A logic for characterizing multiple bounded agents*

John Grant
Dept. of Computer and Information Sciences
and Department of Mathematics
Towson Univ.
Towson, MD 21252 USA
jgrant@towson.edu

Sarit Kraus
Dept. of Mathematics and Computer Science
Bar-Ilan University,
Ramat-Gan, 52900 Israel
and Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742
sarit@umiacs.umd.edu

Donald Perlis
Dept. of Computer Science
Univ. of Maryland, College Park 20742, USA
Phone: +1 (301) 405-2685
e-mail: perlis@cs.umd.edu

Abstract

We describe a meta-logic for characterizing the evolving internal reasoning of various families of agents. We view the reasoning of agents as ongoing processes rather than as fixed sets of conclusions. Our approach utilizes a strongly sorted calculus, distinguishing the application language, time, and various syntactic sorts. We have established soundness and completeness results corresponding to various families of agents. This allows for useful and intuitively natural characterizations of such agents' reasoning abilities. We discuss and contrast consistency issues as in the work of Montague and Thomason. We also show how to represent the concept of focus of attention in this framework.

*This material is based upon work supported by the National Science Foundation under Grant No. IIS-9907482. We wish to thank the referees for their valuable comments and suggestions.

Keywords: logics of knowledge and beliefs, bounded agents, real-time reasoning, multiple agents.

1 Introduction

There is an increasing sophistication of robots and other automated agents, as well as an increasing reliance on interacting groups of agents. These developments call for formal methods to analyze, and even predict, behaviors of such agents and groups of agents. This paper is intended as a step in that direction.

We develop a framework for characterizing the evolving internal reasoning and cooperative activities of various families of agents. This framework is formally described as a meta-logic. We view the reasoning of agents as ongoing processes rather than as fixed sets of conclusions; to facilitate this, time is represented explicitly. As an illustration of our approach, in Section 1.1 we give an example about a situation for preparing to make a phone call involving three agents.

Section 2 describes our framework. Section 2.1 introduces three languages: the language of the subject matter, the language of the agents, and the language of the meta-logic in which we express general statements about the reasoning of the agents. Section 2.2 contains the axioms and axiom schemas of the meta-theory we use to characterize the behavior of agents. Section 2.3 provides a construction for a model of the meta-theory. In fact, this model is a minimal model, and we give two alternative characterizations for it. The existence of a model means that the meta-theory is consistent and we discuss this issue.

In Section 3 we relax the restriction that the agents can apply all the rules in one unit of time. We discuss several families of agents: the agents for which there is a fixed bound on the amount of time it takes them to apply a rule, and the agents which will eventually apply a rule. We show how the axioms of the meta-theory can be rewritten to account for these properties of the agents. As in Section 2, we construct a model for the axioms and extend the alternative characterizations for it.

Section 4 deals with incorporating the concept of Focus of attention into our framework. The idea is that at any given time only some of the agent's beliefs are in the Focus and the agent can use only what is in the Focus for its reasoning and actions. New knowledge is brought into the Focus but stays there only for a fixed amount of time. We incorporate this concept both into the framework of Section 2 and then sketch two different ways of handling the families of agents discussed in Section 3.

Section 5 surveys artificial intelligence research on logical axiomatizations for rational agents. It may be useful for the reader to at least skim through this section first. Section 6 provides the conclusion and future work. The details of the telephoning example and some of the axioms are given in the Appendix.

1.1 Example: preparing to make a phone call

The major example we use to illustrate the use of our framework is a situation for preparing to make a phone call. This example can also be viewed as a deadline problem although we don't treat it that way here. Suppose that an agent B is in Israel and wants to phone Jack. B knows Jack's local phone number, but not Jack's country code and region (area) code. B also knows that there is an agent C that knows country codes and an agent A that knows area codes. Hence B asks agent C for Jack's country code and agent A for Jack's area code. Agent C knows the country codes of various people and has general knowledge about country codes. In particular, agent C knows that Jack and Jill live in the same country, that Jill lives in the USA, and that the USA country code is 1.

Let $CC(u_1, u_2)$ mean that the country code of individual u_1 is u_2; $Co(u_1, u_3)$ means that u_1 lives in country u_3; and $SC(u_1, u_2)$ means that u_1 and u_2 live in the same country. We can formalize the relevant axioms of agent C as follows:

$$\alpha_1 : \forall u_1 u_2 u_3 (SC(u_2, u_1) \wedge Co(u_1, u_3) \rightarrow Co(u_2, u_3))$$
$$\alpha_2 : \forall u_1 (Co(u_1, usa) \rightarrow CC(u_1, 1))$$
$$\alpha_3 : Co(jill, usa) \qquad\qquad \alpha_4 : SC(jack, jill).$$

It takes one use of a conjunction rule and two sequential uses of (versions of) modus ponens applied to these axioms to derive $CC(jack, 1)$ and this should be taken into consideration when reasoning about the time in which B will be able to make the phone call. Agent A's axioms are a default version of agent C's axioms. Agent A knows that Jack and James live in the same neighborhood, that James lives in Bethesda, and that typically people living in the same neighborhood have the same area code. The relevant axioms of agent A can be formalized as follows: Here N stands for neighborhood, SN for same neighborhood, AC for area code, and bth for Bethesda:

$$\beta_1 : \forall u_1 u_2 u_3 (SN(u_2, u_1) \wedge N(u_1, u_3) \rightarrow N(u_2, u_3))$$
$$\beta_2 : \forall u_1 (N(u_1, bth) \hookrightarrow AC(u_1, 301))$$
$$\beta_3 : N(james, bth) \qquad\qquad \beta_4 : SN(jack, james).$$

Here β_2 expresses the belief via the default implication $\hookrightarrow$ that 301 is the typical (default) area code in Bethesda.[1] The same reasoning steps with $\hookrightarrow$ substituted for $\rightarrow$ derive $AC(jack, 301)$. Once agents C and A figure out the country and area codes, they tell their results to agent B who then makes the phone call. Because of the explicit representation of time, we will know when B will be able to place the call.

2 A sorted logic for agents

Our meta-logic will describe the reasoning used by commonsense agents. Thus we must discuss both the agents' logics and the meta-logic. Consider the telephone example again. An agent may at some time, say $t = 12$, have among its beliefs the wffs $Co(jill, usa)$, i.e., at time 12 the agent knows Jill lives in the USA. It also may have among its beliefs at time 12 that it already had that belief that Jill lives in the USA at time 10. Thus the agent logic is concerned with both the application domain (telephoning) and the agent's own knowledge[2] and the knowledge of other agents.

[1] An axiomatization for $\hookrightarrow$ is given in Section 2.2.

[2] We use the notion of knowledge in its weak colloquial sense with no assumption of correctness of belief. We do not distinguish between knowledge and belief and use the K predicate for that purpose.

The meta-logic is also concerned with the agents' knowledge and reasoning. For instance, we want to be able to express general ideas about agents, such as "if agent G believes α and $\alpha \rightarrow \beta$ at time t, then at time t+1 G believes β." This requires expressing connectives between unspecified wffs α and β and is best done with special function symbols.

2.1 The basic syntax

In this section we describe three languages. The first, L_{Su}, is the language of the subject matter and contains the symbols needed for writing statements about the application domain. For the telephoning example, for instance, there would be the predicate symbols Co, SC, and CC as well as the constant symbols 1, $jill$, $jack$, and usa; and the formula $Co(jack, usa)$ would be in L_{Su}. The second, L_{Ag}, is the language of the agents. It includes, in addition to the symbols of L_{Su}, predicate and constant symbols allowing the agents to reason about time, their own beliefs and the other agents' beliefs. So the formula $K(A, 5, \lceil Co(jill, usa) \rceil)$ would be in L_{Ag}. The notation $\lceil Co(jill, usa) \rceil$ is used as a constant symbol to represent the formula $Co(jill, usa)$. The third, L_{Me}, is the language of the meta-logic; it allows us to express general statements about the reasoning of the agents. The formula $\forall zx K(z, x, now(x))$ expressing the Clock axiom, to be discussed later, is an example of a formula in L_{Me}.

L_{Su} is an unsorted language.[3] L_{Ag} and L_{Me} are sorted languages. All the languages are first order languages with equality and have the logical symbols $\forall, \exists, \neg, \wedge, \rightarrow$. L_{Su} also contains the symbol $\hookrightarrow$ for default implication.

Definition 1 L_{Su} is the language of the agents' application and depends on the subject matter; it does not involve time, and includes the following symbols:
variables: $u_1, u_2, \cdots$; predicate, constant and function symbols as needed for the application.
$\mathcal{F}_{Su}$ is the set of formulas of L_{Su} obtained by using the standard syntax of first-order logic.

Formulas such as $Co(jill, usa)$ and $\forall u_1(Co(u_1, usa) \rightarrow CC(u_1, 1))$ are in L_{Su}.

Definition 2 L_{Ag} is the language of the agents and contains 4 sorts:
S_a: the sort representing the application domain,
S_n: the sort representing the names of agents,
S_t: the sort representing time,
S_f: the sort representing formulas.

In order to clearly distinguish variables of different sorts of L_{Ag} we use different letters for variables of different sorts:
$u_1, u_2, \cdots$ for variables of sort S_a,
$z_1, z_2, \cdots$ for variables of sort S_n,
$x_1, x_2, \cdots$ for variables of sort S_t,
$y_1, y_2, \cdots$ for variables of sort S_f.

Additional symbols are identified by their sorts. For example, the predicate symbol K, to be given below, is written as $K(S_n, S_t, S_f)$. This means that K is a ternary predicate symbol

[3] We do this for simplicity but L_{Su} could be a sorted language also.

with first argument in S_n, second argument in S_t, and third argument in S_f. Below we specify the symbols corresponding to various sorts. The sort S_a represents the application domain and includes constant, function and predicate symbols of the application. The sort S_n represents names of agents and includes a constant symbol for the name of each agent. The sort S_t represents time and includes the following symbols: (a) the "now" unary predicate symbol, Now (S_t); (b) the "before" binary predicate symbol, $< \ (S_t, S_t)$; (c) the constant symbol, 0, which is the starting time; and (d) the successor unary function symbol, $+ : S_t \rightarrow S_t$. For any ℓ, we will denote $x\overbrace{+\cdots+}^{\ell}$ by $x+^{\ell}$. We write $0+^{\ell}$ as ℓ. The sort S_f represents formulas used as parameters to the knowledge predicates. It includes a set of formulas $\mathcal{F}_{Ag}$ and a corresponding set of constant symbols C_{Ag} which are defined recursively, simultaneously. For every formula α placed in $\mathcal{F}_{Ag}$, a constant symbol $\lceil\alpha\rceil$ of sort S_f is created and added to C_{Ag}. $\mathcal{F}_{Ag}$ is the smallest set that contains $\mathcal{F}_{Su}$ and is closed under the application of the constant symbol 0, the function symbol $+$, the predicate symbols Now, $<$, K, A, OBS, $Need$, Ask, and $Tell$, and the usual logical symbols, all appropriately sorted. The predicate symbols, in addition to $Now(S_t)$ and $< (S_t, S_t)$ are knowledge: $K(S_n, S_t, S_f)$; awareness: $A(S_n, S_t, S_f)$; observation: $OBS(S_n, S_t, S_f)$; need to know: $Need(S_n, S_t, S_f)$; ask: $Ask(S_n, S_n, S_t, S_f)$; and tell: $Tell(S_n, S_n, S_t, S_f)$.

For example: $OBS(6, 8, \lceil Co(jill, usa)\rceil)$ means intuitively that agent 6 observes at time 8 that Jill's country is the USA; $K(7, 10, \lceil C(jill, usa)\rceil)$ means intuitively that agent 7 at time 10 knows that the country of Jill is the USA; $Tell(10, 11, 8, \lceil C(jill, usa)\rceil)$ means intuitively that agent 10 tells agent 11 at time 8 that the country of Jill is the USA; $Need(3, 5, \lceil Co(jill, u_1)\rceil)$ means intuitively that agent 3 at time 5 needs to know Jill's country; and $K(6, 8, \lceil Ask(5, 6, 3, \lceil C(jill, u)\rceil)\rceil)$ means intuitively that agent 6 knows at time 8 that agent 5 asked agent 6 at time 3 what Jill's country is.

We will now define the logic which the designers of the agents may use in reasoning about the behavior the agents.

Definition 3 L_{Me} is the meta-language of the agents and contains five sorts: S_n, S_t, S_f, S_v, and S_g. S_n, S_t, and S_f are defined as in the definition of L_{Ag}. The other sorts are used as follows: S_v is the sort representing variable sequences; S_g is the sort representing ground term sequences. L_{Me} contains various symbols, in addition to the symbols of L_{Ag}, that allow us to deal with parts of the formulas of the agents and to express the reasoning of the agents. Additional variable symbols are used for the new sorts: $v_1, v_2, \cdots$ for variables of sort S_v; and $w_1, w_2, \cdots$ for variables of sort S_g. The functions applied to the sort S_f are: one unary function symbol, *neg*: $S_f \rightarrow S_f$, for the negation of the formula; three binary function symbols: *con*:$(S_f, S_f) \rightarrow S_f$, for the conjunction of formulas; *imp*: $(S_f, S_f) \rightarrow S_f$, for the implication of formulas; and $defimp$: $(S_f, S_f) \rightarrow S_f$, for the default implication of formulas.

The function and predicate symbols applied to both S_t and S_f are: one unary function symbol *now*: $S_t \rightarrow S_f$ which constructs a name for a Now atom; and a ternary function symbol *k*: $(S_n, S_t, S_f) \rightarrow S_f$ which constructs a name for a K atom.

Sort S_v includes a constant symbol $\lceil \bar{v} \rceil$ for every sequence of variables $\bar{v}$ of sort S_n, S_a, S_t, and S_f. Sort S_g includes a constant symbol $\lceil \bar{g} \rceil$ for every sequence of ground terms $\bar{g}$ of sort S_n, S_a, S_t, and S_f.

Additional function symbols of L_{Me} are: a ternary function symbol *sub*: $(S_f, S_v, S_g) \rightarrow S_f$ which substitutes ground terms for variables in a formula; a binary function symbol $forall$: $(S_f, S_v) \rightarrow S_f$ which universally quantifies a formula; a binary function symbol $exists$: $(S_f, S_v) \rightarrow S_f$ which existentially quantifies a formula and a binary function symbol $mgci$: $(S_f, S_f) \rightarrow S_f$ which constructs the most general common instance of two formulas (using the subscript ordering of variables).

For example, $\forall zxK(z, x, now(x))$ is in L_{Me}. Its intuitive meaning is that every agent at every time knows what the time is now. For another example, $\forall zxy_1y_2(K(z, x, y_1) \wedge K(z, x, imp(y_1, y_2)) \rightarrow K(z, x+, y_2))$ is in L_{Me}. Its intuitive meaning is that it takes an agent one step (time unit) to apply modus ponens. As we will show in the next section, if y_1 and y_2 are interpreted as the names of the formulas α and β in L_{Ag} respectively, i.e., $\lceil \alpha \rceil$ and $\lceil \beta \rceil$, then $imp(y_1, y_2)$ is interpreted as $\lceil \alpha \rightarrow \beta \rceil$, which is the name of the formula $\alpha \rightarrow \beta$ in L_{Ag}. Similarly, if y_1 and y_2 are interpreted as the names of the formulas $CC(u_1, 1)$ and $CC(jack, u_2)$ then $mgci(y_1, y_2)$ is interpreted as the name of the formula $CC(jack, 1)$.

2.2 Axioms of L_{Me}

Next we provide the theory $\mathcal{T}$ over L_{Me}, that is, the axioms and axiom schemas that we will use in this paper. Actually, our definition provides a class of theories, $\mathcal{T}_{Me}$, one for each L_{Me}, depending on L_{Su}. We assume a certain type of behavior for the agents that is reflected in our axioms. We include some of the axioms in the body of the text and some in the Appendix. We explain each axiom briefly.

Axioms *A*1-*A*18 are given in the Appendix. These axioms define the time axioms, give the inequality of different symbols, and make the formula forming functions behave in the proper way to form the correct symbols.

The additional axioms presented below describe the reasoning of a given set of agents associated with the theory $\mathcal{T}$. $\mathcal{T}$ may include different axioms based on different behaviors by different agents. We have chosen inference rules for the reasoning of agents that are useful for our examples. Performing each inference rule takes one unit of time. In Section 3, where we deal with families of agents, we relax this restriction.

AR1. Clock:

$$\forall zxK(z, x, now(x))$$

Every agent always knows the time (through some clock mechanism).

AR2. Observation:

$$\forall zxy(OBS(z, x, y) \rightarrow K(z, x+, y))$$

Every agent will know in one unit of time what it observes.

AR3. Modus Ponens and Conjunctions: The first three rules specify that the agents use modus ponens, conjunction, and detachment. The fourth rule is a modus ponens like rule for default implication. In all the axioms reaching a conclusion takes one time period. Note, that if the agent knows that $a \wedge b \rightarrow c$ and it knows a and it knows b, it will take it two time periods to know c. First, using the conjunction rule it will conclude $a \wedge b$ and only in the next time period, using modus ponens, it will conclude c.

AR3a. Modus Ponens:

$$\forall z x y_1 y_2 (K(z, x, y_1) \wedge K(z, x, imp(y_1, y_2)) \ \rightarrow \ K(z, x+, y_2))$$

AR3b. Conjunction:

$$\forall z x y_1 y_2 (K(z, x, y_1) \wedge K(z, x, y_2) \ \rightarrow \ K(z, x+, con(y_1, y_2)))$$

AR3c. Detachment:

$$\forall z x y_1 y_2 (K(z, x, con(y_1, y_2)) \ \rightarrow \ K(z, x+, y_1) \wedge K(z, x+, y_2))$$

AR3d. Default Modus Ponens:

$$\begin{aligned} \forall z x y_1 y_2 ((K(z, x, y_1) \wedge K(z, x, defimp(y_1, y_2)) \wedge \neg K(z, x, neg(y_2)) \wedge \\ \neg \exists y_3 (K(z, x, y_3) \wedge y_2 = neg(y_3))) \ \rightarrow \ K(z, x+, y_2)) \end{aligned}$$

Intuitively, this axioms means that if agent G believes at a given time i that if a is true, then usually b is true (i.e., $K(G, i, \ulcorner a \hookrightarrow b \urcorner)$, and if it knows a and it doesn't know that b is false (i.e., $\neg K(G, i, \ulcorner \neg b \urcorner)$ then it will conclude b in the next time period. We do not apply this rule if b is of the form $\neg c$, because the agent might believe c but not $\neg\neg c$.

AR4. Universal Rules: The following three rules specify that the agents use universal modus ponens (for implications and default implications) and universal instantiation

AR4a. Universal Modus Ponens:

$$\begin{aligned} \forall z x y_1 y_2 y_3 v_1 w_1 (K(z, x, sub(y_1, v_1, w_1)) \wedge K(z, x, forall(imp(y_1, y_2), v_1)) \\ \wedge \ y_3 = sub(y_2, v_1, w_1) \ \rightarrow \ K(z, x+, y_3)) \end{aligned}$$

An example of an instantiation of this axiom in the country code application is as follows. Suppose that at a given time t agent G knows that Jill lives in the USA (i.e., $K(G, t, \ulcorner Co(jill, usa) \urcorner)$, and that the country code for someone who lives in the USA is 1, (i.e., $K(G, t, \ulcorner \forall u_1 (Co(u_1, usa) \rightarrow CC(u_1, 1)) \urcorner))$. Then, by substituting u_1 with $jill$ it will conclude in the next time period that Jill's country code is 1 (i.e., $K(G, t+, \ulcorner CC(jill, 1) \urcorner)$.)

AR4b. Universal Instantiation

$$\forall zxy_1y_2vw(K(z,x,forall(y_1,v)) \wedge y_2 = sub(y_1,v,w) \;\rightarrow\; K(z,x+,y_2))$$
(where w is elementwise sort-compatible with v)

Consider the above example. If agent G knows at time period t that the country code of someone who lives in the USA is 1, i.e., $K(G,t,\ulcorner \forall u_1(Co(u_1,usa) \rightarrow CC(u_1,1))\urcorner)$, then by substituting u_1 with $jill$, he will know that if Jill lives in the USA, then her country code is 1, i.e., $K(G,t+,\ulcorner Co(jill,usa) \rightarrow CC(jill,1))\urcorner)$. Note that an agent whose inference rules are specified without the universal modus ponens ($AR4a$) will reach similar conclusions to that of an agent whose specification includes the universal modus ponens, but it will take it additional time period.

AR4c. Universal Default Modus Ponens:

$$\begin{aligned}\forall zxy_1y_2y_3v_1w_1(K(z,x,sub(y_1,v_1,w_1)) \wedge K(z,x,forall(defimp(y_1,y_2),v_1))\\ \wedge y_3 = sub(y_2,v_1,w_1) \wedge \neg K(z,x,neg(y_3)) \wedge \neg\exists y_4(K(z,x,y_4) \wedge\\ y_3 = neg(y_4))) \;\rightarrow\; K(z,x+,y_3))\end{aligned}$$

AR5. Introspection: We begin by giving axioms for the A(ware) predicate. An agent will be aware of every subformula of a known formula. $AR5a$-$AR5f$ are in the Appendix. Awareness is needed for negative introspection stating that an agent aware of but not knowing a formula at time t will know at the next time period that it did not know the formula at time t. This is useful in the solution of various problems such as the three wise men problem [16].

AR5g. Positive introspection: Positive introspection means that an agent that knows a formula at time t will know at the next time period that it knew the formula at time t.

$$\forall zxy(K(z,x,y) \;\rightarrow\; K(z,x+,k(z,x,y)))$$

AR5h. Negative introspection:

$$\forall zxy(A(z,x,y) \wedge \neg K(z,x,y) \;\rightarrow\; K(z,x+,neg(k(z,x,y))))$$

For example, if agent G is aware of a at time period t (i.e., $A(G,t,\ulcorner a\urcorner)$), but it does not know a at t (i.e., $\neg K(G,t,\ulcorner a\urcorner)$), then in the next time period it will know that it didn't know a at t (i.e., $K(G,t+,\ulcorner \neg K(G,t,\ulcorner a\urcorner)\urcorner)$.)

AR6. Inheritance: The following two axioms are for inheritance. These formulas assure that formulas which do not present a contradiction are inherited at the next time period. The first axiom considers the inheritance of positive formulas while the second axiom considers negated formulas.

AR6a.

$$\forall z x_1 y_1 (K(z,x_1,y_1) \wedge \neg K(z,x_1,neg(y_1)) \wedge \forall x_2 (y_1 \neq now(x_2)) \wedge \forall y_2 (y_1 \neq neg(y_2)) \;\rightarrow\; K(z,x_1+,y_1))$$

For example, if the agent knows a at time period t and it does not know $\neg a$, then it will continue to know a at the next time period (given that a is not equal to $now(t)$ or starts with a negation - the latter case is *AR6b*.)

AR6b.

$$\forall z x_1 y_1 (K(z,x_1,y_1) \wedge \neg K(z,x_1,neg(y_1)) \wedge \exists y_2 (y_1 = neg(y_2) \wedge \neg K(z,x_1,y_2)) \;\rightarrow\; K(z,x_1+,y_1))$$

For example, if the agent knows $\neg a$ at time period t and it does not know a at time period t, then it will continue to know $\neg a$ at $t+1$.

AR7. Truthfulness: The following axiom states that the agents are trustworthy; if an agent tells another agent a formula, the second agent adds the formula to its knowledge base.

$$\forall z_1 z_2 x y (Tell(z_1,z_2,x,y) \;\rightarrow\; K(z_2,x+,y))$$

AR8. Helpfulness: This axiom states that agents are helpful: if an agent asks another agent a question, at the first time period when the second agent knows the answer it will tell that answer to the first agent.

$$\forall z_1 z_2 x_1 x_2 y_1 y_2 v_1 w_1 (K(z_2,x_1,y_2) \wedge y_2 = sub(y_1,v_1,w_1) \wedge \exists x_3 (x_3 < x_1 \wedge Ask(z_1,z_2,x_3,y_1) \wedge \neg \exists x_4 (x_3 < x_4 \wedge x_4 < x_2 \wedge Tell(z_2,z_1,x_4,y_2)) \;\rightarrow\; Tell(z_2,z_1,x_2+,y_2)))$$

In the country code example, suppose agent B asks agent C for Jack's country code at time period 5, i.e., $Ask(B,C,5,\lceil CC(jack,u_2)\rceil)$ and suppose that at time period 7 C concluded that Jack's country code is 1, i.e., $K(C,7,\lceil CC(jack,1)\rceil)$, then it will tell this to B in the next time period, i.e., $Tell(C,B,8,\lceil CC(jack,1)\rceil)$ (assuming it didn't tell it in periods 6 or 7.)

Another group of axioms *AR9* involve the *Need* predicate and are in the Appendix. In addition to these general axioms there may be special axioms about the agents, such as axioms that deal with the capabilities of certain agents. An example will illustrate this later. We assume that all special axioms have the form of the *AR* axioms, i.e. $\forall \bar{t_1}(\phi(\bar{t_1},\bar{t_2}) \rightarrow Pred(\bar{t_3}))$ where $Pred \in \{K, Need, Ask, Tell\}$, $\bar{t_1}$, $\bar{t_2}$, and $\bar{t_3}$ are variable sequences of various sorts, $\bar{t_3} \subseteq \bar{t_1}$, ϕ is a formula, and every occurrence of a time variable in ϕ must be less than the occurrence of the time variable in *Pred*.

For each observation by an agent we add an axiom to $\mathcal{T}$ stipulating the fact of that observation. We assume that an agent can have only finitely many observations in a given time period. So, for *AO*1 below, we assume that for each n and each i there are finitely many j's specifying the agent's observations in a given scenario.

AO1. Observation:

$$OBS(n, i, \lceil \alpha^j_{n,i} \rceil) \text{ for each } n,\ i, \text{ and } j.$$

Let's go back now to the application involving country codes presented in the previous section. The country-code domain axioms α_k must be introduced as observations of agent G. So there must be t_k for $k = 1, 2, 3$, and 4, such that we have $OBS(G, t_k, \lceil \alpha_k \rceil)$ as axioms. Suppose that all $t_k = 0$, i.e., agent G believes the α_k at time 0. Note that all α_k can be inherited indefinitely by $AR6$. Then, by $AO1$, $\mathcal{T} \vdash K(G, 1, \lceil \alpha_i \rceil)$. Using $AR4a$, we get $\mathcal{T} \vdash K(G, 2, \lceil CC(jill, 01) \rceil)$, as an example. Also, using $AR3b$, we obtain $\mathcal{T} \vdash K(G, 2, \lceil SC(jack, jill) \wedge Co(jill, usa) \rceil)$. Then, using $AR4a$, we obtain $\mathcal{T} \vdash K(G, 3, \lceil C(jack, usa) \rceil)$. Now by using $AR4a$ again we obtain $\mathcal{T} \vdash K(G, 4, \lceil CC(jack, 1) \rceil)$, as claimed informally earlier.

The default example about area codes follows the same way except for a substitution of $AR4a$ by $AR4c$.

2.3 The basic semantics

This section deals with the semantics of the meta-theory $\mathcal{T}$ (actually $\mathcal{T}_{Me}$) for agents which was presented earlier. We start by showing, in Theorem 1, that $\mathcal{T}$ is consistent; for this purpose we construct a model. (We consider only Herbrand models, that is, models that contain only elements that are ground terms of L_{Me}). Actually, we construct a special type of minimal model, which contains a minimal set of positive atoms that can exist in a model. This model, in fact, models the reasoning of the agents, assuming our axioms. See Theorem 2 below. Our construction is based on the construction of the unique perfect model of a locally stratified theory in logic programming [8]. Such a model is important because only that model reflects exactly the reasoning and actions of the agents. (Other models of the theory would include, for example, the knowledge of formulas not necessarily obtained by the axioms.) Then, we provide restrictions on the models to obtain this intended model in two different ways.

First we define the notion of *knowledge supported model* in which each positive atom is thereby an application of an axiom. The model obtained this way is the same minimal model constructed earlier. The second approach uses the concept of *completion* (this is basically the *Clark completion* [8]) to achieve the same purpose. This is done by writing the completion axioms for all the predicates. For example, the original axioms for the A predicate are all implications defining it. To make sure that each positive atom is in a model by an application of an axiom, the reverse implication is needed. This is given in the completion axiom which is a disjunction of the different ways that A is constructed.

Theorem 1 *$\mathcal{T}$ has a minimal model, for any given L_{Me}.*

Proof: We sketch the construction for a minimal model, M, where minimization is done separately for each time value. We omit the specific details showing how each rule generates atoms in M. The key idea is to minimize each predicate for one time period (strata) at a time starting with 0 and proceeding by induction. We define a priority relationship ($\succ$) over atoms of the Herbrand universe, where, $R(n, i, \lceil\alpha\rceil) \succ S(m, j, \lceil\beta\rceil)$ if $i < j$, for any ternary predicates R and S, and similarly for Ask and $Tell$. In the construction we choose atoms of lower priority before atoms of higher priority, wherever possible. As an example, consider the axiom of negative introspection, $AR5h$, and suppose that $A(n, i, \lceil\alpha\rceil)$ is already in the model, but $K(n, i, \lceil\alpha\rceil)$ is not in the model. At this point, using the axiom we could add either $K(n, i, \lceil\alpha\rceil)$ or $K(n, i+, neg(k(i, \lceil\alpha\rceil)))$ to the model. Since $K(n, i, \lceil\alpha\rceil) \succ K(n, i+, neg(k(i, \lceil\alpha\rceil)))$, we add $K(n, i+, neg(k(i, \lceil\alpha\rceil)))$.

The construction is possible because all the axioms are either (quantified) atoms ($AR1$ and $AO1$) or implications where the time component in the consequent is greater than the time component in the antecedent. The construction starts by writing all atoms at time 0 that must hold in every model. These are the OBS atoms with time 0 that are axioms from $AO1$, i.e. $OBS(n, 0, \lceil\alpha\rceil)$ and $K(m, 0, \lceil Now(0)\rceil)$ by $AR1$ for every agent constant m, and possibly additional special axioms.

In the induction step we assume that the portion of the model M that contains all its atoms with time value i has already been constructed. The details of the construction at time value $i + 1$ involve the application of each relevant axiom to atoms in M(and in some cases atoms not in M) to generate atoms at time value $i + 1$ using the preference relation discussed earlier. ▮

The model we have constructed is the intended minimal model. Hence, by including a set of observations and questions of the agents that is finite for each time period, we can generate the minimal model one period at a time and thus determine, for example, what different agents will know at various times. It is also possible to choose a different set of inference rules in which case the knowledge and actions of the agents might be different. We used a priority relationship to construct the intended model. We now sketch another technique that can be used to define the same model. The idea of *knowledge supported model* is that every positive fact in the model must be there because of the application of an axiom. Again, a model that is not knowledge supported would contain additional statements. For example, such a model might have $K(1, 2, \lceil Co(jill, usa)\rceil)$, even though there is no way for agent 1 at time 2 to know that Jill's country is the USA.

Definition 4 A model M' of $\mathcal{T}$ is knowledge supported if

1. whenever $M' \models P(n, i, \lceil\alpha\rceil)$, either
   (a) $\alpha = Now(i)$, or
   (b) $P = OBS$ and $OBS(n, i, \lceil\alpha\rceil)$ is an axiom, or
   (c) $P \in \{K, A, OBS, Need\}$, and there exists an axiom of the form $\forall \bar{t_1}(\phi(\bar{t_1}, \bar{t_2}) \rightarrow P(\bar{t_3}))$ and a substitution θ such that $\bar{t_3}\theta =< n, i, \alpha >$ and there is a substitution θ', $\theta \subseteq \theta'$, such that $M' \models \phi(\bar{t_1}, \bar{t_2})\theta'$, and

2. a similar condition to 1(c) (with an extra agent value) for the Ask and $Tell$ predicates.

The definiteness of the axioms (no disjunction) guarantees that there is a unique knowledge supported model which is the same as the model constructed in Theorem 1.

Theorem 2 *The minimal model M defined in the proof of Theorem 1 is the unique knowledge supported model of T.*

Proof: Let M be the minimal model constructed in the proof of Theorem 1. The proof follows the inductive construction of this minimal model by observing that each time a new atom is added to M, one of the conditions for a knowledge supported model must hold. Conversely, suppose that M' is knowledge supported. By induction on time values we can show that the priority relationship places the same atoms into M that are in M'. ∎

Another approach for obtaining the same minimal model adds *completion* axioms for all the predicates. We show the completion axioms here for some representative predicates. We note that in order to be able to write the completion axioms, the original axioms must satisfy some finiteness criterion. In particular, the number of OBS facts must be finite for each time value.

1. The completion axiom(s) for OBS states that all OBS facts are obtained from OBS axioms. If there are finitely many OBS axioms, one completion axiom suffices:
   $\forall zxy(OBS(z,x,y) \rightarrow OBS(z,x,\lceil\alpha^1\rceil) \vee \cdots \vee OBS(z,x,\lceil\alpha^t\rceil))$
   where there are t OBS axioms.
   Otherwise, for each time value i and each agent n for which there is at least one OBS axiom, we write
   $< i,n >$: $\forall y(OBS(n,i,y) \rightarrow OBS(n,i,\lceil\alpha^1_{n,i}\rceil) \vee \cdots \vee OBS(n,i,\lceil\alpha^{t_{i,n}}_{n,i}\rceil))$ where the number of axioms for agent n with time i for OBS is the finite integer $t_{i,n}$.

2. The completion axiom for A is (intuitively) the reverse of the axioms in $AR5$ for A.

$$\begin{aligned}&\forall zxy(A(z,x+,y) \rightarrow K(z,x,y) \vee \exists y_1y_2(y_2 = imp(y,y_1) \wedge A(z,x,y_2))\vee\\ &\quad \exists y_1y_2(y_2 = imp(y_1,y) \wedge A(z,x,y_2)) \vee \quad \exists y_1y_2(y_2 = con(y,y_1) \wedge A(z,x,y_2))\vee\\ &\quad \exists y_1y_2(y_2 = con(y_1,y) \wedge A(z,x,y_2)) \vee \exists y_1(y_1 = neg(y) \wedge A(z,x,y_1))\vee\\ &\quad \exists y_1z(y_1 = exists(y,z) \wedge A(z,x,y_1)) \vee \exists y_1v(y_1 = forall(y,v) \wedge A(x,y_1))).\end{aligned}$$

3. We write two completion axioms for K: one for time 0 and the other for a generic time $x+$.
   $\forall zy(K(z,0,y) \rightarrow y = \lceil Now(0)\rceil)$.
   $\forall zxy(K(z,x+,y) \rightarrow D_{AR1} \vee \cdots \vee D_{AR7})$
   containing a disjunct for all axioms that have a K atom after $\rightarrow$. For example, the disjunct D_{AR3a} is:
   $\exists y_1(K(z,x,y_1) \wedge K(z,x,imp(y_1,y)))$.
   For another example, the disjunct D_{AR4a} is:
   $\exists y_1y_2vw(K(z,x,sub(y_1,v,w)) \wedge K(z,x,forall(imp(y_1,y_2),v)) \wedge y = sub(y_2,v,w))$.

The effect of the completion axioms is the same as what we get by using the concept of knowledge supported model, that is, to ensure that if an atom is in a model of the completion, it must be there because of an axiom. Hence we obtain

Theorem 3 *The minimal model M defined in the proof of Theorem 1 is the unique model of $\mathcal{T} \cup C(\mathcal{T})$ (the theory together with its completion).*

¿From the previous two theorems we obtain an equivalence between the concept of knowledge supported model and the addition of the completion to the theory. This provides a soundness and completeness result.

Theorem 4 $\mathcal{T} \cup C(\mathcal{T}) \vdash \alpha$ *iff for every knowledge supported model M of $\mathcal{T}, M \models \alpha$.*

Proof: (Soundness: $\Rightarrow$) Immediate from the soundness of first-order logic since every knowledge supported model is a model.

(Completeness: $\Leftarrow$) This is proved by showing that every model of $T \cup C(T)$ is a knowledge supported model using the definitions. The proof proceeds by induction on the time value. ∎

2.4 Consistency

Above we characterized the intended model in three different ways. The existence of a model has another important consequence, the consistency of $\mathcal{T}$. This provides an interesting contrast with a result of Montague [44]. He showed that under certain fairly intuitive conditions a syntactic formalization of modal notions such as knowledge becomes contradictory. Thomason later showed [54] that under even less restrictive conditions a syntactic formalization yields a contradictory set of beliefs. In Montague's case the meta-theory is inconsistent, and in Thomason's the object (agent) theory is inconsistent.[4]

Yet our meta-theory is always consistent. Moreover an agent theory need not by any means be inconsistent. Nevertheless, it is an interesting property of our approach that agent theories may indeed be (or become) inconsistent — e.g., through observations — and this is one of the advantages of our treatment: commonsense reasoning can proceed in the presence of inconsistency. The time-situated character of our logic offers "protection" from inconsistency, in two ways: it affords a consistent meta-theory, and it allows for an inconsistent but still useful agent theory. (While we have not concentrated on the latter aspect in the present paper, other work makes substantial use of this idea [46, 15, 43].)

Why is our meta-theory consistent, in light of negative results by Montague, Thomason, and others? The main intuition is that Montague and Thomason tried to model ideal agents, while our main goal is to model bounded agents. Therefore, not all the axioms which they present are used in our system. Some of these axioms are modified by adding the time of the reasoning into consideration e.g., Thomason and Montague's scheme $K(\alpha) \wedge K(\alpha \rightarrow \beta) \rightarrow K(\beta)$ is modified and applied in our model as: $\forall zxy_1y_2(K(z,x,y_1) \wedge K(z,x,imp(y_1,y_2)) \rightarrow K(z,x+,y_2))$. Other axioms such as $K(K(\alpha \rightarrow \alpha))$ are not used at all. In particular, we have stratified time, not all beliefs are held at the same time, or inherited once held. In addition, we have no "infinity point" at which there is logical closure.

[4] Partial solution to these problems can be obtained by restricting the language of the agents [45, 11]. The language of our agents is not restricted.

2.5 Complexity concerns

Our meta-theory is first-order, hence semi-decidable, but in general we cannot guarantee any low degree of complexity. This however may not be troubling, since the intended practical use of such a formalism is not for online agent-use at all, but rather for offline beta-testing and analysis beforehand, and done only once (per agent family-type).

3 Families of agents

The assumption that the agents can apply all the rules in one unit of time is often unrealistic. So in the telephone example agent B may not be able to count on agent C (a program) to complete its computation within four units of time. In general, one may not be able to guarantee that all the rules, such as modus ponens, will always be executed in one time unit. But it may be possible to guarantee that a rule, like modus ponens, will be executed in at most m time periods for some fixed m. Therefore, in this section we consider a modification of the rules, by allowing axioms that may act over several time units. We refer to them as *indefinite* axioms. For example, the following version of the axiom for $AR3a$: $\forall zxy_1y_2(K(z,x,y_1) \wedge K(z,x,imp(y_1,y_2)) \rightarrow (K(z,x+,y_2) \vee K(z,x++,y_2))$ applies Modus Ponens within two time periods. We may think of such a rule as characterizing families of agent groups rather than an agent group. So one use of this axiom may be thought of as standing for two agents with the same name; one applying MP within one time period and the other applying MP during the second time period; or more generally to agents that may take either one or two time units to apply MP.

There are several families of such indefinite agents that can be characterized depending on how much time it takes them to apply an inference rule. We have identified two main cases, with two subcases in each. We consider agents where there is a fixed bound on the amount of time it takes them to apply a rule (the *fixed case*). We also consider agents for which it is simply known that the rules will be applied eventually (the *eventual case*). Within each case we distinguish two subcases: the first is where the formulas need to be known only at the initial time, and the second is where the formulas need to be known for every time period, starting with the initial time, until the rule is applied. Intuitively, in the first case an agent has some mechanism to remember which inference rules it needs to do, e.g., by putting it in a queue. The second case is more realistic. In each time period, an agent uses its current beliefs and applies its inference rules. It is only capable of applying a few rules at a time, but it does so in a systematic way to ensure "fair" treatment of all the formulas.

We start by motivating each case and writing the corresponding version for MP. The other AR axioms would be modified the same way.

Fixed Case 1 (one-time knowledge). This version states that if certain formulas are known at some time then within ℓ time periods something else will be known. Here it is for MP. $\forall zxy_1y_2(K(z,x,y_1) \wedge K(z,x,imp(y_1,y_2)) \rightarrow K(z,x+,y_2) \vee \cdots \vee K(z,x+^{\ell},y_2))$

Fixed Case 2 (continuous knowledge). This version is similar to 1, but the formulas known at the initial time must continue to be known in order for the rule to be applied. The version

for MP is
$\forall zxy_1y_2(K(z,x,y_1) \wedge K(z,x,imp(y_1,y_2)) \rightarrow K(z,x+,y_2) \vee \cdots \vee (K(z,x+,y_1) \wedge K(z,x+,imp(y_1,y_2)) \wedge \cdots \wedge K(z,x+^{\ell-1},y_1) \wedge K(z,x+^{\ell-1},imp(y_1,y_2)) \rightarrow K(z,x+^{\ell},y_2)))$.

Eventual Case 1 (one-time knowledge). This version states that if certain formulas are known at some time then eventually something else will be known. Here is the version for MP.
$\forall zxy_1y_2(K(z,x,y_1) \wedge K(z,x,imp(y_1,y_2)) \rightarrow \exists x'(x < x' \wedge K(z,x',y_2)))$.

Eventual Case 2 (continuous knowledge). In this version the formulas known at the initial time must continue to be known continuously in order for the rule to be applied. The version for MP is given below
$\forall zxy_1y_2(K(z,x,y_1) \wedge K(z,x,imp(y_1,y_2)) \rightarrow \exists x'(x < x' \wedge \forall x''(x < x'' \wedge x'' < x' \wedge K(z,x'',y_1) \wedge K(z,x'',imp(y_1,y_2)) \rightarrow K(z,x',y_2))))$.

When $\mathcal{T}$ includes indefinite axioms there will be more than one intended model, as the various intended models represent the actions of different families of agents. Later in this section we will show how to construct the class of all intended models. For now, let's consider the concept of minimality in this context. Consider Fixed Case 1 above, where $l = 2$, and suppose that y_2 can be inherited to $x++$. Consider the substitution of n for z, i for x, $\lceil\alpha\rceil$ for y_1 and $\lceil\alpha'\rceil$ for y_2 and models that satisfy $K(n,i,\lceil\alpha\rceil)$. In this case choose M such that $M \not\models K(n,i+,\lceil\alpha'\rceil)$, but $M \models K(n,i{+}{+},\lceil\alpha'\rceil)$. The other model, with respect to this axiom, M' will be such that $M' \models K(n,i+,\lceil\alpha'\rceil)$ and assuming inheritance, $M' \models K(n,i++,\lceil\alpha'\rceil)$, hence the former is minimal (with respect to the application of this specific axiom). For the minimality of a model the same process would have to be applied to all the axioms.

In general, for Fixed Case 1, suppose the maximal number of time units for the agent to apply MP is ℓ. If α can be inherited, the minimal model will have $M \models K(n,i+^{\ell},\lceil\alpha\rceil)$, but not any $M \models K(n,i+^{m},\lceil\alpha\rceil)$, $m < \ell$. On the other hand, if α cannot be inherited, there will be several different minimal models, such as M_1 and M_2, where $M_1 \models K(n,i+,\lceil\alpha\rceil)$, but $M_1 \not\models K(n,i++,\lceil\alpha\rceil)$, and $M_2 \models K(n,i++,\lceil\alpha\rceil)$, but $M_2 \not\models K(n,i+,\lceil\alpha\rceil)$.

The minimality of a model may differ between Fixed Case 1 and 2. For example, in the Fixed Case 1 we may have (1) $K(n,i,\lceil\alpha\rceil)$, (2) $K(n,i,imp(\lceil\alpha\rceil,\lceil\beta\rceil))$, (3) $K(n,i+,\lceil\beta\rceil)$, (4) $K(n,i+,neg(\lceil\beta\rceil))$ (5) $K(n,i++,\lceil\beta\rceil)$ in a model which is not minimal because (5) or (3) may be deleted. However, in the Fixed Case 2, where the formulas must continue to be known continuously, this situation cannot arise: we will not have (5) because α is not known at time $i++$.

Consider Eventual Case 1. In this case, if α can be inherited indefinitely, then there will not be a minimal model because we can construct an infinite chain of models .... $\subseteq M_2 \subseteq M_1$, where $M_j \models K(n,i+^{j},\lceil\alpha\rceil)$ and $M_j \not\models K(n,i+^{h},\lceil\alpha\rceil)$ for any $h < j$. Again, if α cannot be inherited, there will be many minimal models.

Returning to the telephone example in the paragraph before Section 2.3, we can use the above indefinite approach to formalize a family of agents, each of which may take, say, one or two time units to perform the $AR3$ axioms, but a single time unit to perform the axiom $AR2$. Then in our meta-theory $\mathcal{T}$ for the fixed case starting at time 0, we obtain instead of

$K(G, 4, \lceil CC(jack, 1)\rceil)$,

$$K(G,4,\lceil CC(jack,1)\rceil) \vee K(G,5,\lceil CC(jack,1)\rceil) \vee$$
$$K(G,6,\lceil CC(jack,1)\rceil) \vee K(G,7,\lceil CC(jack,1)\rceil)$$

whereas in the eventual case we instead can prove $\exists x K(G, x, \lceil CC(jack, 01)\rceil)$. A similar result holds for default implication.

The concepts of knowledge supported model and completion axioms also apply to the indefinite case. Let's consider the definition of *knowledge supported model* first. The portion that needs to be changed is "there exists an axiom of the form $\forall \bar{t_1}(\phi(\bar{t_1}, \bar{t_2}) \to P(\bar{t_3}))$" because the indefinite axioms have a different form.

Definition 5 M' is a knowledge supported model of $\mathcal{T}$ if both 1. and 2. hold

1. whenever $M' \models P(n, i, \lceil\alpha\rceil)$, either
   (a) $\alpha = Now(i)$, or
   (b) $P = OBS$ and $OBS(n, i, \lceil\alpha\rceil)$ is an axiom, or
   (c) $P \in \{K, A, OBS, Need\}$, and there exists an axiom of the form
   Definite Case $\forall \bar{t_1}(\phi(\bar{t_1}, \bar{t_2}) \to P(\bar{t_3}))$ and a substitution θ such that $\bar{t_3}\theta =< n, i, \alpha >$ and there is a substitution θ', $\theta \subseteq \theta'$, such that $M' \models \phi(\bar{t_1}, \bar{t_2})\theta'$,
   Fixed Case 1 $\forall \bar{t_1}(\phi(\bar{t_1}, \bar{t_2}) \to D)$ where D is a disjunction of atoms one of which is $P(\bar{t_3})$ and a substitution θ such that $\bar{t_3}\theta =< n, i, \alpha >$ and there is a substitution θ', $\theta \subseteq \theta'$, such that $M' \models \phi(\bar{t_1}, \bar{t_2})\theta'$,
   Fixed Case 2 $\forall \bar{t_1}(\bigvee_{j=1}^{\ell} (\phi^j(\bar{t_1}, \bar{t_2})) \to P(\bar{t_3^j}))$ and a substitution θ such that $\bar{t_3^j}\theta =< n, i, \alpha >$ and there is a substitution θ', $\theta \subseteq \theta'$, such that $M' \models \phi^j(\bar{t_1}, \bar{t_2})\theta'$,
   Eventual Case 1 $\forall x\bar{t_1}(\phi(x, \bar{t_1}, \bar{t_2}) \to \exists x'(x < x' \wedge P(x', \bar{t_3})))$ and a substitution θ such that $P(x', \bar{t_3})\theta = P(n, i, \lceil\alpha\rceil)$ and a substitution θ', $\theta \subseteq \theta'$, such that $M' \models \phi(x, \bar{t_1}, \bar{t_2})\theta'$ and $x < x'$,
   Eventual Case 2 $\forall x\bar{t_1}(\phi(x, \bar{t_1}, \bar{t_2}) \to \exists x'(x < x' \wedge \forall x''(x < x'' \wedge x'' < x' \wedge \phi(x'', \bar{t_1}, \bar{t_2})) \to P(x', \bar{t_3})))$, and a substitution θ such that $P(x', \bar{t_3})\theta = P(n, i, \lceil\alpha\rceil)$ and a substitution θ', $\theta \subseteq \theta'$, such that $M' \models \phi(x, \bar{t_1}, \bar{t_2})\theta'$ and $x < x'$, and for all x'' such that $x < x'' < x'$, $M' \models \phi(x'', \bar{t_1}, \bar{t_2})\theta'$

2. a similar condition to 1(c) (with an extra agent value) for the Ask and $Tell$ predicates.

Again, the definition of knowledge supported model assures that in such a model every atom is there because of the application of an axiom. In the previous section we showed that another way to accomplish the same purpose is to add completion axioms for all the predicates. For the completion axiom for K we gave the disjunct corresponding to D_{AR3a}. Here we show how to handle this disjunct for the four types of indefiniteness.

Fixed Case 1 $\exists x y_1(K(z, x, y_1) \wedge K(z, x, imp(y_1, y)) \wedge (x+ = i \vee \ldots \vee x+^{\ell} = i))$

Fixed Case 2 $\exists xy_1((x+ = i \wedge K(z,x,y) \wedge K(z,x,imp(y_1,y)))$
$\wedge \ldots (x+^{\ell} = i \wedge K(z,x,y_1) \wedge \ldots \wedge K(x+^{\ell},y_1) \wedge K(x,imp(y_1,y)) \wedge \ldots \wedge K(x+^{\ell},imp(y_1,y)))$

Eventual Case 1 $\exists xy_1(x < i \wedge K(z,x,y_1) \wedge K(z,x,imp(y_1,y)))$

Eventual Case 2 $\exists xy_1(x < i \wedge K(z,x,y_1) \wedge K(z,x,imp(y_1,y)) \wedge$
$\forall x''(x < x'' \wedge x'' < i \rightarrow K(z,x'',y_1) \wedge K(z,x'',imp(y_1,y))))$

In a similar way the completion axiom can be written for each predicate in the indefinite cases. Essentially, the completion axioms formalize within first-order logic the concept of knowledge supported model. Hence, writing $C(\mathcal{T})$ for the completion of $\mathcal{T}$, we obtain

Theorem 5 *M is a knowledge supported model of $\mathcal{T}$ if and only if $M \models \mathcal{T} \cup C(\mathcal{T})$.*

Proof: (The idea) The proof proceeds by strong induction on the time component. For time $i = 0$ there is no indefiniteness and the proof is the same as the corresponding part of the proof of Theorem 4. For the inductive part the equivalence of the completion axioms with the concept of knowledge supported model must be shown. ▮

The concept of knowledge supported model (and its equivalent version via completion axioms) is useful but suffers from the inclusive interpretation of disjunction in first-order logic. Earlier, we gave an example of a knowledge supported model for Fixed Case 1 that included (1) $K(n,i,\lceil\alpha\rceil)$, (2) $K(n,i,imp(\lceil\alpha\rceil,\lceil\beta\rceil))$ (3) $K(n,i+,\lceil\beta\rceil)$, (4) $K(n,i+,neg(\lceil\beta\rceil))$ and (5) $K(n,i{+}{+},\lceil\beta\rceil)$. If we think of an indefinite axiom as standing for a deduction that may take one of several time periods but where the result appears once for that deduction only, then the knowledge supported model above would not be an intended model, i.e., it would not represent any of the agents in the family. To end this section we sketch the construction for a class, $\mathcal{M}$, of models, which interpret disjunction exclusively. This construction is similar to the construction of the minimal model in Theorem 1, except that for each disjunctive axiom a choice is made about which disjunct to make true in the consequent.
Construction of an intended model:
At time 0 the construction is the same as in Theorem 1. In the induction step assume that the portion of M that contains all its atoms with time value less than i has already been constructed. Consider now the construction of the portion of the model at time value i. The definite axioms are handled in the same way as in Theorem 1. For each possible application of an indefinite axiom we choose a positive integer as follows. For a Fixed Case axiom, choose a positive integer n, $1 \leq n \leq \ell$, and for an Eventual Case axiom, choose any positive integer n. Consider now any indefinite axiom, and suppose that the axiom is applicable at time j, i.e., $M \models \phi(j,\bar{t_1},\bar{t_2})$, where ϕ is the antecedent. If $j + n = i$ and M satisfies the antecedent of the axiom for all $j < i$, apply the axiom by adding the appropriate atom $\rho(i,\bar{t_3})$ to M. This way we assure that for each disjunctive axiom we pick a single disjunct only.

4 Agents with Focus of attention

Up to now we assumed that an agent can use all of its knowledge every time for its reasoning and actions. However, with inheritance and introspection, the knowledge base can grow very large. In this section we sketch the addition of a focus of attention to the original framework. The idea is that at any given time only some of the agent's beliefs are in the focus and the agent can use only what is in the focus for its reasoning and actions. As new knowledge is obtained, it is brought into the focus, but stays there only for a fixed amount of time.

We handle the concept of Focus by introducing a new predicate $F : (S_n, S_t, S_f, S_t)$ into L_{Me}. We read $F(G, i, \lceil\alpha\rceil, j)$ as saying that at time i the formula α is in agent G's focus with j time units remaining. Formulas are brought into the Focus through an observation of the agent, a deduction of the agent, because the agent is told the formula, or because the agent needs to know the formula. When a formula is placed into the Focus it stays there for c units of time, where c is some fixed constant.

We start by dealing with the definite case. We add two new axioms and modify some of the AR axioms to include the F predicate.

AF1. Inheritance in the Focus: This axiom is the inheritance axiom for F. It states that at the next time period a formula no longer stays in Focus once the number of time units remaining is 0; otherwise it stays in Focus, but with one less time unit remaining.

$$\forall z x_1 x_2 x_3 y (F(z, x_1, y, x_2) \wedge x_2 \neq 0 \wedge x_2 = x_3+ \ \rightarrow \ F(z, x_1+, y, x_3))$$

Entrance to the Focus:

AF2. This axiom shows that a formula gets into Focus when the agent needs to know it.

$$\forall zxy(Need(z, x, y) \ \rightarrow \ F(z, x+, y, c))$$

AF3. This axiom shows that a formula gets into the focus when it is related in a certain way to a known formula and to a needed formula.

$$\forall zxy_1y_2vw(Need(z, x, y_1) \wedge y_2 = sub(y_1, v, w) \wedge K(z, x, y_2) \ \rightarrow \ F(z, x+, y_2, c))$$

Returning to the country code example. If agent C needs to know someone who lives in the USA at period 9, i.e., $Need(C, 9, \lceil Co(u_1, usa)\rceil)$, and it knows that Jill lives in the USA, i.e., $K(C, 9, \lceil Co(jill, usa)\rceil)$, then the formula indicating the Jill lives in the USA will enter into its Focus in the next time period, i.e., $F(C, 9, \lceil Co(jill, usa)\rceil, 20)$.

AF4. This axiom shows an additional way in which a formula gets into the Focus when it is related in a certain way to a known formula and to a needed formula.

$$\begin{aligned}\forall zxy_1y_2y_3y_4vw(K(z, x, y_3) \wedge y_3 = forall(imp(y_1, y_2), v) \wedge Need(z, x, y_4)\\ \wedge y_4 = sub(y_2, v, w) \ \rightarrow \ F(z, x+, y_3, c))\end{aligned}$$

Next we show how to rewrite earlier axioms with Focus by placing the Focus predicate in the consequent. We give only some representative examples and do not rewrite all the axioms:

ARF1. Clock in Focus:

$$\forall zx(K(z,x,now(x)) \wedge F(z,x,now(x),0))$$

ARF2. Observation in Focus:

$$\forall zxy(OBS(z,x,y) \ \rightarrow \ K(z,x+,y) \wedge F(z,x+,y,c))$$

Later, when we use an axiom such as $ARF2$ we will use it as an abbreviation for two axioms, one for each conjunct in the consequent. Next is a representative example for $AR3$ and $AR4$. The idea is that if an agent deduces a formula, where deduction is possible only for formulas in the Focus, then it is placed in the Focus. The following version for Modus Ponens states that if both y_1 and $imp(y_1,y_2)$ are known at time x_1 and are in Focus with time remaining at most x_3, then at the next time period y_2 will be known and in Focus with the time remaining equal to $x_3 - 1$:

ARF3a. Modus Ponens with Focus:

$$\begin{aligned}
&\forall zx_1x_2x_3x_4y_1y_2(K(z,x_1,y_1) \wedge F(z,x_1,y_1,x_2) \\
&\wedge K(z,x_1,imp(y_1,y_2)) \wedge F(z,x_1,imp(y_1,y_2),x_3) \\
&\wedge((x_2 < x_3 \vee x_2 = x_3) \wedge x_4+ = x_3) \vee (x_3 < x_2 \wedge x_4+ = x_2))) \\
&\rightarrow \ (K(z,x_1+,y_2) \wedge F(z,x_1+,y_2,x_4))
\end{aligned}$$

The axioms $AR5$ and $AR6$ need not change. We change $AR7$ to the following:

ARF7 Truthfulness and Focus:

$$\forall z_1z_2xy(Tell(z_1,z_2,x,y) \ \rightarrow \ K(z_2,x+,y) \wedge F(z_2,x+,y,c))$$

$AR8$ and $AR9$ are not changed.

When Focus is added to the language and to the axioms, we obtain a different theory, $\mathcal{TF}$. However, this new theory has much the same properties that the original $\mathcal{T}$ had. In particular, $\mathcal{TF}$ has a minimal model. Also, the concepts of knowledge supported model and completion axioms can be carried over to $\mathcal{TF}$ and the theorems proved earlier for $\mathcal{T}$ are also true for $\mathcal{TF}$. We do not give the details of extending the concept of knowledge supported model, completion, and intended model where Focus is in the language.

In the previous section we considered the case of families of agents with indefinite rules. We sketch two different ways of handling such rules in the presence of Focus.

4.1 Reasoning with formulas in Focus takes an indefinite amount of time

The first method models a situation where the reasoning of the agents for formulas that are already in the Focus takes an indefinite amount of time. We change the ARF axioms (except for $ARF1$) to reflect this indefiniteness. To illustrate we show in two of the four cases how $ARF3a$ would be changed:

Fixed Case 1 $\forall z x_1 x_2 x_3 x_4 y_1 y_2 (K(z, x_1, y_1) \wedge F(z, x_1, y_1, x_2)$
$\wedge K(z, x_1, imp(y_1, y_2)) \wedge F(z, x_1, imp(y_1, y_2), x_3)$
$\wedge ((x_2 < x_3 \vee x_2 = x_3) \wedge x_4 + = x_3) \vee (x_3 < x_2 \wedge x_4 + = x_2)))$
$\rightarrow (K(z, x_1+, y_2) \wedge F(z, x_1+, y_2, x_4) \vee \ldots \vee K(z, x_1+^{\ell}, y_2) \wedge F(z, x_1+^{\ell}, y_2, x_4)))$.

Eventual Case 1 $\forall z x_1 x_2 x_3 x_4 y_1 y_2 (K(z, x_1, y_1) \wedge F(z, x_1, y_1, x_2)$
$\wedge K(z, x_1, imp(y_1, y_2)) \wedge F(z, x_1, imp(y_1, y_2), x_3)$
$\wedge ((x_2 < x_3 \vee x_2 = x_3) \wedge x_4 + = x_3) \vee (x_3 < x_2 \wedge x_4 + = x_2)))$
$\rightarrow \exists x' (x < x' \wedge K(z, x', y_2) \wedge F(z, x', y_2, x_4)))$

We give a result indicating the importance of the value for c as used in Focus. We consider the case where all formulas are inherited and the reasoning involves n periods. For example, if an agent knows α, $\alpha \rightarrow \beta$, and $\beta \rightarrow \gamma$, then, in $n = 2$ periods, using $ARF3a$, the agent will deduce γ. Because of the inheritance of formulas, there is no difference between Fixed Case 1 and 2.

Theorem 6 *Suppose that all relevant formulas in a chain of reasoning are always inherited and suppose the reasoning involves n uses of deduction rules, i.e., starting with certain formulas $\alpha_1, \ldots, \alpha_k$ known at time t, the formula β is deduced at time $t + n$ if each rule takes one unit of time to do and there is no Focus. Then, using this first method for Focus, suppose that initially $\alpha_1, \ldots, \alpha_k$ are known and in Focus with c time units remaining.*

Definite Case *If $c \geq n - 1$ the deduction will be successful.*

Fixed Case *Let ℓ be the largest time value for an indefinite axiom used in the deduction. If $c \geq n \cdot \ell - 1$ the deduction will be successful.*

Eventual Case 1 *The success of the deduction is not assured if $n > 1$.*

Eventual Case 2 *The success of the deduction is not assured.*

Proof:

Definite Case All formulas needed for the deduction are in Focus at time t with c time units remaining. The last deduction will take place at time $t+n-1$. Hence, if $c \geq n-1$, the formulas will remain in Focus long enough to complete the deduction.

Fixed Case Similar to the Definite Case, however a deduction may take up to ℓ units of time. So if $c \geq n \cdot \ell - 1$, the formulas will remain in Focus long enough to complete the deduction.

Eventual Case 1 No matter how large c is, in case a step in the deduction takes time longer than c time units, a formula needed in another deduction step may no longer be in Focus. For $n = 1$ the deduction will succeed because the formulas are in Focus at time t.

Eventual Case 2 The same as Eventual Case 1 except that even for $n = 1$ the formulas needed in the deduction may go out of Focus before the deduction which then will not take place.

▮

4.2 Getting into Focus takes indefinite amount of time

The second method of dealing with families of agents models the situation where the agents are able to do all the inferences on formulas in the Focus in one step, but it may take an indefinite amount of time for formulas to get into the Focus. For this method we change the axioms $AF2$ - $AF4$, but not the ARF axioms. To illustrate, we show how $AF2$ would be changed in two cases:

Fixed Case 1 $\forall zxy(Need(z,x,y) \rightarrow F(z,x+,y,c) \vee \ldots \vee F(z,x+^{\ell},y,c))$.

Eventual Case 1 $\forall zxy(Need(z,x,y) \rightarrow \exists x'(x < x' \wedge F(z,x',y,c))$

For this method if all the formulas needed for the deduction are known and in Focus initially, all four indefinite cases reduce to the definite case because the indefiniteness appears in the Focus only in connection with the $Need$ predicate. So we set up the theorem in this case in such a way that none of the formulas needed in the deduction is in the Focus initially. Instead, we start by placing the formula β in the $Need$ predicate.

Theorem 7 *Suppose that all relevant formulas in a chain of reasoning are always inherited and suppose the reasoning involves n uses of deduction rules, i.e., starting with certain formulas $\alpha_1, \ldots, \alpha_k$ known at time t, the formula β is deduced at time $t+n$ if each rule takes one unit of time to do and there is no Focus. Then, using this second method for Focus, suppose that initially $\alpha_1, \ldots, \alpha_k$ are known and β is needed.*

Definite Case *If $c \geq 2 \cdot n - 1$ the deduction will be successful.*

Fixed Case *Let ℓ be the largest time value for an indefinite axiom used in the deduction. If* $c \geq 2 \cdot n + \ell - 2$ *the deduction will be successful.*

Eventual Case *The success of the deduction is not assured.*

Proof:

Definite Case The deduction proceeds for the first n time periods by placing all the formulas needed for the deduction in the $Need$ predicate and by the following time period all these formulas are in the Focus. The deduction itself may take n time units. Altogether, there are $2 \cdot n + 1$ time periods involved. Assume that the initial time is t. If a formula that comes into Focus in the first time period is needed for a deduction in the last time period, it must remain in Focus from time $t + 1$ to time $t + 2 \cdot n$. Hence, if $c \geq 2 \cdot n - 1$, all formulas will be in Focus long enough to complete the deduction.

Fixed Case As in the Definite Case the deduction proceeds for the first n time periods by placing all the formulas needed for the deduction in the $Need$ predicate. Again suppose that the initial time period is t. At time $t + n$ a formula may take up to ℓ time units to get into Focus. Once the formulas are in Focus, processing is as in the Definite Case. Hence the deduction may take up to $n + \ell + n$ time units. So a formula may have to remain in Focus from time $t + 1$ to time $t + 2 \cdot n + \ell - 1$. Hence, if $c \geq 2 \cdot n + \ell - 2$, all formulas will be in Focus long enough to complete the deduction.

Eventual Case Similar to the Eventual Case of the previous theorem, but even for Eventual Case 1, in case $n = 1$ the needed formulas may come into Focus at times whose distance from each other is greater than c and hence the deduction will not take place.

∎

5 Related Work

Numerous efforts in artificial intelligence research try to formalize a logical axiomatization for rational agents (see [57] for a good survey). This is usually accomplished by formalizing a model for agent behavior in terms of beliefs, desires, intentions (BDI). In this paper we concentrated on the ways the agents' beliefs evolve over time, however our model can be easily extended to include intentions and desires and to study how they are changing over time.

Most of the BDI research adopts Kripke possible worlds semantics (e.g., [26, 33, 34, 10].) This yields the known problem of logical omniscience for belief and knowledge, i.e., the agent's knowledge and belief are closed under consequences. In particular, the multi-modal logic KL_n, contains connectives for representing both time and knowledge for multi-agents [25, 20, 13, 27]. It is based on a linear discrete temporal logic [17][5] and contains, in addition, an indexed set of unary modal connectives for knowledge. These connectives satisfy analogues of the modal system S5, which is widely recognized as a logic of idealized knowledge. We

[5]Surveys of research on temporal reasoning include [5, 6, 3, 36, 56, 2, 9].

study the behaviour of bounded agents and therefore do not apply modal logic, but rather choose a syntactic approach for the modalities.

Georgeff and Rao [48, 47] consider complex possible worlds semantics (and thus also suffer from the omniscience problem). In their formalism, the belief-desire-intention accessible worlds are branching time structures. We do not use the modal temporal logic approach [40, inter alia] where time periods cannot be expressed in the language. We use first order logic, and in our framework time is expressed using a special sort which provides an explicit mechanism for expressing time points time variables which can be quantified (as in, for example, [53]). Thus, it is easier to express in our meta-logic axioms such as the ones in Section 4 on focus of attention. Also, our agents can express time explicitly in their reasoning about their knowledge.

Doyle [14] discusses the complementary roles that the economics theory of rationality and mathematical logic play in the development of rational automated agents. In particular Doyle explores some major limitations of agents that influence their rationality. Our approach was developed to overcome such limitations of automated agents.

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

Konolige [31] suggests a deduction model that is deductively closed under a set of rules, though not necessarily *consequentially* closed, since the rules may not be logically complete. The assumption of deductive closure greatly simplifies the technical problems by disregarding any particular control strategy. It is suggested that systems that are not deductively closed can be modeled by employing a *low cost bound* on derivations, thereby deriving only those proof trees whose depth is bounded; this extension is not part of the model. The model ignores the time element present in the inferencing, assuming that the agent can perform the necessary computations in a time interval which is relatively short with respect to its ability to act.

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

Our approach gives an account of one's explicit set of beliefs. Most of the formalizations of reasoning in the literature deal only with one's implicit set of beliefs. Levesque [38], however, does distinguish these two, giving an intuitively plausible semantic account of both *implicit* and *explicit* beliefs. An agent's implicit beliefs include all valid formulas, its explicit beliefs, and the logical consequences of these explicit beliefs. The agent's explicit beliefs, on the other hand, are closed under a much weaker set of conditions. Using the set of explicit beliefs, Levesque is able to describe an agent who is not logically omniscient.

Levesque's logic, however, does not allow meta-reasoning about one's own beliefs or reasoning about other agents' beliefs. These abilities are needed in many situations, including planning and goal-directed behavior, where one may have to reason about the knowledge that one has as well as the knowledge that others may possess. Fagin and Halpern ([19]) extend Levesque's notion of implicit and explicit beliefs to allow for multiple agents and beliefs of beliefs.

In another related approach, Lakemeyer [35] presents a logic to model the beliefs of a limited reasoner. His model combines features from both possible-world semantics and relevance semantics. Fagin and Halpern [19] propose a model of awareness in which an agent is aware of p if it believes the proposition $p \vee \neg p$. They regard explicit belief as the conjunction of implicit belief and awareness. An agent then may explicitly believe p and $p \rightarrow q$ while not explicitly believing q.

Hoek et. al. [55] provide a framework in which many aspects of rational agency are integrated. They present actions an agent can take which lead to changes in its beliefs, but do not address the problem of logical omniscience.

Huang et al. [29] develop an action logic, ALX, for agents with bounded rationality. ALX avoids the problems similar to the logical omniscience problem in the context of actions and goals: i.e., the necessitation rule for goals (every theorem must be a goal) and the closure of goals under logical implication. They do not consider issues of knowledge and time.

Giunchiglia and Serafini [21] develop formal systems allowing multiple distinct languages and call them *MultiLanguage systems (ML)*. They focus on a particular class of ML systems which allow a hierarchy of first-order languages containing names for the language below and propose them as an alternative to modal logics. Their results also support the claim that we make in this paper (see Section 2.4) that first-order logic is all that is needed to give a consistent theory of propositional attitudes and provability of rational agents. However, their main interest is to provide foundations to the implementation of "intelligent" reasoning systems while we focus in this paper on modeling the reasoning process of agents over time.

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

Durfee and his colleagues [24] develop a logic of knowledge and belief to model multiagent coordination. As in our work, their framework permits an agent to reason not only about the world and its own actions, but also to simulate and model the behavior of other agents in the environment. However, they do not consider temporal issues.

Our logic models agents reasoning about their own reasoning, e.g., about the time being taken by their own reasoning and other agents' reasoning and the reasoning of a designer of an agent on its agent's reasoning. This is then meta-reasoning. Metareasoning has been one of the most studied research topics in formal reasoning (e.g., [30, 7, 28, 12, 32]).

In artificial intelligence, meta-reasoning has become almost synonymous with nonmonotonic reasoning. This is largely due to the fact that nonmonotonic formalisms always rely on some assessment of what a reasoner does not know (or cannot prove), this state of ignorance itself being a piece of meta-information about the reasoner. Sometimes this ignorance is explicit (as in McDermott and Doyle's approach [42]) and sometimes implicit (as in Reiter's and McCarthy's approaches [49, 41]). However, in general, meta-reasoning is a broader notion, involving introspective beliefs in general, whether defeasible or not. For instance, positive introspection means that if I know α then I also know that I know α: $Know(I, \alpha)$ is in my knowledge base, or at least in the logical closure of my knowledge base.

Our approach explicitly represents such an introspection predicate, K, which however, has a time parameter. This is, among other things, the means by which our meta-theory endows its reasoner with a history. If the belief α occurs at time t, then at later times the belief $K(I, t, \lceil\alpha\rceil)$ can be present, recording the fact that α had been a belief at time t. We say "can be present" because this is not required as part of the definition of our meta-theory; we have found it (or its negative counterpart: infer $\neg K(I, t, \lceil\alpha\rceil)$ when α is not a time t belief) useful in many cases, but it also leads to an overabundance of unnecessary introspective conclusions.[6]

Giunchiglia and Traverso [22] describe a meta-theory of mechanized object theory, i.e., a meta-theory which can be put in correspondence with the code implementing the object theory. Barringer et al. [4] present an executable meta-language for temporal logic. Its usage is in the development of interpretation, debugging, loop-checking, and simulation tools, and it provides a representation for providing control knowledge in the execution of planning and scheduling programs. In contrast, we focus on modeling the inference mechanisms of the agents and the ways their beliefs change over time.

Chapter 10 of [20] discusses a very general approach which allows for the possibility of fairly arbitrary relationships between "stages" of reasoning. The generality is such that very little can be said about specific inferences without making further assumptions leading to a more concrete version. In particular, an indexical time-passage aspect is not required.

Doyle's truth maintenance notion was the first serious effort at belief revision; curiously, however, belief revision was largely ignored in the flurry of work on nonmonotonicity, until the later work by Alchourron, Gardenfors, and Makinson (often referred to as the AGM approach) [1, 23] and especially showing a strong connection between NMR and belief revision. The AGM treatment focuses on theory change rather than simply belief change. That is, a theory is viewed as having a fixed set of beliefs (axioms, theorems); this is of course the traditional notion of theory in mathematical logic. Beliefs change by means of a change in axioms, notably via new incoming information that christens a new theory; some old axioms (and theorems) may be lost in the reshuffle, whence the nonmonotonicity. But the process of adjudicating between new and old, and of reasoning in general, is not addressed, and there is no associated time-situated aspect.

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

Engelfriet and Treur [18] formalize default reasoning using branching time temporal mod-

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

els, in which an information state at a certain point in time describes what has been derived until that moment. The branching character of the model reflects the fact that at a certain point in time there might be a number of (conflicting) default rules which can be selected to be applied. Our default Modus Ponens is similar to a normal default rule of Reiter's Default Logics [49]. However, as mentioned above, consistency is checked only after a default conclusion is drawn, using the Inheritance axioms. In Engelfriet and Treur's default rules are applied according to what is possible in the future.

In [37] Lespérance and Levesque present a logical theory of knowledge and action that distinguishes indexical and objective knowledge. They formalize the notions of indexical knowledge, time, action, and ability, and provide a formal semantics for the knowledge operator. Although the logic has a notion of "now", this notion does not evolve as the reasoning progresses, making the logic incapable of addressing a key feature of our approach, namely, reasoning about the *process* of reasoning. Another aspect of Lespérance and Levesque's logic left unaddressed is the problem of logical omniscience.

Linear logic [39, 50, inter alia] is a *resource sensitive* logic that provides accounting of process states, events, and resources. Thus, it can be used to study agents with focus of attention. However, our approach to focus of attention by the use of an extra predicate F allows us to stay within a uniform framework for the meta-logic that we use throughout the paper. Linear logic contains additional connectives and modal operators that fall outside of this framework. Also, an important emphasis in linear logic is the number of times that a formula is used in a proof, while our concern in Section 4 is to model a situation where we limit the number of distinct formulas that the agent can deal with at any one time.

6 Conclusions and future work

We have described a formal system for characterizing the ongoing reasoning of families of time-bounded agents, and shown it largely immune to problems of inconsistency that have arisen in other syntactic approaches to knowledge and belief. We introduced knowledge supported models and provided soundness and completeness results. Future work will involve a notion of space-limited focus of attention, planning and acting (in particular, the knowledge prerequisites for actions, as well as common belief). We also will examine more closely the boundaries between consistent and inconsistent syntactic formalizations of belief.

While we have not explicitly addressed intentions and desires in this paper, these are also naturally formalizable in our meta-theory, along with belief. A further aspect of future work will be to complete this formalization and contrast it with other work (eg, [52, 48, 51]) in which the time-situated character of reasoning is generally less explicitly dealt with.

A The Axioms of L_{Me} not given in Section 2.2

The first 5 axioms of $\mathcal{T}$ define the time axioms as for the natural numbers with 0, the successor function +, and the less-than predicate.[7] For example, $A1$ indicates that 0 is the first time period.

$A1$. $\forall\, x\ (0 \neq x+)$
$A2$. $\forall\, x_1 x_2 (x_1+ \;=\; x_2+ \;\rightarrow\; x_1 = x_2)$
$A3$. $\forall\, x(x < x+)$
$A4$. $\forall\, x_1 x_2 x_3 (x_1 < x_2 \wedge x_2 < x_3 \rightarrow x_1 < x_3)$
$A5$. $\forall\, x_1 x_2 (x_1 < x_2 \rightarrow x_2 \not< x_1)$.

The following three axiom schemas of $\mathcal{T}$ yield the inequality of constant symbols that represent different entities. For example, $A6$ yields that $\lceil C(jill, usa) \rceil \neq \lceil SC(jack, jill) \rceil$.

$A6$. $\lceil \alpha_1 \rceil \neq \lceil \alpha_2 \rceil$, for different $\alpha_1, \alpha_2 \in \mathcal{F}_{Ag}$
$A7$. $\lceil \bar{v}_1 \rceil \neq \lceil \bar{v}_2 \rceil$, for different variable sequences $\bar{v}_1$ and $\bar{v}_2$
$A8$. $\lceil \bar{g}_1 \rceil \neq \lceil \bar{g}_2 \rceil$, for different ground term sequences $\bar{g}_1$ and $\bar{g}_2$.

The following ten axioms of $\mathcal{T}$ make the formula forming functions behave in the proper way to form the correct symbols. For example, axiom $A11$ specifies that the *imp* function, applied to two constant symbols representing formulas, gives the constant symbol representing the implication of those formulas, e.g., $imp(\lceil C(u_1, usa) \rceil, \lceil CC(u_1, 01) \rceil) = \lceil C(u_1, usa) \rightarrow CC(u_1, 01) \rceil$. There is a special condition associated with axioms A15-A18 that we note here. The result of the function is $\lceil 0 = 1 \rceil$ (a false formula) if in A15-A17 the condition after "where" is not met or if in A18 the most general common instance does not exist.

$A9$. $neg(\lceil \alpha \rceil) = \lceil \neg \alpha \rceil$ for all $\alpha \in \mathcal{F}_{Ag}$
$A10$. $con(\lceil \alpha_1 \rceil, \lceil \alpha_2)\rceil \;=\; \lceil \alpha_1 \wedge \alpha_2 \rceil$ for all $\alpha_1, \alpha_2 \in \mathcal{F}_{Ag}$
$A11$. $imp(\lceil \alpha_1 \rceil, \lceil \alpha_2 \rceil) = \lceil \alpha_1 \rightarrow \alpha_2 \rceil$, for all $\alpha_1, \alpha_2 \in \mathcal{F}_{Ag}$
$A12$. $defimp(\lceil \alpha_1 \rceil, \lceil \alpha_2 \rceil) = \lceil \alpha_1 \hookrightarrow \alpha_2 \rceil$, for all $\alpha_1, \alpha_2 \in \mathcal{F}_{Su}$
$A13$. $now(t) = \lceil Now(t) \rceil$ for all ground terms $t \in S_t$
$A14$. $k(t_1, t_2, t_3) = \lceil K(t_1, t_2, t_3) \rceil$ for all ground terms $t_1 \in S_n, t_2 \in S_t, and\ t_3 \in S_f$
$A15$. $sub(\lceil \alpha \rceil, \lceil \bar{v} \rceil, \lceil \bar{g} \rceil) = \lceil \alpha(\bar{g}) \rceil$ for all $\alpha \in \mathcal{F}_{Ag}, \bar{v} \in S_v, \bar{g} \in S_g$ where $\bar{g}$ is of the right sort for substitution in α for $\bar{v}$
$A16$. $forall(\lceil \alpha \rceil, \lceil \bar{v} \rceil) = \lceil \forall \bar{v} \alpha \rceil$ for all $\alpha \in \mathcal{F}_{Ag}, \bar{v} \in S_v$ where $\bar{v}$ contains variables free in α
$A17$. $exists(\lceil \alpha \rceil, \lceil \bar{v} \rceil) = \lceil \exists \bar{v} \alpha \rceil$ for all $\alpha \in \mathcal{F}_{Ag}, \bar{v} \in S_v$, where $\bar{v}$ contains variables free in α
$A18$. $mgci(\lceil \alpha_1 \rceil, \lceil \alpha_2 \rceil) =$ the most general common instance of α_1 and α_2 using the subscipt ordering of variables.

Axioms for Awareness:

[7] The time axioms are variants of Peano's axioms for natural numbers. These axioms are in the metatheory: hence the agents need not use them; for instance, the agents may use built-in functions instead.

AR5a.

$$\forall zxy(K(z,x,y) \;\rightarrow\; A(z,x+,y))$$

For example, if an agent knows b at time period t, then it is aware of b at $t+1$.

AR5b.

$$\forall zxy_1y_2y_3(A(z,x,y_1) \wedge y_1 = imp(y_2,y_3) \;\rightarrow\; A(z,x+,y_2) \wedge A(z,x+,y_3))$$

For example, if the agent is aware of $a \rightarrow b$, then in the next time period it will be aware of a and will be aware of b. Note that if it knows $a \rightarrow b$ it will take it two time periods to be aware of a and to be aware of b.

AR5c.

$$\forall zxy_1y_2y_3(A(z,x,y_1) \wedge y_1 = con(y_2,y_3) \;\rightarrow\; A(z,x+,y_2) \wedge A(z,x+,y_3))$$

For example, if the agent is aware of $a \wedge b$, then in the next time period it will be aware of a and will be aware of b.

AR5d.

$$\forall zxy_1y_2(A(z,x,y_1 \wedge y_1 = neg(y_2) \;\rightarrow\; A(z,x+,y_2))$$

For example, if the agent is aware of $\neg a$, then in the next time period it will be aware of a.

AR5e.

$$\forall zxy_1y_2v(A(z,x,y_1) \wedge y_1 = forall(y_2,v) \;\rightarrow\; A(z,x+,y_2))$$

For example, if the agent is aware of $\forall ua$, then in the next time period it will be aware of a.

AR5f.

$$\forall zxy_1y_2v(A(z,x,y_1) \wedge y_1 = exists(y_2,v) \;\rightarrow\; A(z,x+,y_2))$$

For example, if the agent is aware of $\exists ua$, then in the next time period it will be aware of a.

AR9. Need to Know: The following seven axioms involve the *Need* predicate.

AR9a. This axiom states that if an agent is asked a question to which it does not know the answer, the agent will need to know the answer which by *AR9b* will trigger asking itself the question.

$$\forall z_1z_2xy_1vw(Ask(z_1,z_2,x,y_1) \wedge \neg\exists y_2(y_2 = sub(y_1,v,w)) \wedge K(z_2,x,y_2) \;\rightarrow\; Need(z_2,x+,y_1))$$

AR9b.

$$\forall zxy(Need(z,x,y) \;\rightarrow\; Ask(z,z,x+,y))$$

AR9c. $AR9c - e$ state how an agent that needs to know an answer works backwards, based on its knowledge, to find a simpler formula to which it will then need to know the answer.

$$\forall zxy_1y_2y_3y_4y_5v(Need(z,x,y_1) \wedge K(z,x,y_2) \wedge y_2 = forall(imp(y_3,y_4),v) \wedge \\ y_5 = mgci(y_1,y_4) \wedge y_5 \neq \ulcorner 0 = 1 \urcorner \ \rightarrow\ Need(z,x+,y_5))$$

For example, if agent C needs to know Jack's country code at period 6, i.e., $Need(C,6,\ulcorner CC(jack,u_2)\urcorner)$, and it knows that the country code of people who live in the US is 1, i.e., $K(C,6,\ulcorner \forall u_1(Co(u_1,usa) \rightarrow CC(u_1,1)\urcorner)$, then since $CC(jack,1)$ is the most general common instance of the formulas, it will be needed by C, i.e., $Need(C,7,\ulcorner CC(jack,1)\urcorner)$.

AR9d.

$$\forall zxy_1y_2y_3y_4y_5vw(Need(z,x,y_1) \wedge K(z,x,y_2) \wedge y_2 = forall(imp(y_3,y_4),v) \\ \wedge y_1 = sub(y_4,v,w) \wedge y_5 = sub(y_3,v,w) \ \rightarrow\ Need(z,x+,y_5))$$

Returning to the country codes example, suppose that agent C needs to know the country code of Jack at time period 6, i.e., $Need(C,6,\ulcorner CC(jack,u_2)\urcorner)$ and it knows that the country code of a person who lives in the USA is 1, i.e., $K(C,6,\ulcorner \forall u_1(Co(u_1,usa) \rightarrow CC(u_1,1))\urcorner)$. Substituting u_1 with jack and u_2 with usa, it will conclude that it needs to know that Jack's country is the USA, that is, $Need(C,7,\ulcorner Co(jack,usa)\urcorner)$.

AR9e.

$$\forall zxy_1y_2y_3(Need(z,x,y_1) \wedge y_1 = con(y_2,y_3) \ \rightarrow\ Need(z,x+,y_2) \wedge Need(z,x+,y_3))$$

AR9f.

$$\forall zxy_1y_2y_3y_4y_5v(Need(z,x,y_1) \wedge K(z,x,y_2) \wedge y_2 = forall(defimp(y_3,y_4),v) \wedge \\ y_5 = mgci(y_1,y_4) \wedge y_5 \neq \ulcorner 0 = 1 \urcorner \ \rightarrow\ Need(z,x+,y_5))$$

AR9g.

$$\forall zxy_1y_2y_3y_4y_5vw(Need(z,x,y_1) \wedge K(z,x,y_2) \wedge y_2 = forall(defimp(y_3,y_4),v) \\ \wedge y_1 = sub(y_4,v,w) \wedge y_5 = sub(y_3,v,w) \ \rightarrow\ Need(z,x+,y_5))$$

B Details of the Telephoning Example

We illustrate our concepts by giving some details of the telephoning example. Recall that agent B wants to call Jack and knows Jack's local number. Agent B then asks agent A for the area code and agent C for the country code. Agents A and C are able to obtain this information, tell agent B, who then will know Jack's number.

In the deduction we omit all formulas deduced over time that are not needed later. We also show the deduction of agents B and C; agent A's steps are essentially the same as agent C's, except for the use of default implication. We will not explicitly indicate inheritance, since all formulas are inherited. We take $c = 20$ for the Focus predicate and note that a value such as $c = 10$ would not allow the deduction to take place.

We also use additional axioms for agents A, B, and C. We omit the four axioms for agent A that are similar to $C1$ - $C4$ for agent C (see β_1 - β_4 in Section 2). Axioms $A0$ and $C0$ direct any agent to ask agent A for an area code and agent C for a country code. Axioms $B1$ - $B3$ and $C1$ - $C4$ are the observations of the agents and agent B's need to get Jack's phone number. The function f represents the concatenation of the country code, area code, and local number.

B.1 The Axioms

A0. $\forall xyu_1u_2w(Need(z,x,y) \wedge y = sub(\ulcorner AC(u_1,u_2)\urcorner, w) \rightarrow Ask(z,A,x+,y))$
C0. $\forall xyu_1u_2w(Need(z,x,y) \wedge y = sub(\ulcorner CC(u_1,u_2)\urcorner, w) \rightarrow Ask(z,C,x+,y))$
B1. $OBS(B,0,\ulcorner LN(jack, 555-1234)\urcorner)$
B2. $OBS(B,0,\ulcorner \forall u_1u_2u_3u_4(CC(u_1,u_2) \wedge AC(u_1,u_3) \wedge LN(u_1,u_4) \rightarrow PN(u_1,f(u_2,u_3,u_4)))\urcorner)$
B3. $Need(B,0,\ulcorner PN(jack,u)\urcorner)$
C1. $OBS(C,0,\ulcorner \forall u_1u_2u_3(SC(u_2,u_1) \wedge Co(u_1,u_3) \rightarrow Co(u_2,u_3))\urcorner)$
C2. $OBS(C,0,\ulcorner \forall u_1(Co(u_1,usa) \rightarrow CC(u_1,1))\urcorner)$
C3. $OBS(C,0,\ulcorner Co(jill,usa)\urcorner)$
C4. $OBS(C,0,\ulcorner SC(jack,jill)\urcorner)$

B.2 The Deduction

Steps (1)-(8) represent the basic knowledge of agents B and C. In steps (9)-(12) agent B figures out what to ask agents A and C. We show the deduction of agent C in steps (15)-(33). Agents A and C then provide the information about area code and country code in steps

(34)-(35). Agent B puts the information together in steps (36)-(47).

(1)	$K(B, 1, \lceil LN(jack, 555-1234)\rceil)$	$B1, ARF2$
(2)	$F(B, 1, \lceil LN(jack, 555-1234)\rceil, 20)$	$B1, ARF2$
(3)	$K(B, 1, \lceil \forall u_1 u_2 u_3 u_4 (CC(u_1, u_2) \wedge AC(u_1, u_3)$ $\wedge LN(u_1, u_4) \rightarrow PN(u_1, f(u_2, u_3, u_4)))\rceil)$	$B2, ARF2$
(4)	$F(B, 1, \lceil \forall u_1 u_2 u_3 u_4 (CC(u_1, u_2) \wedge AC(u_1, u_3)$ $\wedge LN(u_1, u_4) \rightarrow PN(u_1, f(u_2, u_3, u_4)))\rceil, 20)$	$B2, ARF2$
(5)	$K(C, 1, \lceil \forall u_1 u_2 u_3 (SC(u_2, u_1) \wedge Co(u_1, u_3) \rightarrow Co(u_2, u_3))\rceil)$	$C1, ARF2$
(6)	$K(C, 1, \lceil \forall u_1 (Co(u_1, usa) \rightarrow CC(u_1, 1)\rceil)$	$C2, ARF2$
(7)	$K(C, 1, \lceil Co(jill, usa)\rceil)$	$C3, ARF2$
(8)	$K(C, 1, \lceil SC(jack, jill)\rceil)$	$C4, ARF2$
(9)	$Need(B, 2, \lceil CC(jack, u_2) \wedge AC(jack, u_3) \wedge LN(jack, u_4)\rceil)$	$(3), B3, AR9d$
(10)	$Need(B, 3, \lceil CC(jack, u_2) \wedge AC(jack, u_3)\rceil)$	$(9), AR9e,$
(11)	$Need(B, 4, \lceil CC(jack, u_2)\rceil)$	$(10), AR9e$
(12)	$Need(B, 4, \lceil AC(jack, u_3)\rceil)$	$(10), AR9e$
(13)	$Ask(B, C, 5, \lceil CC(jack, u_2)\rceil)$	$(11), C0$
(14)	$Ask(B, A, 5, \lceil AC(jack, u_3)\rceil)$	$(12), A0$
(15)	$Need(C, 6, \lceil CC(jack, u_2)\rceil)$	$(13), AR9a$
(16)	$Need(C, 7, \lceil CC(jack, 1)\rceil)$	$(6), (15), AR6, AR9c$
(17)	$Need(C, 7, \lceil Co(jack, usa)\rceil)$	$(6), (15), AR6, AR9d$
(18)	$F(C, 8, \lceil \forall u_1 (Co(u_1, usa) \rightarrow CC(u_1, 1)\rceil, 20)$	$(6), (16), AR6, AF4$
(19)	$F(C, 8, \lceil Co(jack, usa)\rceil, 20)$	$(16), AF2$
(20)	$Need(C, 8, \lceil SC(jack, u_1) \wedge Co(u_1, usa)\rceil)$	$(5), (17), AR6, AR9d$
(21)	$Need(C, 9, \lceil SC(jack, u_1)\rceil)$	$(20), AR9e$
(22)	$Need(C, 9, \lceil Co(u_1, usa)\rceil)$	$(20), AR9e$

(23) $F(C, 9, \ulcorner \forall u_1 u_2 u_3 (SC(u_2, u_1) \wedge Co(u_1, u_3)$
$\rightarrow Co(u_2, u_3))\urcorner, 20)$ — $(5), (19), AR6, AF4$
(24) $F(C, 10, \ulcorner SC(jack, u_1)\urcorner, 20)$ — $(21), AF2$
(25) $F(C, 10, \ulcorner Co(u_1, usa)\urcorner, 20)$ — $(22), AF2$
(26) $F(C, 11, \ulcorner SC(jack, jill)\urcorner, 20)$ — $(8), (24), AR6, AF3$
(27) $F(C, 11, \ulcorner Co(jill, usa)\urcorner, 20)$ — $(7), (25), AR6, AF3$
(28) $K(C, 12, \ulcorner SC(jack, jill) \wedge Co(jill, usa)\urcorner)$ — $(7), (8), (26), (27), AR6, ARF3c$
(29) $F(C, 12, \ulcorner SC(jack, jill) \wedge Co(jill, usa)\urcorner, 19)$ — $(7), (8), (26), (27), AR6, ARF3c$
(30) $K(C, 13, \ulcorner Co(jack, usa)\urcorner)$ — $(5), (23), (28), (29), AR6, ARF4a$
(31) $F(C, 13, \ulcorner Co(jack, usa)\urcorner, 18)$ — $(5), (23), (28), (29), AR6, ARF4a$
(32) $F(C, 13, \ulcorner \forall u_1 (Co(u_1, usa) \rightarrow CC(u_1, 1)\urcorner, 15)$ — $(18), AF1$
(33) $K(C, 14, \ulcorner CC(jack, 1)\urcorner)$ — $(6), (30), (31), (32), AR6, ARF4a$
(34) $Tell(C, B, 15, \ulcorner CC(jack, 1)\urcorner)$ — $(13), (33)$
(35) $Tell(A, B, 15, \ulcorner AC(jack, 301)\urcorner)$ — $agent\ \ A's\ \ reasoning$
(36) $K(B, 16, \ulcorner CC(jack, 1)\urcorner)$ — $(34), ARF7$
(37) $F(B, 16, \ulcorner CC(jack, 1)\urcorner, 20)$ — $(34), ARF7$
(38) $K(B, 16, \ulcorner AC(jack, 301)\urcorner)$ — $(35), ARF7$
(39) $F(B, 16, \ulcorner AC(jack, 301)\urcorner, 20)$ — $(35), ARF7$
(40) $K(B, 17, \ulcorner CC(jack, 1) \wedge AC(jack, 301)\urcorner)$ — $(36), (37), (38), (39), ARF3c$
(41) $F(B, 17, \ulcorner CC(jack, 1) \wedge AC(jack, 301)\urcorner, 19)$ — $(36), (37), (38), (39), ARF3c$
(42) $F(B, 17, \ulcorner LN(jack, 555 - 1234)\urcorner, 4)$ — $(1), (2), AR6, AF1$
(43) $K(B, 18, \ulcorner CC(jack, 1) \wedge AC(jack, 301) \wedge$
$LN(jack, 555 - 1234)\urcorner)$ — $(1), (40), (41), (42), AR6, ARF3b$
(44) $F(B, 18, \ulcorner CC(jack, 1) \wedge AC(jack, 301) \wedge$
$LN(jack, 555 - 1234)\urcorner 18)$ — $(1), (40), (41), (42), AR6, ARF3b$
(45) $F(B, 18, \ulcorner \forall u_1 u_2 u_3 u_4 (CC(u_1, u_2) \wedge AC(u_1, u_3)$
$\wedge LN(u_1, u_4) \rightarrow PN(u_1, f(u_2, u_3, u_4)))\urcorner, 3)$ — $(4), AF1$
(46) $K(B, 19, \ulcorner PN(jack, f(1, 301, 555 - 1234))\urcorner)$ — $(3), (43), (44), (45), ARF4a$
(47) $F(B, 19, \ulcorner PN(jack, f(1, 301, 555 - 1234))\urcorner, 17)$ — $(3), (43), (44), (45), ARF4a$

References

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

[2] R. Alur and T. A. Henzinger. Logics and Models of Real Time: A Survey. In J. W. de Bakker, C. Huizing, W. P. de Roever, and G. Rozenberg, editors, *Proceedings of Real-Time: Theory in Practice*, volume 600 of *Lecture Notes in Computer Science*, pages 74–106. Springer-Verlag, Berlin, Germany, June 1992.

[3] A. B. Baker and Y. Shoham. Nonmonotonic Temporal reasoning. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*. Oxford University Press, 1995.

[4] Howard Barringer, Michael Fisher, Dov Gabbay, and Anthony Hunter. Meta-reasoning in executable temporal logic. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 40–49, San Mateo, CA, USA, April 1991. Morgan Kaufmann Publishers.

[5] J. van Benthem. *The logic of time*. Kluwer Academic Publishers, 1991.

[6] J. van Benthem. Temporal logic. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, pages 241–350. Oxford University Press, 1995.

[7] A. Brogi and F. Turini. Metalogic for knowledge representation. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 61—69, San Mateo, CA, USA, April 1991. Morgan Kaufmann Publishers.

[8] S. Ceri, G. Gottlob, and L. Tanca. *Logic Programming and Databases*. Springer-Verlag, 1990. Sec. 11.2.

[9] L. Chittaro and A. Montanari. Editorial: Temporal representation and reasoning. *Annals of Mathematics and Artificial Intelligence*, 22:1–4, 1998.

[10] P. Cohen and H. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:263–310, 1990.

[11] J. de Rivières and H. Levesque. The consistency of syntactical treatments of knowledge (how to compile quantificational modal logics into classical FOL. *Computational Intelligence*, 4:31–41, 1988.

[12] Juergen Dix, V. S. Subrahmanian, and George Pick. Meta agent programs. Technical Report CS-TR-3927, University of Maryland, College Park, September 1998.

[13] C. Dixon, M. Fisher, and M. Wooldridge. Resolution for temporal logics of knowledge. *Journal of Logic and Computation*, 8(3):345–372, 1998.

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

[15] J. Elgot-Drapkin, M. Miller, and D. Perlis. Memory, reason, and time: the step-logic approach. In R. Cummins and J. Pollock, editors, *Philosophy and AI: Essays at the Interface*. MIT Press, 1991.

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

[17] E. Allen Emerson. Temporal and modal logic. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, pages 995–1072. Elsevier Science Publishers, Amsterdam, The Netherlands, 1990.

[18] J. Engelfriet and J. Treur. Semantics for default logic based on specific branching time models. In *Proc. of ECAI96*, pages 60–64, 1996.

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

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

[21] F. Giunchiglia and L. Serafini. Multilanguage hierarchical logics (or: how we can do without modal logics). *Artificial Intelligence*, 65:29–70, 1994.

[22] F. Giunchiglia and P. Traverso. A Metatheory of a Mechanized Object Theory. *Artificial Intelligence*, 80(2):197–241, 1996.

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

[24] P. Gmytrasiewicz, E. H. Durfee, and D. Wehe. A logic of knowledge and belief for recursive modeling: Preliminary report. In *Proc. of AAAI-92*, pages 628–634, California, 1992.

[25] J. Y. Halpern. Using reasoning about knowledge to analyze distributed systems. *Annual Review of Computer Science*, 2, 1987.

[26] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for model logics of knowledge and belief. *Artificial Intelligence*, 54(3):319–379, 1992.

[27] J. Y. Halpern and M. Y. Vardi. The complexity of reasoning about knowledge and time 1: lower bounds. *Journal of Computer and System Sciences*, 38(1):195–237, 1989.

[28] Eric Horvitz and Adrian Klein. Reasoning, metareasoning, and mathematical truth: Studies of theorem proving under limited resources. In Besnard, Philippe and Steve Hanks, editors, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI'95)*, pages 306–314, San Francisco, CA, USA, August 1995. Morgan Kaufmann Publishers.

[29] Z. Huang, M. Masuch, and L. Polos. ALX, an action logic for agents with bounded rationality. *Artificial Intelligence*, 82(1-2):75–127, 1996.

[30] G. Hunter. *Metalogic: An introduction to Metatheory of Standard First Order Logic*. University of California Press, Berkely and Los Angeles, 1971.

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

[32] R. Kowalski. Using metalogic to reconcile reactive with rational agents. In K. Apt and F. Turini, editors, *Meta-Logics and Logic Programming*. MIT Press, 1995.

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

[34] S. Kraus and V.S. Subrahmanian. Multiagent reasoning with probability, time and beliefs. *International Journal of Intelligent Systems*, 10:459–499, 1995.

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

[36] H. Lee, J. Tannock, and J. S. Williams. Logic-based reasoning about actions and plans in artificial intelligence. *The Knowledge Engineering Review*, 11(2):91–105, 1993.

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

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

[39] P. Lincoln Linear logic *ACM SIGACT News*, 23(2):29-37, 1992.

[40] Z. Manna and A. Pnueli. *Temporal Logic of Reactive and Concurrent Systems*. Addison Wesley, 1992.

[41] John McCarthy. Circumscription, a form of non monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

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

[43] M. Miller and D. Perlis. Presentations and this and that: logic in action. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society*, Boulder, Colorado, 1993.

[44] R Montague. Syntactical treatments of modality, with corollaries on reflection prinnciples and finite axiomatizability. In *Modal and Many-Valued Logics (Acta Philosophica Fennica, vol. 16)*. Academic Bookstore, Helsinki, 1963. Reprinted in R. Montague (1974). *Formal Philosophy*, New Haven, pp. 286-302.

[45] M. Morreau and S. Kraus. Syntactical treatments of propositional attitudes. *Artificial Intelligence journal*, 106:161–177, 1998.

[46] M. Nirkhe, S. Kraus, M. Miller, and D. Perlis. How to (plan to) meet a deadline between *now* and *then*. *Journal of logic computation*, 7(1):109–156, 1997.

[47] A. Rao and M. P Georgeff. Asymmetry thesis and side-effect problems in linear-time and branching-time intention logics. In *Proc. of IJCAI-91*, pages 498–504, Australia, 1991.

[48] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In James Allen, Richard Fikes, and Eric Sandewall, editors, *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR-91)*, pages 473–484, Cambridge, MA, May 1991.

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

[50] A. Scedrov. Linear Logic and Computation: A Survey. In H. Schwichtenberg, editor, *Proof and Computation, Proceedings Marktoberdorf Summer School 1993*. NATO Advanced Science Institutes, Series F, Springer-Verlag, Berlin, 1994.

[51] Yoav Shoham. Agent-oriented programming. *Artificial Intelligence*, 60:51–92, 1993.

[52] Munindar Singh. A semantics for speech acts. *Annals of Mathematics and Artificial Intelligence*, 8:47–71, 1993.

[53] B. Thomas, Y. Shoham, A. Schwartz, and S. Kraus. Preliminary Thoughts on an Agent Description Language. *International Journal of Intelligent Systems*, 6(5):497–508, August 1991.

[54] R. Thomason. A note on syntactical treatments of modality. *Synthese*, 44:391–395, 1980.

[55] W. van der Hoek, B. van Linder, and J.-J. Ch. Meyer. An integrated modal approach to rational agents. In M. Wooldridge and A. Rao, editors, *Foundations of Rational Agency*, pages 37–75. Kluwer, Dordrecht, 1999.

[56] L. Vila. A Survey on Temporal Reasoning in Artificial Intelligence. *AI Communications*, 7(1):4–28, March 1994.

[57] M. J. Wooldridge and N. R. Jennings. Agent theories, architectures and languages: A survey. In *Intelligent Agents*, Lecture Notes in Artificial Intelligence No. 890, pages 1–39. Springer-Verlag, 1995.