Executable Temporal Logic for Nonmonotonic Reasoning[*]

Joeri Engelfriet and Jan Treur

Free University Amsterdam
Department of Mathematics and Computer Science
De Boelelaan 1081a, 1081 HV Amsterdam,
The Netherlands
tel. +31 20 4447756, fax. +31 20 4447653
email: {joeri, treur}@cs.vu.nl

ABSTRACT: In this paper we view nonmonotonic reasoning as a (special kind of) process. As temporal logic is a common formalism to specify and derive properties of processes, we introduce a variant of temporal logic as a general specification language for reasoning processes. We show that it is possible to execute finite specifications in this language, which leads to executability of a large class of finite nonmonotonic reasoning processes.

1 Introduction

Nonmonotonic reasoning is used when an agent wants to draw conclusions about the world based on its partial knowledge of the world. Classical logic usually permits too little conclusions for the agent to base its actions upon. Using nonmonotonic reasoning the agent can extend its partial knowledge to a set of hypothetical beliefs. In general there will be more than one set of beliefs which is a reasonable extension of the (sure) knowledge. Starting from the set of initial facts the agent can construct all or one of the possible belief sets using nonmonotonic inference rules. It may then commit to one of these views (and, if needed, change its commitment later), or it may focus on the intersection of these views.

---

[*] Part of this work has been supported by SKBS and the ESPRIT III Basic Research project 6156 DRUMS II.

The construction of a belief set is a stepwise process. The reasoning agent starts with a set of initial facts to which it applies a number of nonmonotonic inference rules to obtain a larger set of beliefs or conclusions. To this set it may again apply nonmonotonic rules to arrive at a next set of beliefs. Viewed in this way, a nonmonotonic reasoning process is (just) a special kind of process. Temporal logic is recognized as a valuable tool for specifying processes and reasoning about their properties. For each class of processes, a different version of temporal logic might be most appropriate. As we are interested in specifying reasoning processes, we will introduce a version of temporal logic suited to describe reasoning processes. In particular, since a state in a reasoning process should contain the beliefs or conclusions the reasoning agent has arrived at in that state, we propose a temporal epistemic logic as a general specification language for nonmonotonic reasoning processes.

Using temporal logic for specifying processes allows verification and validation of a process. The benefit of using an executable temporal logic is that processes specified in this logic can also be directly executed, so that reasoning about a process and actually executing it, can be done in the same language. It will be shown that specifications in our version of temporal logic can be executed, leading to a general algorithm for executing nonmonotonic reasoning processes. Although there are similarities with existing executable temporal logics (see [Ga89], [Fi94], [FO95], [OM94]), there are some differences as well. The complexity of this algorithm is complete in the complexity class where many nonmonotonic formalisms reside.

In Section two we will formalize the reasoning processes we want to specify using reasoning frames and in Section three we introduce our temporal specification language for these reasoning frames. In Section four we will show that any finite specification is executable, leading to executability of finite (nonmonotonic) reasoning processes. Section five contains conclusions and suggestions for further research.

2 Reasoning Frames

We will start by describing semantically the reasoning processes we would like to specify, reason about, and execute. Classical (propositional) logic lies at the base of most nonmonotonic formalisms, so we will assume we have a propositional language, $\mathbf{L}$, its corresponding set of models, $\mathbf{Mod}$, and the (semantic) consequence relation $\models \subseteq \mathbf{Mod} \times \mathbf{L}$. Furthermore, for a set $\mathbf{X} \subseteq \mathbf{L}$, we define the models of $\mathbf{X}$: $\mathbf{Mod(X)} = \{ \mathbf{m} \in \mathbf{Mod} \mid \mathbf{m} \models \varphi \text{ for all } \varphi \in \mathbf{X}\}$; the consequences of a set $\mathbf{X} \subseteq \mathbf{L}$ are defined by $\mathbf{Cn(X)} = \{ \varphi \in \mathbf{L} \mid \mathbf{Mod(X)} \subseteq \mathbf{Mod}(\varphi)\}$. For a subset $\mathbf{K}$ of $\mathbf{Mod}$, the *theory* of $\mathbf{K}$ is defined by

$\mathbf{Th(K)} = \{ \varphi \mid m \vDash \varphi \text{ for all } m \in K \}$. A set of models $\mathbf{K}$ is called *closed* if $\mathbf{K} = \mathbf{Mod(Th(K))}$, or equivalently, if $\mathbf{K}$ is the set of models of a theory.

Intuitively, the path from initial set of formulae to final conclusions can be seen as the behaviour of a reasoning process which starts with the initial formulae, then makes some nonmonotonic inferences to arrive at a new state, again makes some inferences, et cetera, possibly ad infinitum. The final conclusions of such a process can be seen as the union of all conclusions drawn at all stages. A formalization of such reasoning behaviour would have to describe which formulae have been derived at each stage. We will formalize the notion of state in a reasoning process semantically by the notion of an information state, which should describe the formulae which the agent has concluded (or which the agent believes) in that state:

Definition 2.1 (Information State)

a) An *information state* $\mathbf{M}$ is a non-empty closed set of propositional models, that is, there is a consistent theory of which it is the model class. The truth of a propositional formula α in such a state is defined by:

$\mathbf{M} \vDash \alpha \quad \Leftrightarrow \quad m \vDash \alpha$ for each $m \in \mathbf{M}$

b) The *theory* of an information state $\mathbf{M}$ is defined by:

$\mathbf{Th(M)} = \{ \alpha \mid \alpha$ is a propositional formula and $\mathbf{M} \vDash \alpha \}$

c) The *refinement ordering* $\leq$ on information states is defined by:

$\mathbf{M}_1 \leq \mathbf{M}_2 \quad \Leftrightarrow \quad \mathbf{M}_2 \subseteq \mathbf{M}_1$

d) The *set of all information states* is denoted by $\mathbf{IS}$.

So the theory of an information state contains the formulae the agent believes in that state. Using these notions we can now define a reasoning trace as a specific type of sequence of such information states. We assume that the reasoning at least is conservative: conclusions constructed in earlier stages will be persistent during the reasoning trace. Although many reasoning processes are finite (after a while, no new nonmonotonic inferences can be made), for simplicity we will only consider infinite traces. A process which is finite will give rise to a trace which is constant after a certain point. In particular, if the propositional signature is finite, all traces will be constant eventually.

Definition 2.2 (Reasoning Trace and Limit Model)

a) A *reasoning trace* $\mathcal{M}$ is a function from the set of natural numbers to $\mathbf{IS}$ such that for all $i \in \mathbb{N}$:

(i) $\mathcal{M}_i \leq \mathcal{M}_{i+1}$

(ii) $\mathcal{M}_i = \mathcal{M}_{i+1} \quad \Rightarrow \quad \mathcal{M}_i = \mathcal{M}_j$ for all $j \geq i$.

b) The *refinement ordering* $\leq$ on reasoning traces is defined by:

$\mathcal{M} \leq \mathcal{N} \quad \Leftrightarrow \quad \mathcal{M}_i \leq \mathcal{N}_i$ for all $i \in \mathbb{N}$

c) The *limit model* , $\mathbf{lim}\,\mathcal{M}$ of a reasoning trace $\mathcal{M}$ is defined by

$$\lim \mathcal{M} = \bigcap_{i=0}^{\infty} \mathcal{M}_i$$

d) A reasoning trace $\mathcal{M}$ is sometimes denoted by $(\mathcal{M}_i)_{i \in \mathbb{N}}$.

e) A reasoning trace is called *finitely generated* if each $\mathbf{Th}(\mathcal{M}_i)$ is finitely generated over $\mathbf{Th}(\mathcal{M}_0)$, i.e., if $\mathbf{Th}(\mathcal{M}_i) = \mathbf{Cn}(\mathbf{Th}(\mathcal{M}_0) \cup \{\alpha_i\})$ for some formula α_i.

Since we assume a countable language $\mathbf{L}$, each theory can be approximated by a chain of finitely generated theories. Therefore:

Proposition 2.3

a) For any reasoning trace its limit is an information state.

b) Any information state is the limit model of a finitely generated reasoning trace.

A (nonmonotonic) type of reasoning can now be described by giving its intended reasoning traces. Given a set of initial formulae, there may of course be several traces leading to different conclusion sets. We do, however, assume that the reasoning is deterministic in the sense that given the set of initial formulae and the final conclusion set, the trace between them is uniquely determined. This can be explained in the sense that at each stage of the reasoning process all conclusions that possibly can be drawn, actually are drawn in the next step. Moreover, we do not allow two distinct traces leading to limit models of which one is a refinement of the other (non-inclusiveness of traces).

Definition 2.4 (Reasoning Frame)

a) A *reasoning frame* is a tuple $(\mathbf{L}, \mathbf{Mod}, \models, \mathcal{T})$ with $\mathcal{T}$ a set of reasoning traces such that for all $\mathcal{M}$ and $\mathcal{N}$ in $\mathcal{T}$: if $\mathcal{M}_0 = \mathcal{N}_0$ and $\lim \mathcal{M} \leq \lim \mathcal{N}$ then $\mathcal{M} = \mathcal{N}$.
For shortness, sometimes we also call $\mathcal{T}$ by itself a reasoning frame.

b) If for all sets of formulae $\mathbf{X}$ there exists a trace $\mathcal{M}$ in $\mathcal{T}$ such that $\mathbf{Th}(\mathcal{M}_0) = \mathbf{Cn}(\mathbf{X})$ then $\mathcal{T}$ is called a *complete* reasoning frame. Otherwise it is called *partial*. In the complete case the mapping sending every $\mathbf{X} \subseteq \mathbf{L}$ to the set of traces $\{ \mathcal{M} \in \mathcal{T} \mid \mathbf{Th}(\mathcal{M}_0) = \mathbf{Cn}(\mathbf{X}) \}$ is called the *reasoning trace operator* related to the reasoning frame $\mathcal{T}$.

As an example we consider default logic:

Example 2.5 (Default Logic)

Let D be a set of defaults. For $X \subseteq L$ let $\mathbb{E}(<X, D>)$ denote the set of (Reiter) extensions of the default theory $<X, D>$. For a given X and $E \in \mathbb{E}(<X, D>)$ the following reasoning trace $\mathcal{M}$ can be associated in a canonical manner:

$$\mathcal{M}_i = \mathrm{Mod}(E_i)$$

with $E_0 = \mathrm{Cn}(X)$, and for all $i \geq 0$

$$E_{i+1} = \mathrm{Cn}(E_i \cup \{ \omega \mid (\alpha : \beta_1, \ldots, \beta_n) / \omega \in D \text{ is applicable at level } i \})$$

where a default $(\alpha : \beta_1, \ldots, \beta_n) / \omega \in D$ is applicable at level i if

$$\alpha \in E_i \text{ and } \neg\beta_1 \notin E, \ldots, \neg\beta_n \notin E$$

Note that this is a trace definition based on the given set of defaults D. If we take the set of all these reasoning traces for $E \in \mathbb{E}(<X, D>)$ we get a partial reasoning frame associated to $<X, D>$. Taking the union of these frames for all $X \subseteq L$ we get a complete reasoning frame associated with the set of defaults D.

3 A Temporal Specification Language for Reasoning Frames

A simple observation allows us to find a natural description language for reasoning traces: the steps in a reasoning trace can be viewed as temporal steps. This means that the transition from an information state to the next one (as the result of a number of inference steps) can be seen as a temporal one. In this view a trace is a temporal model based on the set of natural numbers as the flow of time. An obvious candidate for describing these models is (some variant of) temporal epistemic logic. However, the full (tense) logic will turn out to be not completely appropriate: on the one hand it can describe models which are not traces but on the other hand it is not powerful enough. Therefore we will introduce a limited fragment of infinitary tense epistemic logic.

Definition 3.1 (Temporal Epistemic Model)

i) A *temporal epistemic model* is a function $\mathcal{M} : \mathbb{N} \rightarrow IS$.

ii) A temporal epistemic model $\mathcal{M}$ is *conservative* if $\mathcal{M}_i \leq \mathcal{M}_{i+1}$ for all $i \in \mathbb{N}$.

iii) The *refinement ordering* $\leq$ on temporal epistemic models is defined by:

$$\mathcal{M} \leq \mathcal{N} \Leftrightarrow \mathcal{M}_0 = \mathcal{N}_0 \text{ and for all } i : \mathcal{M}_i \leq \mathcal{N}_i .$$

iv) The *limit model* , $\lim \mathcal{M}$, of a temporal epistemic model $\mathcal{M}$ is the information state defined by

$$\lim \mathcal{M} = \bigcap_{i=0}^{\infty} \mathcal{M}_i$$

v) A temporal epistemic model $\mathcal{M}$ is sometimes denoted by $(\mathcal{M}_i)_{i \in \mathbb{N}}$.

Note that the notion of a temporal epistemic model as such is close to the notion of a reasoning trace: any reasoning trace can be considered a conservative temporal model with the property that if it stabilises one step, then it stabilises forever. However, in a temporal epistemic model temporal operators and temporal formulae are interpreted. The temporal language we will use is a modal propositional language with the operators $\mathbf{F}, \mathbf{G}, \mathbf{C}, \mathbf{H_0}$ and infinite conjunctions and disjunctions. An epistemic operator is implicit in these temporal operators: for instance $\mathbf{F}\varphi$ means that sometimes in the future the reasoning agent will know (or have derived) φ. Furthermore, $\mathbf{C}\varphi$ stands for "currently the agent knows (has derived) φ". The truth of a formula φ in a temporal epistemic model $\mathcal{M}$ at time point $\mathbf{i}$, denoted as $(\mathcal{M}, \mathbf{i}) \models \varphi$ is defined inductively:

Definition 3.2 (Temporal Interpretation)

a) For a formula φ without temporal operators:

$(\mathcal{M}, \mathbf{i}) \models \varphi \quad \Leftrightarrow \quad \mathcal{M}_{\mathbf{i}} \models \varphi$

b) For temporal formula φ:

$(\mathcal{M}, \mathbf{i}) \models \mathbf{F}\varphi \quad \Leftrightarrow \quad$ there exists $\mathbf{j} \in \mathbb{N}$, $\mathbf{j} > \mathbf{i}$ such that $(\mathcal{M}, \mathbf{j}) \models \varphi$

$(\mathcal{M}, \mathbf{i}) \models \mathbf{G}\varphi \quad \Leftrightarrow \quad$ for all $\mathbf{j} \in \mathbb{N}$ with $\mathbf{j} > \mathbf{i}$: $(\mathcal{M}, \mathbf{j}) \models \varphi$

$(\mathcal{M}, \mathbf{i}) \models \mathbf{C}\varphi \quad \Leftrightarrow \quad (\mathcal{M}, \mathbf{i}) \models \varphi$

$(\mathcal{M}, \mathbf{i}) \models \mathbf{H_0}\varphi \quad \Leftrightarrow \quad (\mathcal{M}, \mathbf{0}) \models \varphi$

$(\mathcal{M}, \mathbf{i}) \models \neg\varphi \quad \Leftrightarrow \quad$ it is not the case that $(\mathcal{M}, \mathbf{i}) \models \varphi$

c) For a set $\mathbf{A}$ of temporal formula:

$(\mathcal{M}, \mathbf{i}) \models \bigwedge \mathbf{A} \quad \Leftrightarrow \quad$ for all $\varphi \in \mathbf{A}$: $(\mathcal{M}, \mathbf{i}) \models \varphi$

d) A formula φ is true in a model $\mathcal{M}$, denoted $\mathcal{M} \models \varphi$, if for all $\mathbf{i} \in \mathbb{N}$: $(\mathcal{M}, \mathbf{i}) \models \varphi$

e) A set of formulae $\mathbf{T}$ is true in a model $\mathcal{M}$, denoted $\mathcal{M} \models \mathbf{T}$, if for all $\varphi \in \mathbf{T}$, $\mathcal{M} \models \varphi$. We call $\mathcal{M}$ a model of $\mathbf{T}$.

Furthermore the connectives $\vee$ and $\rightarrow$ are introduced as the usual abbreviations. Note that for a propositional formula φ, $\mathbf{C}\neg\varphi$ is not equivalent to $\neg\mathbf{C}\varphi$: $\mathbf{C}\neg\varphi$ means that the agent currently knows $\neg\varphi$, whereas $\neg\mathbf{C}\varphi$ means that the agent currently does not know φ.

The temporal language we have just introduced is still too powerful: we want to use only a fragment to describe models which can be seen as reasoning traces. So the question is: which fragment is appropriate for nonmonotonic reasoning? As steps in a reasoning process are taken whenever a number of (nonmonotonic) inference steps is used, it seems that temporal rules should prescribe taking the equivalent of (nonmonotonic) inference steps in the temporal model. So the next question is what the nature is of a generalized nonmonotonic inference step that a reasoning process can execute. A general format of (temporal) inference rules is $\alpha \rightarrow G\beta$ where α is a condition for the inference, and β is its conclusion: if the condition α is fulfilled, the conclusion β can be drawn, and will be known henceforth. The condition α may of course include reference to the initial facts and the facts which have been derived earlier (and therefore are still true at the present moment). But in nonmonotonic reasoning there is often also a kind of global consistency check. In default logic for instance, a rule is applicable if certain formulae, called the justifications, are consistent with the final outcome of the reasoning process (called an extension), which means they should be consistent throughout the entire reasoning process. Consistency of a formula usually means that its negation should not be true. Therefore we also allow conditions which state that a certain formula should never (in the future of the reasoning process) be true.

Definition 3.3 (Reasoning Theories)

a) A formula is called a *(nonmonotonic) reasoning formula* if it is of the form

$$\alpha \wedge \beta \wedge \varphi \wedge \psi \rightarrow G\gamma,$$

where

$\alpha = \bigwedge\{ H_0\varepsilon \mid \varepsilon \in A \}$ for a set of propositional formulae A.

$\beta = \bigwedge\{ \neg H_0\delta \mid \delta \in B \}$ for a set of propositional formulae B.

$\varphi = \bigwedge\{ \neg F\theta \mid \theta \in D \}$ for a set of propositional formulae D.

$\psi = \bigwedge\{ C\zeta \mid \zeta \in F \}$ for a set of propositional formulae F.

γ is a propositional formula.

A reasoning formula is called *finitary* if all sets of formulae involved are finite; otherwise it is called *infinitary*.

b) A set Th of reasoning formulae is called a *theory of reasoning*. It is called *finitary* if all its elements are; otherwise it is called *infinitary*.

So a reasoning formula prescribes the truth of a formula in the future based on knowledge of initial facts, truth of current facts and consistency of facts in the future (if $\neg F\theta$ is true, then θ is never known in the future, so $\neg\theta$ remains consistent with the agent's beliefs throughout the future).

Definition 3.4 (Conservativity)

The theory $C = \{ C\alpha \rightarrow G\alpha \mid \alpha \text{ a propositional formula} \}$ is a theory of reasoning expressing conservativity of temporal models.

A theory of reasoning prescribes that certain formulae have to be known (derived) in the future, analogously to inference steps. But what about facts which become known at a point in time spontaneously, that is without any inference rule prescribing their truth? We should have a way to make sure that this does not happen: we want the models to have minimal information in the sense that nothing becomes known if there are no rules saying so (see also [Sh88], [KLM90]). This leads to the following notion of minimal models:

Definition 3.5 (Minimal Temporal Models)

A temporal epistemic model $\mathcal{M}$ is called a *minimal model* of a theory **Th** if it is a model of **Th** and for any model $\mathcal{N}$ of **Th**, if $\mathcal{N} \leq \mathcal{M}$ then $\mathcal{N} = \mathcal{M}$.

A minimal model of a theory is a model for which there are no smaller models of the theory, so they contain a minimum of information.

Given the fragment of temporal logic we have defined, a natural property to investigate is whether this fragment is suited for describing reasoning traces.
From now on we will assume that any theory includes the theory **C** described in Definition 3.4. The first result is that all minimal models of any theory are reasoning traces. On the other hand, any reasoning frame is the set of minimal models of a theory of reasoning:

Theorem 3.6

a) For any theory of reasoning **Th** its minimal models constitute a (partial) reasoning frame.

b) For any (partial) reasoning frame $\mathcal{F}$ there exists a theory of reasoning whose minimal models are exactly $\mathcal{F}$.

If there is only a finite number of atomic proposition symbols, then this theory of reasoning can be taken finite and finitary.

Now we give an explicit theory of reasoning for the reasoning frame associated with a default theory as described in example 2.5 (see [ET93]):

Example 3.7 (Default Logic)

Given a default theory $\mathbf{\Delta} = <\mathbf{X}, \mathbf{D}>$, define its theory of reasoning $\mathbf{T(\Delta)}$ by:

$\mathbf{T(\Delta)} = \{\, \mathbf{C}\alpha \wedge \neg\mathbf{F}\neg\beta \rightarrow \mathbf{G}\gamma \mid (\alpha,\beta)/\gamma \in \mathbf{D} \,\} \cup \{\, \mathbf{C}\alpha \mid \alpha \in \mathbf{X} \,\} \cup \mathbf{C}$.

Then the minimal models of $\mathbf{T(\Delta)}$ constitute the partial reasoning frame associated to $\mathbf{\Delta}$. Each minimal model of $\mathbf{T(\Delta)}$ corresponds with an extension of $\mathbf{\Delta}$.

4 Executing Theories of Reasoning

In order to execute a theory of reasoning we interpret its temporal rules as inference rules. If the condition of such a rule is met, we introduce its conclusion at the next step. The condition of the rule pertaining to the initial facts and the present can be checked in a straightforward manner. The only problem are the consistency checks. The way to deal with them is to: a) either assume they will be met and add the conclusion. In this case we will have to check at all later steps that they are still met. b) Otherwise we assume they are not met and do not add the conclusion. In this case the consistency must be violated at some later time. If this does not happen than the execution is not correct. If the theory of reasoning is infinite, it is in general not possible at any point in time to be sure we are executing correctly.

Notice that a reasoning formula is not of the form "PAST implies FUTURE", a form used often for executable temporal logic (see [Ga89]). Of course we could move the consistency checks to the right of the implication. Using the notation of definition 3.3 we would obtain a rule

$$\alpha \wedge \beta \wedge \psi \rightarrow G\gamma \vee \bigvee\{ F\theta \mid \theta \in D \}$$

which is clearly in the desired format. To execute this rule if the conditions are met, we could either introduce γ at the next moment in time, or introduce one of the elements of D at any future time point. But this is not the correct way of executing such a formula, since the consistency checks are meant to be *declarative* and not *imperative*. So instead of the slogan "declarative past implies imperative future" ([Ga89]) we use the slogan "declarative past and future imply imperative future".

We will now informally describe the general algorithm for executing a theory of reasoning Th. We assume that we have a set of initial facts.

Algorithm

1. Mark all rules as unused, set t to 0.
2. If the current facts are contradictory, backtrack to the previous time point.
3. Check all constraints **never_true(θ)**. If θ is entailed by the current facts, backtrack to the previous time point.
4. For each unused rule

$$\alpha \wedge \beta \wedge \varphi \wedge \psi \rightarrow G\gamma,$$

where

$\alpha = \bigwedge\{ H_0\varepsilon \mid \varepsilon \in A \}$ for a set of propositional formulae A.

$\beta = \bigwedge\{ \neg H_0\delta \mid \delta \in B \}$ for a set of propositional formulae B.

$\varphi = \bigwedge\{ \neg F\theta \mid \theta \in D \}$ for a set of propositional formulae D.

$\psi = \bigwedge\{ C\zeta \mid \zeta \in F \}$ for a set of propositional formulae F.

```
            $\gamma$ is a propositional formula.
    do:
        If all formulae in $\mathbf{F}$ are entailed by the current facts, and
           all formulae in $\mathbf{A}$ are entailed by the initial facts, and
           all formulae in $\mathbf{B}$ are not entailed by the initial facts
        then this rule is applicable: mark this rule as used and do either of:

        * introduce $\mathbf{next(\gamma)}$ and constraints $\mathbf{never\_true(\theta)}$ for each $\theta \in \mathbf{D}$.
        * introduce constraints $\mathbf{sometimes\_true(D)}$.

    If we backtracked to this time point, make a choice for all the applicable rules at
    this time point which has not been made before. If this is not possible, backtrack
    one more step. If this is not possible ($\mathbf{t = 0}$) then abort.
    5. $\mathbf{t := t + 1}$; for each formula $\mathbf{next(\gamma)}$, introduce $\gamma$ and delete $\mathbf{next(\gamma)}$;
    If there are no formulae $\mathbf{next(\gamma)}$, check for each constraint $\mathbf{sometimes\_true(D)}$
    whether some $\theta \in \mathbf{D}$ is entailed by the current facts; if not, backtrack one step.
    goto 2.
```

If for all constraints $\mathbf{sometimes\_true(D)}$, some $\theta \in \mathbf{D}$ is entailed by the facts at some point in time, the execution of the algorithm is called *correct.*

As mentioned before, if at each step new formulae are added, we can never be sure during execution, if the execution will be correct. However in case the theory of reasoning is finite, there will be a point in time when no new facts are added: then the execution will always be correct. This is always the case when the propositional language contains a finite number of atoms: then there is a finite number of non-equivalent propositional formulae; if this number is $\mathbf{n}$, then after $\mathbf{n}$ steps of the algorithm, no new facts can be added. For correct executions we have the following:

Theorem 4.1

Let a theory of reasoning $\mathbf{Th}$ be given. For an execution of the algorithm for $\mathbf{Th}$, let $\mathbf{T_i}$ denote the set of propositional formulae derived at time $\mathbf{t = i}$. Define a reasoning trace $(\mathcal{M}_i)_{i \in \mathbb{N}}$ by $\mathcal{M}_i = \mathbf{Mod(T_i)}$. Let $\mathfrak{T}$ be the set of these traces for all possible correct executions of the algorithm for $\mathbf{Th}$. Then we have:

$\mathfrak{T}$ is exactly the set of minimal models of $\mathbf{Th}$.

So in the case of a finite theory of reasoning, our algorithm can construct precisely its minimal models. Since any reasoning process described by a reasoning frame has a theory of reasoning describing it, which can be executed by our algorithm, we have a general algorithm for executing these processes. Applying this algorithm to the theory of reasoning of a default theory, we obtain an algorithm very similar to the ones meant especially for default logic: it picks a subset of so-called *generating defaults*, and checks whether it indeed induces an extension.

Using this algorithm to obtain either one or all of the minimal models of a theory of reasoning, one can verify various properties of the reasoning process described by the theory of reasoning. For instance it can be checked whether a certain formula is derived in one or in all possible runs. Thus for nonmonotonic reasoning processes, sceptical and credulous entailment can be calculated. It can be shown (using the above algorithm) that the complexity of sceptical entailment using theories of reasoning is complete in Π_2^P . This is the complexity of sceptical entailment of many nonmonotonic formalisms like autoepistemic logic, McDermott and Doyle's nonmonotonic logic and nonmonotonic logic N (see [En95]).

A special class of theories of reasoning is obtained when we do not allow consistency checks (the set $\mathbf{D}$ in definition 3.3 is empty). The induced reasoning process will then be monotonic, and the algorithm will be much more efficient.

In specific instances of theories of reasoning one would like to make the algorithm more efficient. This can be done by using heuristic knowledge to make smart choices at each point in time. In particular, if at the current point in time one of the $\theta \in \mathbf{D}$ is already entailed by the facts, only the second choice in the algorithm makes sense. In the case of default logic, one could use priorities between default rules or specificity of rules to restrict the number of possible choices (see [Br94]).

The set of possible runs of our algorithm can be parametrized by selection functions. Such functions describe the choices which have to be made at each point in time (in a similar fashion as in [TT92]). Then the "good" selection functions make use of heuristic knowledge to guide the reasoning process.

5 Conclusions and Further Research

The formalism of temporal logic is used to specify and verify processes in general. If a class of processes can be specified accurately by a form of temporal logic which can be executed, we have a general execution mechanism for this class. In this paper we studied the class of nonmonotonic reasoning processes (see also [EHT95], [ET95]). These processes can be described semantically by the reasoning traces they produce. Viewing these traces as temporal models, they can be specified by temporal rules. We have shown that a fragment of infinitary temporal epistemic logic is suitable for describing any set of traces (see [ET93] and [ET94] for specifications of a number of specific types of nonmonotonic reasoning). In the (important) case when the signature is finite we have given an algorithm which can execute any specification of reasoning. Although the execution of temporal rules is similar to existing executable temporal logics, the consistency checks are treated differently. They are treated as declarative tests in the future, not as imperative commands for the future.

As our specification language describes reasoning processes, a natural question to investigate is whether it is possible to find a generic meta-level architecture capable of executing specifications of reasoning. In such an architecture, the meta-level should check the rules for applicability and select the conclusions to be added at the object-level. Heuristic knowledge at the meta-level can be used to control the selection of conclusions. There should be an easy and natural way to translate specifications of reasoning into rules to be used by this architecture, thus instantiating the generic architecture.

Specification languages for such meta-level architectures can be found in the research on formal specification languages for complex reasoning systems: see [TW93] for an overview of eight of these languages, illustrated on the basis of a common nonmonotonic example reasoning pattern. One of these languages is DESIRE (see also [LPT92], [GT94]); in [TT92] it is shown how DESIRE can be used to specify a reasoning system creating nonmonotonic reasoning patterns. Here explicit control knowledge is (required to be) added at the meta-level; this control knowledge specifies a selection function on (possible) default conclusions. The software environment supporting DESIRE provides (automatically generated) implementation code (PROLOG code that can be executed).

Acknowledgements

Part of this work has been supported by SKBS and the ESPRIT III Basic Research project 6156 DRUMS II. Discussions with Heinrich Herre have played a stimulating role in developing part of the material in this paper.

References

[Ga89] Gabbay, D.: "The Declarative Past and Imperative Future: Executable Temporal Logic for Interactive Systems", in: B. Banieqbal, H. Barringer, A. Pnueli (eds.), *Temporal Logic in Specification*, Lecture Notes in Computer Science, Vol. 398, Springer-Verlag, 1989, pp. 409-448.

[Br94] Brewka, G.: "Adding Priorities and Specificity to Default Logic", in: C. MacNish, D. Pearce, L.M. Pereira (eds.), *Logics in Artificial Intelligence*, Proceedings of the 4th European Workshop on Logics in Artificial Intelligence, JELIA ’94, Springer-Verlag, 1994, pp. 247-260.

[EHT95] Engelfriet, J., Herre, H., Treur, J. : "Nonmonotonic Belief State Frames and Reasoning Frames (extended abstract)", in: C. Froidevaux, J. Kohlas (eds.), *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, Proc. ECSQARU'95, Springer-Verlag, 1995, pp. 189-196.

[En95] Engelfriet, J: "Minimal Temporal Epistemic Logic", Technical Report IR-388, Department of Mathematics and Computer Science, Vrije Universiteit Amsterdam, 1995

[ET93] Engelfriet, J., Treur, J.: "A Temporal Model Theory for Default Logic", in: M. Clarke, R. Kruse, S. Moral (eds.), Proc. 2nd European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, ECSQARU '93, Springer-Verlag, 1993, pp. 91-96. Extended version: Report IR-334, Vrije Universiteit Amsterdam, Department of Mathematics and Computer Science, 1993

[ET94] Engelfriet, J., Treur, J. : "Temporal Theories of Reasoning", in: C. MacNish, D. Pearce, L.M. Pereira (eds.), *Logics in Artificial Intelligence*, Proceedings of the 4th European Workshop on Logics in Artificial Intelligence, JELIA ’94, Springer-Verlag, 1994, pp. 279-299. Extended version to appear in: Journal of Applied Non-Classical Logics.

[ET95] Engelfriet, J., Treur, J. : "Specification of Nonmonotonic Reasoning", Technical Report, Vrije Universiteit Amsterdam, 1995.

[Fi94] Fisher, M.: "A Survey of Concurrent METATEM - The Language and its Applications", in: D.M. Gabbay, H.J. Ohlbach (eds.), *Temporal Logic*, Proceedings ICTL'94, Springer-Verlag, 1994, pp. 480-505.

[FO95] Fisher, M., Owens, R. (eds.), *Executable Modal and Temporal Logics*, Proceedings of the IJCAI'93 workshop, Springer-Verlag, 1995.

[GT94] Gavrila, I.S., Treur, J.: "A formal model for the dynamics of compositional reasoning systems", in: A.G. Cohn (ed.), *Proc. 11th European Conference on Artificial Intelligence, ECAI'94*, Wiley and Sons, 1994, pp. 307-311.

[KLM90] Kraus, S., D. Lehmann, M. Magidor: "Nonmonotonic Reasoning, Preferential models and cumulative logics"; A.I. 44 (1990), 167 - 207

[LPT92] Langevelde, I.A. van , Philipsen, A.W., Treur, J. : "Formal Specification of Compositional Architectures", in: B. Neumann (ed.), *Proceedings of the 10th European Conference on Artificial Intelligence*, ECAI'92, John Wiley & Sons, Chichester, 1992, pp. 272-276.

[MT93] Marek, V., M. Truszczynski: *Nonmonotonic Logic*, Springer-Verlag, 1993.

[OM94] Orgun, M.A., Ma, W.: "An Overview of Temporal and Modal Logic Programming", in: D.M. Gabbay, H.J. Ohlbach (eds.), *Temporal Logic*, Proceedings ICTL'94, Springer-Verlag, 1994, pp. 445-479.

[Sh88] Shoham, Y.: *Reasoning about Change*, MIT Press, 1988

[TT92] Tan, Y.H., Treur, J. : "Constructive Default Logic and the Control of Defeasible Reasoning", in: B. Neumann (ed.), *Proc. 10th European Conference on Artificial Intelligence*, ECAI-92, Wiley and Sons, 1992, pp. 299-303.

[TW93] Treur, J., Wetter, Th. (eds.), *Formal Specification of Complex Reasoning Systems*, Ellis Horwood, 1993

[Vo93] Voorbraak, F.: "Preference-based Semantics for Nonmonotonic Logics", in: Bajcsy, R. (ed.), *Proc. 13th International Joint Conference on Artificial Intelligence*, IJCAI-93, Morgan Kaufmann, 1993, pp.584-589.