æ

Active Logics: A Unified Formal Approach to Episodic Reasoning

Jennifer Elgot-Drapkin[1]
Sarit Kraus[2]
Michael Miller[3]
Madhura Nirkhe[3]
Donald Perlis[3]
[1]Arizona State University, Box 875406, Tempe, AZ 85287-5406
[2]Bar Ilan University, Israel
[3]University of Maryland, College Park, MD 20742

March 16, 1995

Contents

First-order logic doesn't have
a good concept of time.
Jerry Feldman, *circa* 1980
robots with a life of their own...
Nils Nilsson, 19??
Past is Prologue
Anon

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 to 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 (e.g., Minsky [**?**]). 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 from within a unified framework.

In particular, we address the issue of formal, integrated, and longitudinal reasoning: inferentially-modeled behavior that incorporates a fairly wide variety of types 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.

Out of the frying pan and into the fire?

1 Preliminary concepts

1.1 Motivation

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.

In particular, 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 concern, and many highly distinct engines may implement the same (static) logic. Some engines are slow, some fast; some use resolution, and some do not; etc.

Yet, as we shall argue below, in commonsense reasoning, the two cannot be separated: many commonsense "theorems" depend crucially on their manner (e.g., time) of production (in order to be theorems). A case in point: "I've been working on this chore for 15 minutes now, and still have not finished it (maybe I should give up)"—such an assertion 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 your phone number and you come in the door (so I can put away the phone book)." In general, explanations or commentaries on one's activities tend to 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 doing. This paper is aims to present that thesis, to draw conclusions about underlying mechanisms relevant to the enterprise of formalizing such activity, and to illustrate an array of technical devices that we have developed toward such an enterprise.

Over the last eight years we have introduced and explored new logics and applied them to a number of distinct application areas. Here we bring them all together...

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, mosify its goals as unforseen 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 proposal (which we are already embarked upon) we next present an imaginary scenario designed to illustrate the kind of behavior we think essential to the AI dream of truly intelligent systems, and which we think active logics are well suited. To date active logics have been successfully applied to most of the technical issues in this scenario, but not yet fully integrated into a single systems of axioms, rules of inference, and procedures. Such integration is in itself be a major undertaking, but we think that many of the essential KR issue has been solved. It is the aim of this paper to motivate formal work toward such integrated systems, to illustrate one such formalism, 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 a sense we are proposing *case-based learning* in the form of trial-and-error, advice-taking, and ... as suitable and essential aspects of CSR.

The specific capabilities illustrated in the scenario, and solved in isolation by existing active-logic studies are:...

1. keep temporal indexicals up to date during reasoning
2. re-establish communication using a focal algorithm to choose a place to meet
3. perform word-sense disambiguation, especially between two agents
4. reason in real time about others' evolving knowledge and reasoning
5. perform deadline-coupled real-time planning
6. reason effectively in the presence of contradictions
7. perform effective non-monotonic reasoning
8. find sensible explanations for past events
9. make sensible predictions (YSP)

The following scenario 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 ridiculously 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 below and which appears to be the standard model of formal commonsense reasoning performed in a highly idealized setting without benefit (or detriment) of real-world interactions that may require rethinking of those idealizations.

One day in the life of AL

AL and his friend Sue have the task of painting a barn today. They make the following plan: AL 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. AL heads off to the paint store to make his purchase. At a bridge he intends to cross he notices a sign saying the bridge is under repair and uncrossable. This forces him to back up a considerable distance and take an alternate and much less direct route. As he goes, he realizes they will need two ladders and a plank, and that he should get a second ladder and a plank at the paint store.

He then sees another sign saying that all Main Street stores are closed for the day due to a water main outage, and he assumes Sue will now try to go to the more expensive paint store to get a ladder. He also assumes Sue does not know about the bridge being out, since else she surely would have warned him. AL 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. He 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.

He arrives at the paint store but the street sign reads "Ellis Avenue" rather than "Ellis Street"—he 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 he can get paint and brushes there.

He 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. He is puzzled by the apparent contradiction, then reasons that she must have found the Main Street hardware store open after all. Then he sees that the ladder has a tag on it reading "Main Steet Hardware II, Harwood Lane location." He does not know how Sue found out about the second location of the store, but evidently she did. Since Harwood Lane is very close, he decides to make a quick trip there for a second ladder and plank. Then he reasons that he should wait instead, since it is by now 12:25 and Sue will be here at any moment.

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 for ease of exposition we shall dub "AL"). 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* AL's beliefs but are not themselves candidates for those beliefs. Thus we are critiquing the "direct use of static logics for commonsense reasoning".

While AL may differ in important ways from a real-world reasoner, nevertheless he (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 AL leaves out these details: all that is found in him 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.[1] Yet it does no such thing. 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 AL's theory initially and T_2 a subsequent theory held by AL; just how it is that AL gives up T_1 and adopts T_2 is left up in the air, except to say that something changed his beliefs; we are asked to suppose that at least one new belief is handed AL, and this apparently causes the rest of the change. Thus AL undergoes "bursts" of reasoning, each characterized by a beginning and an end. First AL has beliefs (corresponding to) T_1, then something happens, and eventually AL 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 AL'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.

In particular, we believe (and will argue) that direct static logics are *inherently incapable* of representing many aspects of reasoning essential to commonsense reasoning, including:

1. real-time planning with deadlines

[1] quote McDermott/Doyle, Reiter, etc

2. change of belief (McDermott-Doyle, Reiter)

3. change of terminology (McCarthy-Lifschitz)

4. enlargement of language

5. contradiction-resolution (Roos)

6. recognition of errors

7. tractable NMR

8. learning

Active logics (to be defined below) do appear to be capable of representing all the above kinds of reasoning; versions of all the above have been formalized using active logics and applied to (simple) examples.

To the best of our knowledge, the first active logics studied were the step-logics of Elgot-Drapkin and Perlis. 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.

We now offer a very general definition of an active logic; this will include the various versions of step-logic to date:

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 haven't) been proven so far (this is also true of static logics) 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.

In step-logics in particular, 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).

1.2 Old Introduction

Rosenschein et al have looked a bit at embedded reasoning, but not quite in the same way...

A particularly vexing aspect of this type of reasoning is what we call the *swamping problem*—namely that from a contradiction all wffs are concluded. For this reason most formal studies of reasoning deliberately avoid contradictions; those that do not (e.g., Doyle [**?**]), provide a separate device

for noting contradictions and revising beliefs while the "main" reasoning engine sits quiescent. In general, however, this will not do, since the knowledge needed to resolve conflicts will depend on the same wealth of world knowledge used in any other reasoning. Thus reasoning about birds involves inference rules applied to beliefs about birds, whether used to resolve a conflict or simply to produce non-conflicting conclusions. We contend, then, that one and the same on-going process of reasoning should be responsible both for keeping itself apprised of contradictions and their resolution, and for other forms of reasoning.

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. Thus Konolige [**?**] studies agents with fairly arbitrary rules of inference, but assumes logical closure for the agents with respect to those rules, ignoring the effort involved in performing the deductions. Similarly, Levesque [**?**] and Fagin and Halpern [**?**] provide formal treatments of limited reasoning, so that, for instance, a contradiction may go unnoticed; but the conclusions that *are* drawn are done so instantaneously, i.e., the steps of reasoning involved are not explicit. Fagin and Halpern in particular postulate a notion of awareness, so that if α and $\alpha \rightarrow \beta$ are known, still β will not be concluded unless the agent is aware of β; just how it is that β fails to be in the awareness set in unclear. 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 [**?**] comes a little closer to meeting our desiderata but still maintains a largely final-tray-like perspective.

Ordinary logic serves well the purpose of modelling a reasoning agent's activity from *afar*, as a meta-theory about the agent. This is a useful thing; still, it is also of interest to have a direct representation of the evolving process of the very reasoning itself. This can be done in ordinary logic if the representation is in the meta-theory, say by means of a time argument to a predicate representing the agent's proof process. Indeed, the most elementary step-logic we have proposed is just of this sort. However, in order for the agent to reason about the passage of time that occurs *as* it reasons, time arguments must be put into the agent's own language. That is, such an agent's logic (a step-logic) would evolve and represent that evolving history at the same time. Can this be anything at all like a traditional logic? It can, and not merely by implementing a deductive engine and watching it go through states one by one. This is not so very surprising, for this seems to be what humans do: we are constantly going on in time, and yet reasoning in time, even reasoning about time as we go on in time.

There will be salient differences from ordinary logic, however. Since time goes on as the agent reasons, and since this phenomenon is part of what is to be reasoned about, the agent will need to take note of facts that come and go, e.g., "It is now 3pm and I am just starting this task ... Now it is no

longer 3pm, but rather it is 3:15pm, and I still have not finished the task I began at 3pm." So, as time (and the agent's reasoning) goes on, the former conclusion that "It is now 3pm" needs to be retracted, in favor of the new conclusion "It is now 3:15pm". This immediately puts us in a non-traditional setting, for we lose monotonicity: as the history evolves, conclusions may be lost.[2] Their loss, however, need not be considered a weakness, but rather a strength, based on a reasoned assessment of a changing situation. It is clear, then, that a step-logic cannot in general retain or inherit all conclusions from one step to the next. We caution the reader to keep this in mind in our examples. Despite this feature, we will see that step-logic is primarily a deductive apparatus.[3]

The issue of representing time within a logic has been studied intensively, e.g., by Allen [**?**], McDermott [**?**], and McKenzie and Snodgrass [**?**]. However, such representations of time are not related in any obvious way to the process of actually producing theorems in that *same* logic. In effect, we want to augment logic with a notion of "now", which appropriately changes as deductions are performed. It turns out that this is not an easy task. While there are many issues related to the general approach we are advocating, we will concentrate here on describing some useful technical devices in time-situated reasoning that pertain to negative introspection. æ

We begin with a description of the active logic approach and philosophy. The subsequent sections then describe particular types of problems we have been able to solve using this formal approach. Section 3 .... In Section 5 .... Section **??** .... In Section **??** .... Section **??** .... In Section 15 we briefly describe our techniques for handling limited resources of time, space, and computation. Section **??** describes how active logics have been used for multi-agent coordination without communication through the use of focal points. (See [**?**] for more details.) We then conclude in Section **??** with future directions for our research in active logics.

æ

[2] That is, the new information that "it is now 3:15pm" can be thought of as erasing the old information that "it is now 3pm". While this is not strictly non-monotonic in the usual sense, it has a similar flavor.

[3] To be sure, non-monotonic formalisms already exist in the literature [**?**, **?**, **?**]. However, they do not explicitly treat on-going processes in the reasoning modelled. We suspect that the finely honed non-monotonicities in those studies may amount to a kind of temporal reasoning that would be brought out if they were applied to problems in which an agent comes across conflicts; see [**?**].

1.3 Related work

Halpern-Moses-Vardi

SOAR

CYC

OSCAR

1.4 Active logic approach and philosophy

2 Active logic approach and philosophy

2.1 How can a logic keep track of time as theorems are proven?

Step-logics were introduced in [**?**, **?**, **?**] to model a commonsense agent's ongoing process of reasoning in a changing world. They have since been extended and renamed as *active logics* to allow several new features, including limited short-term memory (see Section 15, and the introduction of new expressions into the language over time (see Section **??**.

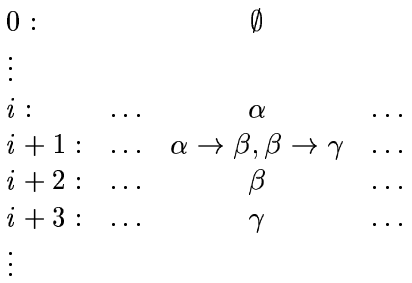

Figure 1: Step-logic studies

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 $i-1$.

The agent's world knowledge is 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 [**?**] illustrates three steps in an active logic with *Modus Ponens* ($\frac{i:\alpha, \alpha \to \beta}{i+1:\beta}$) as one of its inference rules.

The following features of this framework relate and contrast it to conventional commonsense reasoning systems:[4]

Thinking takes time: Reasoning actions occur concurrently with other physical actions of the agent and with the ticking of a clock. The agent 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. 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 the agent's ongoing lifetime of changing views of the world. This leads to the issue of contradictions below.

Omniscience: An agent reasoning with active logic 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.[5]

Handling contradictions: Consider Fermat's Last Theorem (FLT). Suppose G believes FLT is true (after reading so in the New York Times). But (let us suppose) in fact FLT is false; then G 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, 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 (the swamping problem), 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 that the commonsense agent must be capable of [**?**].

This next section details the basics of our formalism.

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

[5]Konolige [**?**], Levesque [**?**] and Fagin and Halpern [**?**] proposed systems in which the agents are not omniscient. However, the inference time is not explicitly captured in their systems.

2.2 The Basics

We return now to the idea that there are two distinct types of formalisms of interest, that occur in pairs: 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 step-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 a step-logic *pair*.

We propose three major mechanisms to study as possible aspects of an agent-theory: self-knowledge, time, and retraction. Since it is important for the agent to reason about its own processes, a self, or belief, predicate is needed. We employ a predicate symbol, K, for this purpose: $K(i, \text{`}\alpha\text{'})$ is intended to mean that the agent knows wff α at time i.[6] K may or may not be part of the agent's own language; however, many kinds of reasoning require that it be. Note that 'α' is a name for α, i.e., a constant term.[7] We drop the quotes in $K(i, \text{`}\alpha\text{'})$ in the remainder of the paper.

In order for the agent to reason about time, a time predicate is needed. This not only amounts to a parameter such as i in $K(i, \alpha)$ as we just saw, but information as to how i relates to the on-going time as deductions are performed. Thus the agent should have information as to what time it is *now*, and this should change as deductions are performed. We use the predicate expression $Now(i)$ to mean the time currently is i. Again, this may or may not be part of the agent's language, but in many cases of interest it is.

Finally, since we want to be able to deal with commonsense reasoning, the agent will have to use default reasoning. That is, a particular fact may be believed if there is no evidence to the contrary; however, later, in the face of new evidence, the former belief may be retracted. For this, we need some kind of a retraction device. Retraction will be facilitated by focusing on the dual: inheritance. We do *not* assume that all deductions at time i are inherited (retained) at time $i+1$. By carefully restricting inheritance we achieve a rudimentary kind of retraction. The most obvious case is that of $Now(i)$. If at a given step the agent knows the time to be i, by having the belief $Now(i)$, then that belief shall not be inherited to the next time step.

Here we encounter a general phenomenon of temporal constraint that will pervade the rest of our development. Consider the process of concluding by default, on the basis of not knowing X "now," that X is false (where X is any assertion, possibly dependent on time). But how, at time i, can an

[6] We are not distinguishing here between belief and knowledge. See [?] for a discussion of belief vs. knowledge.

[7] In this paper we do not address the case of α having free variables in $K(i, \text{`}\alpha\text{'})$.

agent determine that it does not know X at time i? Intuitively, certain beliefs have accumulated at time i, and only *then* can the further belief be formed, that X is not among the former. Thus the negative introspective conclusion seems to come *after* the time at which X is in fact not present: it is concluded, say, at time $i+1$, that X was not known at i. Now this introspective time-delay may seem to be a mere quibble; but if we ignore it, trouble arises. For suppose that we write the above default as follows:

$$(\forall t)[(Now(t) \wedge \neg K(t, X)) \rightarrow \neg X]$$

That is, if we don't currently know X, then conclude $\neg X$. If this is one of the beliefs present at time i, and if the beliefs $Now(i)$ and $\neg K(i, X)$ are also present at time i, then indeed the conclusion $\neg X$ may be derived by some appropriate rule of inference in the next step, $i+1$. But now, let's consider the belief $\neg K(i, X)$. This appears (through negative introspection) in the set of beliefs at time i, on the basis of X *not* being in that same set. There are problems with this, for we then are not really dealing with a fixed set for time i, but rather a two-stage production in which beliefs are gathered initially and then an introspective process is allowed to add to that set, playing fast and loose with the meaning of "not being known at time i." This in turn leads to severe ambiguities, in that the very process of inserting, say, $\neg K(i, X)$ into the beliefs at time i results in something being known after all, something that was *not* really known at time i, namely $\neg K(i, X)$ itself.

But suppose we grant that some oracle manages to place all negative introspective conclusions *about* the time-i belief set *into that very same set.* This unfortunately forces an infinite set of beliefs into that set, since there are infinitely many unknown formulas at any step. Yet our approach of real-time reasoning commits us to a finite belief set at all steps. Thus we must forego the luxury of having the agent be able to know that it doesn't know a given fact *now*; instead the best that can be done is to know that it didn't know the fact a moment ago, when it last was able to scan its belief set. The act of scanning has changed the world, at least in the sense that it has taken time. Thus the agent's self-knowledge lags slightly behind. We then will represent the above default reasoning in the following altered form:

$$(\forall t)[(Now(t) \wedge \neg K(t-1, X)) \rightarrow \neg X]$$

If we didn't know X a moment ago, then conclude $\neg X$. Suppose this is a belief at time i. If at time $i-1$, we did not have the belief X, then, using the revised notion of introspection, at time i we can negatively introspect to produce the belief $\neg K(i-1, X)$. If we also have the belief $Now(i)$ at time i, a suitable form of *modus ponens* allows us to conclude $\neg X$ at time $i+1$.

Aside from obvious real-time relevance, the Now predicate is important in other ways. For instance, above we illustrated its use in representing

default information; more will appear on this later, in Section 3. The *Three-wise-men problem* involves drawing the conclusion that one has a white spot, on the basis of the behavior of others over time and in particular on how much time has elapsed; proposed solutions that do not use something like a *Now* predicate assume omniscience of the reasoners and thus lose much of the sense of the original problem. Finally, the *Nell and Dudley problem* requires a changing time so that Dudley will be able to recognize when it has become too late to do anything.

In [**?**] we proposed eight step-logic pairs, arranged in increasing sophistication, with respect to the three mechanisms above (self-knowledge, time, and retraction). In our current notation, these are $< SL_0, SL^0 >$, ..., $< SL_7, SL^7 >$. SL_0 has none of the three mechanisms, and SL_7 has all. Of the eight agent-theory/meta-theory pairs, only SL^0 and SL_7, the simplest meta-theory and the most complex agent-theory, have been studied in any detail.[8] The meta-theories all are consistent, first-order theories, and therefore complete with respect to standard first-order semantics. However, their associated agent-theories are another matter. These we do not even *want* in general to be consistent, for they are (largely) intended as formal counterparts of the reasoning of fallible agents. SL_0 is an exception, for it, as an initial effort, was constructed to do merely propositional (tautological) reasoning so we could more easily test its meta-theory, SL^0.

A notion of completeness for the meta-theory is defined as follows:

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* $SL^n \vdash \neg K(i, \alpha)$.[9]

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 step-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 will be able to establish for all our agent logics in which the logical axioms do not contain the predicate symbol "*Now*".

[8] We describe SL^0 in Section 2.4 and SL_7 in Section 2.5.

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

2.3 Definitions and Theorems

We now present several definitions, most of which 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* $OBS : \mathbf{N} \rightarrow \mathcal{P}(\mathcal{W})$, *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* $INF : \mathcal{H} \rightarrow \mathcal{P}(W)$, *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 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.[10] Definitions 2.5 and 2.6 formalize these concepts in terms of a step-logic SL_n.

Definition 2.5 *An* SL_n-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,* $SL_n(OBS, INF)$, *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* $SL_n(\cdot, INF)$.

Let $SL_n(OBS, INF)$ be an SL_n-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* $SL_n(OBS, INF) \vdash_i \alpha$ *to mean* α *is an i-theorem of* $SL_n(OBS, INF)$.[11]

[10] 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.

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

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:

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

Figure 2: Example of a particular OBS and INF

Definition 2.7 *Given a theory $SL_n(OBS, INF)$, a corresponding SL^n-*theory*, written $SL^n(OBS, INF)$, is a first-order theory having binary predicate symbol K,[12] numerals, and names for the wffs in $\mathcal{L}$, such that*

$$SL^n(OBS, INF) \vdash K(i, \alpha) \quad \textit{iff} \quad SL_n(OBS, INF) \vdash_i \alpha.$$

Thus in $SL^n(OBS, INF)$, $K(i, \alpha)$ is intended to express that α is an i-theorem of $SL_n(OBS, INF)$.[13]

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 $SL_n(OBS, INF) \vdash_i \alpha$.*

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

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

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 2.1 *If $SL_n(OBS, INF)$ has a step-model, then it is step-wise consistent.*[14]

Proof: Let $SL_n(OBS, INF)$ have a step-model $M = < M_0, M_1, \ldots, M_i, \ldots >$. Let $j \in \mathbf{N}$ be arbitrary. Then for each α in the set of j-theorems, $M_j \models \alpha$. This means that the set of j-theorems is consistent, since it has a (standard first-order) model M_j. ■

[14]This result will be useful in showing certain step-logics are consistent; however, by the same token, since many interesting step-logics are *inconsistent* (and in fact derive much of their interest from their inconsistency), step-models are not sufficiently general as defined. We intend to explore a broader concept of step-model in future work.

Theorem 2.2 (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 $M \models_i \alpha$.

Proof: Let α be an i-theorem of $SL_n(OBS, INF)$, and let M be a step-model of $SL_n(OBS, INF)$.
$SL_n(OBS, INF) \vdash_i \alpha$, so by definition of step-model, $M_i \models \alpha$, and hence (by definition of i-true) $M \models_i \alpha$. ■

2.4 SL_0 and SL^0

The first step-logic pair we investigated was $< SL_0, SL^0 >$. The language of SL_0 is propositional, where the propositional letters are $P_0, P_1, P_2, \ldots$. 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 [**?**] we formalized the meta-theory SL^0 for describing the steps taken by such an agent.[15]

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 [**?**]), 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.

Theorem 2.3 *SL^0 is analytically complete.*

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

SL^0 was studied to gain an understanding of the underlying idea of step-logic, and to gain some practical experience.[16] Although SL^0 was studied in some detail, SL_0 is not an appropriate step-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

[15] Although there we did not yet use the notational distinction of SL_0 and SL^0.

[16] An implementation of SL^0 has been written in PROLOG, and was run on an IBM PC-AT.

should have only a relatively small number of active beliefs with which to work at each step.[17]

2.5 SL_7

In this section we outline what is so far the most ambitious step-logic: SL_7.[18] SL_7, 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 in Section **??**). 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 step-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; $K(i, \alpha)$ can be thought of as stating that α is known [19] at step i; and $Contra(\{\alpha, \beta\}, i)$[20] means 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:

$$\begin{aligned} i: &\quad \ldots, \alpha \\ i+1: &\quad \ldots, \beta \end{aligned}$$

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 not assumed to be inherited or retained in

[17] 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.

[18] The earlier SL_i's are weaker versions, missing either time or retraction or belief/knowledge predicates, and therefore too weak for many types of commonsense reasoning problems. Also, SL_7, unlike SL_0, is intended not for derivation of tautologies but rather for the study of particular default capabilities; in particular, tautologies and other logical axioms are not generally employed in SL_7. Finally, we use the notation SL_7 for any of a family of step-logics whose OBS and INF involve the predicates Now and K and contain a retraction mechanism. Choosing OBS and INF therefore fixes the theory within the family.

[19] known, believed, or concluded. The distinctions between these (see [?, ?, ?]) will not be addressed here.

[20] Note this was written as $Contra(i, \alpha, \beta)$ in [?]. We change the notation for convenience.

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.

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.[21] 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 below. 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.[22] We regard the closed sub-formulas at step i as approximating the wffs that the agent is "aware of" at i.[23] 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 [?] 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.

Retractions are used to facilitate removal of certain conflicting data. Handling contradictions in a system of this sort can be quite tricky. Currently we handle contradictions by simply not inheriting the formulas directly involved.[24] 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 $SL_7(\cdot, INF_B)$, 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 below.

We formulated $SL_7(\cdot, INF_B)$ with applications such as the *Brother problem* (see Section 3) in mind. This led us 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")

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

[22] 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 Section 2.2.

[23] "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.)

[24] In future work we hope to have a mechanism for tracing the antecedents and consequents of a formula α when α is suspect, a la Doyle and deKleer (see [?, ?]), though in the context of a real-time reasoner.

are those listed in Section 3.

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)	$\frac{i :}{i+1 : Now(i+1)}$	Corresponds to looking at clock
Rule 2(OBS)	$\frac{i :}{i+1 : \alpha}$	If $\alpha \in OBS(i+1)$
Rule 3(MP)	$\frac{i : \alpha, \alpha \rightarrow \beta}{i+1 : \beta}$	Modus ponens
Rule 4(XMP)	$\frac{i : P_1\overline{a}, \ldots, P_n\overline{a}, (\forall \overline{x})[(P_1\overline{x} \wedge \ldots \wedge P_n\overline{x}) \rightarrow Q\overline{x}]}{i+1 : Q\overline{a}}$	Extended modus ponens
Rule 5(INTRO)	$\frac{i :}{i+1 : \neg K(i, \beta)}$	Negative introspection[a]
Rule 6(CONTRA)	$\frac{i : \alpha, \neg\alpha}{i+1 : Contra(\{\alpha, \neg\alpha\}, i)}$	Presence of (direct) contradiction
Rule 7(INH)	$\frac{i : \alpha}{i+1 : \alpha}$	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, and where α is not of the form $Now(\beta)$. That is, contradictions and time are not inherited.

The intuitive reason time is not inherited is that time changes at each step. (Clearly, in general one would want a stronger restriction on the inheritance of time. It is not obvious, however, what that should be. For the purposes of illustrating certain behaviors, however, a stronger restriction is not necessary.)

The intuitive reason contradicting 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 time or a contradiction, and (intuitively) it expresses a fact (that there was a contradiction at step i) that remains true.

Note that central to our approach is the idea that, for at least some conclusions that our agent is to make, the time the conclusion is drawn is important. For instance, if it concluded at time (step) 5 that some wff B is unknown, we prefer the agent to conclude $\neg K(5, B)$ rather than simply $\neg K(B)$. The reason for this is that it may indeed be true that B is unknown at time 5, but that later B becomes known; this latter event however should not force the agent to forget the (still true) fact that *at time 5*, B was unknown. On the other hand, if we put time stamps on *all* conclusions,

then B itself, once concluded, will require more complex inheritances in order to carry B on from step to step as a continuing truth. Thus it seems preferable not to time-stamp every conclusion. This leaves us with a problem of deciding which conclusions to stamp; currently we are stamping only introspections, contradictions, and "clock look-ups".

It is worth amplifying on the use of Contra. Suppose that at step i the agent has the wffs $\neg\alpha, \neg\beta$, and $\alpha \vee \beta$. (They are all i-theorems.) While these are indeed mutually inconsistent, they do not form a *direct* contradiction; it takes some further work to see the contradiction. If, for instance, at step $i+1$ the agent deduces β (say, from a further wff $\neg\alpha \wedge (\alpha \vee \beta) \rightarrow \beta$ also present at step i), then at step $i+1$ there would be a direct contradiction. This would then be noticed (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 $i+3$. 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 $i+1$, 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. See Section **??** for more on how we handle contradictions.

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.*

Proof: See the appendix for the details. The idea is to show $SL_7(OBS, INF_B)$ has a step-model, and apply Theorem 2.5. ■

Remark 2: When OBS is empty (i.e. $\forall i, OBS(i) = \emptyset$), $SL_7(OBS, INF_B)$ reduces to a "clock", i.e.,
$\forall i$, $SL_7(OBS, INF_B) \vdash_i \alpha$ iff $\alpha = Now(i)$.

æ æ

æ

2.6 Semantics

2.7 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 foremeost, 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 metatheorems) 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 metatheorem 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 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 [**?**] to give insights into the structure of the theorem-set. In the case of Icarus, 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 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 (prefered-model semantics) second-order logic, there can be no effective proof-theory that is complete.

Thus 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.

==========

Reason is inference, not truth. A model of reasoning is a model of inference processes, not of logical (semantic) consequence. E.g., deadlines.

Inconsistency is ok, or should be. No need for Gilmore-Kripke avoidance of paradox if we are modeling inference as opposed to truth.

Inference of contra is ok, as process not semantically characterized. Use semantics simply for meaning, for truth-specs of language, not as inf-char.

Toggle: infer, revise. Classically these are kept separate, e.g., in TMS or in NMR. But in fact revision is a paramount example of inference; much of our world knowledge is needed in the revision process. Thus again we have a case of self-reference: belief revision involves using our beliefs and inferential tools to revise some of those self-same beliefs derived with those self-same tools. Active logics allow this (indeed were designed with this in mind.)

What is logic? First, a language. Then meanings for the language, as well as inference rules. The two need not match, and will not match in commonsense reasoning. Some idealization is required to get a match, and with it one loses plausibility. See NKP (in preparation) for an attempted compromise.

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 inferene are linked in the very

defintion 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, its embedding only requires very particular elements, most notably a clock; also it can require an environment to supply observations. Ongoing work also aims at spatial features. 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 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. 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 due to Michael Miller.

We can attempt a very general definition of logic, to include all existing cases: we will need a collection of languages, to allow for language change as Icarus learns new expressions; and a collection of theorem-sets, to allow not only for changes in Icarus' beliefs but also for cautious reasoning (sanctioned alternative belefs). 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 Sem, Inf, and Comp:

FOL: —L— = 1, —Th— = 1, Sem, Inf, Comp

SOL: —L— = 1, —Th— = 1, Sem, Inf, non-Comp

DL: —L— = 1, —Th— = oo, Inf

CIRC: —L— = 1, —Th— = 1, Sem, Inf, partial-Comp

AEL: —L— = 1, —Th— = 1, Sem

AL: —L— = oo, —Th— = oo, Sem, Inf, partial-Comp

But if we simply aim for meaning, this is easy: we use first-order semantics, where in the intended model Now(t) means the time is now t. Indeed, in the "intended agent model" (or rather the time-sequence of intended models) each t-inference A has the meaning that A is a belief of the agent at time t. If the agent infers Now(t) at time 5, then that "belief" is true iff t=5. There will of course be all sorts of unintended models (as nearly always happens for interesting logics). We also may consider simply belief models, in which agent beliefs are true; and if there is a contradiction, there are no such models, as we would want.

Revision: semantic shift, belief revision, dialogue, use-mention, error correction, abduction (mult explanations, imagination), context.

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 shift- ing 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 ac- tive logics keep track of time taken by inference, thereby allow- ing 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 -P, is inferred at time t, then even though all formulas may be inferrable 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" , "-P")

2.8 A truth-oriented semantics

Below are some formal details of a traditional truth-oriented semantics for active logics.

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 2.18 *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 2.19 *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 2.20 *A wff α is* i-true *in a step-model M (written $M \models_i \alpha$) if $M_i \models \alpha$.*

Definition 2.21 *$SL_n(OBS, INF)$ 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 2.22 *$SL_n(OBS, INF)$ is* eventually consistent *if $\exists i$ such that $\forall j > i$, the set of j-theorems is consistent.*

Definition 2.23 *An observation-function OBS is* finite *if $\exists i$ such that $\forall j > i$, $OBS(j) = \emptyset$.*

Definition 2.24 *$SL_n(\cdot, INF)$ is* self-stabilizing *if for every finite OBS, $SL_n(OBS, INF)$ is eventually consistent.*

Remark 3:

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 2.5 *If $SL_n(OBS, INF)$ has a step-model, then it is step-wise consistent.*[25]

Proof: Let $SL_n(OBS, INF)$ have a step-model $M = < M_0, M_1, \ldots, M_i, \ldots >$. Let $j \in \mathbf{N}$ be arbitrary. Then for each α in the set of j-theorems, $M_j \models \alpha$. This means that the set of j-theorems is consistent, since it has a (standard first-order) model M_j. ■

Theorem 2.6 (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 $M \models_i \alpha$.

[25] This result will be useful in showing certain step-logics are consistent; however, by the same token, since many interesting step-logics are *inconsistent* (and in fact derive much of their interest from their inconsistency), step-models are not sufficiently general as defined. We intend to explore a broader concept of step-model in future work.

Proof: Let α be an i-theorem of $SL_n(OBS, INF)$, and let M be a step-model of $SL_n(OBS, INF)$.
$SL_n(OBS, INF) \vdash_i \alpha$, so by definition of step-model, $M_i \models \alpha$, and hence (by definition of i-true) $M \models_i \alpha$. ■

2.9 Nonmonotonicity

3 The Brother problem

We now turn to a particular problem for which active logics seem well-suited. We use Moore's *Brother problem* (see [**?**]) to provide examples of SL_7 at work. In particular, we focus on the ease with which default reasoning can be done in a step-logic. 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 step-logic, in that the negative reflection problem (determining when something is not known) reduces to a simple look-up.[26]

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 the SL_7 in Figure 3 on page 27 for this example. Each scenario has its own 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.[27] 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 $(\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B]$[28]

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.

[26] Remember, all step-logics force ((need better word)) only a finite number of beliefs at any given step.

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

[28] It appears that some arithmetic is involved here, but 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.

- 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; this is an area we are currently exploring).

3.1 Simple negative introspection succeeds

In this example 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. See Figure 4. Here, and in Sections 3.2 and 3.3, 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 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 autoepistemic knowledge stated above (Axiom 1) and the use of the alternate version of modus ponens, Rule 4.

---

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

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

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

Figure 4: 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 5.

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

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

Figure 5: Negative introspection fails appropriately

Note that a traditional final-tray-like ((phrase used before?)) approach could produce quite similar behavior to that seen in Figures 4 and 5 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 a step-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 6 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.

---

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

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

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

$i+3$: $\underline{Now(i+3)}, P \rightarrow B, (\forall x)[(Now(x) \wedge \neg K(x-1, B)) \rightarrow \neg B], P, B, \neg K(i, B), \neg K(i, \neg B),$
$\neg B, \neg K(i+1, \neg B), \underline{Contra(\{B, \neg B\}, i+2)}$

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

Figure 6: Introspection conflicts with other deduction and resolves

---

Remark 4: 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[29]).

Proof: We briefly sketch the proof of part 1 of the preceding remark. Parts 2 and 3 are similar; part 3 involves constructing a model for each step after the last inconsistent step (which happens to be step $i+3$).

Since $OBS_{B_1}(j) = \emptyset$, for $j < i$, by Remark 2, if $j < i$, $\alpha \in \vdash_j$ iff $\alpha = Now(j)$. Therefore every step in $SL_7(OBS_{B_1}, INF_B)$ up to and including step $i-1$ is consistent. From step i on we have additional theorems which must be considered. This is due to the fact that $OBS_{B_1}(i)$ is not empty. To show that step i and all subsequent steps are consistent, we propose a model M_j for each step j. In each M_j interpret the predicates in the following way: $K \equiv false, B \equiv false, P \equiv false, Now(k) \equiv k = j$, where P is any predicate other than K, B, or Now. We can then see that we have a model for each of steps i thru $i+2$. Noting that for an arbitrary step $i+k, k > 2$,

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

$$\vdash_{i+k} = \left\{ \begin{array}{l} Now(i+k), \\ P \rightarrow B, \\ (\forall x)[(Now(x) \wedge \neg K(x-1,B)) \rightarrow \neg B], \\ \neg B, \\ \neg K(i,\neg B), \neg K(i+1,\neg B), \\ \neg K(i,B), \ldots, \neg K(i+k-1,B), \\ \neg K(i,P), \ldots, \neg K(i+k-1,P) \end{array} \right\}$$

we see that, again, M_{i+k} is an appropriate model. Therefore, by Theorem 2.5, $SL_7(OBS_{B_1}, INF_B)$ is step-wise consistent. ■

æ

4 Reasoning about others' reasoning

5 The Three wise men problem

In this section we present a variation of this classic problem which was first introduced to the AI literature by McCarthy in [**?**]. This version best illustrates the type of reasoning that is so characteristic of commonsense reasoners. We shall see that active logic provides an intuitive solution to this problem.

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

We assume in this puzzle that the wise men do not lie, that they all have the same reasoning capabilities, and that they can all think at the same speed. We then can postulate that the following reasoning took place. Each wise man knows there is at least one white card. If the cards of wise man #2 and wise man #1 were black, then wise man #3 would have been able to announce immediately that his card was white. They all realize this (they are all truly wise). Since wise man #3 kept silent, either wise man #2's card is white, or wise man #1's is. At this point wise man #2 would be able to determine, if wise man #1's were black, that his card was white. They all realize this. Since wise man #2 also remains silent, wise man #1 knows his card must be white.

It is clear that it is important to be able to reason in the following manner:

> If such and such were true at that time, then so and so *would have realized it by this time.*

So, for instance, if wise man #2 is able to determine that wise man #3 would have already been able to figure out that wise man #3's card is white, and wise man #2 has heard nothing, then wise man #2 knows that wise man #3 does *not* know the color of his card. Step-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 [**?**, **?**, **?**]) from the perspective of a final state of reasoning, and thus are not able to address this temporal aspect of the problem: assessing what others have been able to

conclude *so far*. Elgot-Drapkin [**?**] provides a solution based on step-logic to a version of this problem in which there are only two men.

5.1 Formulation

The step-logic used to model the *Three-wise-men problem* is defined in Figures 7 and 8. The problem is modeled from wise man #1's point of view. The observation-function contains all the axioms that wise man #1 needs to solve the problem, and the inference-function provides the allowable rules of inference.

OBS_{W_3} is defined as follows.

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

Figure 7: OBS_{W_3} for the Three-wise-men Problem

We use an SL_5 theory. An SL_5 theory gives the reasoner knowledge of

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

its own beliefs as well as knowledge of the passage of time.[30] The language of SL_5 is first-order, having binary predicate symbols K_j and U, and function symbol s. $K_j(i, \text{`}\alpha\text{'})$ expresses the fact that "$\alpha$ is known[31] 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. $U(i, \text{`}x\text{'})$ expresses the fact that an utterance of x is made at step i.[32] $s(i)$ 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

[30] For more details on SL_n theories, see [?, ?].

[31] known, believed, or concluded. The distinctions between these (see [?, ?, ?]) are not addressed here.

[32] For simplicity, in the remainder of the paper we drop the quotes around the second argument of predicates U and K_j.

inference rule. Note that Rule 8 in Figure 8, does provide an unrestricted form of inheritance.[33]

We note several points about the axioms which wise man #1 requires. (Refer to Figure 7.) Wise man #1 knows the following:

1. Wise man #2 knows (at every step) that wise man #3 uses the rule of *modus ponens*.
2. Wise man #2 uses the rules of *modus ponens* and *modus tolens*.
3. Wise man #2 knows (at every step) that if both my card and his card are black, then wise man #3 would know this fact at step 1.
4. Wise man #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.[34]
5. Wise man #2 knows (at every step) that if there's no utterance of W_3 at a given step, then wise man #3 did not know W_3 at the previous step. (Wise man #2 knows (at every step) that there will be an utterance of W_3 the step after wise man #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 wise man #2 will know this at the next step.[35]
8. If my card is black, then wise man #2 knows this (at every step).
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 wise man #2 knows his card is white.)

Note the following concerning the inference rules:

1. Rule 6 is a rule of introspection. Wise man #1 can introspect on what utterances have been made.[36]

[33]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.

[34]In other words, if wise man #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 wise man #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 [?, ?].

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

[36]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.

2. The rule for extended *modus ponens* allows an arbitrary number of variables.

3. Rule 7 is a rule of instantiation. If wise man #1 knows that wise man #2 knows α at *each* step then, in particular, wise man #1 will know at step $i+1$ that wise man #2 knew α at step i.

4. The rule of inheritance is quite general: *everything* is inherited from one step to the next.[37]

5.2 Solution

The solution to the problem is given in Figure 9. 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.

In step 1 all the initial axioms ($OBS_{W_3}(1)$) have been inferred through the use of Rule 1.[38] Nothing of interest is inferred in steps 2 through 4. In step 5, wise man #1 is able to negatively introspect and determine that no utterance of W_3 was made at step 3. Note the time delay: wise man #1 is able to prove *at step 5* that he did not know *at step 4* of an utterance made *at step 3*.[39] The remaining wffs shown in step 5 were all inferred through the use of Rule 7, the rule of instantiation. Wise man #1 needs to know that wise man #2 knows these particular facts at step 4. The reasoning continues from step to step. Note that at step 11, wise man #1 has been able to deduce that wise man #2 knows that if wise man #1's card is black, then his is white. From this step on, we essentially have the *Two-wise-men problem.* (See [**?**].) In step 17 wise man #1 is finally able to deduce that his card is white.

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

Several points of contrast can be drawn between this version and the two-wise-men version.

[37] For other commonsense reasoning problems, a far more restrictive version of inheritance is necessary.

[38] To save space we have not repeated them in the figure. See Figure 7 for the individual axioms.

[39] For a detailed description of this phenomenon, see [**?**].

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)	$K_2(s^4(0), (\forall i)(\forall x)(\forall y)$ $[K_3(i, x \rightarrow y) \rightarrow (K_3(i, x) \rightarrow K_3(s(i), y))])$	(R7,1a)
	(c)	$K_2(s^4(0), K_3(s(0), (B_1 \wedge B_2) \rightarrow W_3))$	(R7,1b)
	(d)	$K_2(s^4(0), (\forall i)[\neg U(s(i), W_3) \rightarrow \neg K_3(i, W_3)])$	(R7,1e)
6:	(a)	$\neg U(s^3(0), W_3)$	(R3,5a,1f)
	(b)	$K_2(s^5(0), K_3(s(0), B_1 \wedge B_2) \rightarrow K_3(s^2(0), W_3))$	(R3,5b,5c,1j)
7:	(a)	$K_2(s^4(0), \neg U(s^3(0), W_3))$	(R3,6a,1g)
	(b)	$K_2(s^6(0), (B_1 \wedge B_2) \rightarrow K_3(s(0), B_1 \wedge B_2))$	(R7,1c)
8:	(a)	$K_2(s^5(0), \neg K_3(s^2(0), W_3))$	(R3,7a,5d,1k)
	(b)	$K_2(s^7(0), \neg(B_1 \wedge B_2) \rightarrow (B_1 \rightarrow W_2))$	(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:		$K_2(s^8(0), B_1 \rightarrow W_2)$	(R3,10,8b,1i)
12:	(a)	$(K_2(s^8(0), B_1) \rightarrow K_2(s^9(0), W_2))$	(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 9: Solution to the Three-wise-men Problem

1. In the two-wise-men version, wise man #1 needs only to know about a *single* rule of inference used by wise man #2. In this version wise man #1 needs to know *several* rules used by wise man #2: *modus ponens*, extended *modus ponens*, and *modus tolens*. Because wise man #1 reasons within first-order logic, these three rules required the use of six axioms.

2. In the two-wise-men version, it is sufficient for wise man #1 to know that wise man #2 has certain beliefs *at step 1*. In the three-wise-men version, this is not sufficient—wise man #1 must know that wise man #2 *always* holds these beliefs.

3. What wise man #2 needs to know about wise man #3 is analogous to what wise man #1 needs to know about wise man #2 in the two-wise-men version. So, for instance, wise man #2 must know that wise man #3 uses the rule of *modus ponens* (and this is the only rule of wise man #3's about which wise man #2 must know). Also wise man #2 needs only to know that wise man #3 has certain beliefs *at step 1*.

Many formulations of the *Three-wise-men problem* have involved the use

of common knowledge or common belief (see [**?**] and [**?**] in particular). For instance, a possible axiom might be $C(W_1 \vee W_2 \vee W_3)$: it is common knowledge that at least one card is white. Adding the common knowledge concept here introduces unnecessary complications due, to a large degree, to the fact that the problem is modeled *from wise man #1's point of view*, rather than using a meta-language that describes the reasoning of all three (as [**?**, **?**] have both done). This 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 [**?**].

Our next section details how an active logic can deal successfully with contradictions.

æ

6 Reasoning in the face of contradictions

7 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. In Elgot-Drapkin's work, this is called 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 step-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 [**?**], but only in an ad hoc way. There a conjecture was formulated, to the effect that, under (unspecified) circumstances, a step-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- (or step-) 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 *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.

7.1 The lingering consequences and causes of contradictions

Early step-logics rely heavily on the rules OBS, MP, and INH (See rules 2, 3, and 7 of Figure 3 on page 27).[40]

Suppose we apply these rules to the observation function Obs_1:

$$Obs_1(j) = \begin{cases} P, P \to Q & \text{if } j = k \\ \neg P & \text{if } j = k + n \\ \emptyset & \text{otherwise} \end{cases}$$

for fixed $k, n > 0$. Notice what happens (see Figure 10): P and $P \to Q$ 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 $\neg P$.

⋮	⋮
Step k:	$\underline{P, P \to Q}$
Step k+1:	$P, P \to Q, \underline{Q}$
⋮	⋮
Step k+n:	$P, P \to Q, Q, \underline{\neg P}$
Step k+n+1:	$P \to Q, Q, \underline{Contra(\{P, \neg P\}, k+n)}$
⋮	⋮

Figure 10: A belief (Q) based on a questionable former belief (P) persists.

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:

$$Obs_2(j) = \begin{cases} Q, Q \to R, Q \to \neg R & \text{if } j = k \\ \emptyset & \text{otherwise} \end{cases}$$

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

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 11).[41]

⋮	⋮
Step k:	$\underline{Q, Q \rightarrow R, Q \rightarrow \neg R}$
Step k+1:	$Q, Q \rightarrow R, Q \rightarrow \neg R \underline{R, \neg R}$
Step k+2:	$Q, Q \rightarrow R, Q \rightarrow \neg R, \underline{Contra(\{R, \neg R\}, k+1), R, \neg R}$
Step k+3:	$Q, Q \rightarrow R, Q \rightarrow \neg R, Contra(\{R, \neg R\}, k+1), \underline{Contra(\{R, \neg R\}, k+2), R \neg R}$
⋮	⋮

Figure 11: 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 12, in which we again use Obs_2 and augment MP and INH with these new stipulations).[42]

A more comprehensive solution must take into account the way inference is chained over the course of steps in step-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

[41] 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.

[42] These new stipulations are nevertheless beneficial and will be used in the logic described shortly. (See Inf_{deriv}, Figure 13 on page 56.)

$\vdots$ $\vdots$

Step k: $\underline{Q, Q \rightarrow R, Q \rightarrow \neg R}$

Step k+1: $Q, Q \rightarrow R, Q \rightarrow \neg R \underline{R, \neg R}$

Step k+2: $Q, Q \rightarrow R, Q \rightarrow \neg R, \underline{Contra(\{R, \neg R\}, k+1)}$

Step k+3: $Q, Q \rightarrow R, Q \rightarrow \neg R, Contra(\{R, \neg R\}, k+1)$

Step k+4: $Q, Q \rightarrow R, Q \rightarrow \neg R, Contra(\{R, \neg R\}, k+1) \underline{R, \neg R}$

Step k+5: $Q, Q \rightarrow R, Q \rightarrow \neg R, Contra(\{R, \neg R\}, k+1), \underline{Contra(\{R, \neg R\}, k+3)}$

$\vdots$ $\vdots$

Figure 12: The contradiction (R, $\neg R$) will alternately arise and then be disinherited.

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.

7.1.1 *dc*-recovery: Some Preliminary Definitions

Let $SL(Inf, Obs)$ be an arbitrary step-logic with inference rules all of the form:[43]

$$\begin{array}{ll} \mathbf{k}: & \beta_1, \ldots, \beta_n \\ \hline \mathbf{k+1}: & \alpha \end{array}$$

Definition 7.1 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-

[43]The following definitions can be extended to apply to the more general rule schema discussed in [**?**].

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 7.2 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 7.2 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 7.3 *A step-logic $SL(Inf, Obs)$ dc-recovers* if $\exists j$ such that $\forall k > j$ $\neg\exists\alpha \vdash_{k+1} Distr(\alpha, k)$.

Definition 7.3 says this: a step-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 7.4 *A step-logic $SL(Inf, Obs)$* is *eventually free of direct contradictions* if $\exists j$ such that $\forall k > j$ and $\forall\alpha$, either $\nvdash_k \alpha$ or $\nvdash_k \neg\alpha$.

Lemma 7.5 If $SL(Inf, Obs)$ dc-recovers then $SL(Inf, Obs)$ is eventually free of direct contradictions.

Proof: If $SL(Inf, Obs)$ dc-recovers then $\exists j$ such that $\forall k > j$ no k-theorem is $k+1$ distrusted by definition 7.3. Thus $\forall k < j, \alpha$ either $\nvdash_k \neg\alpha$ $\nvdash_k \alpha$ by definition 7.2(i). Hence $SL(Inf, Obs)$ is eventually free of direct contradictions. ■

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

7.1.2 Extending Step-logic: Active-logic

In this section we will develop a new 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 13.

Rule 1:	$\mathbf{i}:$ _____ $\mathbf{i+1}:$ α		If $\alpha \in OBS(i+1)$
Rule 2:	$\mathbf{i}:$ $\alpha[S]$ $\mathbf{i+1}:$ $\alpha[S]$		Inheritance[a]
Rule 3:	$\mathbf{i}:$ $\alpha[S_1], \alpha \rightarrow \beta[S_2]$ $\mathbf{i+1}:$ $\beta[\{\alpha, \alpha \rightarrow \beta\} \cup S_1 \cup S_2]$		MP[b]
Rule 4:	$\mathbf{i}:$ $\alpha[S_1],\ \neg\alpha[S_2]$ $\mathbf{i+1}:$ $Distr(\alpha, i), Distr(\neg\alpha, i)$		Contradiction Distrusted
Rule 5:	$\mathbf{i}:$ $\alpha < S_1, \ldots, S_m >$, $Distr(\beta_1, i-1), \ldots, Distr(\beta_n, i-1)$ $\mathbf{i+1}:$ $Distr(\alpha, i)$		Distrust Consequences[c]
Rule 6:	$\mathbf{i}:$ $\alpha[S_1], \neg\alpha[S_2], \beta[S_3]$ $\mathbf{i+1}:$ $Distr(\beta, i)$	$\beta \in S_1$ 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 13: Inf_{deriv}

In Figure 13 the following abbreviations are used:

(1) α abbreviates $\alpha[\emptyset]$; i.e., we simply write α when α's derivation is the empty set.[44]

[44] (Since the limitations we will place on *Obs* (see the statement of the *dc*-recovery theorem, section 7.2) makes the derivation of any theorem of the form $Distr(\alpha, i)$ irrelevant, we annotate wffs of the form $Distr(\alpha, i)$ with $[\emptyset]$.)

(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.

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 14. Derivations are in **bold** type. Notice the two instances of Q at step 2 each with a dis-

Step 1: $\underline{P, P \rightarrow Q, R, R \rightarrow Q}$

Step 2: $P, P \rightarrow Q, R, R \rightarrow Q,$
$\underline{\neg P, Q[\{\mathbf{P}, \mathbf{P} \rightarrow \mathbf{Q}\}], Q[\{\mathbf{R}, \mathbf{R} \rightarrow \mathbf{Q}\}]}$

Step 3: $P \rightarrow Q, R, R \rightarrow Q,$
$Q[\{\mathbf{P}, \mathbf{P} \rightarrow \mathbf{Q}\}], Q[\{\mathbf{R}, \mathbf{R} \rightarrow \mathbf{Q}\}]$
$\underline{Distr(P, 2), Distr(\neg P, 2)}$

Step 4: $P \rightarrow Q, R, R \rightarrow Q,$
$Q[\{\mathbf{R}, \mathbf{R} \rightarrow \mathbf{Q}\}], Distr(P, 2), Distr(\neg P, 2)$

$\vdots$ $\vdots$

Figure 14: Inf_{deriv} at work.

tinct 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 $i > 4$).

7.2 The *dc*-recovery Theorem

$SL(Inf_{deriv}, Obs)$ is the first non-trivial active-logic that we have developed that has the *dc*-recovery property given that *Obs* satisfies certain reasonable constraints. The following definitions describe those constraints.

Definition 7.6 An observation function *Obs* is *finite* if $\exists i$ such that $\forall j > i$, $Obs(j) = \emptyset$.

Definition 7.7 A wff α is *P-free* if α does not contain the predicate symbol P.

Definition 7.8 An observation function *Obs* is *P-free* if $\forall i \alpha$ if $\alpha \in Obs(i)$ then α is P-free.

Theorem 7.9 (*dc*-**Recovery Theorem for** $SL(Inf_{deriv}, Obs)$) Let *Obs* be finite and *Distr*-free then $SL(Inf_{deriv}, Obs)$ *dc*-recovers. (The proof can be found in [**?**].)

7.3 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 $SL(Inf_{deriv}, Obs)$, 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 $\text{THM}_{Obs} = \emptyset$. On the other hand it is not always the case that $\text{THM}_{Obs} = \emptyset$ as illustrated in Figure 14.

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.

æ

8 Language change

9 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?"
"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 [**?**], followed by others [**?**], [**?**], 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 [**?**].

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.

Here we do not offer a new logic per se; rather we borrow an existing one (step-logic [**?**], [**?**]) and apply it 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,[45] 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

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

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.[46] 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.[47]

9.1 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 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

[46]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.

[47]See [?] and [?] for more complete details.

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.[48] 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.[49]

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,[50] 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

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

[48] We assume that beliefs are symbolically represented inside the head in some mental language. [?]

[49] The term *compression* is borrowed from [?].

[50] 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.

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.[51] Instead, a simplifying assumption is to postulate a *tutor* or an advisor that can tell us about our errors.[52] 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.

9.2 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.

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.[53] Perhaps she is talking about a different John. To

[51] 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).

[52] See [?] for a discussion about programs and advice taking.

[53] 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

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:

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

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:

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

another version, in which Sally uses the name "John" at the outset.

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

Figure 15 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.,

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

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.[54]

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.

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

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

Step 2: $\ldots, \underline{BL(j), M(j)}$
(**Sally:** "...his leg is broken and his wife...")

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

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

Step 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.")

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

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

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

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

Figure 15: Sketch of stepped-reasoning in the *Two Johns* story.

9.3 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 [**?**] and [**?**] 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.

RS:

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

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 15. (Others are treated fully in [**?**]).[55]

"**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: $\dfrac{\mathbf{i}:}{\mathbf{i+1}:\quad \alpha}$ 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 15).

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

In figure 15 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:

[55] Among those not discussed here are rules for inheritance (of beliefs from one step to the next), modus ponens, contradiction handling and other belief disinheritance, and negative introspection.

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 15 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.

æ

10 Focal Points

11 Active Logic algorithm for 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.

[**?**, **?**] 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.

11.1 Focal Points

Originally introduced by Schelling [**?**, **?**], 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 [**?**, **?**].) 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 agreement 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, 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 fails to provide 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.

11.2 The Active-Logic Focal Point Algorithm

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.

That is, 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.

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. We denote the agent's predicate's set at step i by $Pred_i$ and its set of term constants at step i by $Term_i$.

We present here one of the rules for focal point identification.

Uniqueness:

An object may be a focal point if it is the only object with a given property. Formally, if in $i-1$ 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. Similar rules are given for *Uniqueness Complement*, *Centrality* and *Extreme*.

These rules 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. For example,

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

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.

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.[56]

To summarize, the Focal Point algorithm that was developed allows for the uncovering of focal points through the use of active-logic, special inference rules, and sets of predicates, functions, and terms that change over time. The technique is particularly well-suited for modeling the time-dependent nature of focal point search.

((More details are needed in this section.)) æ

[56]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.

12 Deadline planning

13 Fully deadline-coupled planning

In planning situations involving tight deadlines a commonsense reasoner may spend substantial amount of the available time in reasoning toward and about the (partial) plan. This reasoning involves, but is not limited to, partial plan formulation, making decisions about available and conceivable alternatives, plan sequencing, and also plan failure and revision. The key observation is that the *time taken in reasoning about a plan brings the deadline closer.* The reasoner should therefore take account of the passage of time during that *same* reasoning, and this accounting must continuously affect every decision under time-pressure. Step-logics were introduced as a mechanism for reasoning situated in time. In this section we give a brief overview of work in fully deadline-coupled planning [**?**, **?**, **?**] 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.

To elaborate on the fully deadline-coupled planning problem, we present an illustrative domain, which we call the *Nell & Dudley Scenario*[57]: 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.

Our approach has many concerns in common with existing research in planning and temporal reasoning. [**?**, **?**, **?**, **?**, **?**]. However, these works do not account for the time taken for meta-planning. Indeed, this is stated in [**?**] (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".

Dean [**?**] proposed a computational approach to reasoning about events and their effects occurring over time. Dean, Firby and Miller [**?**] subsequently designed FORBIN, a planning architecture that supports hierarchical planning involving reasoning *about* deadlines, travel time, and resources. Dean and Boddy [**?**] formulated an algorithmic approach to the solution of time-dependent planning problems by introducing "anytime algorithms"

[57] This problem was first mentioned in the context of time-dependent reasoning by McDermott [**?**], and more recently discussed in [**?**].

which capture the notion that utility is a monotonic function of deliberation time. Here also, the time for computation is not accounted for : "The time required for deliberation scheduling will not be factored into the overall time allowed for deliberation. For the techniques we are concerned with, we will demonstrate that deliberation scheduling is simple, and, hence, if the number of predicted events is relatively small, the time required for deliberation can be considered negligible," [?] (page 50). [?] demonstrated deliberation scheduling for a time-dependent planning problem involving tour and path planning for a mobile robot.

[?] formulated a world model based on temporal logic which allows the problem solver to gather constraints on the ordering of actions without having to commit to it when a conflict is detected. [?] discusses how a planner can reason about the difficulty of its tasks, and depending on available time, produce reasonable if not optimal solutions. [?] and [?] use a first-order temporal logic model to describe complex synchronization properties of parallel multiagent domains. [?] explore the relation between agent design and environmental factors. They use plans to constrain the reasoning. In [?] a computational approach to temporal reasoning is presented in which a problem solver is forced to make predictions and projections about the future and plan in the face of uncertainty and incomplete knowledge; time-maps are described here. [?] examine the complexity of temporal reasoning problems involving events whose order is not completely known. We refer the reader to [?] for a general survey of related work on planning.

However, a *fully* deadline-coupled planner has an important qualification that these efforts fail to meet: in addition to determining the current time, estimating the expected execution time of partially completed plans and being able to discard alternatives that are deadline-infeasible, it must also have 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! Step-logics are used here as a way to do this without the vicious circle of "meta-meta-meta..." hierarchies.

ADD A SUMMARY OF THE PLANNING AND TEMPORAL REASONING FORMALISM HERE (ABOUT 1.5 PAGES)

14 Memory issues

15 Limited resources: time, space and computation bounded reasoning

An agent under severe time-pressure may spend 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.

15.1 Shortcomings

The active logics formalism described thus far, and as applied to the planning domain in the earlier section 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 either to the development of the agent's current thread of reasoning. Step-logics and our treatment of deadline-coupled planning in the past have 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. Step-logics 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 paper we propose an extension of the original step-logic formalism to take into consideration space and computation constraints. We revisit the fully deadline-coupled planning problem in the light of this new framework.

15.2 A limited span of attention

We propose a solution to the space problem partially based on [**?**] 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[58].

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 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[59] 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[60]. The $focus$ and Now which are crucial to time-situated

[58] 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 [**?**].

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

[60] This has the feature that all thinking does not pass through the STM unless it is

reasoning are always accesible 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. ((cite here some work on scope)) 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 some details of this mechanism in section ???. 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 [**?**] for some of the specific heuristic strategies employed for the tightly time-constrained planner.

15.3 A limited think capacity

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 [61].

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

relevant to the focus.

[61] 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.

the inference rules either completely nondeterministically up to the engine capacity I, 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.

((More details are needed in this section.))

æ

16 Conclusions and Future Work

17 Conclusions and Future Work

To be sure, active logics studied to date do not do all that one would like. Their biggest shortcoming is that they indulge in litterbugging (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 as we shall argue, 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 15; but it is a hard problem to come up with plausible candidates. Thus litterbugging is not forced on us by active logics; it is our understanding of key issues such as relevance, focus of attention, and memory management that is the sticking point. Our belief is that as more is learned about these issues, it will be possible to incorporate them into the active logic framework.

æ