Agent Programming with Temporally Extended Goals

Koen V. Hindriks
Delft University of Technology
P.O.Box 5031 Delft, The Netherlands
k.v.hindriks@tudelft.nl

Wiebe van der Hoek
University of Liverpool
Liverpool L69 3BX, United Kingdom
wiebe@csc.liv.ac.uk

M. Birna van Riemsdijk
Delft University of Technology
P.O.Box 5031 Delft, The Netherlands
m.b.vanriemsdijk@tudelft.nl

ABSTRACT

In planning as well as in other areas, temporal logic has been used to specify so-called *temporally extended goals*. Temporally extended goals refer to desirable *sequences of states* instead of a set of desirable *final states* as the traditional notion of achievement goal does, and provide for more variety in the types of goals allowed. In this paper, we show how temporally extended goals can be integrated into the agent programming language GOAL. The result is that GOAL agents may now have both beliefs about the future as well as have temporally extended goals. We propose a new decision making mechanism that takes temporally extended goals into account, and investigate properties of this framework.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, languages and structures*; I.2.5 [**Artificial Intelligence**]: Programming Languages and Software

General Terms

Theory, Languages

Keywords

Agent Programming, Goals, Formal Semantics, Temporal Logic

1. INTRODUCTION

The core component of a rational agent is its capability to make rational choices of action, in order to satisfy its design objectives. In agent programming languages for rational agents, such choices are derived from the agent's beliefs and goals. That is, the agent should act towards realization of its goals, taking into account its beliefs when making choices of action.

There has been much work in recent years on how goals can be used in agent programming frameworks (see, e.g., [13, 26, 5, 8, 10, 21, 24, 15, 16]). Techniques for reasoning with goals to decide on action have become increasingly sophisticated. Moreover, while in most earlier approaches the focus was on achievement goals (goals to reach a certain state of affairs), there is a growing attention also for other goal types such as maintenance goals (goals to maintain a certain state of affairs).

Cite as: Agent Programming with Temporally Extended Goals, Koen Hindriks, Wiebe van der Hoek and Birna van Riemsdijk, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. XXX-XXX.

Our long-term aim is the development of an agent programming language with *expressive* means for the representation of various types of goals, and built-in *computational reasoning mechanisms* that use these goals for rational decision making. Generally speaking, one can take two approaches to address the former issue: one can design separate representational tools for each goal type, or one can use a single language in which several goal types can be represented. An advantage of the first approach is that it facilitates the design of specialized reasoning methods tailored towards specific goal types. An advantage of the second approach is that it provides for a unifying approach, allowing the representation of multiple goal types within a single framework. In this paper, we take the latter approach.

As in [16], we use linear temporal logic for the representation of goals. This is inspired by the use of *temporally extended goals* in planning, in which the use of linear temporal logic has been shown to be very useful for specifying planning problems as well as for specifying heuristic information for efficiently finding better quality plans (see, e.g., [1, 2, 4, 23]). An important advantage of the use of temporal logic is that it facilitates establishing a connection between the agent programming language and agent logics, as done in [14], which in turn paves the way for verification of agent programs. Moreover, it provides for additional reasoning power.

The general idea of the representation of goals using temporal logic is that an agent should choose its actions such that it produces computation traces satisfying the goals. An agent can do this by looking ahead and envisioning which execution paths it could take, and then choosing an action on a path on which its goals are satisfied.

However, agents have limited time and resources for reasoning, i.e., agents are *boundedly rational* [22]. This means they cannot look infinitely far ahead and then choose the best action, i.e., they need a finite *bound* or *lookahead horizon*, restricting the reasoning. The solution that was proposed in [15, 16] is that the agent programmer specifies a fixed, finite lookahead horizon, consisting of the number of steps that an agent can look ahead. However, choosing such a number is relatively arbitrary and error prone.

The main contribution of this paper is that we show how temporal logic can be used for a qualitative and more intuitive specification of horizons, and we propose a computational mechanism for reasoning with temporally extended goals in an agent programming framework, based on the use of these horizons (Section 5). This mechanism needs an agent to have beliefs about the future, and consequently an important part of the effort is concerned with the incorporation of temporal beliefs in the programming framework, and establishing the relation between beliefs and goals in this temporal setting (Sections 3 and 4). We also investigate (logical) properties of our framework. In Section 2, we provide some further motivation and background, and we discuss related research and conclude the paper in Section 6.

2. BOUNDED GOALS AND TEMPORAL BELIEFS

In this section, we sketch the main issues addressed in this paper using a simple example. Two important goal types are achievement goals and maintenance goals. For example, an employed agent may have the achievement goal to be at work, and since he sometimes takes the car to work, he has the maintenance goal of keeping enough fuel in the tank. Using linear temporal logic, achievement goals can be represented using the $\diamond$ ("eventually") operator, e.g., $\diamond atWork$, and maintenance goals can be represented using the $\Box$ ("always") operator, e.g., $\Box$*fuel* (cf. [3] for a discussion of various ways of representing maintenance goals in temporal logic).

Both of these goals have an "unbounded" aspect, i.e., the goal $\diamond atWork$ specifies that the agent wants to be at work at *some point in the future*, and the goal $\Box$*fuel* specifies that the agent *always* wants to have enough fuel in the tank. In order to determine whether particular choices of action will realize these goals, the agent may have to look infinitely far ahead. That is, if a particular finite sequence of actions does not realize the achievement goal, it may always be the case that the *next* action does, and similarly, if this sequence of actions maintains the maintenance goal, it may always be the case that the *next* action violates the maintenance goal.

We can thus see that these unbounded goals pose problems for boundedly rational agents, since they cannot always determine whether choosing a certain action will lead to realization of the goals. In this paper, we therefore suggest to focus on goals that *do* have a bound, and we maintain that many unbounded goals in fact can be endowed with a natural bound to aid the agent in its reasoning.

Take, for example, the achievement goal to be at work. Here, it is probably the case that the agent should be at work before, e.g., 9 o'clock, i.e., the achievement goal has a deadline providing for a bound on the goal. This *deadline goal* can be represented in temporal logic as $atWork$ $\mathbf{before}$ *9:00*. A natural bound for the maintenance goal of keeping enough fuel in the tank may be "until the agent arrives at a gas station". This *bounded maintenance goal* can be represented in temporal logic as $fuel$ $\mathbf{until}$ $atGasStation$. Using the expressive power of temporal logic, it is thus possible to specify qualitative bounds or horizons.

The mechanism we propose for using temporally extended goals in the agent's decision making is an adaptation of the lookahead mechanism of [15, 16]. Instead of looking ahead a fixed number of steps the idea is that the programmer may specify qualitative bounds that define the lookahead horizon using bounded goals. When looking ahead until this horizon, the agent can use its goals to avoid selecting those actions that prevent the realization of (some of) these goals. For example, if the agent has deadline goal $atWork$ $\mathbf{before}$ *9:00*, the agent looks ahead until 9 o'clock, and should try choosing its actions such that it achieves the goal of being at work before the deadline. It should, e.g., not fall asleep again after the alarm rings, since this would lead to a violation of the deadline goal. The mechanism is similar for bounded maintenance goals, in which case the agent should look ahead and maintain a state of affairs until the bound of the goal.

For an operationalization of bounded goals in a way as described above, it is important that the bounds of these goals *do* occur eventually, since the agent would otherwise keep looking ahead without ever encountering the bound. Put differently, the agent should be able to determine that a bound of a bounded goal occurs eventually, of course, without actually doing the lookahead. We propose to allow the agent to have beliefs about the future, i.e., *temporal beliefs*, such that the agent can determine whether a bound occurs eventually by temporal reasoning over his beliefs.

3. REPRESENTATION OF BELIEFS AND GOALS

We use the agent programming language GOAL [9] as a basis for our framework. We maintain that investigating the consequences of certain choices in a specific framework also sheds light on the general principles. GOAL is a language for programming *rational agents*. GOAL agents maintain a mental state of declarative *beliefs* and *goals* and derive their choice of action from their beliefs and goals. A GOAL agent program defines the initial beliefs and goals of an agent, specifies the preconditions and effects of the actions available to the agent, and contains a set of *action rules* to select actions for execution at runtime. Action rules define a *strategy or policy* of the agent for acting. The beliefs and goals of an agent are dynamic and change over time. The action specifications and action rules are static.

GOAL does not commit to any particular knowledge representation language but assumes such a language with associated inference relation and update operators is given. In the current implementation of GOAL Prolog has been used with a STRIPS-like [12] specification of actions. This follows current practice in agent programming where so far essentially propositional languages like Prolog are used to specify an agent's beliefs and/or goals.

However, as discussed above, such languages do not provide the expressive power to express temporally extended goals, nor temporal beliefs. Such temporal goals and beliefs have been usefully applied in planning and other areas, and in this section we show how beliefs and goals can be represented using linear temporal logic (LTL) in the agent programming framework GOAL. A base language $\mathcal{L}_0$ of classical propositional formulae over a set of atoms At, with typical element ϕ, is assumed that includes $\top, \bot \in \mathcal{L}_0$ denoting respectively the true and false sentence.

Definition 1. (Linear Temporal Logic)
The language of linear temporal logic $\mathcal{L}_{LTL}$, with typical element χ, is defined by:

$$\begin{array}{lll} \phi & ::= & \text{any element from } \mathcal{L}_0 \\ \chi & ::= & \phi \mid \neg\chi \mid \chi \wedge \chi \mid \bigcirc\chi \mid \chi\ \mathbf{until}\ \chi \end{array}$$

The semantics of LTL formulas is as usual defined on traces, which are infinite sequences of *valuations*. Given a set of propositional atoms At, *state* is a valuation, if for every $p \in$ At, exactly one of the literals p and $\neg p$ is in *state*. For example, if t is a trace then $\bigcirc\chi$ is satisfied on t in state i, expressed as $t, i \models_{LTL} \bigcirc\chi$, if $t, i+1 \models_{LTL} \chi$. See, e.g., [11] for further details.

The eventuality operator $\diamond\varphi$ is introduced as an abbreviation for $\top$ $\mathbf{until}$ φ and its dual $\Box\varphi$ is defined as $\neg\diamond\neg\varphi$. We argue that goals with deadlines may be effectively used in an agent's decision-making in Section 5. To be able to easily express in particular deadline goals, we also introduce the φ $\mathbf{before}$ ψ operator. This operator does not introduce additional expressivity and can be defined in terms of $\mathbf{until}$ by $\neg(\neg\varphi\ \mathbf{until}\ \psi)$.

3.1 Mental States

Rational agents need to maintain a rational balance among their beliefs and goals [7]. That is, the beliefs and goals of a rational agent need to be reasonable and ideally are justified in some way. Allowing temporal formulae as beliefs and goals raises particular issues related to maintaining such a balance. In particular, arguing that imposing particular constraints on the relation between beliefs and goals is reasonable, is much harder in a temporal setting. It has been the subject of much debate. Two of the most influential papers in which this relation is defined are [7, 19], in which temporal logics are proposed that incorporate modal operators for beliefs and goals, and for other mental attitudes.

In this paper, we follow the approach proposed in [7] when it comes to defining *rationality constraints* on beliefs and goals and their relation. That is, we provide primitives for the representation of an agent's beliefs and goals, imposing only a few basic constraints at this level on the relation between beliefs and goals. Using these primitives, other variants of these mental attitudes can be defined that can be used for rational action selection.

As usual in GOAL, we use a belief base, typically denoted by Σ, and goal base, typically denoted by Γ, as representational structures for beliefs and goals. Both Σ and Γ consist of LTL formulas in this paper. Following [7] (and other approaches, for that matter [19, 14]), we require the belief base and goal base each to be consistent. Again following [7], we maintain that a rational agent should not want to change the inevitable. This is called *realism* in [7]. Informally, things that the agent believes will happen inevitably are represented by an agent's beliefs about the future, whereas a goal expresses something that an agent wants to achieve at some moment in time in the future. That is, the goals of an agent should determine a condition that is more specific than what is believed to be inevitable; since the more specific entails the less specific, an agent's goals should entail its beliefs. Goals are thus more specific than beliefs, in the sense that they add desired properties that can be influenced or controlled by the agent to the inevitable beliefs.

The belief base and goal base of a GOAL agent together make up its *mental state*. The following definition formally defines mental states and the accompanying rationality constraints.

Definition 2. (Mental States)
A mental state of a GOAL agent, typically denoted by m, is a pair $\langle\Sigma,\Gamma\rangle$ with $\Sigma \subseteq \mathcal{L}_{LTL}$ the belief base, and $\Gamma \subseteq \mathcal{L}_{LTL}$ the goal base. Additionally, mental states need to satisfy the following *rationality constraints*:

(i) The belief base is consistent: $\Sigma \not\models_{LTL} \bot$,
(ii) The goal base is consistent: $\Gamma \not\models_{LTL} \bot$,
(iii) Goals refine (inevitable) beliefs: $\Gamma \models_{LTL} \Sigma$.

Note that it follows from this definition that the belief base and goal base are also mutually consistent, i.e., $\Sigma \cup \Gamma \not\models \bot$, which means that the agent cannot have something as a goal that is never realizable according to its beliefs. Also, note that the constraint $\Gamma \models_{LTL} \Sigma$ allows to derive the goal $\Diamond p$ from a disjunctive goal $\Diamond p \vee \Diamond q$ and the belief that $\Box\neg q$, establishing interaction between disjunctive goals and beliefs.

3.2 Mental State Conditions

A GOAL agent needs the means to inspect its beliefs and goals in order to derive its choice of action from these. To do so, so-called *mental state conditions* are introduced to reason about the agent's beliefs and goals. The language $\mathcal{L}_m$ of mental state conditions extends $\mathcal{L}_{LTL}$ with a belief $\mathbf{B}$ and (primitive) goal $\mathbf{G}$ operator, which can be used to express conditions on the mental state of an agent. That is, the set of mental state conditions consists of Boolean combinations of formulae of the form $\mathbf{B}\chi$ and $\mathbf{G}\chi$ with $\chi \in \mathcal{L}_{LTL}$.

Definition 3. (Mental State Conditions: Syntax)
The language $\mathcal{L}_m$, with typical element ψ, of *mental state conditions* is defined by:

$$\begin{array}{lcl} \chi & ::= & \text{any element in } \mathcal{L}_{LTL} \\ \psi & ::= & \mathbf{B}\chi \mid \mathbf{G}\chi \mid \neg\psi \mid \psi\wedge\psi \end{array}$$

Note that it is not allowed to nest the operators $\mathbf{B}$ and $\mathbf{G}$, nor to use temporal operators outside the scope of these operators. The semantics of mental state conditions is defined with respect to mental states.

Definition 4. (Mental State Conditions: Semantics)
Let $\langle\Sigma,\Gamma\rangle$ be a mental state. The semantics of mental state conditions is defined by:

$$\begin{array}{lll} \langle\Sigma,\Gamma\rangle \models_m \mathbf{B}\chi & \text{iff} & \Sigma \models_{LTL} \chi, \\ \langle\Sigma,\Gamma\rangle \models_m \mathbf{G}\chi & \text{iff} & \Gamma \models_{LTL} \chi, \\ \langle\Sigma,\Gamma\rangle \models_m \neg\psi & \text{iff} & \langle\Sigma,\Gamma\rangle \not\models_m \psi, \\ \langle\Sigma,\Gamma\rangle \models_m \psi\wedge\psi' & \text{iff} & \langle\Sigma,\Gamma\rangle \models_m \psi \text{ and } \langle\Sigma,\Gamma\rangle \models_m \psi'. \end{array}$$

Using the belief and primitive goal modalities $\mathbf{B}$ and $\mathbf{G}$ it is possible to *define* several related notions of goals. First, we define an operator $\mathbf{Goal}\chi$ by $\mathbf{G}\chi \wedge \neg\mathbf{B}\chi$, i.e., $\mathbf{Goal}\chi$ holds if χ follows from the agent's goal base, and is not believed to occur inevitably. The operator $\mathbf{Goal}\chi$ corresponds more closely to the intuitive notion of a goal as being something that the agent should put effort into bringing about.

Using this operator, we can make several additional classifications of types of goals that an agent may be said to have. For example, χ is said to be an *achievement goal* whenever $\mathbf{Goal}\Diamond\chi$, and we write $\mathbf{G}_{ach}\chi$. Note that if an agent has an achievement goal $\mathbf{G}_{ach}\chi$ and believes that χ always implies χ', i.e. $\mathbf{B}\Box(\chi\rightarrow\chi')$, although we have $\mathbf{G}\Diamond\chi'$, it does not follow that $\mathbf{G}_{ach}\chi'$ since the agent may for instance believe that χ'. Achievement goals are particular instances of *deadline goals* of the form $\mathbf{Goal}(\chi_1 \ \mathbf{before}\ \chi_2)$ with $\chi_2 = \bot$. An agent is said to have a *bounded maintenance goal* when $\mathbf{Goal}(\chi_1 \ \mathbf{until}\ \chi_2)$ holds. Finally, it should be noted that the goal operator $\mathbf{G}$ used in the presentation of GOAL in [9] is different from the goal operator $\mathbf{G}$ introduced here; the operator $\mathbf{G}$ in [9] is best read as an achievement goal operator, an interpretation formally justified in [14].

Proposition 1. The following formulae are valid on mental states:

1. $\neg\mathbf{B}\bot \wedge \neg\mathbf{G}\bot$
2. $\mathbf{B}(\chi_1\rightarrow\chi_2)\rightarrow(\mathbf{B}\chi_1\rightarrow\mathbf{B}\chi_2)$
3. $\mathbf{B}\chi\rightarrow\mathbf{G}\chi$
4. $(\mathbf{B}(\chi_1 \ \mathbf{before}\ \chi_2)\wedge\mathbf{B}\Diamond\chi_2)\rightarrow\mathbf{B}\Diamond\chi_1$
5. $(\mathbf{G}(\chi_1 \ \mathbf{before}\ \chi_2)\wedge\mathbf{B}\Diamond\chi_2)\rightarrow\mathbf{G}\Diamond\chi_1$
6. $\mathbf{Goal}\chi\leftrightarrow(\mathbf{G}\chi\wedge\neg\mathbf{B}\chi\wedge\neg\mathbf{B}\neg\chi)$

Item 1 expresses that both beliefs and goals are consistent; item 2 says that $\mathbf{B}$ is a 'normal modal operator', 3 implies realism. Item 5 (4 is similar) expresses the following rationality of goals: if the agent has a goal that χ_1 will happen before χ_2, and he indeed believes that χ_2 will sometime occur, then he has as a goal that χ_1 will sometime occur. Item 6 explains why $\mathbf{Goal}$ can be considered to model goals that the agent is willing to act upon: any χ for which $\mathbf{Goal}\chi$ holds is in the agent's goal base, and not to believed to be guaranteed or impossible. Proofs are omitted for reasons of space.

Some other desirable properties follow rather straightforwardly from those above, e.g., we have that if an agent has a goal, it does not believe that the opposite is inevitable, i.e., $\mathbf{G}\chi\rightarrow\neg\mathbf{B}\neg\chi$, or, equivalently, $\mathbf{B}\neg\chi\rightarrow\neg\mathbf{G}\chi$.

4. DYNAMICS OF BELIEFS AND GOALS

In the previous section, we showed how beliefs and goals can be represented using LTL. In this section, we discuss how beliefs and goals change during execution of the agent. In Section 4.1, we show how to represent preconditions and effects of actions in LTL and in Section 4.2 we specify how beliefs and goals change as the result of the execution of actions.

4.1 Action Theories

Both in planning and in agent programming, actions are typically specified by defining their preconditions and effects. A well-known problem in artificial intelligence is the *frame problem*, which is the problem of how to specify what is *not* changed by the execution of actions. This problem arises in particular in the context of logical specifications of actions.

In this paper, we base the specification of actions on [17], in which an encoding of Reiter's solution to the frame problem [20] in LTL is proposed for use in planning. For this, we extend the set of atoms At over which $\mathcal{L}_{LTL}$ is defined (see Definition 1) with a set of actions Act such that $\mathsf{Act} \cap \mathsf{At} = \emptyset$ [17]. If $\mathtt{a} \in \mathsf{Act}$, the intended meaning of $\mathtt{a}$ holding at a state of a computation trace is that the action $\mathtt{a}$ is performed in this state. In order to make a clear distinction between actions and other propositional atoms, we write $\mathbf{do}(\mathtt{a})$ rather than $\mathtt{a}$, when referring to actions in LTL formulas. We call the resulting language $\mathcal{L}_{LTL}^{\mathsf{Act}}$.

Action preconditions are specified in $\mathcal{L}_{LTL}^{\mathsf{Act}}$ by means of formulae of the form

$$\Box(\mathbf{do}(\mathtt{a}) \rightarrow pre_{\mathtt{a}})$$

where $pre_{\mathtt{a}} \in \mathcal{L}_0$ expresses the precondition for doing $\mathtt{a}$. Intuitively, these formulae express that at any time, action $\mathtt{a}$ may be performed only if its preconditions $pre_{\mathtt{a}}$ hold. For example, the precondition of the action of driving to work may be that a car is available, specified as $\Box(\mathbf{do}(\mathtt{driveToWork}) \rightarrow \mathit{car})$.

Moreover, *action effects* are represented by temporal logic encodings of Reiter's successor state axioms [20]. The basic idea of Reiter's solution to the frame problem is that a propositional atom $p \in \mathsf{At}$ may change its truth value only if an action is performed that affects this truth value. Two cases are distinguished: (i) actions that have p as effect and (ii) actions that have $\neg p$ as effect. For each proposition $p \in \mathsf{At}$ the first set of actions $\mathtt{a}_1, \ldots, \mathtt{a}_m$ is collected and a disjunction is formed of the form $\mathbf{do}(\mathtt{a}_1) \vee \ldots \vee \mathbf{do}(\mathtt{a}_m)$ denoted by A_p^+ and, similarly, the second set of actions $\mathtt{a}_{m+1}, \ldots, \mathtt{a}_n$ is collected and a disjunction is formed of the form $\mathbf{do}(\mathtt{a}_{m+1}) \vee \ldots \vee \mathbf{do}(\mathtt{a}_n)$ denoted by A_p^-.

Then, for each proposition p a successor state axiom of the form

$$\Box(\bigcirc p \leftrightarrow (A_p^+ \vee (p \wedge \neg A_p^-)))$$

is introduced. Intuitively, such formulae express that at any time, in the next state p holds iff an action is performed that has p as effect (i.e. A_p^+ holds) or p is true in the current state and no action that has $\neg p$ as effect is performed (i.e. $\neg A_p^-$ holds). For example, one may specify that the agent is at work, either if it just drove or cycled to work, or if it was already at work and did not go home, i.e., $\Box(\bigcirc \mathit{atWork} \leftrightarrow (A^+_{\mathit{atWork}} \vee (\mathit{atWork} \wedge \neg A^-_{\mathit{atWork}})))$, where $A^+_{\mathit{atWork}} = \mathbf{do}(\mathtt{driveToWork}) \vee \mathbf{do}(\mathtt{cycleToWork})$ and $A^-_{\mathit{atWork}} = \mathbf{do}(\mathtt{driveHome}) \vee \mathbf{do}(\mathtt{cycleHome})$.

A difference between agents and planners is that the former perform actions indefinitely and potentially produce infinite action sequences whereas a planner is assumed to only produce a *finite* sequence of actions. This difference between agents and planners is captured by the formula

$$\Box(\mathbf{do}(\mathtt{a}_1) \vee \ldots \vee \mathbf{do}(\mathtt{a}_n))$$

where $\mathtt{a}_1, \ldots, \mathtt{a}_n$ are actions which together exhaust all the actions in Act. We call this the *always acting* axiom. Naturally, this requires that at least one action can be executed at any time. This can be achieved, e.g., by including an action that can always be executed, such as a `wait` action that can always be executed and that has no effect, other than the passing of time.

An *action theory* $\mathcal{A} \subseteq \mathcal{L}_{LTL}^{\mathsf{Act}}$ now consists of action preconditions for all actions $\mathtt{a} \in \mathsf{Act}$, successor state axioms for all propositions $p \in \mathsf{At}$, which indirectly specify the effects of actions, and the axiom of always acting.

Using LTL for the specification of actions in our setting has two main advantages. First, it contributes to the uniformity of our framework, which uses LTL as the basis for representing mental attitudes. Second, it provides a solution to the issue of *where temporal beliefs come from.* That is, the idea is that we refine the proposal of Section 3 by not allowing arbitrary LTL formulas in the belief base. Rather, the belief base consists of those formulas used for specifying the action theory, and a representation of what is currently the case, i.e., a representation of the *current state*.

This is defined formally as follows. A belief base Σ, containing LTL formulas defined over a set of atoms and actions $\mathsf{At} \cup \mathsf{Act}$ is a set of the form *state* $\cup\, \mathcal{A}$, where *state* is a valuation[1] over At describing the current state and $\mathcal{A} \subseteq \mathcal{L}_{LTL}^{\mathsf{Act}}$ is an action theory. The restriction of the current state to literals is usual in planning. In the sequel, we will assume belief bases to be of this form.

An action theory may allow that multiple actions are executed simultaneously. Obviously, this is only allowed when the preconditions of these actions hold and the effects of the actions are consistent. We use $\mathsf{Do}(A)$ as a shorthand for $\bigwedge_{\mathtt{a} \in A} \mathbf{do}(\mathtt{a})$, and $\mathsf{NotDo}(B)$ as a shorthand for $\bigwedge_{\mathtt{b} \in B} \neg\mathbf{do}(\mathtt{b})$. Let $B = \mathsf{Act} \setminus A$ and assume that belief base Σ has the form *state* $\cup\, \mathcal{A}$ with $\mathcal{A} \subseteq \mathcal{L}_{LTL}^{\mathsf{Act}}$. We say that a set of actions $A \subseteq \mathsf{Act}$ *can be executed consistently* in Σ if $\mathcal{A} \cup \{\mathsf{Do}(A)\} \cup \{\mathsf{NotDo}(B)\} \not\models_{LTL} \bot$ and $\forall \mathtt{a} \in A : \Sigma \models_{LTL} pre_{\mathtt{a}}$; we say A is *maximal* if there is no set $A' \supset A$ that can be consistently executed.

Logical Properties.

Since action theories are part of an agent's beliefs, the following validities are immediate.

Proposition 2. We have the following, where $\ell(p)$ is a variable over $\{p, \neg p\}$ with $p \in \mathsf{At}$, and $\phi, \phi_1, \phi_2 \in \mathcal{L}_0$ (the base propositional language, see Section 3).

1. $\mathbf{B}\mathbf{do}(\mathtt{a}) \rightarrow \mathbf{B}pre_{\mathtt{a}}$
2. $\mathbf{B} \bigcirc p \leftrightarrow \mathbf{B}(A_p^+ \vee (p \wedge \neg A_p^-))$
3. $\neg\mathbf{B}\ell(p) \leftrightarrow \mathbf{B}\neg\ell(p)$
4. $\mathbf{B}(\phi_1 \vee \phi_2) \leftrightarrow (\mathbf{B}\phi_1 \vee \mathbf{B}\phi_2)$
5. $\mathbf{B}\neg\phi \leftrightarrow \neg\mathbf{B}\phi$
6. $(\mathbf{B}\Box\chi \wedge \mathbf{B}(\mathbf{do}(\mathtt{a}) \rightarrow \bigcirc\neg\chi)) \rightarrow \mathbf{B}\neg\mathbf{do}(\mathtt{a})$

Item 3 of Proposition 2 is an instance of the Closed World Assumption for literals. It entails Items 4 and 5. Also, note that the closed world assumption does not imply that an agent believes each temporal formula or its negation; for example, we can have $\neg\mathbf{B}\Diamond p$ without also having $\mathbf{B}\neg\Diamond p$.

Moreover, it is important to note that what can be derived as a belief from an action theory and a current state may be relatively weak. For example, the agent will only be able to derive $\mathbf{B}\mathbf{do}(\mathtt{a})$, if $\mathtt{a}$ is the only executable action, i.e., the only action for which the precondition is believed to hold in the current state. If, e.g., the action $\mathtt{b}$ is also executable, it will only be able to derive the weaker $\mathbf{B}(\mathbf{do}(\mathtt{a}) \vee \mathbf{do}(\mathtt{b}))$. This reflects the fact that an action theory only specifies what could in principle be executed, and what would be the effects of that. If multiple actions are executable, the agent still has to make a choice and select the actions that *will* be executed.

A similar discussion applies to the derivation of (temporal) beliefs about propositions. For example, assume that $\mathsf{Act} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$,

[1] See Section 3.

$A_p^+ = \mathbf{do}(\mathtt{a}) \vee \mathbf{do}(\mathtt{b}) \vee \mathbf{do}(\mathtt{c})$, and that a and b are the only executable actions in the current state, according to the belief base, i.e., $\mathbf{B}(\mathbf{do}(\mathtt{a}) \vee \mathbf{do}(\mathtt{b}))$ holds. Further, assume that $\mathbf{B}\neg p$. The agent can then derive $\mathbf{B} \bigcirc p$, since both a and b have p as an effect. This is reflected in Item 2 of Proposition 2, i.e., the agent would be able to derive $\mathbf{B}A_p^+$ from $\mathbf{B}(\mathbf{do}(\mathtt{a}) \vee \mathbf{do}(\mathtt{b}))$, and from this be able to derive $\mathbf{B} \bigcirc p$. However, if b would not have p as an effect, i.e., $A_p^+ = \mathbf{do}(\mathtt{a}) \vee \mathbf{do}(\mathtt{c})$, the agent would not be able to derive $\mathbf{B}A_p^+$ and consequently neither $\mathbf{B} \bigcirc p$.

Intuitively, the action theory implicitly specifies all possible futures, i.e., all possible sequences of action execution and their effects that are in principle possible, given the action specifications. An agent will only be able to derive $\mathbf{B}\varphi$ if φ is the case in all possible futures, i.e., no matter which actions the agent will choose, φ will be the case. In particular, $\mathbf{B}\Diamond\phi$ will only be derivable if ϕ will eventually be the case according the beliefs, no matter what actions the agent chooses for execution.[2] These kinds of beliefs will be used for reasoning with bounded goals in Section 5.

4.2 Progression of Belief and Goal Bases

An agent's belief base specifies when actions can be executed and what their effects are, and it specifies the current state. In a *programming framework* for rational agents, we also need to provide a computational mechanism that specifies how an agent's beliefs and goals change, if the agent selects actions for execution, i.e., we have to specify how the belief base and goal base are *progressed* when actions are executed.

Progression of Beliefs.

When a set of actions A is executed, only the part of the belief base describing the current state is updated. That is, the action theory remains unchanged. The new current state consists of all literals p and $\neg q$ for which $\bigcirc p$ or $\bigcirc\neg q$ can be derived from the belief base, if it is assumed that A will be executed.

Definition 5. (Progression of Belief Base) Given a belief base Σ consisting of a current state *state* and action theory $\mathcal{A} \subseteq \mathcal{L}_{LTL}^{\mathsf{Act}}$, we define the effects of the execution of a set of actions $A \subseteq \mathsf{Act}$ as follows. Let $B = \mathsf{Act} \setminus A$. Then

$$\begin{array}{rll} \mathit{state}' = & \{ \quad p & \mid \Sigma \cup \{\mathsf{Do}(A)\} \cup \{\mathsf{NotDo}(B)\} \models_{LTL} \bigcirc p\} \cup \\ & \{ \quad \neg q & \mid \Sigma \cup \{\mathsf{Do}(A)\} \cup \{\mathsf{NotDo}(B)\} \models_{LTL} \bigcirc \neg q\} \end{array}$$

and $\mathit{Progr}_B(\Sigma, A) = \mathit{state}' \cup \mathcal{A}$.

The following theorem justifies this definition of progression.

THEOREM 1. *Let $A \subseteq \mathsf{Act}$ be a set of actions that can be consistently executed in Σ, and let $B = \mathsf{Act} \setminus A$ and $\phi \in \mathcal{L}_0$. Then we have:*

$$\langle \Sigma, \Gamma \rangle \models_m \mathbf{B}((\mathsf{Do}(A) \wedge \mathsf{NotDo}(B)) \rightarrow \bigcirc\phi) \textit{ iff } \mathit{Progr}_B(\Sigma, A) \models_{LTL} \phi$$

The following corollary specifies that progression of the belief base does not violate the first rationality constraint of Definition 2, i.e., consistency of the belief base.

COROLLARY 1. *Let $A \subseteq \mathsf{Act}$ be a set of actions that can be consistently executed in Σ. Then $\mathit{Progr}_B(\Sigma, A) \not\models_{LTL} \bot$.*

Progression of Goals.

Before we explain how goals are progressed, we refine the proposal of Section 3 for the definition of the goal base. Without loss of generality, we require that for mental states $\langle \Sigma, \Gamma \rangle$ it holds that $\Sigma \subseteq \Gamma$. The satisfaction of the third rationality constraint of Definition 2, i.e., $\Gamma \models_{LTL} \Sigma$, is then trivially satisfied. The idea is that goals $\chi \in \Gamma \setminus \Sigma$ can be used to refine the inevitable beliefs.

There are two main purposes of progression of the goal base: (i) it should incorporate a mechanism for making sure that the agent does not put effort into trying to reach goals that are already believed to be reached, and (ii) it should ensure consistency of the goal base, and inclusion of the belief base in the goal base.

Regarding (i), the idea is that if, e.g., $\Diamond\phi \in \Gamma$, and at some point $\Sigma \models_{LTL} \phi$, the agent believes that this goal is reached.[3] It should then no longer put effort into trying to reach ϕ. In [16], we have proposed a progression mechanism that syntactically transforms individual LTL formulas in the goal base after each execution step, to take care of this.

The mechanism is a slight adaptation from [1, 2]. For example, $\mathit{Progr}_{\mathit{indvG}}(\Diamond\chi, \Sigma)$ is defined as $\mathit{Progr}_{\mathit{indvG}}(\chi, \Sigma) \vee \chi$, and $\mathit{Progr}_{\mathit{indvG}}(\phi, \Sigma)$ for $\phi \in \mathcal{L}_0$ is defined as $\top$ if $\Sigma \models \phi$, and $\bot$ otherwise. Therefore, if $\Sigma \models_{LTL} \phi$, $\mathit{Progr}_{\mathit{indvG}}(\Diamond\phi, \Sigma) = \top \vee \Diamond\phi$, which, using boolean simplifications, can be reduced to $\top$. The operator $\mathit{Progr}_{\mathit{indvG}}$ is defined similarly for the other connectives. We refer to [16] for further details, and omit them here for reasons of space. Progression of formulas in the goal base as described above only needs to be applied to goals $\chi \in \Gamma \setminus \Sigma$. Progression of Σ has already been specified in Definition 5.

Regarding (ii), i.e., consistency of the goal base, this is an issue that is not solved by $\mathit{Progr}_{\mathit{indvG}}$. For example, $\mathit{Progr}_{\mathit{indvG}}(\phi, \Sigma)$ is $\bot$ if ϕ does not follow from Σ, yielding an inconsistent goal. While this may be easily detected through a syntactic analysis, two goals $\Diamond p, \bigcirc q \vee \Box\neg p \in \Gamma \setminus \Sigma$ may progress to $\Diamond p$ and $\Box\neg p$ whenever $\Sigma \not\models_{LTL} p$ and $\Sigma \not\models q$, which means that Γ would become inconsistent. This is not detected by $\mathit{Progr}_{\mathit{indvG}}$, since this operator simply performs a *syntactic* transformation.

While this could be resolved by constraining the kinds of formulae in the goal base and performing boolean simplifications, this will not prevent that $\Gamma \setminus \Sigma$ becomes inconsistent with Σ. It could be the case that the agent believes a goal to be reachable, but, after choosing a particular action, no longer believes this goal to be reachable. To resolve this issue, we assume a selection function *sel* that, given a goal base Γ and belief base Σ chooses a maximal set of goals[4] λ generated by applying $\mathit{Progr}_{\mathit{indvG}}(\cdot, \Sigma)$ to Γ, such that $\Sigma \cup \lambda$ is still consistent.

Definition 6. (Progression of Goal Base)
The progression of a goal base Γ with respect to a belief base Σ where $\Sigma \subseteq \Gamma$ is defined as follows. Let $\gamma = \cup_{\chi \in \Gamma \setminus \Sigma} \mathit{Progr}_{\mathit{indvG}}(\chi, \Sigma)$. Also, let the set of candidates for new goals be $\Lambda =$

$$\{\lambda \mid \lambda \subseteq \gamma \;\&\; \lambda \cup \Sigma \not\models \bot \;\&\; \forall \gamma'(\lambda \subset \gamma' \subseteq \gamma \Rightarrow \Sigma \cup \gamma' \models \bot)\}$$

that is, Λ is the set of new goals λ such that it is a maximal subset of γ where $\lambda \cup \Sigma$ is still consistent. Finally, let *sel* be a selection function on such candidate sets. Then we define $\mathit{Progr}_G(\Gamma, \Sigma) = \mathit{sel}(\Lambda) \cup \Sigma$.

[2] Note that the kind of reasoning as described above can be done repeatedly, to derive beliefs such as $\mathbf{B}\Diamond\phi$.

[3] Note that there is a difference between believing that the goal $\Diamond\phi$ *has been* reached, which could be said to be the case if at some point $\mathbf{B}\phi$ holds, and believing that the goal *will be* reached, which is the case if $\mathbf{B}\Diamond\phi$ holds.

[4] Other kinds of selection functions can be considered as well.

Definition 7. (Mental State Transformer $\mathcal{M}$)
The *mental state transformer* function $\mathcal{M}$ is a mapping from a mental state $m = \langle \Sigma, \Gamma \rangle$ (with $\Sigma \subseteq \Gamma$), and a set of actions $A \subseteq \mathsf{Act}$ to a new mental state. Let $\Sigma' = \mathit{Progr}_B(\Sigma, A)$. Then $\mathcal{M}(m, A) =$

$$\begin{cases} \langle \Sigma', \mathit{Progr}_G(\Gamma, \Sigma') \rangle & \text{if } A \text{ can be consistently executed in } \Sigma \\ \text{undefined} & \text{otherwise} \end{cases}$$

Proposition 3. Let A be a set of actions and m be a mental state, i.e., a state that satisfies the rationality constraints of Definition 2. Then $\mathcal{M}(m, A)$ also satisfies the rationality constraints of Definition 2 whenever it is defined.

5. DECISION MAKING WITH TEMPORALLY EXTENDED GOALS

From the actions that are executable in a mental state, a GOAL agent has to make a choice as to which actions it will actually execute. The basic mechanism available in GOAL that allows an agent to make this choice, is a *rule-based action selection mechanism*. This mechanism will be described in Section 5.1. In Section 5.2, we define a mechanism on top of the rule-based mechanism, which is based on *looking ahead*.

5.1 Rule-based Action Selection

Action rules have the form **if** ψ **then** $\mathbf{do}(\mathsf{a})$ and are used to specify that action a may be selected by the agent for execution if mental state condition ψ holds; if that is the case we say that action a is *applicable*. If the preconditions of an applicable action also hold, we say that the action is *enabled*. We introduce a special predicate $\mathbf{enabled}(\mathsf{a})$ and write $m \models \mathbf{enabled}(\mathsf{a})$ when a is enabled. Formally, if **if** ψ_1 **then** $\mathbf{do}(\mathsf{a}), \ldots,$ **if** ψ_n **then** $\mathbf{do}(\mathsf{a})$ are all the action rules for action a, then $m \models \mathbf{enabled}(\mathsf{a})$ is defined as $m \models (\psi_1 \vee \ldots \vee \psi_n) \wedge \mathbf{B}\mathit{pre}_{\mathsf{a}}$. We assume that the action rules are defined such that in any mental state, at least one action is enabled.

Action rules allow agents to derive their choice of action from their beliefs and goals in the *current mental state*. Using these rules, the agent selects a subset of actions from all actions that might be executed in the state. For example, the rule **if** $\mathbf{B}(\mathit{home} \wedge \mathit{raining}) \wedge \mathbf{G}(\mathit{atWork})$ **then** $\mathbf{do}(\mathit{driveToWork})$ can be used to specify that if the agent is home and it is raining, and if he has the goal to be at work, he can select the action of driving to work.

The semantics of action selection and execution are formally specified in GOAL by means of an operational semantics [18]. In each state a GOAL agent non-deterministically selects a *maximal* subset of the enabled actions that can be executed consistently. This is formally defined in the following transition rule, which describes how an agent moves from one mental state to another.

Definition 8. (Action Rule Semantics)
Let $m = \langle \Sigma, \Gamma \rangle$ be a mental state. The labelled transition relation $\longrightarrow$ is the smallest relation induced by the following transition rule.

$$\frac{\begin{array}{c} A \subseteq \mathsf{Act} \text{ is a maximal set that can be executed consistently in } \Sigma \\ E = \{\mathsf{a} \in \mathsf{Act} \mid m \models \mathbf{enabled}(\mathsf{a})\} \end{array}}{m \xrightarrow{A \cap E} \mathcal{M}(m, A \cap E)}$$

The set $A \cap E$ in the transition rule is a maximal set of actions that are enabled, i.e. selected by action rules, and that can be consistently executed. Note that there does not need to be a unique maximal set $A \cap E$ and in that case one of these sets is non-deterministically chosen.

The execution of a GOAL agent results in a *computation trace*. We define a trace as a sequence of mental states, such that each mental state can be obtained from the previous by applying the transition rule of Definition 8. As GOAL agents are non-deterministic, the semantics of a GOAL agent is defined as the *set* of possible computations of the GOAL agent, where all computations start in the initial mental state of the agent.

Definition 9. (Meaning of a GOAL *Agent)*
A *trace* t is an infinite sequence $m_0, A_0, m_1, A_1, \ldots$ of mental states m_i and action sets A_i such that $m_i \xrightarrow{A} m_{i+1}$. The meaning $\mathcal{R}_{Agt}$ of a GOAL agent named Agt with initial mental state m_0 is the set of all traces starting in that state.

Properties.

The purpose of the rule-based action selection mechanism is to allow the agent programmer to provide the agent with a means to choose actions for execution from an available set of executable actions. That is, rather than leaving it completely up to the agent to choose which actions to execute, the rules can be used to reduce the options an agent has, i.e., the rules specify when it may make sense to execute an action.

Informally, the traces that can be generated when an agent Agt executes, i.e., the traces in $\mathcal{R}_{Agt}$, should thus be a *subset* of the traces implicitly specified by the action theory of the agent: the action theory implicitly describes *all* traces that are in principle possible according to the action theory, and the rule-based action selection mechanism allows an agent to choose between these options.

In order to make this precise, we need to transform the traces of GOAL agents, since, for the comparison, we only take into account those parts of GOAL traces that refer to what actually happens, i.e., the part of the belief base describing the current state, and the actions that are executed.

Definition 10. (GOAL *traces to LTL traces*) Let Agt be a GOAL agent with initial mental state $m_0 = \langle \Sigma_0, \Gamma_0 \rangle$ where Σ_0 is of the form $\mathit{state}_0 \cup \mathcal{A}$ and $\mathcal{A}$ is an action theory defined over the set of actions Act, and let $t = m_0, A_0, m_1, A_1, \ldots \in \mathcal{R}_{Agt}$. The function $\mathcal{T}$ takes each pair m_i, A_i in t, where $m_i = \langle \Sigma_i, \Gamma_i \rangle$ where Σ_i is of the form $\mathit{state}_i \cup \mathcal{A}$ and $B_i = \mathsf{Act} \setminus A_i$, and yields the set $s_i = \mathit{state}_i \cup \bigcup_{a_i \in A_i} \mathbf{do}(a_i) \cup \bigcup_{b_i \in B_i} \neg\mathbf{do}(b_i)$, returning $s_0, s_1, \ldots$ as the result. $\mathcal{T}$ is lifted to sets of traces in the obvious way.

The traces resulting from the application of $\mathcal{T}$ are LTL traces, where each state s_i consists of a part representing the current state, and a part representing which actions are executed in that state, i.e., the states are valuations over $\mathsf{At} \cup \mathsf{Act}$. This allows us to formulate the following theorem, which specifies that the transformed traces of a GOAL agent are a subset of the traces that satisfy the action theory. The proof is by induction on the length of prefixes of traces.

THEOREM 2. *Let Agt be a* GOAL *agent with initial mental state $m_0 = \langle \Sigma_0, \Gamma_0 \rangle$ and action theory $\mathcal{A} \subseteq \Sigma_0$, and let $\mathcal{M}(\mathcal{A}) = \{t \mid t, 0 \models_{LTL} \mathcal{A}\}$*[5]*. We then have $\mathcal{T}(\mathcal{R}_{Agt}) \subseteq \mathcal{M}(\mathcal{A})$.*

This means that the beliefs of an agent, which are derived from the current state and the action theory, are "justified" in the sense that they correspond to the actual traces as generated through the rule-based action selection mechanism. That is, if the agent has a particular belief in some mental state, it is the case that no matter what actions the agent chooses for execution, the trace will develop according to this belief. For example, if the agent believes $\mathbf{B}\Diamond\phi$ in some state, then it is the case that at some point in the future, it will believe $\mathbf{B}\phi$.

[5] See Section 3 for the notation.

5.2 Action Selection using Lookahead

Action rules as introduced in the previous section allow an agent to derive actions from what he believes and what he has as goals in the current mental state, but we argue that this does not yet account for the full role that such goals can have in the action selection mechanism of a rational agent.

As explained in Section 2, the additional mechanism we propose for using temporally extended goals in the agent's decision making is a lookahead mechanism, where the idea is to use the bounds of bounded goals as a lookahead horizon. The agent can use his belief base for determining whether he believes these bounds will eventually occur, and, from Theorem 2, we know that then the agent will eventually believe the bound. For example, an agent may believe that it will eventually be 9 o'clock, i.e., that $\mathbf{B}(\Diamond \mathit{9{:}00})$ holds. By Theorem 2, we then have that $\Diamond\mathbf{B}(\mathit{9{:}00})$ holds on all traces of the agent, i.e., no matter what the agent does, it will eventually believe that it is 9 o'clock. This means that the agent can safely look ahead until 9 o'clock, since the deadline will occur eventually.

Although the agent can only use its bounded goals of which the deadline is believed to occur eventually, to determine a lookahead horizon, it may still also take its other goals into account when choosing an action. For example, if the agent has the deadline goal $\mathit{breakfast}$ **before** atWork it may be the case that its deadline atWork will not always occur, e.g., if the agent decides to stay in bed. This means that this deadline cannot be used for looking ahead. However, if the agent can either have breakfast before going to work (while still arriving there before 9 o'clock), or not have breakfast before going to work, it should choose to have breakfast first, thereby satisfying *both* deadline goals. The approach we propose below to achieve this desired behavior, is to take bounded goals for which the bounds are believed to occur eventually as a basis. However, if the agent looks ahead until such a bound, it will also take its other goals into consideration when choosing an action. That is, first the agent uses its bounded goals to determine a lookahead horizon that is believed to occur eventually, and if multiple such bounds exist, it takes the one that is furthest away. Then, it takes *all* goals into consideration when choosing an action, trying to minimize violation of goals until the lookahead horizon.

In order to formalize this approach, we need to introduce several notions. If t is a trace, we use t^i to denote the i-th state of t, $t^{\langle h\rangle}$ to denote the prefix of t of length h, $t[i]$ to denote the suffix of t from state i, and we use $t \models_{LTL} \chi$ to abbreviate $t, 0 \models_{LTL} \chi$. We use $AG(m) = \{\chi \mid m \models \mathbf{Goal}(\chi)\}$ to denote all goals of the agent in mental state m. Note that we use the operator $\mathbf{Goal}$ in this definition, reflecting the fact that we only use those goals for decision making that will not be reached inevitably, i.e., excluding goals such as $\mathit{8{:}00}$ **before** $\mathit{9{:}00}$. We use $BG(m)$ to denote the set of bounded goals of a mental state m of which the agent believes the bound to occur eventually, defined as $\{\varphi\ \mathbf{before}\ \psi \mid \varphi\ \mathbf{before}\ \psi \in AG(m)\ \&\ m \models \mathbf{B}\Diamond\psi\} \cup \{\varphi\ \mathbf{until}\ \psi \mid \varphi\ \mathbf{until}\ \psi \in AG(m)\ \&\ m \models \mathbf{B}\Diamond\psi\}$. Given a set of bounded goals BG, we define the set of horizons of BG as $H(BG) = \{\psi \mid \varphi\ \mathbf{before}\ \psi \in BG$ or $\varphi\ \mathbf{until}\ \psi \in BG\}$. Given a trace t and a set of horizons H, we define the maximum horizon on t as $\mathit{maxH}(t, H) = i$, if there is an i is such that there is a $\psi \in H$ with $t^i \models \mathbf{B}\psi$ and for all $\psi' \in H$ there is $j \leq i$ with $t^i \models \mathbf{B}\psi'$ and there is no $j < i$ with $t^j \models \mathbf{B}\psi$, and $\mathit{maxH}(t, H) = 0$, otherwise, i.e., if $H = \emptyset$.

Let $\mathbf{Goal}\chi$ be a goal. We define that this goal is violated on a finite prefix of length h of a trace $t \in \mathcal{R}_{Agt}$ if there does not exist an infinite continuation t' of the transformed prefix, such that $\mathcal{T}(t)^{\langle h\rangle}t' \models_{LTL} \chi$. We use $VG(X, t, h)$, where X is a set of goals, to denote the set of goals from X that are violated on $t^{\langle h\rangle}$.

The following function σ^{min}_{Agt} is used to formally specify the semantics of a GOAL agent Agt that uses lookahead for decision making with temporally extended goals. Broadly speaking, the function takes the traces $\mathcal{R}_{Agt}$ that are generated by the action selection rules of agent Agt, and filters out all traces that do not minimize violation of goals until the maximum lookahead horizon. We define the function inductively on the set of traces $\mathcal{R}_{Agt}$ and on the time point i on such a trace. An agent will use its goals to choose an appropriate action from the start, i.e. time 0, which explains why the base case of the inductive definition starts at -1. The base case defines the starting point of traces $\mathcal{R}_{Agt}$ that need to be filtered. The idea is that the filter function $\sigma^{min}_{Agt}(i)$ returns all traces t that minimize violation of the goals of the agent on a finite prefix of length $\mathit{maxH}(t, H(BG(t^i)))$, starting from state i.

Definition 11. (Decision-Making Function σ) Let Agt be an agent. σ^{min}_{Agt} is a function that chooses an action that minimizes violation of goals, as far as it can see this coming within the maximum horizon of a trace, defined as follows:

$$\begin{array}{lcl}
\sigma^{min}_{Agt}(-1) & = & \mathcal{R}_{Agt}, \\
\sigma^{min}_{Agt}(i) & = & \{t \in \sigma^{min}_{Agt}(i-1) \mid \neg\exists t' \in \sigma^{min}_{Agt}(i-1) : \\
 & & VG(AG(t'^{i}), t, \mathit{maxH}(t, H(BG(t'^{i})))) \subset \\
 & & VG(AG(t^i), t, \mathit{maxH}(t, H(BG(t^i))))\}
\end{array}$$

The semantics of a GOAL agent Agt that uses lookahead for choosing actions, on top of a rule-based action selection mechanism, is then defined as follows.

Definition 12. (Meaning of GOAL *Agent using Lookahead)* The meaning R^{LA}_{Agt} of a GOAL agent Agt is defined by:

$$\mathcal{R}^{LA}_{Agt} = \bigcap_{i=-1}^{\infty} \sigma^{min}_{Agt}(i)$$

One could define refinements of this semantics, e.g., such that the agent tries to minimize violation of goals, but also maximize achievement of goals, or such that it takes into account importance of goals rather than simply minimizing the number of goals that is violated. However, for reasons of space we do not include such definitions.

The following proposition states that the lookahead mechanism of Definition 12 does not change the rule-based semantics if the agent never has any bounded goals of which he believes the bound to occur eventually.

Proposition 4. Let Agt be an agent. If for all $t \in \mathcal{R}_{Agt}$ and i it holds that $BG(t^i) = \emptyset$, then $\mathcal{R}^{LA}_{Agt} = \mathcal{R}_{Agt}$.

6. CONCLUSION

The main contribution of this paper is that we have shown how temporal logic can be used for a qualitative and more intuitive specification of horizons for bounded goals, and we have proposed a lookahead mechanism for reasoning with bounded goals in an agent programming framework, based on the use of these horizons. We have shown how temporal beliefs can be used to ensure that the agent can derive whether horizons will eventually occur, making sure that he will only look ahead until a horizon if this is the case. Moreover, since agents need to maintain a rational balance among their beliefs and goals, we have defined several rationality constraints, and we have provided progression mechanisms for beliefs and goals that maintain the rationality constraints.

The main difference between this work and [16] is that instead of looking ahead a fixed number of steps as proposed in [16], in this paper we propose the specification of qualitative bounds that define the lookahead horizon using bounded goals. Moreover, in contrast with [16] we propose the use of temporal beliefs and mechanisms for maintaining a rational balance between beliefs and goals in this context.

This work has been inspired by the use of temporally extended goals in planning. An important difference between agents and planners is that agents more explicitly differentiate between their beliefs and their goals, warranting the definition of rationality constraints and mechanisms for maintaining these during execution of the agent. Also, while planners provide a finite plan to realize a set of initial goals, agents continually take the changing beliefs and goals into account during execution, when making decisions. There are thus differences between planning and agent programming, and historically agent programming languages have even been proposed to circumvent the high computational complexity of planning. Nevertheless, we believe that the current state of the art in both agent programming and planning may yield interesting results when it comes to synergies between planning and agent programming (see also [21]). Deadlines have also been considered in the context of deontic logic [6], in which it is investigated what it means that an agent is obliged to meet a deadline. To the best of our knowledge, our work is the first to focus on the use of deadline goals in agent programming.

As for future research, more work is needed with respect to the computational properties of the framework and on how to implement the ideas efficiently, in particular also for the selection function of Definition 6. We envisage that planning techniques can be adapted to implement lookahead, but more research is needed to investigate exactly how to do this. More experimentation is needed to identify common temporal goal patterns and to develop techniques for supporting the programmer when using temporally extended goals, e.g., comparable to [25], in which a declarative service flow language is proposed that has an appealing graphical notation, but that is grounded in temporal logic.

We conclude by remarking that this paper extends the work reported in [14]. In [14] it is shown that the GOAL programming logic of [9] can be embedded in a minor variation of the logic of [7]. The same embedding result presented in [14] is applicable to GOAL agents as defined here. In this paper we have provided a concrete instantiation of the abstract agents introduced in [14]. As a result, the connection with [7] has become even stronger as we now also have the principle $\mathbf{B}\chi \rightarrow \mathbf{G}\chi$ in our semantics.

7. REFERENCES

[1] F. Bacchus and F. Kabanza. Planning for temporally extended goals. pages 1215–1222, 1996.

[2] F. Bacchus and F. Kabanza. Using temporal logics to express search control knowledge for planning. *Artificial Intelligence*, 16:123–191, 2000.

[3] C. Baral, T. Eiter, M. Bjäreland and M. Nakamura. Maintenance goals of agents in a dynamic environment: Formulation and policy construction. *Artificial Intelligence*, 172(12-13):1429–1469, 2008.

[4] M. Bienvenu, C. Fritz, and S. A. McIlraith. Planning with qualitative temporal preferences. In *KR'06*, pages 134–144, 2006.

[5] L. Braubach, A. Pokahr, D. Moldt, and W. Lamersdorf. Goal representation for BDI agent systems. In *ProMAS'04*, volume 3346 of *LNAI*, pages 44–65. Springer, 2005.

[6] J. Broersen, F. Dignum, V. Dignum, and J.-J. C. Meyer. Designing a deontic logic of deadlines. In *DEON'04*, volume 3065 of *LNCS*, pages 43–56. Springer, 2004.

[7] P. R. Cohen and H. J. Levesque. Intention Is Choice with Commitment. *Artificial Intelligence*, 42:213–261, 1990.

[8] M. Dastani, M. B. van Riemsdijk, and J.-J. Ch. Meyer. Goal types in agent programming. In *ECAI'06*, volume 141 of *Frontiers in Artificial Intelligence and Applications*, pages 220–224. IOS Press, 2006.

[9] F. de Boer, K. V. Hindriks, W. van der Hoek, and J.-J. C. Meyer. A Verification Framework for Agent Programming with Declarative Goals. *Journal of Applied Logic*, 5(2):277–302, 2007.

[10] S. Duff, J. Harland, and J. Thangarajah. On proactivity and maintenance goals. In *AAMAS'06*, pages 1033–1040, Hakodate, 2006.

[11] E. Emerson. Temporal and Modal Mogic. In J. v. Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B. North-Holland Publishing Company, 1990.

[12] R. Fikes and N. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[13] K. V. Hindriks, F. S. de Boer, W. van der Hoek, and J.-J. C. Meyer. Agent Programming with Declarative Goals. In *ATAL'00*, pages 228–243, 2000.

[14] K. V. Hindriks and W. van der Hoek. Goal agents instantiate intention logic. In *JELIA'08*. 2003, 2008.

[15] K. V. Hindriks and B. van Riemsdijk. Satisfying maintenance goals. In *DALT'07*, volume 4897, 2008.

[16] K. V. Hindriks and B. van Riemsdijk. Using temporal logic to integrate goals and qualitative preferences into agent programming. In *DALT'08*, 2008.

[17] M. C. Mayer, C. Limongelli, A. Orlandini, and V. Poggioni. Linear temporal logic as an executable semantics for planning languages. *Journal of Logic, Lang and Information*, 16, 2007.

[18] G. D. Plotkin. A Structural Approach to Operational Semantics. Technical Report DAIMI FN-19, University of Aarhus, 1981.

[19] A. S. Rao and M. P. Georgeff. Intentions and Rational Commitment. Technical report, AAII, 1993.

[20] R. Reiter. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in honor of John McCarthy*, pages 359–380. Academic Press, New York, 1991.

[21] S. Sardiña, L. de Silva, and L. Padgham. Hierarchical planning in BDI agent programming languages: a formal approach. In *AAMAS'06*, pages 1001–1008, 2006.

[22] H. Simon. Bounded rationality and organizational learning. *Organization Science*, 2(1):125–134, 1991.

[23] T. C. Son and E. Pontelli. Planning with preferences using logic programming. *Theory and Practice of Logic Programming*, 6(5):559–607, 2006.

[24] J. Thangarajah, J. Harland, D. Morley, and N. Yorke-Smith. Suspending and resuming tasks in BDI agents. In *AAMAS'08*, pages 405–412, 2008.

[25] W. M. P. van der Aalst and M. Pesic. DecSerFlow: Towards a truly declarative service flow language. In *WS-FM'06*, volume 4184 of *LNCS*, pages 1–23. Springer, 2006.

[26] M. Winikoff, L. Padgham, J. Harland, and J. Thangarajah. Declarative and procedural goals in intelligent agent systems. In *KR2002*, Toulouse, 2002.