contradictions in certain cases. As long as the root cause of a contradiction persists, and no other action is taken, the contradiction will periodically re-arise (see Figure 13, in which we again use OBS_2 and augment MP and INH with these new stipulations).[49]

A more comprehensive solution must take into account the way inference is chained over the course of steps in active logics. Any given i-theorem α may have been proven in any number of ways, where each distinct proof is based on (other) theorems appearing at previous steps. We can view α as the root of a proof tree whose nodes are the theorems used in "deriving" α and whose branches represent distinct proofs of α. If we record the collection of wffs which appear on each branch of α's proof tree, along with α (at each step at which α appears), then we can use this information, in some cases, to (i) remove unwarranted consequences of contradictions and (ii) prevent a contradiction from re-emerging.

[48] At the same time both R and $\neg R$ are also disinherited at each step beyond $k+2$ because of the stipulation placed on INH which prohibits the inheritance of any $Contra$-ed theorems.

[49] These new stipulations are nevertheless beneficial and will be used in the logic described shortly. (See INF_{deriv}, Figure 14 on page 31.)

$$\begin{array}{rl}
\mathbf{k}: & \underline{Q, Q \rightarrow R, Q \rightarrow \neg R} \\
\mathbf{k+1}: & Q, Q \rightarrow R, Q \rightarrow \neg R\underline{R, \neg R} \\
\mathbf{k+2}: & Q, Q \rightarrow R, Q \rightarrow \neg R, \underline{Contra(\{R, \neg R\}, k+1)} \\
\mathbf{k+3}: & Q, Q \rightarrow R, Q \rightarrow \neg R, Contra(\{R, \neg R\}, k+1) \\
\mathbf{k+4}: & Q, Q \rightarrow R, Q \rightarrow \neg R, Contra(\{R, \neg R\}, k+1)\underline{R, \neg R} \\
\mathbf{k+5}: & Q, Q \rightarrow R, Q \rightarrow \neg R, Contra(\{R, \neg R\}, k+1), \underline{Contra(\{R, \neg R\}, k+3)}
\end{array}$$

Figure 13: The contradiction $\{R, \neg R\}$ will alternately arise and then be disinherited.

5.1.1 *dc*-recovery: Some Preliminary Definitions

Let $SL(OBS, INF)$ be an arbitrary active logic with inference rules all of the form:

$$\frac{k: \quad \beta_1, \ldots, \beta_n}{k+1: \quad \alpha}$$

Definition 5.1 *If* $\vdash_{i+1} \alpha$ *resulted from the application of an inference rule whose i antecedents are* $\beta_1, \ldots, \beta_n$ *then a* derivation of α at step $i+1$ *is a (possibly empty) set of theorems S containing exactly each of* $\beta_1, \ldots, \beta_n$ *and each wff in every derivation* S_j *(at step i) of* β_j*, for* $1 \leq j \leq n$*. (When a step number is understood we will simply say "derivation" instead of "derivation at step i". When we wish to call attention to the derivation S of* α *we write* $\alpha[S]$*.)*

Note that a theorem α may have more than one derivation at a step. For instance if MP is a rule of the logic and P, R, $P \rightarrow Q$, $R \rightarrow Q$ are all k-theorems then Q may have two different derivations at $k+1$; one including P and $P \rightarrow Q$ (and the theorems appearing in each of their respective derivations), and the other including R and $R \rightarrow Q$ (and the theorems appearing in each of their respective derivations).

Definition 5.2 *Let* $\vdash_i \alpha$*, then* α *is* distrusted *at step* $i+1$ *iff:*

(i) $\vdash_i \neg\alpha$ *or if* α *is of the form* $\neg\beta$ *and* $\vdash_i \beta$ *(that is* α *is part of a direct contradiction which appears at step i), or*

(ii) $\exists\beta$ *such that both* $\vdash_i \beta[S_1]$ *and* $\vdash_i \neg\beta[S_2]$ *and* $\alpha \in S_1$ *or* $\alpha \in S_2$*, or*

(iii) each derivation of α *at step i contains at least one wff which itself is distrusted at step* $i-1$*.*

We will use the predicate symbol Distr to assert that α *is distrusted at some step k as in* $Distr(\alpha, k)$*.*

Intuitively definition 5.2 says that an i-theorem is considered distrusted at step $i+1$ if either (i) its negation is also an i-theorem, (ii) it led to a direct contradiction, or (iii) each of its derivations contains a distrusted theorem.

Definition 5.3 *An active logic* $SL(OBS, INF)$ dc-recovers *if* $\exists j$ *such that* $\forall k > j\ \neg\exists\alpha \vdash_{k+1}\ Distr(\alpha, k)$*.*

Definition 5.3 says this: an active logic dc-recovers if there is a step j such that for any subsequent step k, if α is a k-theorem then α will not be distrusted at step $k+1$.

Definition 5.4 *An active logic* $SL(OBS, INF)$ *is* eventually free of direct contradictions *if* $\exists j$ *such that* $\forall k > j$ *and* $\forall\alpha$*, either* $\not\vdash_k \alpha$ *or* $\not\vdash_k \neg\alpha$*.*

Lemma 5.5 *If $SL(OBS, INF)$ dc-recovers then $SL(OBS, INF)$ is eventually free of direct contradictions.*

Proof: If $SL(OBS, INF)$ dc-recovers then $\exists j$ such that $\forall k > j$ no k-theorem is $k+1$ distrusted by definition 5.3. Thus $\forall k < j, \alpha$ either $\not\vdash_k \neg\alpha \not\vdash_k \alpha$ by definition 5.2(i). Hence $SL(OBS, INF)$ is eventually free of direct contradictions. ■

Notice that the logics discussed thus far in this section do not *dc*-recover.

5.1.2 An Active-logic with the dc-recovery Property

In this section we will develop an active logic which does dc-recover given certain restrictions on its OBS function. (It will turn out that both OBS_1 and OBS_2 given earlier satisfy these constraints.)

We begin by introducing derivations formally into the logic. This is done using the inference function INF_{deriv} given in Figure 14.

Rule 1	$\dfrac{\mathbf{i}:}{\mathbf{i+1}:\alpha}$	, if $\alpha \in OBS(i+1)$
Rule 2	$\dfrac{\mathbf{i}:\alpha[S]}{\mathbf{i+1}:\alpha[S]}$	Inheritance[a]
Rule 3	$\dfrac{\mathbf{i}:\alpha[S_1], \alpha \rightarrow \beta[S_2]}{\mathbf{i+1}:\beta[\{\alpha, \alpha \rightarrow \beta\} \cup S_1 \cup S_2]}$	MP[b]
Rule 4	$\dfrac{\mathbf{i}:\alpha[S_1],\ \neg\alpha[S_2]}{\mathbf{i+1}:Distr(\alpha, i), Distr(\neg\alpha, i)}$	Contradiction Distrusted
Rule 5	$\dfrac{\mathbf{i}:\alpha < S_1, \ldots, S_m >, Distr(\beta_1, i-1), \ldots, Distr(\beta_n, i-1)}{\mathbf{i+1}:Distr(\alpha, i)}$	Distrust Consequences[c]
Rule 6	$\dfrac{\mathbf{i}:\alpha[S_1], \neg\alpha[S_2], \beta[S_3]}{\mathbf{i+1}:Distr(\beta, i)}$	, if $\beta \in S_1$ or S_2. Distrust Antecedents

[a] Where $\not\vdash_i Distr(\alpha, i-1)$, $\not\vdash_i \neg\alpha$, and for each $\beta \in S \not\vdash_i Distr(\beta, i-1)$. Also, if α is of the form $\neg\gamma$ then this rule does not apply if $\vdash_i \gamma$.

[b] The stipulations placed on the antecedent of rule 2 apply to each of $\alpha[S_1]$, $\alpha \rightarrow \beta[S_2]$, and β here.

[c] Where each S_k contains at least one of $\beta_1, \ldots, \beta_n$ and α is not of the form $Distr(\gamma, j)$.

Figure 14: INF_{deriv}

In Figure 14 the following abbreviations are used:

(1) α abbreviates $\alpha[\emptyset]$; i.e., we simply write α when α's derivation is the empty set.[50]

(2) $\vdash_i \alpha < S_1, \ldots, S_n >$ if and only if $\vdash_i \alpha[S_1], \ldots, \alpha[S_n]$ and there is no S such that $\forall k$, $1 \leq k \leq n$, $S \neq S_k$ and $\vdash_i \alpha[S]$; that is $S_1, \ldots, S_n$ are *exactly* all of α's derivations at step i.

Notice that derivations distinguish instances of theorems so that if $\vdash_i \alpha$ and α has multiple derivations at i, say $S_1, \ldots, S_n$, then each of $\alpha[S_1]$,…,$\alpha[S_n]$ will appear as i-theorems.

[50](Since the limitations we will place on OBS (see the statement of the dc-recovery theorem, Section 5.2) makes the derivation of any theorem of the form $Distr(\alpha, i)$ irrelevant, we annotate wffs of the form $Distr(\alpha, i)$ with $[\emptyset]$.)

The idea behind each of the rules of INF_{deriv} is this:

- **Rule 1:** (OBS) The derivation of an observation is empty indicating that no other beliefs have been used to derive it.
- **Rule 2:** (INH) The derivation of an inherited belief is unaffected. Inheritance only applies to trustworthy beliefs: Namely, $\alpha[S]$ is inherited from step i to $i+1$ if it is not distrusted, its direct contradiction does not also appear at step i, and no $\beta \in S$ is distrusted. (See stipulation (a) in the figure.)
- **Rule 3:** (MP) The derivation of a belief inferred via MP includes the wffs in the antecedent of MP (i.e., α and $\alpha \rightarrow \beta$) and all wffs contained in each antecedents' respective derivation. MP is applied only to trustworthy wffs as in rule 2 above. (See stipulation (b) in the figure.)
- **Rule 4:** This rule marks a wff as distrusted at step $i+1$ when both it and its direct contradiction appear at step i. (Note: The predicate symbol *Contra* is not used here but it will return in the next chapter.)
- **Rules 5 and 6:** These rules track down the consequences of *Distr*-ed beliefs (rule 5) and the antecedents of contradictory (distrusted) beliefs (rule 6). Rule 5 marks as *Distr*-ed at step $i+1$ any belief whose only derivations each contain a theorem distrusted at step $i-1$. (Notice that if *any* of an i-theorem's derivations contain an distrusted wff, those instances of the wff will not appear at step $i+1$ due to the stipulations placed on rules 2 and 3, regardless of the applicability of rule 5.) Rule 6 marks as *Distr*-ed any (antecedent) wff which appears in the derivation of a contradictory wff. That is, beliefs leading to a contradiction are themselves marked as distrusted.

A very simple example of INF_{deriv} at work is based on the following observation function:

$$OBS_3(j) = \begin{cases} P, P \rightarrow Q, R, R \rightarrow Q & \text{if } j = 1 \\ \neg P & \text{if } j = 2 \\ \emptyset & \text{otherwise} \end{cases}$$

The resulting sequence of steps is shown in Figure 15. Derivations are in **bold** type.

$$\begin{array}{ll}
\mathbf{1:} & \underline{P, P \rightarrow Q, R, R \rightarrow Q} \\[1em]
\mathbf{2:} & P, P \rightarrow Q, R, R \rightarrow Q, \\
 & \underline{\neg P, Q[\{\mathbf{P}, \mathbf{P} \rightarrow \mathbf{Q}\}], Q[\{\mathbf{R}, \mathbf{R} \rightarrow \mathbf{Q}\}]} \\[1em]
\mathbf{3:} & P \rightarrow Q, R, R \rightarrow Q, \\
 & Q[\{\mathbf{P}, \mathbf{P} \rightarrow \mathbf{Q}\}], Q[\{\mathbf{R}, \mathbf{R} \rightarrow \mathbf{Q}\}] \\
 & \underline{Distr(P, 2), Distr(\neg P, 2)} \\[1em]
\mathbf{4:} & P \rightarrow Q, R, R \rightarrow Q, \\
 & Q[\{\mathbf{R}, \mathbf{R} \rightarrow \mathbf{Q}\}], Distr(P, 2), Distr(\neg P, 2)
\end{array}$$

Figure 15: INF_{deriv} at work.

Notice the two instances of Q at step 2 each with a distinct derivation, one of which contains P which itself contradicts $\neg P$, also appearing at step 2. At step 3 the contradictands P and $\neg P$ are marked as distrusted and have not been inherited, though one derivation of Q at this step contains the distrusted contradictand P. This instance of Q, the one with P in its derivation, is disinherited at step 4 by stipulation (a) placed on INH which restricts inheritance to those instances of theorems containing no distrusted wffs in their derivations. By step 4 then, only one "clean" derivation of Q remains (and will continue to persist for all steps $i > 4$).

5.2 The dc-recovery Theorem

$SL(OBS, INF_{deriv})$ is the first non-trivial active-logic that we have developed that has the dc-recovery property, given OBS satisfies certain reasonable constraints.

Theorem 5.6 (**dc-Recovery Theorem for** $SL(OBS, INF_{deriv})$) *Let OBS be finite and Distr-free*[51]*. Then $SL(OBS, INF_{deriv})$ dc-recovers.*

The proof can be found in [92].

5.3 Discussion

Though dc-recovery is a desirable property of active logics, so too is is the property that those wffs not "involved" in direct-contradictions remain unaffected by the dc-recovery process. We are currently working on a characterization of the set of theorems which survive the recovery process of $SL(OBS, INF_{deriv})$, which we denote by THM_{OBS}. We note two characterizations which do *not* apply to THM_{OBS}: First, *if O is the set of all theorems introduced by OBS, and M is a minimal subset of O whose complement, $\overline{M}$, is consistent, then $\overline{M} \subseteq THM_{OBS}$.* (Nor, should it be.) To see this let O be $\{P, \neg P\}$. Then $\overline{M} = \{P\}$ or $\{\neg P\}$. But notice that regardless of the step at which each of P and $\neg P$ is introduced via OBS, they will simultaneously appear at some step i. Thus they will both be disinherited at $i+1$, never to re-appear. Hence $\mathrm{THM}_{OBS} = \emptyset$. On the other hand it is not always the case that $\mathrm{THM}_{OBS} = \emptyset$ as illustrated in Figure 15.

The logic described in this section maintains and searches through derivations at every step in the deductive process. It might be argued that this is too computationally expensive a task. This is true of reasoning that is comprised of long chains of inferences, as in mathematical reasoning where derivations may be very long. It is also true of reasoning that relies on many simultaneous corroborations of the same hypothesis, as in some scientific reasoning where there are many derivations of the same theorem. But commonsense reasoning seems to be a different sort of process, one that is often (though not always) characterized by lots of world knowledge and rather (i) short chains of reasoning and (ii) limited or lazy corroborations of beliefs.

One way to look at this "short-chain" hypothesis is that commonsense reasoners frequently touch base with reality, by getting external inputs, e.g., direct observation, testing, questioning, etc. Thus the reasoning gets regular validations or corrections, which can perhaps appropriately be treated a bit like new axioms. Of course, in commonsense reasoning axioms do not have a rigidly fundamental character as in mathematics, since we need to be able to account for error even in observations. Observations, then, may begin new chains of reasoning. Maintaining these short chains (derivations) has a computationally negligible effect.

The "lazy-corroboration" hypothesis asserts that we typically do not seek many independent corroborations or "proofs" (derivations) of our beliefs. This is not to say that we deliberately avoid corroborations, nor that we always feel content with just one or two. There are times when it becomes extremely important to secure as much evidence as possible before accepting a belief; say a plan to escape in a life and death situation. But, in general, we tend to readily accept beliefs and seek corroborations only as needed; we take a "lazy" approach to belief corroboration. As I look out the window and see what I think is my truck in the parking lot I simply believe that it *is* my truck. I don't have to go outside and try the key in the door, or check the vehicle's identification number, or peek through the windshield to see the empty coffee mug I left in there this morning to help verify that it is, indeed, my truck.

6 Language changes

> "Did you hear that John broke his leg?"
> "No, really? That's a shame!"
> "Yes, and his wife now has to do
> everything for him."
> "Wife? John isn't married. Which
> John are you talking about?"

[51] See pages 20 and 20 for definitions of *finite* and *Distr-free*, respectively

"I'm talking about John Jones."
"Oh, I thought you meant John Smith."

The above apparently mundane conversation hides some very tricky features facing any formal representational and inferential mechanism, whether for use in natural language processing, planning, or problem-solving. For here occurs an implicit case of language control. As it dawns on the two speakers above that they are using the name "John" differently they need to reason about usage and adopt a strategy to sort out the confusion, e.g., by using last names too.

The ability of a reasoning agent to exercise control of its own reasoning process, and in particular over its language, has been hinted at a number of times in the literature. Rieger seems to have been the first to enunciate this, in his notion of referenceability [121], followed by others [108], [85], etc.

The underlying idea, as we conceive it here, is that the tie between linguistic entities (e.g., words) and their meanings (e.g., objects in the world) is a tie that the agent had better know about and be able to alter when occasion demands. This has a number of important commonsense uses, which have been listed elsewhere [109].

The formal point, though, is that a new treatment is called for so that rational behavior via a logic can measure up to the constraint that it be able to change usage, employ new words, change meanings of old words, and so on. The usual fixed language with a fixed semantics that is the stock-in-trade of AI seems inappropriate to this task.

Active logic seems to adapt well to the specific issue of language change. Referenceability, to stick with Rieger's terminology, demands that the agent – and therefore the agent's language – have expressions available to denote expressions themselves (e.g., via quotation) and also to denote the tie between an expression and what it stands for. The form that this word-object tie takes seems to vary according to context,[52] and that is what this paper will focus on, by examining several specific commonsense settings.

Traditional descriptions of nonmonotonic reasoning envision nonmonotonicity as a relationship between theories: from one theory certain theorems follow that do not follow when that theory is augmented with additional information (axioms). However, this relationship is expressed only in the meta-theory; the usual logics pay attention to behavior only within a given theory. On the other hand, "theory change" is the central feature of the *step-logic formalism*. In brief, a step-logic models belief reasoning by sanctioning inference one-step-at-a-time, where the time of reasoning is integral to the logic. Complicated reasoning made of many successive inferences in sequence take as many steps as the sequence contains. Error, change of mind, change of language, and change of language usage all are time-tempered in that they are appropriately characterized only with regard to a historical account of beliefs, language, and its usage. The one-step-at-a-time approach offers a natural account of such histories.

A key informal idea for us will be that of a *presentation*, which means roughly a situation or context in which attention has been called to a *presumed* entity, but not necessarily an entity we have a very clear determination of at first.[53] This, we argue, is the case in virtually all situations initially, until we get our bearings. But before we actually make an identification we determine (perhaps unconsciously) that there is *something* for us to deal with. This is a small point as far as initial matters go, but becomes important if later we decide to change our usages. Some examples will help. We have devised a formalism that "solves" these example problems and have implemented our solution to some of the problems in Prolog. Space allows only a brief sketch of certain underlying mechanisms.[54]

6.1 Rosalie's Car

A car flashes by us, and we quickly identify it as Rosalie's car (which for simplicity we denote rc). We may be unaware of any recognition process, thinking simply that we see rc flash by. Then we notice that the license plate on the car is not what we would expect to see on rc, and we re-assess our belief that we are seeing rc. Something, we tell ourselves, made us think *this* (the car we see driving away) is *that* (the car rc we already knew of from earlier times). Once we have produced appropriate internal tokens, we can then say that we mistook *this* for *that*. The something-or-other that brought about our

[52] Recently, McCarthy and others have been investigating formal theories of context ([83], [46]). The implications this may have for our work are, at this point, unclear.

[53] The vagueness in our notion of presentation does not, at this stage, hinder our formal treatment. However, we believe it will be necessary to clarify this notion. This is the focus of ongoing work. Among other things, it will involve a focus of attention, as hinted at by our informal "this" and "that" description below.

[54] See [92] and [93] for more complete details.

mistake is what we call a presentation. It will not play a formal role for us, but simply a motivational one in leading us to our formal devices.

How can we formalize the notion of taking *this* for *that*? We begin by looking into the relationship between the two – not a physical relationship, as in features that the two cars may share (though this may ultimately have a bearing on belief revision) but rather a cognitive relationship between the entities. This relationship is suggested in the case of the mistaken car by the English statement, "I mistook *this* car to be *that* (Rosalie's)." The *this* here can be viewed as a demonstrative which (together with an appropriate demonstration) is used to pick out the mistaken car, the one which passed by. The *that* can be viewed as another demonstrative which is used to pick out rc. The statement, "I mistook this car to be Rosalie's", indicates a cognitive tie between two objects, automobiles in this case, that are in a sense linked in a (former) belief by the term rc.

Essentially what has happened is this: Initially, we are aware of an interest in one car only: Rosalie's; then later, in two: Rosalie's and the car that flashed by (i.e., the car *mistakenly identified to be* Rosalie's). In a sense, the term 'rc' in the original belief 'rc just went by' refers to both of these cars.[55] That is, we had rc in mind but connected a "mental image" of it to the wrong car, the one that flashed by. As such, beliefs about the incident reflect an unfortunate mental conflation or compression of these two cars that must be torn apart in the reasoning process.[56]

We use the 4-*ary* predicate symbol **FITB** to state that an object of perception (presented at some time or *step*) is at first identified to be some (other) object, thereby producing a (set of) belief(s), i.e., $FITB(x, y, S, i)$ says that object of perception, x, which was presented at step i, is at first identified to be y producing the beliefs in the set S. Then we use Russell's ι-operator à la Russell to pick out the *this* that was mistaken for *that*, e.g., $\iota x FITB(x, rc, \{FlashedBy(rc)\}, t)$ – "the unique object of presentation, presented at step t, which was at first identified to be rc which produced the belief $FlashedBy(rc)$." This *reality term* is used to denote what a reasoner currently takes to be some entity, possibly filling in for a previously held, but incorrect description of the same entity. (As a shorthand convention we use $tfitb(y, S, i)$, "the thing (object of presentation) which was at first identified to be ...", in place of $\iota(x)FITB(x, y, S, i)$.) By incorporating reality terms we are able to express certain errors of object misidentification reflected in one's former beliefs, for instance: $tfitb(rc, \{FlashedBy(rc)\}, t) \neq rc$ – "the unique object of presentation which was at first identified to be rc at step t, which produced the belief $FlashedBy(rc)$, is not rc. (We abbreviate assertions of the form $tfitb(t, S, i) \neq t$, (which we call *tutorials*) by $MISID(t, S, i)$.) Asserting the error sets in motion a belief revision process which is characterized, in part, by the following: The earlier belief $FlashedBy(rc)$ is disinherited,[57] i.e., the step-logic ceases to have that belief, although it does retain (as a belief) the historical fact that it once had that belief, and

$$FlashedBy(tfitb(rc, \{FlashedBy(rc)\}, t))$$

is produced.

Just how does one come to suspect and detect erroneous beliefs? We have already alluded to one answer, namely that we come to suspect an error upon noting competing or incoherent beliefs. We may suspend the use of potentially problematic beliefs, perhaps speculating and hypothesizing about alternative views of the world, in an effort to hash out the difficulty. How does one decide just which alternative to have faith in? In some cases one may use a hypothesize-and-test process to ferret out the problem from the set of possible errors that might have been made. A complete principled account of how one speculates and then confirms or denies her suspicions is beyond the scope of this paper.[58] Instead, a simplifying assumption is to postulate a *tutor* or an advisor that can tell us about our errors.[59] The tutor plays the role of a friend who says, "Hey, that's not Rosalie's car". How the agent comes to represent and use the friend's advice is the issue we are addressing.

6.2 One and Two Johns

Our **One John** example is very similar to that of rc above, but will help us in moving toward the third example below. Here we imagine that we are talking to Sally about a third person, whom we initially come to identify as our friend John, merely in virtue of matching John to Sally's description of the person, or the context of the conversation, etc., but not in virtue of hearing Sally use the name "John". Later we find out it is not John, but someone else.

[55] We assume that beliefs are symbolically represented inside the head in some mental language. [32]
[56] The term *compression* is borrowed from [78].
[57] Disinheritance is a fundamental feature of step-logic. In particular, when two simultaneously held beliefs are in direct contradiction, neither is inherited to the next step, although either may later be re-proven by other means. Another way disinheritance allows the agent to cease believing a wff, that we introduce here, is based on a misidentification.
[58] It is likely that default reasoning is involved as is knowledge about the likelihood of errors (e.g., a car is likely to be misidentified since there are typically many similar looking cars).
[59] See [80] for a discussion about programs and advice taking.

There is no appropriate entity before us in *perception* which has been misidentified as in the case of the mistaken car; rather it is an abstract entity, a someone-or-other, still an object of presentation, the person that Sally had in mind. There is *this* someone that has been taken to be *that*, John. Our formalism treats abstract (objects of) presentation(s) of this sort much like the case of rc.

Now let us extend this to the **Two Johns** case: We are in a situation in which we are presented with a notion of a person, whom we (come to) think is our friend John. Then we are led to believe that he has a broken leg and his wife has to do everything for him. Later we suspect that there is a confusion, that not everything we are hearing makes sense. (John, our friend, is not married.) Is Sally wrong? Or have we got the wrong person in mind? Now here is the twist: Sally starts employing the name "John" to refer to this person.[60] Perhaps she is talking about a different John. To even consider this option we need to be able to "relax" our usage so that "John" is not firmly tied to just one referent. And later when Sally says that she is talking about John Jones, not our friend, John Smith, we need a way to refer to the two entities without *using* the term $John$. We may continue to *mention* the name, but judiciously, as it is ambiguous.

We can try to employ the same formal strategy that the agent used above. Namely, we may initially come to suspect that

$$tfitb(john, BrokenLeg(john)) \neq john$$

which has the English reading: "the unique object of presentation which was at first identified to be John, producing the belief $BrokenLeg(john)$, is not John." But then once we hear Sally use the the name "John" to refer to the person with the broken leg, whom we now believe is not our friend John, more must be done – the name "John" must be disambiguated.

This is where we must exhibit control over our language and language usage. First the ambiguity must be recognized. That is, we must come to see that *this* and *that* share the same name. Once that is done, new terms should be created, each to unambiguously denote one of the two Johns.

Proper naming and the use of names is made explicit with the the predicate symbol **Names**. We write $Names(x, y, i)$ to state that x names object y which first came to be known (by the reasoner) at time or step i; this could be weakened to time $\leq i$, or time $\geq i$, etc., if the exact time is not known. Including the third argument is somewhat non-standard, though not without a commonsense basis. We usually have at least a vague idea of when we come to know about someone. We can think of $Names(x, y, i)$ as collapsing $IsNamed(x, y) \wedge FirstLearnedAbout(I, y, i)$, where I is intended to be the first person pronoun.

To make ambiguity precise the binary predicate symbol **Amb** is used to state that a name does not refer uniquely beyond a certain step. Axiom **AM** expresses this:

AM : $(\forall x)(\exists yzij)\{(Names(x, y, i) \wedge$
$Names(x, z, j) \ \wedge y \neq z \ \wedge \ i \leq j) \rightarrow$
$Amb(x, j)\}$

It says that if two different objects share a name, then the name is ambiguous for the reasoner once he became aware of both objects.

Once an ambiguity arises, our reasoner will need to disambiguate any belief using the ambiguous term. We use $RTA(x, y, i)$ to state that object x is referred to as y prior to step i. In particular if $Names(x, y, j)$ then $RTA(x, y, k)$ for $k > j$, $trta(y, i)$ is used an abbreviation for:

$$\iota x RTA(x, y, i)$$

"the unique thing referred to as y prior to step i", itself a non-ambiguous reality term.

Figure 16 gives a brief sketch of the evolution of reasoning we have in mind. In the figure we use M, BL, j, and 'j to abbreviate $Married$, $BrokenLeg$, $john$, and '$john$ respectively. Also $j1$ is used to abbreviate the expression $trta(`j, 2)$, i.e.,

$$j1 = \iota x RTA(x, `j, 2)$$

namely " the unique thing referred to as 'john' prior to step 2", and $j2$ is used to abbreviate the expression $tfitb(trta(`j, 2), \{M(j), B(j)\}, 2)$, i.e.,

$$j2 = \iota x FITB(x, \iota y RTA(y, `j, 2), \{M(j), B(j)\}, 2)$$

[60] The sequence of events here is different than that reflected in the dialogue at the beginning of this abstract. Specifically, Sally uses the name "John" here only *after* we come to think that she is talking about our friend John. In the full paper we also discuss another version, in which Sally uses the name "John" at the outset.

namely "the unique thing which was first identified to be the the unique thing referred to as 'john' prior to step 2, which produced the beliefs $Married(john)$ and $BrokenLeg(john)$ at step 2." The predicate symbol $Contra$ indicates a contradiction between its arguments, a signal to the reasoner that something is amiss thereby initiating a belief revision process.[61]

---

1 : $\underline{\neg M(j), Names(`j, j, -\infty)}, AM$

2 : $\ldots, \underline{BL(j), M(j)}$
(**Sally** : "$...hislegisbrokenandhiswife...$")

3 : $AM, \underline{Names(`j, j, -\infty), Contra(\neg M(j), M(j))}$
(**Agent** : "$Impossible! He isn't married.$")

4 : $\ldots, \underline{MISID(j, \{M(j), B(j)\}, 2)}$
(**Sally** : "$You misidentified who I'm talking about.$")

5 : $AM, \underline{M(tfitb(j, \{M(j), BL(j)\}, 2))}, \underline{BL(tfitb(j, \{M(j), BL(j)\}, 2))}$
(**Agent** : "$So that's what's wrong.$")

6 : $\ldots \neg M(j)$
(<Reinstate Marital Belief>)

7 : $\ldots, \underline{Names(`j, tfitb(j, \{M(j), BL(j)\}, 2), 2)}$
(**Sally** : "$I'm talking about John.$")

8 : $\ldots, \underline{Amb(`j, 2)}$
(**Agent** : "$Oh, they have the same name!$")

9 : $AM, \underline{\neg M(j1), M(j2), BL(j2), Names(`j, j1), Names(`j, j2)}, \underline{j2 \neq j1},$
(**Agent** : "$Now I've got it.$")

Figure 16: Sketch of the *Two Johns* story.

We can view each step as a discrete moment in the reasoning process. Formulae associated with each step are intended to be (some of) those relevant to the story as time passes. At each step, underlined wffs reflect beliefs newly acquired at that step. Others, in step-logic terminology, are *inherited* from the previous step. Ellipses indicate that *all* beliefs shown in the previous step are inherited to the current step.

Beliefs at step 1 are those held before the agent's conversation with Sally and those at step 9 reflect an unambiguous account of the two Johns, one now denoted by $j1$ and the other by $j2$, once the problem is sorted out. In between are steps whose beliefs reflect information acquired via the conversation with Sally (steps 2 and 7) and via her advice (step 4); steps whose beliefs reflect that problems have been noted (a contradiction is noted in step 3 and the ambiguity is noted in step 8); and steps reflecting disinheritance (going from step 2 to 3, and from step 5 to 6).

The indicated steps have the following intuitive gloss: (1) the agent believes that John is not married, and is named "John". Then (2) comes to believe his leg is broken and he is married. This produces a contradiction, noted in (3), so neither marital belief is retained. Advice is then taken that John has been misidentified (4) which leads to the retraction (disinheritance) of the belief that John has a broken leg (6). The agent learns that the 'other person' is named "John" (7), notes the ambiguity (8), and takes corrective action (9) by creating and incorporating the unambiguous terms $j1$ and $j2$, one for each John.

[61] E.g., suspending the use of potentially problematic beliefs, in particular the contradictands and their consequences. See [27] for details.

6.3 Formal Treatment

There are several notable features of the stepped approach to reasoning illustrated in the previous section which will need to be preserved in a formal device applied to the specific issue of reasoning about former beliefs. Most conspicuous is that the reasoning be situated in a temporal context. As time progresses, a reasoner's set of currently accepted beliefs evolves. Beliefs become former beliefs by being situated in an ever changing "now", of which the reasoner is aware.

Secondly, inconsistency may arise and when it does its effect should not be disastrous; rather it should be controllable and remedial, setting in motion a fairly broad belief revision process, which includes belief retraction.

Finally, the logic itself must be specially tailored to be flexible or "active" enough to allow, even encourage, language change and usage change when necessary. As a theoretical tool the general step-logic framework developed in [27] and [22] is well suited to these desiderata.

A step-logic models reasoning by describing and producing inferences (beliefs) one step at a time, where the time of reasoning is integral to the logic. Complicated reasoning made of many successive inferences in sequence take as many steps as that sequence contains. A particular step-logic is a member of a class of step-logic formalisms; each particular step-logic is characterized by its own *inference* and *observation* functions (illustrated below).

One distinguishing feature of step-logics is that only a finite number of beliefs (i.e., theorems) are held at any given discrete time, or *step*, of the reasoning process. Thus we can view each step as a discrete moment in a reasoning process.

Let α, β, and γ (with or without subscripts) be wffs of a first-order language $\mathcal{L}$ and let $i \in \mathbf{N}$. The following illustrates what a step in the modeled reasoning process of a step-logic looks like.

$$\mathbf{i}\text{: } \alpha,\ \beta,\ \gamma,\ \ldots$$

represents the belief set of the agent being modeled at step i, i.e., if it is now step (or time) i then α, β, and γ are currently believed.

A wff becomes an i-theorem (roughly, a belief a step i) in virtue of being proven (inferred) at step i. Proofs are based on a step-logic's inference function, which extends the historical sequence of beliefs one step at a time. An inference function can be viewed as a collection of inference rules which fire in parallel at each step in the reasoning process to produce the next step's theorems. For every $i \in \mathbf{N}$, the set of i-theorems are just those wffs which can be deduced from the previous step(s), each using only one application of an applicable rule of inference.

Inference rules, in their most general form, adhere to the structure suggested by rule schema **RS** below.

$$\textbf{RS:} \qquad \begin{array}{ll} \mathbf{i-j}: & \alpha_{i-j_1}, \ldots, \alpha_{i-j_m} \\ \vdots & \quad \vdots \\ \mathbf{i}: & \alpha_{i_1}, \ldots, \alpha_{i_n} \\ \hline \mathbf{i+1}: & \beta_1, \ldots, \beta_p \end{array}$$

where $i, j \in \mathbf{N}$ and $(i-j) \geq 0$. The idea behind schema **RS** is this: at any step of the reasoning process the inference of β_1 through β_p as $(i+1)$-theorems is mandated when all of α_{i-j_1} through α_{i-j_m} are $(i-j)$-theorems, and all of α_{i-j+1_1} through α_{i-j+1_r} are $(i-j+1)$-theorems, ..., and all of α_{i_1} through α_{i_n} are i-theorems.

Now we apply this to *Two Johns*. We will discuss several of the important step-logic inference rules which come into play in steps 1 through 9 of figure 16. (Others are treated fully in [92]).[62]

"**Observations**" can be thought of as non-logical axioms or facts which the agent acquires over time. Observations are proven in accordance with rule O:

Rule O: $\quad \underline{\mathbf{i}\text{:} \qquad\qquad}$

[62] Among those not discussed here are rules for inheritance, modus ponens, contradiction handling and other belief disinheritance, and negative introspection.

$$\mathbf{i+1}: \quad \alpha \qquad \text{IF } \alpha \in Obs(i+1)$$

where the function Obs is tailored to correspond to the particular problem to be solved. For *Two Johns* Obs is defined by

$$Obs(i) = \begin{cases} \neg M(j), Names(`j, j, -\infty), AM & if\ i=1 \\ M(j), B(j) & if\ i=2 \\ MISID(j, \{M(j), B(j)\}, 2) & if\ i=4 \\ Names(`j, tfitb(j, \{M(j), B(j)\}, 2), 2) & if\ i=7 \\ \emptyset & \text{otherwise} \end{cases}$$

which indicates beliefs which the agent held prior to "talking with Sally" (those in $Obs(1)$) and those acquired while "talking with Sally" (those in $Obs(2)$, $Obs(4)$, and $Obs(7)$). Thus the use of rule O adds new beliefs at steps 1, 2, 4 and 7 in in the solution to *Two Johns* (as depicted in figure 16).

The "**Misidentification Renaming**" rule (M) takes care of the renaming of a misidentified object in the beliefs produced by the presentation. It says this: If α, containing the term t, was produced by a presentation at step k and a misidentification of t comes to the reasoner's attention at a later step i, then at $i+1$ the reasoner will believe that α holds of the misidentified object (of presentation), i.e., $tfitb(t, S, k)$ where S is a set of wffs and $\alpha \in S$.

Rule M:
$$\frac{\mathbf{i}: \quad MISID(t, S, k)}{\mathbf{i+1}: \quad \alpha(t/tfitb(t, S, k))} \quad \text{WHERE } \alpha \in S$$

In figure 16 rule M applies at step 4 to produce the beliefs $M(tfitb(j, \{M(j), BL(j)\}, 2))$ and $BL(tfitb(j, \{M(j), BL(j)\}, 2))$ which appear at step 5.

The "**Ambiguity Renaming**" rule (A) disambiguates name clashes:

Rule A:
$$\frac{\mathbf{i}: \quad Amb(`x, k), \alpha(x)}{\mathbf{i+1}: \quad \alpha(x/trta(`x, k))}$$

This rule takes an antecedent wff $\alpha(x)$ which *uses* the ambiguous term x and eliminates the offending term replacing it with $trta(`x, k)$, which mentions but does not use $x$. In figure 16 rule A applies at step 8 to produce the beliefs $\neg M(j1)$, $M(j2)$, $BL(j2)$, $Names(`j, j1)$, $Names(`j, j2)$ and $j2 \neq j1$ which appear at step 9. (Recall that both $j1$ and $j2$ abbreviate terms which contain the sub-term $trta(`j, 2)$, which is created by rule A.)

Our full system has seven additional inference rules, including a "**Name-use**" rule that can *appropriately* lead the reasoner into contradiction if names are not disambiguated.

7 Focal Points

Coordination is a central theme of Distributed Artificial Intelligence (DAI). Much of the work in this field can be seen as a search for mechanisms that will allow agents with differing views of the world, and possibly with different goals, to coordinate their actions for mutual benefit.[63]

Kraus, Rosenschein and Fenster [66, 31] consider how automated agents could use a coordination technique common to communication-free human interactions, namely *focal points*. Focal points are prominent solutions of an interaction to which agents are drawn. To discover these prominent solutions, agents must be able to use contextual information, and exploit the relative likelihood that their partner will also be drawn to a particular solution. Standard representation techniques (e.g., classical logic, game theory) are unsuitable for focal point search, either because they abstract away context or because they do not capture the difficulty of finding solutions.

[63] See for example, [133, 21, 6, 135, 73, 67, 124, 59].

7.1 The Focal Point Concept

Originally introduced by Schelling [129, 118], focal points refer to prominent solutions of an interaction, solutions to which agents are drawn. His work on this subject explored a number of simple games where, despite surface equivalence among many solutions, human players were predictably drawn to a particular solution by using contextual information.

Here is a "toy example" that illustrates the Focal Point concept clearly (more examples can be found in [66].) Consider two people who have each been asked to divide 100 identical objects into two arbitrarily-sized piles. Their only concern in deciding how much goes into each pile is to match the other person's behavior. If the two agents match one another, they each win $40,000, otherwise they get nothing. Schelling found that most people, presented with this scenario, choose an even division of 50 objects per pile. They reason that, since at one level of analysis all choices are equivalent, they must focus on any uniqueness that distinguishes a particular option (such as symmetry), and rely on the other person's doing likewise.

There are a number of intuitive properties that seem to qualify a given solution as a focal point. Among these properties are uniqueness, symmetry, and extremeness. However, even when we consider these special properties, more must be done to identify focal points. *There are bound to be competing potential focal points, since there is* something *unique about any solution.*

That is, any solution will have something to recommend it—but the less obvious that something is, the less attractive the alternative becomes, precisely because it becomes less obvious that the other agent will duplicate our line of reasoning. For example, in the "toy example" above, the choice of 10–90 recommends itself, since it is the only choice where the number of tens in both piles is a perfect square (1 squared and 3 squared), and where at the same time the first pile is smaller than the second. This is a farfetched example, but the point should be clear: a focal point is produced not only because it satisfies one of the intuitive principles mentioned above, but because it seems computationally more accessible—it seems more likely that the other agent will also recognize the point than that he will recognize competing points.

Standard logic is not appropriate for providing the solution to focal points. One reason is that computational complexity seems central to identifying focal points. Not only must a solution to a given problem satisfy a property like uniqueness in order to qualify as a focal point, it must also be easier to find than other solutions with similar properties. It is therefore necessary to model the computational process itself in the reasoning procedure as we search for focal points. Classical first order logic does not model the computational process. However, *active logic*, dealing explicitly with the passage of time as an agent reasons, is appropriate.

7.2 The Active-Logic Focal Point Algorithm

For simplicity, we assume that there are two agents. They are given a set of objects and both need to choose the same object, without communication.

The intuition behind our active logic focal point algorithm is that the agent, at each step i, will look for candidates in the domain that have certain properties (like uniqueness). If something in the domain has the property, it is a focal point at step i. As time goes on, new beliefs are derived (e.g., through modus ponens), and the domain over which the search is being conducted also expands (through observations or consideration of new conjunctive properties). Then the search for candidate focal points is repeated—and an old focal point may, given the new information, no longer be one. The search for focal points is cut off at some depth of computation, depending on time constraints, at which point the agent attempts to resolve competing focal points.

Let us now consider the details of the above process. We first consider the way in which the agent models the (changing) domain, then the rules that qualify a candidate as a focal point. Finally, we consider the ways in which an agent resolves competing focal points.

7.2.1 Domain of Consideration

Before the process starts, the agent is given two finite sets enumerating the domain constants (one, *Pred*, is a set of predicates, and the second, *Term*, is a set of term constants) over which the focal point computation is going to be done initially. Both lists can grow as the computation progresses.

Example of Dividing 100 Objects into Two Piles: The vectors that sum to 100, with no element less than 0, can be given as an initial finite domain over which properties will be discovered.

It should be noted that these finite sets represent the *explicit* knowledge of the agent, not its implicit knowledge. For example, an agent may implicitly be aware of the infinite set of positive integers, but for the moment only be considering the finite set of integers from 1 to 500. As time goes on, numbers above 500 may come under the explicit scope of consideration.

7.2.2 Addition of Term Constants

There are two mechanisms for adding new explicit terms. The first is observation, where new term constants are observed over time (e.g., a new bridge is observed). The second mechanism is the use of inductive rules, such as a successor rule that generates new integers.

Example 1: At step i, the domain includes Bridge(C125). At step $i+1$ we have Observe{Bridge(C237)}. At step $i+2$ we then have C237 in *Term*.

Example 2: If $Int(x) \rightarrow Int(x+1)$ is a rule at step i, and $Int(5)$ is known at step i, then at step $i+1$ the agent will know $Int(5+1)$. Assuming that the agent has the requisite procedure attached to the symbol $+$, he will (in step $i+2$) add the term 6 to *Term*.

7.2.3 Addition of Predicate Constants

Consider an agent searching for focal points. When he starts, he considers attributes that might be held by only a single object in his domain. For example, there might be only one object that is Red. However, if such a unique object does not exist, then he may consider conjunctions of attributes. For example, there might be only one House that is Red. We want to capture this intuition in our algorithm.

When the process starts, *Pred* is equal to the finite set of predicates provided to the agent. At the second step, the agent considers binary conjunctions of predicates from the original list. At step three, he considers ternary conjunctions of predicates from the original list, and so on. The following lines describe the evolution of *Pred* through successive steps.

$$\text{step 1: } Pred_1 = \{\text{domain constant predicates and their negations}\} = \{P_1, \neg P_1, P_2, \neg P_2 \ldots\}$$

$$\text{step 2 : } Pred_2 = \{\text{binary combinations of predicates of } Pred_1\} =$$

$$\{P_1 \wedge P_2, P_1 \wedge P_3, P_2 \wedge P_3, \ldots, \neg P_1 \wedge P_2 \wedge \neg P_1 \wedge \neg P_2, \ldots\} \cup Pred_1$$

$$\text{step 3 : } Pred_3 = \{\text{ternary combinations of predicates of } Pred_1\} =$$

$$\{P_1 \wedge P_2 \wedge P_3, P_2 \wedge P_3 \wedge P_4, \ldots, \neg P_1 \wedge \neg P_2 \neg P_3, \ldots\} \cup Pred_2$$

7.2.4 Explicit and Easily Computed Knowledge

We want agents, in their search for focal points, to consider both explicit knowledge and "obvious" knowledge that is easily computed from their databases. For example, if "less than" is a predicate that the agent is considering, and both 5 and 6 are terms of which he is aware, then we want the agent to use the knowledge that 5 is less than 6, even though this fact is not explicitly represented in his database.

We therefore use a special notation to signify that a fact is "known" at the previous level. We write $\in^*$ to mean that the fact is either explicitly listed in *Facts* at level i, or that it can be simply computed over the constant terms *Term* known at level i.

The question of what can be simply computed is domain dependent, as well as agent dependent. There is an analogy here with the idea of "operational" in the Explanation Based Learning literature [96]. Checking "less than" might be operational in some machines; in other machines, deciding in a game of chess whether a given board position is reachable from the current state might be operational because of specialized hardware.

7.3 Focal Point Rules

In this section we present the actual rules by which an agent identifies candidates for focal points. We make no claims for completeness here. These rules provide good coverage of the Focal Point examples in [129], but additional rules may be appropriate in other cases.

Identification of focal points is a two stage process. First the agent identifies candidates by looking for meta-characteristics of objects, such as uniqueness. Second, the agent resolves competing candidates to the best of his ability (using other rules) and decides on one or more focal points.

7.3.1 Uniqueness

An object may be a focal point if it is the only object with a given property. Formally, if in $i-1$th step we have $P \in \mathcal{P}red_{i-1}$, and there exists an $x \in \mathcal{T}erm_{i-1}$ such that

$$P(x) \in^* \mathcal{F}acts_{i-1} \forall y \in \mathcal{T}erm, y \neq x [P(y) \notin^* \mathcal{F}acts_{i-1}],$$

then in step i we will have

$$\text{Unique}(x, P, i).$$

Note that Unique is a "meta-predicate" that does not itself appear in the *Pred* set. Note also that the term x is considered unique with respect to the predicate P; this will be important later when competing focal points must be resolved.

Example: This rule would be applicable in the case where we know about only one Bridge, namely C125.

Both x and y can be vectors, in which case they will be denoted by $[x]$ and $[y]$. Another example of uniqueness (using equality on elements of a vector) is the following: $P([x, y]) \equiv x = y$ where the domain is defined to be vectors such that $Sum([x, y]) \equiv x + y = 100$. This causes us to choose the vector [50,50] over all others whose elements sum to 100.

7.3.2 Uniqueness Complement

Lack of information can also cause a solution to be prominent.

An object may be a focal point if it is the only object without a given property. Formally, if in $i-1$th step we have $P \in \mathcal{P}red_{i-1}$, and there exists an $x \in \mathcal{T}erm_{i-1}$ such that

$$P(x) \notin^* \mathcal{F}acts_{i-1} \forall y \in \mathcal{T}erm, y \neq x [P(y) \in^* \mathcal{F}acts_{i-1}],$$

then in step i we will have

$$\text{Unique-Comp}(x, P, i).$$

Example: This rule would be applicable in the case where we know that everybody in the domain is a member of the Democratic Party, except that we have no information one way or the other about John. Although we don't know whether or not John is also a member, this lack of knowledge causes him to be prominent.

7.3.3 Centrality

Another meta-predicate is the concept of Centrality, the intuitive property of a central point around which a domain (or sub-domain) is symmetric.

An object may be a focal point if it is a central object within a given domain. Formally, if in $i-1$th step we have $P \in \mathcal{P}red_{i-1}$, and there exists an $x \in \mathcal{T}erm_{i-1}$ such that

$$P(x) \in^* \mathcal{F}acts_{i-1}$$

$$\forall y \in \mathcal{T}erm, y \neq x \wedge P(y) \in^* \mathcal{F}acts_{i-1},$$

$$\exists z \in \mathcal{T}erm, z \neq y \wedge P(z) \in^* \mathcal{F}acts_{i-1},$$
$$\text{such that } \mathit{Diff}(y,x) = \mathit{Diff}(x,z)$$

where *Diff* is a difference function defined on terms of the domain (e.g., "$-$" in the domain of numbers), then in step i we will have

$$\text{Central}(x, P, i).$$

Example: In the range between 0 and 10, the number 5 is Central (where P is the predicate Integer, and *Diff* is defined to be the minus function).

We introduce centrality as an additional meta-predicate because we want to recognize focal point terms that are central with respect to a given property. For example, a house that is centrally located with respect to all other houses might be a focal point. $House(x)$ is the chosen predicate P in the centrality meta-predicate, while *Diff* is defined to represent spatial distance. Using the meta-predicate Unique would not allow us to recognize the centrality of this point. However, every Unique term is Central according to the above definition (the degenerate case).

We might define a general purpose predicate $central(x) \in^* \mathcal{P}red$ as one that has other y and z terms in the domain that stand in a certain relationship with it (e.g., distance). We might then have a unique central point. But there is no way, using the uniqueness meta-predicate, for us to ensure that the x, y, and z terms all share some other common attribute as a condition for x's centrality. For example, x might be the central house among a set of houses, but y and z being houses could not be captured in the definition of a general purpose predicate $central(x)$.

7.3.4 Extreme

An object can sometimes be prominent because it is the highest object, or the tallest, or the smallest, among the elements of the domain. We consider only those elements of the domain that satisfy some property of *Pred*, expanded to include the identity predicate (always true).

An object may be a focal point if it is an extreme object in a totally-ordered domain. Formally, if in $i-1$th step we have $P, Q \in \mathcal{P}red_{i-1}$, and there exists an $x \in \mathcal{T}erm_{i-1}$ such that

$$P(x) \wedge \forall y \in \mathcal{T}erm_{i-1}, y \neq x \wedge P(y), (Q(x,y) \in^* \mathcal{F}acts_{i-1} \wedge Q(y,x) \notin^* \mathcal{F}acts_{i-1}),$$

then in step i we will have

$$\text{Extreme}(x, P, Q, i).$$

Example: In the range between 1 and 10000, the number 1 is Extreme (with the predicate Q being "less than").

Every object that is unique is also central and extreme, trivially.

7.4 Computing Focal Points—The Resolution Rules

The rules above specify when an object is unique, or extreme, etc. They do not relate directly to the question of when the object is actually a focal point. We thus need a rule to use in tying together these attributes with the notion of focal point.

The most straightforward approach is to relate each of the meta-predicates above with the focal point attribute:

$$\frac{i: \ \text{Unique}(x, P, i)}{i+1: \ \text{FocalPoint}(x, i)}$$

$$\frac{i: \ \text{Unique-Comp}(x, P, i)}{i+1: \ \text{FocalPoint}(x, i)}$$

$$\frac{i: \ \text{Central}(x, P, i)}{i+1: \ \text{FocalPoint}(x, i)}$$

$$\frac{i: \ \text{Extreme}(x, P, i)}{i+1: \ \text{FocalPoint}(x, i)}$$

These rules of course may not supply us with a unique focal point, since there could be a term that satisfies Unique, another that satisfies Unique-Comp, etc. There could even be two separate terms that are Unique with respect to different predicates. Moreover, two separate terms that are (for example) extreme might receive less attention than a single term that is central, precisely because the two extremes are competing with one another. There is still utility for the agent in discovering the set of focal points, since even if the choice is made among them probabilistically, there is an increased chance for coordination among the agents.

We will not attempt here to provide additional rules that guarantee a single focal point. Instead, we illustrate that one could introduce additional rules so as to reduce the size of the focal point set.

It is critical to resolve among focal points so that ones that are discovered more easily have higher priority. Active logic provides us with a natural tool for dealing with this. Using active logic, there are several mechanisms for relating priority to complexity; we here present one.

A focal point might be generated (given the above rules) at a given level, then not be a focal point at a subsequent level. An agent looks for focal points only up to a certain level k. At this level, there might be several competing focal points that are still valid (e.g., arising from different rules, or from different predicates). As an initial winnowing mechanism, the focal points that were generated earliest are kept and the others discarded. The intuition is that, since the other agent may not go as deep in the deduction as we have in looking for a focal point, we are more likely to match the other agent by taking the earliest focal point. It is the solution (that we still believe in) most likely to have been reached by the other agent.[64]

Example: In the range between 1 and 10000, the number 1 is Extreme (with the predicate P being "less than"), and 10000 is Extreme (with the predicate P being "greater than"), after the first step.

If the domain of considered integers grows at each step, 1 will still be extreme while 10000 will no longer be extreme. Thus, at the end of the process, 1 will be chosen since it has been "extreme" for the longest period. This disambiguates between the two extreme ends of a finite domain that is growing in only one direction.

The algorithm only considers "term-property" pairs; if a term was a focal point because of some property at level i, then was no longer a focal point because of that property at level i+1, then again became a focal point because of a different property at level i+2 (and remains a focal point until the end), then it is considered to have been generated at level i+2.[65]

We may also choose to introduce rules that assign a priority to the meta-predicates (like Unique) so that, for example, a unique object gets priority as a focal point over an extreme object.

7.4.1 Convergence Conditions

When interacting human agents search for focal points, there is generally no guarantee that their choices will be identical. When interacting automated agents search for focal points, they are following set algorithms. Depending on their own knowledge, and their knowledge of each other and of the domain, they may be able to reach a guaranteed solution. In other cases there is no guaranteed agreement, but the focal point algorithm can be thought of as a heuristic to prune the search for a focal point.

As with various forms of communication, the agents can benefit from having some common background when they use a focal point algorithm. For example, agents that negotiate the allocation of a common resource should have some common language and some protocol for negotiations. If the protocol is more detailed, the negotiation is more efficient and the chances of reaching mutually beneficial agreement are greater. Similarly, as the agents use more detailed focal point algorithms, and if they have more common background, their probability for convergence increases.

8 Deadline planning

Time is the most obvious critical resource in planning with deadline constraints. There is a given moment d (for deadline) in the future by which a goal G must be achieved, and the agent's task is to find a suitable plan to achieve G and enact it before

[64] Other approaches present themselves, such as considering the *coverage* of a focal point, e.g., if a term is a focal point for much of the deduction, though it is not at the final step, we would still consider it a likely solution. We could also then probabilistically weight the steps of the deduction, so that (for example) earlier steps receive more weight than later steps. These methods are left for future work.

[65] The idea behind looking at term-property pairs to establish the first appearance of a focal point is that once a focal point has disappeared because of other terms with the same property, its prominence because of that original property is completely negated.

time d. This means that both the planning and the enacting of the resulting plan must be take no more than $(d - Now)$ time units, where Now is the initial time at which planning begins. Proper planning often involves "meta-planning", in order to adjudicate between alternative plans, reject infeasible plans, and so on. But that takes time too! Action, which takes time, occurs in the very process of thinking or reasoning, including such meta-reasoning. In [110], it is argued that, traditionally, actions in AI are viewed as separate from the planning process which leads to those actions. Even when the two are intertwined, as in real-time, dynamic or reactive planning, the planning effort is treated as a different kind of beast, not an action itself. Just as it is essential to understand certain features of actions in order to make an intelligent choice of actions in a plan, it is necessary to reflect upon features of planning to make intelligent decisions while planning.

When the reasoning is not carried out *within* but rather only *about* a deadline situation the time for meta-planning does not enter the computation. However, in reality, meta-planning often itself must go on as the deadline approaches. To be sure, in some commonly encountered situations the time taken for meta-planning may be very short. But what of highly novel settings in which one cannot *a priori* assign expected utilities to various conceivable options or refinements? Then the planner is forced to decide on utilities and other factors in real time. In these cases it seems unlikely that such meta-planning will always have a modest time cost. Clearly, the emphasis then is not on searching for a theoretically optimal plan, but one which is speculated to work within the deadline. The reasoner must have the flexibility to interleave planning and execution, not only because there may not be enough time to wait until a complete plan is formulated, but because future planning actions may depend upon the outcomes of earlier executions.

The importance of accounting for time of meta-planning as part of overall time of planning and acting then is real. But in general it may be impossible to determine in advance how long meta-planning will take. An alternate perspective, which we explore, is to simply measure how long planning, meta-planning, and acting are in fact taking, and use this increasing time measure to help decide how to continue in the planning/meta-planning/acting vis-à-vis the approaching deadline.

Thus our approach is not to provide a special technique for precomputing time for meta-reasoning (which we suspect is indeterminate, in general) but rather one in which the reasoning and meta-reasoning are performed together and the time for each is fully accounted for as they occur. We don't pre-compute how long meta-planning will take; we do some rough estimation of time to perform actions, but chiefly, we track how long planning, meta-planning and acting are taking in real-time, as they occur. Simultaneously we compare the evolving time elapsed with the approaching deadline, and this comparison effects decisions about continued planning and acting.[66]

In this section we give a brief overview of work on fully deadline-coupled planning [65, 64, 104] that uses active logics as its underlying framework. It is a mechanism that lets a time-situated reasoner keep track of an approaching deadline as she/he makes (and enacts) her/his plan, thereby treating *all* facets of planning (including plan-formation and its simultaneous or subsequent execution) as deadline-coupled. The approach for planning is deliberately noncommittal with respect to a number of traditional planning issues, such as total or partial order. Indeed, any planning algorithm can be implemented in the active-logic framework. In our illustrations we use total order planning to keep the planning as simple as possible while dealing with the temporal aspects.[67]

To elaborate on the fully deadline-coupled planning problem, we present an illustrative domain, which we call the *Nell & Dudley Scenario*[68]: Nell is tied to the railroad tracks as a train approaches. Dudley must formulate a plan to save her and carry it out before the train reaches her. If we suppose Dudley has never rescued anyone before, then he cannot rely on having any very useful assessment in advance, as to what is worth trying. He must deliberate (plan) in order to decide this, yet as he does so the train draws nearer to Nell. We want to prevent Dudley from spending so much time seeking a theoretically optimal plan to save Nell, that in the meantime the train has run Nell down. Moreover, we want Dudley to do this without much help in the form of expected utilities or other prior computation. Thus he must assess and adjust (meta-plan) his on-going deliberations *vis-a-vis* the passage of time. His total effort (plan, meta-plan and action) must stay within the deadline. He must, in short, reason in time about his own reasoning in time. In particular, we will demonstrate our mechanisms for a planner, with the following simple scenario. Here Dudley knows that Nell is a distance of 30 'paces' from him when he first realizes (at step 0) that the train will reach her in 50 time units. He begins to form a plan, and refines the plan in subsequent steps.

[66] There is an extensive related literature, treating in turn the areas of temporal projection (e.g., [38, 39, 54, 60, 76, 77, 100, 107, 132, 3]), plan interaction (e.g., [134, 136, 128, 138]), and meta-planning (e.g., [125, 7, 8, 16, 114, 58]).

[67] We have not sought to build an optimal planner, not even a state-of-the-art planner; there are many ways to make the planner more sophisticated. Our aim has been first and foremost to couch planning in a fully time-situated framework; further work will be required to incorporate our findings into state-of-the-art techniques for a truly efficient planner. However, in our view, evolving-time is a sufficiently critical issue for real-time deadline coupled planning, that it must be tackled directly (as our effort attempts) no matter what other desirable features may or may not be included (such as partial order planning).

[68] This problem was first mentioned in the context of time-dependent reasoning by McDermott [86], and more recently discussed in [11].

Formulas: A formula $X(s : f, Args)$ consists of a predicate name X which may represent a fluent or an action predicate, with a list of arguments. The first argument denotes the time interval $s : f$ over which the predicate holds, where s and f are the interval's beginning and ending points, respectively. The other arguments of the predicate follow and are denoted by $Args$ for easy reference.

A partial plan: A partial plan is a belief $\mathbf{Ppl}(i, p, Triplet\_List)$ denoting a partial plan at step i with the name p. The $Triplet\_List$ is an ordered list of action triplets. Each triplet, $[C_A, A, R_A]$, consists of an *action*, A, preceded and followed, respectively, by a list of *conditions*, C_A, and *results*, R_A. A is a formula containing an action predicate and C_A and R_A are lists of formulas.[69] An agent may have several partial plans to achieve the same goal. A special plan with the name *null* is a plan with no actions in it.

The following is a partial plan for Dudley. d is Dudley, n is Nell, h denotes home and r the railroad track. We use the shorthand $r - h$ to denote the *distance between* the railroad track (r) and home (h). In this partial plan Dudley intends to release Nell and then pull her from the railroad track[70] The symbol $\bullet\!\!\rightarrow$, as it appears in $X(s : t\bullet\!\!\rightarrow R, \ldots)$, denotes that X is intended to hold beyond $s : t$, and up to R (by default). The term *save* is a label naming the plan Dudley is forming.

$$\mathbf{Ppl}(2, save, \left\{ \begin{bmatrix} At(t_3 : t_4, d, r) \\ Release(t_3 : \overline{t}_4, d, n, r) \\ \neg Tied(t_4 \bullet\!\!\rightarrow t_1, n, r) \end{bmatrix}_1 \begin{bmatrix} \neg Tied(t_1, n, r) \\ Pull(t_1 : \overline{t}_2, d, n, r) \\ Out\_of\_danger(t_2 \bullet\!\!\rightarrow 50, n, r) \end{bmatrix}_2 \right\}),$$

Context of partial plan: Each of the partial plans defines a context within which reasoning can be done about the expected state of the world if the plan were to be carried to completion.

The agent maintains a belief $\mathbf{CS}(i, p, Context\_List)$ denoting the *context set* for each plan p at each step i. The list $Context\_List$ consists of quoted formulas (we omit the quotes for readability), and includes all of the facts (observations), formulas corresponding to actions in the plan, and formulas that the agent deduces to be true in the state of the world resulting from the successful execution of plan p. The context set changes with time as the plan undergoes modification and as inferences are made in the context of the plan.

Projection: [71] At each step i, the belief $\mathbf{Proj}(i, p, Proj\_List)$ denotes the projection that is formed in the context of each partial plan p in progress, based on the default of persistence.[72]

Temporal reasoning rules: We developed three inference rules for temporal reasoning: (i) the temporal projection rule (TP), (ii) the restructured modus ponens rule (RMP), and (iii) the context set revision rule (CSR). The details of these rules are described in [103]. Here we just describe them briefly.

1. The temporal projection rule (TP) effectively *smoothes* beliefs over time intervals which present gaps in the agent's knowledge. Our approach is best described by the term *parallel projection*. Here the entire known state of the world at one moment is used to determine the (expected) state at the next moment.
2. Instead of applying modus ponens (MP) in its familiar form: viz. from α and $\alpha \rightarrow \beta$ deduce β, we use a restructured MP rule (RMP) in accordance with our philosophy to let earlier defaults play out their effects completely to result in an anticipated state of the world to which later defaults may be applied if necessary. (RMP depends on a clause form representation of data.) A formula which is a fact has no justification attached to it. All axioms are treated as facts. A formula α which was derived using one or more projections $\beta_1, \beta_2 \ldots$ is only as feasible as the weakest projection, and is itself classified as a default. Such a formula is annotated with the projections used in its derivation and is written as $\alpha[\beta_1, \beta_2, ...]$. The RMP rule is used in extending the context set. This allows Dudley to compute the extended effects of actions. It also allows him to deduce the future consequences of his planning as it interacts—possibly with the actions of other agents or with events observed in the world. It allows for reasoning with the current projection by letting earlier events play out their consequences in an anticipated future before later events.[73]
3. The CSR rule ensures that the context set is always kept updated to match the most current projection, and the state of the world in which the agent is situated. The problem is that the default reasoning, based on the

[69] Whenever formulas appear in lists such as C_A or R_A and later in beliefs **CS**, **Proj** and **Ppl**, they are in fact treated as if they are "quoted." We omit the quotes to keep the long strings readable. Thus the beliefs of the agent that we will describe shortly are still first order formulas.

[70] The full specification and explanations of the formalization can be found in [103].

[71] Our projection mechanism has commonalities with some of the chronological minimization approaches, notably [132, 77, 60]. See [102] for detailed discussion.

[72] The i denotes the step number, and $Proj\_List$ is a list of quoted formulas.

[73] We used these techniques for solving different versions of the Yale Shooting Problem [102].

projection, may incorporate contradictory formulas into the context of a plan. The CSR rule plays the important role of resolving contradictions.

As explained before, formulas are annotated by the projections which are used to support them in future conjectures. In the event that the projections cease to hold as of "now," the formulas that are supported by them are dropped from the context set in the revision process. The revision is a kind of real-time truth maintenance.

Plan Refinement: We developed several simple refinement rules to refine a partial plan *in time*. We demonstrate here one of them for the refinement of non-primitive action.

The active-logic planner is hierarchical. Abstraction is embodied in the way the axioms encode the knowledge about actions. Skeleton plans at upper levels first synthesized by using higher level actions. These are then broken into more primitive actions by rules such as the action refinement rule described in the rule below. Our design allows for the concurrent processing of levels, and for concurrent refinement of multiple partial plans.

- Refinement of non-primitive action

$$\frac{i : \mathbf{Ppl}(i, p, \left\{ \ldots \begin{bmatrix} C_A \\ A \\ R_A \end{bmatrix} \ldots \right\}), CS(i, p, \{\ldots, Q_1 \wedge \ldots \wedge Q_k \rightarrow A\})}{i+1 : \mathbf{Ppl}(i+1, , p, \left\{ \ldots \begin{bmatrix} C_{Q_1} \\ Q_1 \\ R_{Q_1} \end{bmatrix} \ldots \begin{bmatrix} C_{Q_k} \\ Q_k \\ R_{Q_k} \end{bmatrix} \ldots \right\})}$$

provided every condition $C_A \in \mathbf{CS}_{i,p} \cup \mathbf{Proj}_{i,p}$.[74]

Working estimate of time (WET): The WET (working estimate of time) of a plan is a rough estimate of the total time that the plan will consume. It consists of two parts. The PET (*planning estimate of time*) is the (estimated) time to be spent in reasoning about the plan. This includes plan formulation, refinement, temporal projection and context-based reasoning. The EET (*execution estimate of time*) of the plan is the (estimated) time required to actually execute the actions that have been identified in the plan. Thus, WET = EET + PET. We estimate the WET of a plan based on the estimates of the WET's of the actions that are already part of the plan. We do not have a mechanism to estimate the WET of the unknown portion of a partial plan except for the sliding *Now* which accounts for the time taken to identify the remaining portion of the plan.

We developed several rules for computing the EET and PET of an action based on the type of the action. Here we just present a simple rule to compute the WET.

- Computing the WET

$$\frac{i : \mathbf{Ppl}(i, p, \quad \left\{ \begin{bmatrix} C_{A_1} \\ A_1(s_1 : f_1, \ldots) \\ R_{A_1} \end{bmatrix} \ldots \begin{bmatrix} C_{A_k} \\ A_k(s_k : f_k, \ldots) \\ R_{A_k} \end{bmatrix} \right\}), \ldots}{i+1 : \mathbf{WET}(i, p, \sum_{j=1}^{k} EET(A_j) + PET(A_j))}$$

where the EET and PET for each action A_j is computed based on the criteria describe in [102].

Feasibility: As long as the sum of a (partial) plan's WET + *Now* is within the deadline, Dudley declares the plan **Feasible** using the following rule, and continues refining and/or putting the partial plan into execution.

- Marking a plan "feasible"

$$\frac{i : \mathbf{Ppl}(p, i, \{\ldots\}), \mathbf{Goal}(p, g, d), \mathbf{WET}(i-1, p, \omega)}{i+1 : \mathbf{Feasible}(i, p)}$$

if $\omega + i \leq d$.

If the WET computation indicates the plan is not feasible, the plan is frozen (no longer refined for the time being), but may be used in the future.[75]

[75] Our focus here is not to find an optimal heuristic for producing the best plans, but rather to develop the underlying framework for incorporating passage of time into an inference based approach to planning. Within such a framework, numerous experiments contrasting various heuristics can now be undertaken; this is a direction for our future work.

To give a flavor of the deadline-coupled reasoning, we present more details of the simple scenario of Dudley and Nell case. Only a few of Dudley's beliefs are shown below. For additional axioms and inference rules please consult [103].

Step 0:
$\mathbf{CS}(0, null, \{\ldots, At(0,d,h)_{obs}, r-h=30_{obs}, Tied(0,n,r)_{obs},\})$,
$\mathbf{Proj}(0, null\{\})$,
$\mathbf{Goal}(save, Out\_of\_danger(50,n,r), 50)$,
$\mathbf{Unsolved}(0, Out\_of\_danger(50,n,r))$, ...

(Step 0 represents Dudley's state of mind before planning had begun, but after he learned that Nell is tied to the tracks. The belief $\mathbf{Goal}(save, Out\_of\_danger(50,n,r), 50)$ denotes that the plan save is being developed to meet Dudley's goal to take Nell out of danger by the deadline 50. The subscript obs on a formula indicates that that formula has come in as an observation and thus is not based on a projection.)

Step 1:
$\mathbf{CS}(1, null, \{\ldots, At(0,d,h)_{obs}, r-h=30_{obs}, Tied(0,n,r)_{obs}, t_1=t_2+1\})$,
$\mathbf{Proj}(1, null, \{At(1:\infty,d,h), Tied(1:\infty,n,r)\})$,
$\mathbf{CS}(1, save, \{\ldots, At(0,d,h)_{obs}, Tied(0,n,r)_{obs}\})$,
$$\mathbf{Ppl}(1, save, \left\{ \begin{bmatrix} \neg Tied(t_1,n,r) \\ Pull(t_1:\overline{t}_2,d,n,r) \\ Out\_of\_danger(t_2 \bullet\!\!\rightarrow 50,n,r) \end{bmatrix}_1 \right\}),$$
$\mathbf{Proj}(1, save, \{\})$,
$\mathbf{WET}(1, save, 0)$,
$\mathbf{Feasible}(1, save)$, ...

(A new plan called $save$ is begun and is initially declared to be feasible.)

Step 2:
$\mathbf{CS}(2, save, \{\ldots, At(0,d,h)_{obs}, Tied(0,n,r)_{obs}, Pull(t_1:\overline{t}_2,d,n,r), r-h=30_{obs}, t_2 \leq 50,$
$t_1=t_2-1, t_3=t_4-3, t_4 \leq t_1,\})$,
$$\mathbf{Ppl}(2, save, \left\{ \begin{bmatrix} At(t_3:t_4,d,r) \\ Release(t_3:\overline{t}_4,d,n,r) \\ \neg Tied(t_4 \bullet\!\!\rightarrow t_1,n,r) \end{bmatrix}_1 \begin{bmatrix} \neg Tied(t_1,n,r) \\ Pull(t_1:\overline{t}_2,d,n,r) \\ Out\_of\_danger(t_2 \bullet\!\!\rightarrow 50,n,r) \end{bmatrix}_2 \right\}),$$
$\mathbf{Proj}(2, save, \{At(1:\infty,d,h), Tied(1:\infty,n,r)\})$,
$\mathbf{WET}(2, save, 2)$,
$\mathbf{Feasible}(2, save)$, ...

(Plan refinements begin. Since Dudley doesn't believe that Nell will be not tied, the precondition of the action Pull is not true. Since Dudley believes that if he release Nell, she will become not tied, he adds the action release to his plan.

Now for the first time WET and feasibility are actually computed. The plan in Step 1 includes only one action, namely *Pull*. The PET for *Pull* is the time required to bind its time variables. There are no other uninstantiated variables, and it is not part of a sequence. *Pull* is primitive action which does not need further refinement. Since it takes one time step, EET for *Pull* is 1. Thus WET for *Pull* is 2, which is also the WET for the partial plan *save*. For brevity we suppressed the null plan. The pull action in the partial plan of step 1, is added to the CS of step 2, indicating that the pull action will occur in the context of the plan *save*.

The projection is computed based on the CS of step 1. Dudley projects that it will stay at home forever and Nell will stayed tied forever.)

Step 3:

$$\begin{aligned}
&\mathbf{CS}(3, save, \{\ldots, At(0,d,h)_{obs}, r-h=30_{obs}, Tied(0,n,r)_{obs}, Pull(t_1:\overline{t}_2, d, n, r),\\
&\quad Out_of_danger_c(t_2,n,r), Release(t_3:\overline{t}_4, d, n, r), t_2 \leq 50, t_1 = t_2 - 1,\\
&\quad t_3 = t_4 - 3, t_4 \leq t_1, t_6 < t_7 \leq t_3\}),\\
&\mathbf{Ppl}(3, save, \left\{\begin{array}{l}
\qquad\left[\begin{array}{c} At(t_6,d,l) \\ Run(t_6:\overline{t}_7, d, l:r) \\ At(t_7 \bullet\!\!\rightarrow t_3, d, r) \end{array}\right]_1 \qquad \left[\begin{array}{c} At(t_3:t_3+1, d, r) \\ Release_1(t_3:\overline{t}_3+1, d, n, r) \\ \neg Tied(t_3+1 \bullet\!\!\rightarrow t_1, n, r) \end{array}\right]_2 \ldots \\
\left[\begin{array}{c} At(t_3+2:t_4, d, r) \\ Release_3(t_3+2:\overline{t}_4, d, n, r) \\ \neg Tied(t_4 \bullet\!\!\rightarrow t_1, n, r) \end{array}\right]_4 \ldots
\end{array}\right\})\\
&\mathbf{Proj}(3, save, \{At(1:\infty, d, h), Tied(1:\infty, n, r)\}),\\
&\mathbf{WET}(3, save, 7),\\
&\mathbf{Feasible}(3, save), \quad \ldots
\end{aligned}$$

(Release is a complex action that is refined by replacing it with three primitive actions. The action Run is added to satisfy the precondition that Dudley will be at the railroad tracks by the time of the release.

Since the consequence of the pull action is that Nell will be out of danger, this is added to the CS of step 3 as a result of applying RMP to the appropriate axiom. The WET for *Pull* is 2 as explained in step 2, that does not change. The PET for *Release* is 2 (one to bind the time variables, and another to refine it into primitive actions) and its EET is 3. Thus the WET for *Release* sums to 5, and the WET for the plan (as of the previous step) is 7, as reflected in the WET belief.)

Step 4:

$$\begin{aligned}
&\mathbf{CS}(4, save, \{\ldots, At(0,d,h)_{obs}, r-h=30_{obs}, Tied(0,n,r)_{obs}, Pull(t_1:\overline{t}_2, d, n, r),\\
&\quad Out_of_danger_c(t_2,n,r), Release_1(t_3:\overline{t_3+1}, d, n, r), \ldots, Run(t_6:\overline{t}_7, d, l:r), \neg Tied_c(t_4, n, r),\\
&\quad t_2 \leq 50, t_1 = t_2 - 1, t_3 = t_4 - 3, t_4 \leq t_1, t_6 < t_7 \leq t_3\}),\\
&\mathbf{Ppl}(4, save, \left\{ \left[\begin{array}{c} At(t_6,d,h) \\ Run(t_6:\overline{t}_7, d, h:r) \\ At(t_7 \bullet\!\!\rightarrow t_3, d, r) \end{array}\right]_1 \left[\begin{array}{c} At(t_3:t_3+1, d, r) \\ Release_1(t_3:\overline{t_3+1}, d, n, r) \\ \neg Tied(t_3+1 \bullet\!\!\rightarrow t_1, n, r) \end{array}\right]_2 \ldots \right\}),\\
&\mathbf{Proj}(4, save, \{At(1:\infty, d, h), Tied(1:\infty, n, r), Out_of_danger(t_2+1:\infty, n, r)\}),\\
&\mathbf{WET}(4, save, 9),\\
&\mathbf{Feasible}(4, save), \quad \ldots
\end{aligned}$$

(Planning continues as above. The plan in Step 3 consists of three new primitive actions obtained by refining *Release* into its three components: $Release_1$, $Release_2$, and $Release_3$. Of these, $Release_1$ has a PET of 1, which is the step required to bind the time variables, since it is the first of the sequence of the three actions that constitute the *Release*. Once this is bound, the times of the other two are decided automatically. Thus PET for $Release_2$ and $Release_3$ are subsequently zero. The EET for each of them is 1. The *Run* action has a PET of 3 (one to bind the time variables, 1 to refine it, and 1 to bind the other variables). Thus the WET of the plan is now 9. Also, notice that in this step, the variable l in the *Run* action has been bound to h by looking it up in the projection.)

As was demonstrated in the example, we developed other mechanisms for a planner that reason *in* time which we don't describe here. Our approach has many concerns in common with existing research in planning and temporal reasoning. [87, 49, 56, 55, 125]. However, these works do not account for the time taken for meta-planning. Indeed, this is stated in [125] (page 402): "Here we will not worry about the cost of meta-reasoning itself; in practice, we have been able to reduce it to an insignificant level".

In our work, although we do not make any attempts to optimize plans, our *fully* deadline-coupled planner meets an important criterion: in addition to performing metareasoning for determining the current time, estimating the expected execution time of partially completed plans, and discarding alternatives that are deadline-infeasible, our system also has a built-in way of accounting for all the time spent as a deadline approaches. This means not only accounting for the time of various segments (procedures in the more usual approaches), but also the time for this very accounting for time! Active logics do this without a vicious circle of "meta-meta-meta..." hierarchies.

9 Time-situated Variations of the YSP

In the previous section we described the application time-situated reasoning mechanism to dead-line planning. This mechanism was also applied in [102] to several real-time variations of the Yale Shooting Problem [53] appropriate to active logics.

In the *classical* YSP problem, there is a certain ambiguity about the role of the reasoner. There the reasoning is itself timeless, presumably it takes place after all the events in question. Our treatment is significant in that Dudley the reasoner can reason *in* time about the events in progress and adjust his reasoning to suit new observations.

The first scenario that we considered in [102] is a *witness* scenario where Dudley is a witness to the scene of the crime. In it we show how Dudley draws the intuitive conclusion that Fred must be dead on observing a shoot action, and discard the unintuitive one where the gun mysteriously gets unloaded just before the shooting. Two developments of the witness scenario were presented, which are of a *detective* nature where Dudley must offer a reasonable explanation about actions in the past, to fit his present observations. On seeing Fred alive at a later time, the same mechanism allows him to continue to perform belief revision to account for "why things went wrong"[99]. The last scenario is a *killer* scenario where Dudley formulates a plan to kill Fred by a certain deadline and reasons that Fred is expected to be dead in the context of his plan to carry out a shoot action.

Here we sketch only the simple Witness scenario:

Axioms(These are part of every context set):
$\neg Loaded(T) \vee \neg Shoot(\overline{T}) \vee \neg Alive(T+1)$;
$\neg Alive(T) \vee Alive(0:T)$

0: $\mathbf{CS}(0, null, \{\underline{Alive(0)_{obs}}, \underline{Loaded(0)_{obs}}\})$,
$\mathbf{Proj}(0, null, \{\})$

Dudley observes that a gun is loaded, and that Fred is alive. Dudley is a passive eyewitness. Hence the reasoning context is that of a *null* plan. There is no projection yet regarding either *Alive* or *Loaded*.

1: $\mathbf{CS}(1, null, \{Alive(0)_{obs}, Loaded(0)_{obs}\})$,
$\mathbf{Proj}(1, null, \{\underline{Alive(1:\infty)}, \underline{Loaded(1:\infty)}\})$

Rules yielding new conclusions: TP. There are no new observations.

2: $\mathbf{CS}(2, null, \{Alive(0)_{obs}, Loaded(0)_{obs}\})$,
$\mathbf{Proj}(2, null, \{Alive(1:\infty), Loaded(1:\infty)\})$

Wait period. No new actions or conclusions.

3: $\mathbf{CS}(3, null, \{Alive(0)_{obs}, Loaded(0)_{obs}\})$,
$\mathbf{Proj}(3, null, \{Alive(1:\infty), Loaded(1:\infty)\})$

Wait period.

4: $\mathbf{CS}(4, null, \{Alive(0)_{obs}, Loaded(0)_{obs}, \underline{Shoot(\overline{4})_{obs}}\})$,
$\mathbf{Proj}(4, null, \{Alive(1:\infty), Loaded(1:\infty)\})$

Rules yielding new conclusions: OBS. A shooting is observed by Dudley.

5: $\mathbf{CS}(5, null, \{Alive(0)_{obs}, \underline{\neg Alive_c(5)[Loaded(4)]}, Loaded(0)_{obs}, Shoot(\overline{4})_{obs}\})$,
$\mathbf{Proj}(5, null, \{Alive(1:\infty), Loaded(1:\infty)\})$

Rules yielding new conclusions: RMP. In the RMP application to the axiom $\neg Loaded(T) \vee \neg Shoot(\overline{T}) \vee \neg Alive(T+1)$ in clause form, $Shoot(\overline{4})$ is resolved with first since it is a fact. Next, $Loaded(4)$ is used in favor of $alive(5)$ in the resolution due to the projection being at an earlier time. It is allowed to play its effects before the projection at the next time is considered. The projection $Loaded(4)$ used in the inference is used to annotate the inference $\neg Alive_c(5)[Loaded(4)]$, and the result is noted as a possible point of inflection in the value of the predicate *Alive*.

6: $\mathbf{CS}(6, null, \{Alive(0)_{obs}, \neg Alive_c(5)[Loaded(4)], Loaded(0)_{obs}, Shoot(\overline{4})_{obs}\})$,
$\mathbf{Proj}(6, null, \{\underline{Alive(1:4)}, \underline{\neg Alive(6:\infty)}, Loaded(1:\infty)\})$

Rules yielding new conclusions: TP. Note that *Alive* is no longer projected to infinity but only until time 4 according to the new context set information. Also $\neg Alive$ is projected from time 6 onwards. The projection for *Loaded* remains unchanged.

$\vdots$

The witness version of the YSP gives the intuitive answer: In the context of the *null* plan, Fred must have died at step 5 as a result of the shooting, provided of course, that the default regarding the gun staying loaded up until step 4 is indeed true. $\neg Alive(5)[Loaded(4)]$ is still defeasible, only as good as $Loaded(4)$ really, and is treated as a default.

10 Resource limitations

An agent under severe time-pressure may spend a substantial amount of the available time in reasoning toward and about a plan of action. In a realistic setting, the same agent must also measure up to two other crucial resource limitations as well, namely space and computation bounds. We describe here these concerns and offer some solutions we are currently working on that address them.

The active logic formalism described thus far, and as applied to the planning domain in Section 8, suffers from the following shortcomings.

- **The space problem:** As time advances, more knowledge is gathered as a result of observations from the agent's environment and as a result of the deduction processes within. The knowledge base which is continuously expanding could potentially become so formidable that it would be completely unrealistic to assume that the agent could possibly apply all the inferences to this complete knowledge base. Usually, most of this information is not directly relevant to the development of the agent's current thread of reasoning. Our treatment of active logic for deadline-coupled planning in the past has disregarded the space problem in preference to dealing adequately with time-related issues. The space issue deserves serious attention where the original number of beliefs of the agent is large, and where very many new beliefs are added to the agent's knowledge base over time.
- **Unrealistic parallelism:** A step is defined as the time required by the agent to perform one inference or one primitive physical action in the world. Actions can be carried out in parallel if the sensors and effectors permit. For example, an agent can walk and eat simultaneously. Active logic planners treat 'think' actions within the agent in the same spirit as physical actions and recognize that they sap precious time resources. The original step-logic inference system assumes that during a given step i the agent can apply all available inference rules in parallel, to the beliefs at step $i-1$. There are two problems with this. One is the unrealistic amount of parallelism that potentially allows the agent to draw so many inferences in one time step that the meaning of what constitutes a step begins to blur. Secondly, it is unreasonable to expect that all inference rules would have the same time granularity. For example, it is unlikely that a simple application of Modus Ponens will take just as long to fire as an inference rule to refine a plan or check for plan feasibility, especially as plans become very large. While the representation is uniformly declarative, some rules have more procedural flavor than others, and can be imagined to take more time steps. Just as there is a limit on the physical capabilities of the agent as to how many physical actions can be done in parallel in the same time step, there must be a limit to the parallel capacity of the inference engine as well.

A claim towards fully deadline-coupled reasoning would be a tall one if the model depicts an agent with an infinite attention span and infinite think capacity. In this section we propose an extension of the original step-logic formalism to take into consideration space and computation constraints.

We propose a solution to the space problem partially based on [26] as follows. The agent's current focus of attention is limited to a small fixed number of beliefs forming the STM (short term memory), while the complete belief set is archived away in a bigger associative store, namely, the LTM (long term memory). In addition, we use a QTM which is a technical device to hold the conclusions that result in each step since further inferencing with these must be stalled until the next time step. The size of the STM is a fixed number k[76].

In the most simplistic model, the STM could be represented as a queue, in which case the inference/retrieval algorithm reduces to a simple depth first or breadth first strategy depending upon whether new observations and deductions are added to the head or tail of the queue respectively. It seems that choosing the STM elements without focus consideration may lead the reasoning astray quite easily, and also lead to often incomplete threads of reasoning due to thrashing. We propose to maintain a predicate called **Focus**(...) which keeps track of the current line of reasoning. This is dynamically changed by the agent's inference mechanism and is responsible for steering the reasoning back to a particular thread even when a large

[76] What is a realistic k for a commonsense reasoner? There is psychological basis that suggests that human short-term memory holds seven-plus-or-minus-two 'chunks' of data at one time [91].

number of seemingly irrelevant inferences are drawn. Among the agent's inference rules is a set of *focus changing (FC)* rules, which when fired alter the focus. Those K beliefs from the associative LTM which are most[77] relevant to the current focus are highlighted to form the STM.

In short, the framework can be described as follows. The $QTM_{i/i+1}$ is an intermediate store of formulae that are theorems derived through the application of inference rules to the formulae in STM_i (the STM at step i). They are candidates for the STM at step $i+1$, although only K among them will be selected. Thus the results of the inference rules, can be imagined to fall into $QTM_{i/i+1}$ and are available for selection to form the STM at the next step[78]. The $focus$ and Now which are crucial to time-situated reasoning are always accessible to the agent.

FRAMEWORK:

$$\frac{i : STM_i\{...\}, Now(i), Focus(i, ...), LTM_i\{...\}}{i+1 : STM_{i+1}\{...\}, Now(i+1), Focus(i+1, ...), LTM_{i+1}\{...\}}$$

$QTM_{i/i+1}$ holds β if β is an i-theorem. It includes relevant formulae which are retrieved from the LTM using the retrieval rule. Step i concludes by selecting K formulae from $QTM_{i/i+1}$ which are relevant to $Focus_i$ to form STM_{i+1}. LTM_{i+1} is LTM_i appended with $QTM_{i/i+1}$.

The main problem in limiting the space of reasoning is to decide what should be in the focus. In our planning framework, we have developed a mechanism that is at work to limit the focus to a single feasible plan at a given time step. A list of actions, conditions and results from the plan that need further processing (we call it the active list), form a list of keywords in the focus. We describe the details of this mechanism in [101]. Heuristic rules are proposed to maximize the probability of finding a solution within the deadline. This would correspond to a sort of best first strategy or a beam search of width K in the general framework. Although these heuristic rules are independent of the instance of the problem in question, they are likely to be different depending upon the category of the problem being solved. A deadline-coupled actor-planner is likely to maintain a much narrower focus than a long-range 'armchair' planner. We refer to [105] for some of the specific heuristic strategies employed for the tightly time-constrained planner.

The following theorem demonstrates that under appropriate conditions, any inference derivable in an active logic with no memory limitation, can also be derived in a memory limited active logic. The size of STM (i.e., K) can be as small as two beliefs. Let $SL(OBS, INF)$ denote an active logic with an inference function INF, an observation function OBS, and unlimited memory as described in [27].[79] Let $SL_K^{FET}(OBS, INF)$ denote the corresponding active logic with a limited short-term memory of size K and an algorithm, called FET, describing the strategy for fetching elements into STM.

Theorem 10.1 *Let $K \geq 2$. If all the inference rules in INF are monotonic then it is possible to describe a (simple) algorithm FET such that any theorem of $SL(OBS, INF)$ will eventually appear as a theorem of $SL_K^{FET}(OBS, INF)$. I.e. if $\vdash_i \alpha$ in SL (α was proven at step i) then $\exists j$ such that $\vdash_j \alpha$ in $SL_K^{FET}(OBS, INF)$.*

Note: the requirement of monotonicity in particular entails that the "clock"-rule for Now is left out. Thus the result applies only to Now-free inferences. We also assume that new observations are consistent with previous facts and derivations.

For a slightly different solution to the space problem in active logic (which also is based on [26]), see [137].

Next, we address the bounded computation resource problem. An intelligent agent can be expected to have a sizable reservoir of inference rules acquired during its lifetime. Firing of an inference rule corresponds to a 'think' action. Without a bound on its inferencing power, the agent could fire all the inference rules applicable (termed in conventional production systems as the conflict set) simultaneously during a time step. We limit the inference capacity of the engine to I. Each inference rule j is assigned a drain factor d_j. This is a measure of the drain incurred by the inference engine while firing an instance of this rule. For instance, Modus Ponens and the more elaborate inference rule for plan refinement, would be given different drain factors to reflect this difference in granularity [80].

Our limited-capacity inference engine fires only a subset of the applicable rules in each time step. Among the various alternatives, it is possible to pick the inference rules either completely nondeterministically up to the engine capacity I,

[77] There is then a ranking among the relevant formulae, and the k formulae at the top of the list are picked. In our implementation, we select the k formulae at random from the candidate formulae.

[78] This has the feature that all thinking does not pass through the STM unless it is relevant to the focus.

[79] In this section, familiarity with the notation in [27] is assumed.

[80] How to calibrate the inference rules for the assignment of these drain factors is a separate and interesting issue, but we will not address it presently. Also, how thinking actions compare with physical actions is a technical issue that could be resolved by trying to calibrate the system to check on the relative speed of its inference cycle with that of its sensors and motors. We skip this implementation sensitive issue for the present.

or one could again apply some heuristics to improve the agent's chances. Several parameters, such as agent attitudes, the uncertainty of the environment, or the urgency to act could dictate this choice.

Thus, in effect, during each step, K beliefs are highlighted from the knowledge base (LTM) to constitute the STM. From among the rules applicable to these K beliefs, a subset of rules is chosen such that sum of the drain factors does not exceed the engine's inference capacity I. The results of the inferencing are put in the QTM. Finally, the contents of the QTM are copied to the LTM.

11 Conclusions and Future Work

In this paper we have tried to present a broad picture of the many different issues and problems that can be tackled using active logic. These range from highly general issues to specific applications.

Our overall aim is to provide a formal framework in which large portions of commonsense reasoning can be carried out in a realistic manner. We have described our efforts in the areas of planning in deadline situations, multi-agent reasoning, reasoning with contradictions, and language change. All of these were carried out in the same underlying framework of active logics. Future work includes combining these into a single system, so that a single agent would be able to perform all these activities within one integrated episode *a la* Agenta. A key step we will take is to standardize active logic, and make it publicly available on-line.

To be sure, active logics studied to date do not do all that one would like. Their biggest shortcoming is that they indulge in litter-bugging (as also do static logics): too many unwanted theorems are produced, creating a space problem. This is not a serious problem for static logics *per se* (since the associated idealization relegates such concerns to an independent engine); nor need it be so for active logics: we can view them as well as idealizations, though a bit less removed from realism than their static counterparts. For it is not mere resource-limitations that motivate active logics. Nevertheless, the definition of active logics is general enough not to rule out space-saving ("tidy") versions (see Section 10); but it is hard to come up with plausible candidates. Thus litter-bugging is not forced on us by active logics; it is our poor understanding of key issues such as relevance, focus of attention, and memory management that is the sticking point. Our expectation is that as more is learned about these issues, it will be possible to incorporate them into the active logic framework.

Other areas of ongoing and future work include: pragmatics of discourse [47, 111]; rule-change (allowing the agent to alter not only beliefs but also the inference rules governing belief formation and inheritance); and situated feedback (so that reasoning can be more closely tied to actions, as in deciding it is now time to eat lunch, initiating action toward that goal, and noting the action is taking place).

Our long-range goal of the design and implementation of a highly flexible episodic reasoner, one that can carry on the sort of reasoning presented in the Agenta story of barn-painting, is years away. It involves many difficult issues of both theoretical and practical scope, yet we are convinced both that it is essential to keep such goals well in mind as a guide to research directions, and that active logics (or something similar) will be needed to achieve this end by principled means. Omniscient or prescient thinking (an oxymoron) can neither govern nor usefully model situated episodic reasoning.

References

[1] C. Alchourron, P. Gardenfors, and D. Makinson. On the logic of theory change. *J. Symbolic Logic*, 50:510–530, 1985.

[2] J. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

[3] A. Baker. A simple solution to the Yale Shooting Problem. In *Proc. First Int'l Conf. on Principles of Knowledge Representation and Reasoning*, pages 11–20, 1989.

[4] Afzal Ballim and Yorick Wilks. *Artificial Believers: The Ascription of Belief.* Lawrence Erlbaum Associates, 1991.

[5] M. Boddy and T. Dean. Deliberation scheduling for problem solving in time-constrained environments. *Artificial Intelligence*, 1994. To appear.

[6] A. H. Bond and L. Gasser. An analysis of problems and research in DAI. In A. H. Bond and L. Gasser, editors, *Readings in DAI*, pages 3–35. Morgan Kaufmann Pub., Inc., Ca., 1988.

[7] R. Brooks. Intelligence without reason. In *Proceedings of IJCAI-91, Prepared for Computers and Thought*, 1991.

[8] Rodney Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[9] B. Chellas. *Modal Logic*. Cambridge University Press, 1980.

[10] Keith L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logics and Data Bases*, pages 293–322. Plenum Press, 1978.

[11] Philip R. Cohen and Hector J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(3), 1990.

[12] T. Dean and M. Boddy. An analysis of time-dependent planning. In *Proceedings of the 7th National Conference on Artificial Intelligence*, pages 49–54, St. Paul, MN, 1988. AAAI.

[13] J. deKleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[14] J. Doyle. A truth maintenance system. *Artificial Intelligence*, 12(3):231–272, 1979.

[15] J. Doyle. Some theories of reasoned assumptions: An essay in rational psychology. Technical report, Department of Computer Science, Carnegie Mellon University, 1982.

[16] J. Doyle. Artificial intelligence and rational self-government. Technical Report CS-88-124, Carnegie-Mellon University Computer Science Department, 1988.

[17] J. Doyle. Rationality and its roles in reasoning. *Computational Intelligence*, 8(2):376–409, 1992.

[18] J. Drapkin and D. Perlis. Analytic completeness in SL_0. Technical Report TR-1682, Department of Computer Science, University of Maryland, College Park, Maryland, 1986.

[19] J. Drapkin and D. Perlis. A preliminary excursion into step-logics. In *Proceedings SIGART International Symposium on Methodologies for Intelligent Systems*, pages 262–269. ACM, 1986. Knoxville, Tennessee.

[20] J. Drapkin and D. Perlis. Step-logics: An alternative approach to limited reasoning. In *Proceedings of the European Conf. on Artificial Intelligence*, pages 160–163, 1986. Brighton, England.

[21] Edmund H. Durfee. *Coordination of Distributed Problem Solvers*. Kluwer Academic Publishers, Boston, 1988.

[22] J. Elgot-Drapkin. *Step-logic: Reasoning Situated in Time*. PhD thesis, Department of Computer Science, University of Maryland, College Park, Maryland, 1988. (Directed by D. Perlis.) Available as Technical Report CS-TR-2156, Technical Report UMIACS-TR-88-94, and through UMI order no. 8912283.

[23] J. Elgot-Drapkin. A real-time solution to the wise-men problem. In *Proceedings of the AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, pages 33–40, 1991. Stanford, CA.

[24] J. Elgot-Drapkin. Step-logic and the three-wise-men problem. In *Proceedings of the 9th National Conference on Artificial Intelligence*, pages 412–417, Anaheim, California, 1991.

[25] J. Elgot-Drapkin. Reasoning situated in time II: The three-wise-men problem. In preparation, 1996.

[26] J. Elgot-Drapkin, M. Miller, and D. Perlis. Life on a desert island: Ongoing work on real-time reasoning. In F. M. Brown, editor, *Proceedings of the 1987 Workshop on The Frame Problem*, pages 349–357. Morgan Kaufmann, 1987. Lawrence, Kansas.

[27] J. Elgot-Drapkin and D. Perlis. Reasoning situated in time I: Basic concepts. *Journal of Experimental and Theoretical Artificial Intelligence*, 2(1):75–98, 1990.

[28] R. Fagin and J. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34(1):39—76, 1988.

[29] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about knowledge*. MIT Press, 1995.

[30] R. Fagin, J. Halpern, and M. Vardi. A nonstandard approach to the logical omniscience problem. *Artificial Intelligence*, 79:203–240, 1995.

[31] M. Fenster, S. Kraus, and J. Rosenschein. Coordination without communication: Experimental validation of focal point techniques. In *Proc. of the First International Conference on Multiagent Systems*, pages 102–116, California, USA, 1995.

[32] J. Fodor. *The Language of Thought.* Harvard University Press, 1979.

[33] A. Frisch and P. Haddawy. Anytime deduction for probabilistic logic. *Artificial Intelligence*, 69:93–122, 1994.

[34] P. Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States.* MIT Press, Cambridge, MA, 1988.

[35] A. Garvey and V. Lesser. A survey of research in deliberative real-time artificial intelligence. *Real-Time Systems*, 6:317–347, 1994.

[36] K. Gary. RABIT: A spreading activation approach to real-time commonsense reasoning. Master's thesis, Arizona State University, Tempe, Arizona, 1993.

[37] K. Gary and J. Elgot-Drapkin. RABIT: Bridging formal and implementational approaches to commonsense reasoning. Submitted for publication, 1996.

[38] Michael Gelfond. Autoepistemic logic and formalization of common-sense reasoning. In *Proceedings of the second international workshop on nonmonotonic reasoning*, 1988. Munich 88.

[39] M. P. Georgeff. Many agents are better than one. In F. M. Brown, editor, *Proceedings of the 1987 Workshop on The Frame Problem*, pages 59–75x, Lawrence, Kansas, 1987. Morgan Kaufmann.

[40] M. Ginsberg, editor. *Readings in Nonmonotonic Reasoning.* Morgan Kaufmann, 1987.

[41] Fausto Giunchiglia. An epistemological science of comon sense. *Artificial Intelligence*, 77:371–392, 1995.

[42] A. Globerman and S. Kraus. A modal active logic with focus of attention for reasoning in time. in preperation.

[43] J. Goodwin. *A Theory and System for Non-Monotonic Reasoning.* PhD thesis, Department of Computer and Information Science, Linköping University, Linköping, Sweden, 1987.

[44] S. D. Goodwin and H.J. Hamilton, editors. *Proceedings of the TIME-95 International Workshop on Temporal Representation and Reasoning.* University of Regina, 1995.

[45] R. Guha and D. Lenat. Cyc: a midterm report. *AI Magazine*, 11(3):32–59, 1990.

[46] R. V. Guha. *Contexts: A formalization and some applications.* PhD thesis, Stanford University, Palo Alto, CA, 1991.

[47] J. Gurney, D. Perlis, and K. Purang. Active logic and Heim's rule for updating discourse context. In *IJCAI 95 Workshop on Context in Natural Language*, 1995.

[48] R. Guttman. SOAR and RABIT: A preliminary comparison of two AI architectures. Technical Report TR–94–000, Department of Computer Science and Engineering, Arizona State University, Tempe, AZ, January 1994.

[49] A. Haas. Possible events, actual events, and robots. *Computational Intelligence*, 1, 1985.

[50] A. Haas. A syntactic theory of belief and action. *Artificial Intelligence*, 28:245–292, 1986.

[51] P. Haddawy and A. Frisch. Convergent deduction for probabilistic logic. In *Proceedings of Third Workshop on Uncertainty in Artificial Intelligence*, pages 278–286, Seattle, WA, 1987.

[52] J. Halpern and Y. Shoham. A propositional modal logic of time intervals. In *Proceedings of the Symposium on Logic in Computer Science*, Boston, MA, 1986. IEEE.

[53] S. Hanks and D. McDermott. Nonmonotonic logic and temporal projection. *Artificial Intelligence*, 33:379–412, 1987.

[54] B. Haugh. Simple causal minimizations for temporal persistence and projection. In *Proceedings of the 6th National Conference on Artificial Intelligence*, pages 218–223, Seattle, WA, 1987. AAAI.

[55] E. Horvitz, G. Cooper, and D. Heckerman. Reflection and action under scare resources: Theoretical principles and empirical study. In *Proceedings of IJCAI-89*, pages 1121–1127, Detroit, Michigan, 1989.

[56] E. J. Horvitz. Reasoning under varying and uncertain resource constraints. In *Proceedings of the 7th National Conference on Artificial Intelligence*, pages 111–116, St. Paul, MN, 1988. AAAI.

[57] Eric J. Horvitz and Geoffrey Rutledge. Time-dependent utility and action under uncertainity. In *Uncertainity in Artificial Intelligence 6*, 1991.

[58] F. Ingrand and M. Georgeff. Managing deliberation and reasoning in real-time ai systems. In *Proceedings of 1990 DARPA workshop on Innovative Approaches to Planning, Scheduling and Control*, pages 284–291, San Diego, CA, November 1990.

[59] Nick R. Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence Journal*, 75(2):1–46, 1995.

[60] H Kautz. The logic of persistence. In *Proceedings of the 5th National Conference on Artificial Intelligence*, pages 401–405. AAAI, 1986.

[61] K. Konolige. A deductive model of belief. In *Proceedings of the 8th Int'l Joint Conference on Artificial Intelligence*, pages 377–381, Karlsruhe, West Germany, 1983.

[62] K. Konolige. *A Deduction Model of Belief.* Pitman, London, 1986.

[63] S. Kraus and D. Lehmann. Knowledge, belief and time. *Theoretical Computer Science*, 58:155–174, 1988.

[64] S. Kraus, M. Nirkhe, and D. Perlis. Deadline-coupled real-time planning. In *Proceedings of 1990 DARPA workshop on Innovative Approaches to Planning, Scheduling and Control*, pages 100–108, San Diego, CA, 1990.

[65] S. Kraus, M. Nirkhe, and D. Perlis. Planning and acting in deadline situations. In *Proceedings of AAAI-90 Workshop on Planning in Complex Domains*, 1990.

[66] S. Kraus and J. Rosenschein. The role of representation in interaction: Discovering focal points among alternative solutions. In *Decentralized Artificial Intelligence, Volume 3*, Germany, 1992. Elsevier Science Publishers.

[67] S. Kraus, J. Wilkenfeld, and G. Zlotkin. Multiagent negotiation under time constraints. *Artificial Intelligence*, 75(2):297–345, 1995.

[68] J. E. Laird, A. Newell, and P. S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64, 1987.

[69] G. Lakemeyer. Limited reasoning in first-order knowledge bases. *Artificial Intelligence*, 69:213–255, 1994.

[70] D. Lenat and R. V. Guha. Cyc: a midterm report. *AI Magazine*, 11(3):32–59, 1990.

[71] D. Lenat, R. V. Guha, K. Pittman, D. Pratt, and M. Shepherd. Cyc: Toward programs with common sense. *Communications of the ACM*, 33(8):30–49, August 1990.

[72] Y. Lespérance and H. Levesque. Indexical knowledge and robot action—a logical account. *Artificial Intelligence*, 73(1,2):69–115, 1995.

[73] V.R. Lesser. A retrospective view of FA/C distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics, Special Issue on Distributed Artificial Intelligence*, 21(6):1347–1362, December 1991.

[74] H. Levesque. A logic of implicit and explicit belief. In *Proceedings of the 3rd National Conf. on Artificial Intelligence*, pages 198–202, 1984. Austin, TX.

[75] H. Levesque. All I know: A study in autoepistemic logic. *Artificial Intelligence*, 42:213–265, 1990.

[76] V. Lifschitz. Formal theories of action (preliminary report). In *Proceedings of the 10th Int'l Joint Conference on Artificial Intelligence*, pages 966–972, Milan, Italy, 1987. Morgan Kaufmann.

[77] V. Lifschitz. Pointwise circumscription. In M. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*, pages 179–193. Morgan Kaufmann, 1987.

[78] A. Maida. Maintaining mental models of agents who have existential misconceptions. *Artificial Intelligence*, 50(3):331–383, 1991.

[79] J. Malik and T. Binford. Reasoning in time and space. In *Proceedings of the 8th Int'l Joint Conference on Artificial Intelligence*, pages 343–345, Karlsruhe, West Germany, 1983.

[80] J. McCarthy. Programs with common sense. In *Proceedings of the Symposium on the Mechanization of Thought Processes*, pages 77–84, Teddington, England, 1958. National Physical Laboratory.

[81] J. McCarthy. Formalization of two puzzles involving knowledge. Unpublished note, Stanford University, 1978.

[82] J. McCarthy. Circumscription: A form of non-monotonic reasoning. *Artificial Intelligence*, 13(1,2):27–39, 1980.

[83] J. McCarthy. Notes on formalizing context. In *Second Workshop on Formal Common Sense Reasoning*, 1993.

[84] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, pages 463–502. Edinburgh University Press, 1969.

[85] J. McCarthy and V. Lifschitz. Commentary on McDermott. *Computational Intelligence*, 3(3):196–197, 1987.

[86] D. McDermott. Planning and acting. *Cognitive Science*, 2:71–109, 1978.

[87] D. McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155, 1982.

[88] D. McDermott and J. Doyle. Non-monotonic logic I. *Artificial Intelligence*, 13(1,2):41–72, 1980.

[89] E. McKenzie and R. Snodgrass. Extending the relational algebra to support transaction time. In *Proceedings of the SIGMOD Conference*, pages 454–466, San Francisco, CA, 1987. ACM.

[90] E. Mendelson. *Introduction to Mathematical Logic*. Wadsworth, Belmont, CA, 3rd edition, 1987.

[91] G. Miller. The magical number seven plus or minus two. *The Psychological Review*, 63:81–97, 1956.

[92] M. Miller. *A view of one's past and other aspects of reasoned change in belief*. PhD thesis, Department of Computer Science, University of Maryland, College Park, Maryland, 1993. (Directed by D. Perlis.).

[93] M. Miller and D. Perlis. Language change. In preparation, 1996.

[94] Elijah Millgram. Rational goal acquisition in highly adaptive agents. In *Proceedings of the AAAI-95 Fall Symposium Series on Rational Agency: Concepts, Theories, Models, and Applications*, pages 105–107, 1995.

[95] Marvin Minsky. *The Society of Mind*. Simon & Schuster, 1986.

[96] T. M. Mitchell, R. Keller, and S. Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine Learning*, 1(1):47–80, January 1986.

[97] R. Moore. Semantical considerations on nonmonotonic logic. In *Proceedings of the 8th Int'l Joint Conf. on Artificial Intelligence*, 1983. Karlsruhe, West Germany.

[98] R. Moore. *Formal Theories of the Commonsense World*, chapter A Formal Theory of Knowledge and Action, pages 319–358. Ablex Publishing Company, 1985.

[99] L. Morgenstern and L. A. Stein. Why things go wrong: A formal theory of causal reasoning. In *Proceeding, AAAI-88*, pages 518–523, 1988.

[100] P. H. Morris. The anomalous extension problem in default reasoning. *Artificial Intelligence*, 35:383–399, 1988.

[101] M. Nirkhe. *Time-Situated Reasoning within Tight Deadlines and Realistic Space and Computation Bound*. PhD thesis, Department of Electrical Engineering, University of Maryland, College Park, MD, 1994.

[102] M. Nirkhe and S. Kraus. Formal real-time imagination. *Fundamenta Informaticae, special issue on Formal Imagination*, 23(2,3,4):371–390, 1995.

[103] M. Nirkhe, S. Kraus, M. Miller, and D. Perlis. How to (plan to) meet a deadline between now and then. *Journal of Logic and Computation*, 1996. in press.

[104] M. Nirkhe, S. Kraus, and D. Perlis. Fully deadline-coupled planning: One step at a time. In *Proceedings of the Sixth International Symposium on Methodologies for Intelligent Systems*, Charlotte, NC, 1991.

[105] M. Nirkhe, S. Kraus, and D. Perlis. Situated reasoning within tight deadlines and realistic space and computation bounds. In *Proceedings of the Second Symposium On Logical Formalizations Of Commonsense Reasoning*, Austin, Texas, 1993.

[106] M. Nirkhe, S. Kraus, and D. Perlis. Thinking takes time: A modal active-logic for reasoning in time. In *Proc. of BISFAI-95*, 1995.

[107] J. Pearl. On logic and probability. *Computational Intelligence*, 4:99–103, 1988.

[108] D. Perlis. Languages with self reference I: Foundations. *Artificial Intelligence*, 25:301–322, 1985.

[109] D. Perlis. Putting one's foot in one's head—part I: Why. *Noûs*, 25:325–332, 1991. Special issue on Artificial Intelligence and Cognitive Science.

[110] D. Perlis, J. Elgot-Drapkin, and M. Miller. Stop the world!—I want to think! *International J. of Intelligent Systems*, 6:443–456, 1991. Special issue on temporal reasoning.

[111] D. Perlis, K. Purang, and J. Gurney. Active logic applied to cancellation of gricean implicature. In *AAAI 96 Spring Symposium on Computational Implicature*, 1996.

[112] Donald Perlis. Logic for a lifetime. Technical Report CS-TR-3278 and UMIACS-TR-94-62, University of Maryland, 1994.

[113] Donald Perlis. Sources of, and exploiting, inconsistency: preliminary report. In *1996 Workshop on Commonsense Reasoning*, Stanford, 1996.

[114] M. E. Pollack and M. Ringuette. Introducing the tileworld: Experimentally evaluating agent architectures. In *Proceedings, AAAI-90*, pages 183–189, 1990.

[115] J. Pollock. How to reason defeasibly. *Artificial Intelligence*, 57:1–42, 1992.

[116] J. L. Pollock. *Cognitive Carpentry*. Bradford/MIT Press, 1995.

[117] John Pollock. Oscar: a general theory of rationality. *Journal of Experimental and Theoretical Artificial Intelligence*, 1(3):209–226, 1989.

[118] E. Rasmusen. *Games and Information*. Basil Blackwell Ltd., Cambridge, Ma, 1989.

[119] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1,2):81–132, 1980.

[120] Raymond Reiter. *Nonmonotonic Reasoning*, volume 2 of *Annual Reviews in Computer Science*, pages 147–186. Annual Reviews Inc., 1987.

[121] C. Rieger. *Conceptual Memory: A Theory and Computer Program for Processing the Meaning Content of Natural-Language Utterances*. PhD thesis, Department of Computer Science, Stanford University, Palo Alto, California, 1974.

[122] N. Roos. A logic for reasoning with inconsistent knowledge. *Artificial Intelligence*, 57:69–103, 1992.

[123] P.S. Rosenbloom, J.E. Laird, A. Newell, and R. McCarl. A preliminary analysis of the soar architecture as a basis for general intelligence. *Artificial Intelligence*, 47:289–325, 1991.

[124] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Boston, 1994.

[125] S. Russell and E. Wefald. Principles of metareasoning. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*. Morgan-Kaufman, 1989.

[126] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[127] S. J. Russell and D. Subramanian. Provably bounded-optimal agents. *Journal of Artificial Intelligence Research*, 2:575–609, 1995.

[128] Earl Sacerdoti. The non-linear nature of plans. In *Advance papers for IJCAI – 75*, 1975.

[129] Thomas C. Schelling. *The Strategy of Conflict*. Oxford University Press, New York, 1963.

[130] J. R. Shoenfield. *Mathematical Logic.* Addison Wesley, Reading, MA, 1967.

[131] Y. Shoham. *Reasoning About Change.* MIT Press, Cambridge, MA, 1988.

[132] Yoav Shoham. Chronological ignorance: Experiments in nonmonotonic temporal reasoning. *Artificial Intelligence*, 36:279–331, 1988.

[133] R. Smith and R. Davis. Framework for cooperation in distributed problem solvers. *IEEE Transactions on Systems, Man and Cybernetic*, C-29(12):61–70, 1981.

[134] G.A. Sussman. A computational model of skill acquistion. Technical Report AI-TR-297, MIT AI Lab, 1973.

[135] K. P. Sycara. Persuasive argumentation in negotiation. *Theory and Decisions*, 28:203–242, 1990.

[136] A. Tate. Interacting goals and their use. In *Proc. of IJCAI75*, pages 215–218, 1975.

[137] W. C. Tsai and J. Elgot-Drapkin. Bridging the gap between theoretical and practical commonsense reasoning systems: Part I. Technical Report TR–94–001, Department of Computer Science and Engineering, Arizona State University, Tempe, AZ, January 1994.

[138] S. Vere. Planning in time: Windows and durations for activities and goals. In *Proceedings, IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 246–267. IEEE Computer Society, 1983.

[139] Michael P. Wellman. Rationality in decision machines. In *Proceedings of the AAAI-95 Fall Symposium Series on Rational Agency: Concepts, Theories, Models, and Applications*, pages 154–156, 1995.

[140] R. Weyhrauch. The building of mind. In *1986 Workshop on Meta-Architectures and Reflection*, 1986. Sardinia, Italy.

[141] S. Zilberstein and S. Russell. Constructing utility-driven real-time systems using anytime algorithms. In *Proceedings of the IEEE workshop on imprecise and approximate computation*, pages 6–10, 1992.