Non-monotonic Temporal Goals

Ricardo Gonçalves, Matthias Knorr, João Leite, and Martin Slota

CENTRIA, Universidade Nova de Lisboa, Portugal

Abstract. In this paper we introduce a logic programming based framework which allows the representation of conditional non-monotonic temporal beliefs and goals in a declarative way. We endow it with stable model like semantics that allows us to deal with conflicting goals and generate possible alternatives. We show that our framework satisfies some usual properties on goals and that it allows imposing alternative constraints on the interaction between beliefs and goals. We prove the decidability of the usual reasoning tasks and show how they can be implemented using an ASP solver and an LTL reasoner in a modular way, thus taking advantage of existing LTL reasoners and ASP solvers.

1 Introduction

Mental attitudes such as beliefs, goals, and intentions are well-known to be fundamental for representing autonomous rational agents [4,12,13,5]. Roughly, beliefs represent the agent's knowledge about the state of the world, goals represent states the agent aims at achieving, and intentions are the goals that the agent commits to pursue. We focus on the representation and reasoning about declarative beliefs and goals.

One fundamental ingredient when modeling goals is the notion of time. Goals usually refer to some state of affairs that the agent aims to maintain or achieve sometime in the future. For example, an agent might have the goal to maintain a positive balance on her bank account during the entire month to avoid fines, or to study before the next week's exam. Temporal logic has been shown to be flexible and expressive for representing different goal types [7,10,2], and several works in the literature modeling mental attitudes of agents are based on (extensions of) temporal logic. Namely, [5,10,16,2] are based on Linear Temporal Logic (LTL), while [12] is based on Computational Tree Logic (CTL*).

Another fundamental ingredient is the possibility to model defeasible and conflicting information. As argued in [2], it is quite common that goals have a conditional form and admit exceptions, so the adoption of new beliefs or goals may cause the retraction of some of the agent's current goals. In particular, we may encounter conflicting goals that cannot be pursued at the same time, in which case we have to consider alternative sets of goals. For example, an agent may want to go to Paris for the weekend, but also to London. These goals are conflicting and cannot be achieved together.

Therefore, representation and reasoning about goals would greatly benefit from an approach combining the temporal and non-monotonic aspects. However,

most of the work on beliefs and goal is monotonic [12,5,10,16], and therefore does not allow us to represent defeasible beliefs and goals. The work in [15] introduces a non-monotonic logic for conditional goals based on default logic, but it does not consider temporal formulas. In [2], a non-monotonic extension of LTL is defined that allows for expressing goals with exceptions, but it does not handle conflicting goals nor does it consider beliefs, therefore not allowing us to model the interaction between these and goals.

In this paper, we bridge the gap between temporal and non-monotonic goal languages by introducing a general non-monotonic goal framework. The language obtained is expressive enough to represent conditional beliefs and goals over complex temporal formulas, and it also allows reasoning about conflicting goals. The semantics defined in the spirit of stable models/answer set programming (ASP) [8] not only endows it with a purely declarative semantics, but also supports a novel perspective on dealing with alternative sets of goals in which each stable model can be seen as a possible consistent set of goals that an agent might adopt. Besides satisfying some usual properties on goals, our framework is also general and flexible enough to represent different constraints on the interaction between beliefs and goals in a simple way. We show decidability and how to implement our framework using existing LTL and ASP solvers.

The paper is structured as follows. In Sect. 2, we introduce the basic language for reasoning about beliefs and goals over a temporal logic. Then, in Sect. 3, we define a non-monotonic framework for representing defeasible beliefs and goals, along with its semantics, and prove some properties of our framework in Sect. 4. We discuss decidability and implementation in Sect. 5, compare with related work in Sect. 6, and conclude in Sect. 7.

2 Logic of beliefs and goals

In this section we introduce the language for representing beliefs and goals, which is based on temporal logic. Temporal logic has been shown to be quite flexible and expressive for representing different goal types [7,10,2]. Since our approach is modular on the temporal component, and in order to ease the presentation, we follow [5,10,16,2] and work in this paper with Linear Temporal Logic [11].

2.1 Linear temporal logic

Here, we introduce Linear Temporal Logic (LTL). The *language of LTL*, $\mathcal{L}_{\text{LTL}}$, is built from a set of propositional symbols $\mathcal{P}$ using the usual classical connectives $\mathbf{t}, \sim, \sqcap, \sqcup, \Rightarrow$, the unary temporal operator $\bigcirc$ (next) and the binary temporal operator $\mathcal{U}$ (until). Other temporal operators can be defined by abbreviation: $\Diamond\varphi := \mathbf{t}\,\mathcal{U}\varphi$ (eventually), $\Box\varphi := \sim\Diamond\sim\varphi$ (always), $\varphi\mathcal{B}\psi := \sim(\sim\varphi\,\mathcal{U}\psi)$ (before).

The semantics for $\mathcal{L}_{\text{LTL}}$ is given as interpretation sequences of classical valuations. Formally, an *LTL interpretation* is a sequence $m = (m_i)_{i\in\mathbb{N}}$ where $m_i \subseteq \mathcal{P}$ for each $i \in \mathbb{N}$. The *satisfaction of an LTL formula* by an LTL interpretation $m = (m_i)_{i\in\mathbb{N}}$ at a point i is defined inductively as follows:

- $m, i \Vdash \mathbf{t}$ for every $i \in \mathbb{N}$;
- $m, i \Vdash p$ if $p \in m_i$, for $p \in \mathcal{P}$;
- $m, i \Vdash \sim\varphi$ if $m, i \nVdash \varphi$;
- $m, i \Vdash \varphi_1 \sqcap \varphi_2$ if $m, i \Vdash \varphi_1$ and $m, i \Vdash \varphi_2$;
- $m, i \Vdash \varphi_1 \sqcup \varphi_2$ if $m, i \Vdash \varphi_1$ or $m, i \Vdash \varphi_2$;
- $m, i \Vdash \varphi_1 \Rightarrow \varphi_2$ if $m, i \nVdash \varphi_1$ or $m, i \Vdash \varphi_2$;
- $m, i \Vdash \bigcirc\varphi$ if $m, i+1 \Vdash \varphi$;
- $m, i \Vdash \varphi_1 \mathcal{U} \varphi_2$ if $m, j \Vdash \varphi_2$ for some $j \geq i$ and $m, k \Vdash \varphi_1$ for every $i \leq k < j$.

We say that an LTL interpretation m is a *model* of an LTL formula φ, denoted by $m \Vdash_{\text{LTL}} \varphi$, if $m, 0 \Vdash \varphi$. This is the so-called anchored version of LTL. Given a set Φ of LTL formulas, we denote by $Mod(\Phi)$ the set of LTL models of all formulas in Φ. An LTL formula φ is *valid* if $m \Vdash_{\text{LTL}} \varphi$ for every LTL interpretation m. The consequence relation $\vDash_{\text{LTL}}$ is defined as usual, i.e., $\Phi \vDash_{\text{LTL}} \varphi$ if, for every interpretation m, $m \Vdash \varphi$ whenever $m \Vdash \psi$ for every $\psi \in \Phi$. A set of LTL formulas Φ is an *LTL theory* if, for every $\delta \in \mathcal{L}_{\text{LTL}}$, if $\Phi \vDash \delta$, then $\delta \in \Phi$. We denote by Th_{LTL} the set of all theories over the language $\mathcal{L}_{\text{LTL}}$. Given a set Φ of LTL formulas, we denote by $\Phi^{\vDash_{\text{LTL}}}$ the least LTL theory containing Φ, i.e., the deductive closure of Φ. As it is usual for monotonic logics [19], an important property of Th_{LTL} is the fact that $\langle Th_{\text{LTL}}, \subseteq \rangle$ is a complete lattice, i.e., $\langle Th_{\text{LTL}}, \subseteq \rangle$ is a partial order and for every $\mathcal{A} \subseteq Th_{\text{LTL}}$ the set $\bigcap_{T \in \mathcal{A}} T$ is again a theory over $\mathcal{L}_{\text{LTL}}$.

2.2 Logic of beliefs and goals

We now define the language for specifying beliefs and goals. We start by defining the syntax, which is built on top of the LTL language.

Definition 1. *The* language of beliefs and goals*, denoted by $\mathcal{L}_{BG}$, is defined as:*

$$\delta := \mathbf{B}(\varphi) \mid \mathbf{G}(\varphi) \mid \neg\delta \mid \delta \wedge \delta$$

where φ is an LTL formula. A formula of the form $\mathbf{B}(\varphi)$ is called a belief atom *and one of the form $\mathbf{G}(\varphi)$ is called a* goal atom*.*

Note that the other usual classical connectives can be obtained as abbreviation $\delta_1 \vee \delta_2 := \neg(\neg\delta_1 \wedge \neg\delta_2)$ and $\delta_1 \rightarrow \delta_2 := \neg\delta_1 \vee \delta_2$. We denote by $\mathcal{L}_B$ the set of formulas of $\mathcal{L}_{BG}$ that only contain the belief operator $\mathbf{B}$, i.e., those formulas which are built from belief atoms using the classical connectives. We call $\mathcal{L}_B$ the *belief language* and its elements the *belief formulas*. In the same way, we define the *goal language*, $\mathcal{L}_G$, as the set of formulas of $\mathcal{L}_{BG}$ that only contain the goal operator $\mathbf{G}$. Its elements are called *goal formulas*. Also note that, for simplicity, we follow [7,10,15] and do not allow temporal operators outside the scope of a belief or goal operator, nor nesting of belief and goal operators.

Since the language $\mathcal{L}_{BG}$ is built over belief and goal atoms, we define its semantics based on an *interpretation* $T = \langle T_b, T_g \rangle$, i.e., a pair of LTL theories. The first element of the pair is used to interpret belief atoms and the second element to interpret goal atoms. An interpretation $T = \langle T_b, T_g \rangle$ is *consistent* if both T_b and T_g are different from $\mathcal{L}_{\text{LTL}}$. We define the satisfaction of a formula in $\mathcal{L}_{BG}$ with respect to a pair $\langle T_b, T_g \rangle$ of LTL theories as follows:

$$\begin{array}{ll}
\langle T_b, T_g \rangle \Vdash \mathbf{B}(\varphi) & \text{if } T_b \vDash_{\text{LTL}} \varphi \\
\langle T_b, T_g \rangle \Vdash \mathbf{G}(\varphi) & \text{if } T_g \vDash_{\text{LTL}} \varphi \\
\langle T_b, T_g \rangle \Vdash \neg\delta & \text{if } \langle T_b, T_g \rangle \nVdash \delta \\
\langle T_b, T_g \rangle \Vdash \delta_1 \wedge \delta_2 & \text{if } \langle T_b, T_g \rangle \Vdash \delta_1 \text{ and } \langle T_b, T_g \rangle \Vdash \delta_2
\end{array}$$

Note that we do not impose any constraints on the relation between the LTL theories T_b and T_g, unlike [7,10] where $T_g \vDash_{\text{LTL}} T_b$ is assumed. Our aim here is to be as general as possible, which is witnessed in Section 4 where we show that our framework is flexible enough to impose such constraints in a simple way.

We can now define the consequence relation over the language $\mathcal{L}_{BG}$.

Definition 2. *Given $\delta \in \mathcal{L}_{BG}$ and $\Gamma \subseteq \mathcal{L}_{BG}$, the consequence relation over $\mathcal{L}_{BG}$ is defined as $\Gamma \vDash \delta$ iff, for interpretation $\langle T_b, T_g \rangle$, we have that $\langle T_b, T_g \rangle \Vdash \delta$ whenever $\langle T_b, T_g \rangle \Vdash \psi$ for every $\psi \in \Gamma$. We say that δ is* valid *if $\emptyset \vDash \delta$.*

3 Non-monotonic belief and goal specification

In this section, we present a non-monotonic framework for specifying conditional non-monotonic temporal beliefs and goals. The framework is based on logic programs built over the language $\mathcal{L}_{BG}$ introduced in the previous section.

3.1 Belief and goal bases

Belief bases and goal bases represent the agent's beliefs and goals respectively, and are usually defined as sets of formulas from which we can deduce the beliefs and goals of the agent. For example, in [10], sets of LTL formulas are used.

In this work we generalize this assumption by representing belief and goal bases as sets of non-monotonic rules over $\mathcal{L}_{BG}$, similar to those in Logic Programming, making it possible to represent conditional and defeasible goals.

A *rule* r is of the form

$$\varphi \leftarrow \psi_1, \ldots, \psi_n, \mathit{not}\ \delta_1, \ldots, \mathit{not}\ \delta_m \tag{1}$$

where the *head* of r, φ, and each element of its *body*, $\psi_1, \ldots, \psi_n, \delta_1, \ldots, \delta_m$, is either a goal atom or a belief atom. Like in a logic programming rule, the symbol $\leftarrow$ represents rule implication, the symbol "," represents conjunction and the symbol *not* represents default negation. Thus, r represents that φ holds whenever $\psi_1, \ldots, \psi_n$ hold and $\delta_1, \ldots, \delta_m$ are not known to hold. A rule is called *positive* if it does not contain any occurrence of *not*, and *fact* if its body is empty.

A *belief rule* is a rule of the form (1) where the head φ and all elements of the body are belief atoms. A *goal rule* is a rule of the form (1) where the head φ is a goal atom. A *belief base* $\mathcal{B}$ is a set of belief rules and a *goal base* $\mathcal{G}$ is a set of goal rules. A belief base (goal base) is called *positive* if all its rules are positive.

Definition 3. *An* agent configuration *is a pair* $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$ *where* $\mathcal{B}$ *is a belief base and* $\mathcal{G}$ *is a goal base. An agent configuration* $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$ *is said to be* positive *if both the belief base* $\mathcal{B}$ *and the goal base* $\mathcal{G}$ *are positive.*

Example 4. Consider the simple example about choosing a means of transport.

$$\mathbf{B}(\Diamond strike) \leftarrow \mathbf{B}(strikeInNews) \tag{2}$$
$$\mathbf{G}(\Diamond work) \leftarrow not\ \mathbf{G}(\Diamond beach) \tag{3}$$
$$\mathbf{G}(\Diamond beach) \leftarrow not\ \mathbf{G}(\Diamond work) \tag{4}$$
$$\mathbf{G}(\Diamond bike) \leftarrow \mathbf{G}(\Diamond beach) \tag{5}$$
$$\mathbf{G}(\Diamond car) \leftarrow \mathbf{G}(\Diamond work), \mathbf{B}(\Diamond strike) \tag{6}$$
$$\mathbf{G}(\Diamond(train \sqcup bus)) \leftarrow \mathbf{G}(\Diamond work), not\ \mathbf{B}(\Diamond strike) \tag{7}$$
$$\mathbf{G}(ticket\ \mathcal{B}\ (train \sqcup bus)) \leftarrow \mathbf{G}(\Diamond(train \sqcup bus)) \tag{8}$$
$$\mathbf{G}(money\ \mathcal{B}\ ticket) \leftarrow \mathbf{G}(\Diamond ticket) \tag{9}$$

Informally, an agent believes that there will be a strike if she sees that in the news (2). Rules (3) and (4) represent conflicting goals, i.e., either the agent has the goal to go to the beach or the goal to go to work, not both. In the former case the agent also has the goal to take the bike (5). In the latter case, depending on whether the agent believes that there will be a strike or not, she has the goal to to go by car (6) or the goal to go by train or by bus (7). Moreover, if the agent has the goal to go by train or by bus, then she also has the goal to buy a ticket before that (8). Finally, if the agent has the goal to buy a ticket, then she has the goal to withdraw money first (9).

3.2 Semantics

The definition of a semantics for belief and goal bases is not straightforward due to their complex language. Recall that belief and goal atoms in the rules may contain arbitrary LTL formulas. Thus, unlike, e.g., first-order atoms, belief or goal atoms may not be independent; for example, the goal atoms $\mathbf{G}(\Box(p \vee q))$, $\mathbf{G}(\Diamond \neg p)$ and $\mathbf{G}(\Diamond q)$ are not. To overcome this difficulty, our notion of interpretation accounts for interdependence between such atoms: since $\{\Box(p \vee q), \Diamond \neg p\} \vDash_{\text{LTL}} \Diamond q$ and since any LTL theory is closed under logical consequence, any interpretation $\langle T_b, T_g \rangle$ satisfying both $\mathbf{G}(\Box(p \vee q))$ and $\mathbf{G}(\Diamond \neg p)$ must also satisfy $\mathbf{G}(\Diamond q)$.

Satisfaction of rules in interpretations and the notion of model for agent configurations can thus be defined in a standard way.

Definition 5. *An interpretation* $T = \langle T_b, T_g \rangle$ satisfies *a rule of the form (1), if* $T \Vdash \varphi$ *whenever* $T \Vdash \psi_i$ *for every* $i \in \{1, \ldots, n\}$ *and* $T \nVdash \delta_j$ *for every* $j \in \{1, \ldots, m\}$. *An interpretation is a* model *of an agent configuration* $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$ *if it satisfies every rule of* $\mathcal{B} \cup \mathcal{G}$. *We denote by* $Mod(\mathcal{C})$ *the set of models of* $\mathcal{C}$.

The ordering over interpretations can easily be defined component-wise: given two interpretations $T = \langle T_b, T_g \rangle$ and $T' = \langle T'_b, T'_g \rangle$ we write $T \leq T'$ if $T_b \subseteq T'_b$ and $T_g \subseteq T'_g$. Using this ordering, the notions of minimal and least interpretations can be defined in the usual way.

We are particularly interested in such minimal interpretations and obtain them in a way similar to the stable model semantics. For that purpose, we start by considering positive agent configurations and adapt a well-known result from logic programs saying that every positive agent configuration has a least model.

Theorem 6. *Every positive agent configuration has a least model.*

Based on the semantics for positive agent configurations, we now define the stable model semantics of an agent configuration that can have default negation.

Definition 7. *Let $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$ be an agent configuration and $T = \langle T_b, T_g \rangle$ an interpretation. The agent configuration $\frac{\mathcal{C}}{T}$ is obtained from $\mathcal{C}$ by:*

- *removing from $\mathcal{B}$ and $\mathcal{G}$ all rules which contain not φ such that $T \Vdash \varphi$;*
- *removing not φ from the remaining rules of $\mathcal{B}$ and $\mathcal{G}$.*

Since $\frac{\mathcal{C}}{T}$ is a positive agent configuration, it has a unique least model T'. We define $\Gamma_{\mathcal{C}}(T) = T'$.

An interpretation $T = \langle T_b, T_g \rangle$ is a stable model *of an agent configuration $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$ if it is consistent and $\Gamma_{\mathcal{C}}(T) = T$. We denote by $SM(\mathcal{C})$ the set of all stable models of $\mathcal{C}$.*

Each stable model can be thought of as a possible consistent set of goals an agent might adopt, which is why T_b and T_g are required to be consistent. Thus, from the point of view of multi-agent systems, each such stable model represents a possible consistent alternative that the agent can adopt as her set of intentions, i.e., those goals that the agent commits to. We note that choosing a particular set of intentions is out of the scope of this paper since, as common in agent architectures, this is dealt with on a meta-level. Still, the following (well-known) entailment relations can be defined.

Definition 8. *Let $\mathcal{C}$ be an agent configuration. A formula δ is true under the stable model semantics of $\mathcal{C}$, denoted by $\mathcal{C} \models_{\text{SM}} \delta$, if it is satisfied by every stable model of $\mathcal{C}$. A formula δ is true under the credulous stable model semantics of $\mathcal{C}$, denoted by $\mathcal{C} \models_{\text{CSM}} \delta$, if it is satisfied by some stable model of $\mathcal{C}$.*

From the agent's perspective, skeptical entailment $\models_{\text{SM}}$ represents the goals which she will have independently of the particular set of goals she commits to, while credulous entailment $\models_{\text{CSM}}$ can be used if an agent needs to know whether it is possible that she might adopt that goal.

Example 9. Recall the agent configuration of Ex. 4. Rules (3) and (4) represent the conflicting goals of going to the beach or going to work. This is captured in the semantics by the existence of two stable models. The first one has as consequences the goals $\mathbf{G}(\Diamond beach)$ and $\mathbf{G}(\Diamond bike)$. The second stable model entails

$\mathbf{G}(\Diamond work)$, $\mathbf{G}(\Diamond(train \sqcup bus))$, $\mathbf{G}(ticket\ \mathcal{B}\ (train \sqcup bus))$ and $\mathbf{G}(money\ \mathcal{B}\ ticket)$. Note the fundamental role of complex temporal reasoning in the calculation of the stable models. For example, in the case of the second stable model, rule (9) only fires because $\mathbf{G}(\Diamond ticket)$ follows from $\mathbf{G}(ticket\ \mathcal{B}\ (train \sqcup bus))$ and $\mathbf{G}(\Diamond(train \sqcup bus))$ together, since $\{ticket\ \mathcal{B}\ (train \sqcup bus), \Diamond(train \sqcup bus)\} \models_{\text{LTL}} \Diamond ticket$. Moreover, since the stable models are closed under consequence and since $\{money\ \mathcal{B}\ ticket, \Diamond ticket\} \models_{\text{LTL}} \Diamond money$ we have that the second stable model also entails $\mathbf{G}(\Diamond money)$.

Adding the fact $\mathbf{B}(strikeInNews) \leftarrow$ to the agent configuration of Ex. 4 does not affect the first stable model, but it affects the second. With this extra rule the goals $\mathbf{G}(\Diamond(train \sqcup bus))$, $\mathbf{G}(ticket\ \mathcal{B}\ (train \sqcup bus))$ and $\mathbf{G}(money\ \mathcal{B}\ ticket)$ no longer follow from the second stable model, but now the goal $\mathbf{G}(\Diamond car)$ follows.

4 Properties

We can find a number of properties in the literature that a logical language modeling goals should exhibit. Some are more consensual than others, but that is not the topic of this paper, we rather point to [18]. The aim of this section is to show that some common properties of beliefs and goals hold in our framework and that other approaches that impose additional conditions on the relation between goals and beliefs can be covered.

A usual property of stable models is that they are minimal.

Proposition 10. *Let $\mathcal{C}$ be an agent configuration. If $T = \langle T_b, T_g \rangle$ is a stable model of $\mathcal{C}$, then there is no stable model $T' = \langle T'_b, T'_g \rangle$ of $\mathcal{C}$ such that $T' < T$.*

Modal logic has been used for modeling beliefs and goals of agents [12,5]. The belief operator is usually described using modal logic $KD45$ and the goal operator using the modal logic KD. It is therefore natural to check if these modal axioms hold in our logic. Note that even though we state the following propositions for $\models_{\text{SM}}$, all of them also hold for $\models_{\text{CSM}}$.

Proposition 11. *Let $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$ be an agent configuration. The following holds for all LTL formulas φ and ψ:*

(K_b) $\mathcal{C} \models_{\text{SM}} \mathbf{B}(\varphi \Rightarrow \psi) \rightarrow (\mathbf{B}(\varphi) \rightarrow \mathbf{B}(\psi))$;
(K_g) $\mathcal{C} \models_{\text{SM}} \mathbf{G}(\varphi \Rightarrow \psi) \rightarrow (\mathbf{G}(\varphi) \rightarrow \mathbf{G}(\psi))$;
(D_b) $\mathcal{C} \models_{\text{SM}} \mathbf{B}(\varphi) \rightarrow \neg\mathbf{B}(\sim\varphi)$;
(D_g) $\mathcal{C} \models_{\text{SM}} \mathbf{G}(\varphi) \rightarrow \neg\mathbf{G}(\sim\varphi)$.

The above proposition states that both the axioms K and D hold for both the belief and the goal operator in every agent configuration. Note that we do not consider the modal axioms 4 and 5, which are usually associated with the belief operator. The reason is that these axioms involve formulas with nested beliefs, and therefore cannot be represented in our language.

A property that appears for example in [12] is that beliefs and goals should be closed under implication. Our framework satisfies this property.

Proposition 12. *Let $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$ be an agent configuration. The following holds for all LTL formulas φ and ψ:*

- $\mathcal{C} \models_{\text{SM}} (\mathbf{B}(\varphi \Rightarrow \psi) \wedge \mathbf{B}(\varphi)) \rightarrow \mathbf{B}(\psi)$;
- $\mathcal{C} \models_{\text{SM}} (\mathbf{G}(\varphi \Rightarrow \psi) \wedge \mathbf{G}(\varphi)) \rightarrow \mathbf{G}(\psi)$;

Our notion of agent configuration does not have any built-in constraints on the interaction between beliefs and goals, unlike [12,5]. Our language is designed to be general, in the sense that we do not impose these restrictions, yet expressive enough to allow the representation of such constraints if desired.

A constraint that [5,10] impose is the so-called *realism constraint*. Intuitively this means that an agent should have as goals all her beliefs. Although [12] considers this too restrictive, if we want to impose such a restriction in a given agent configuration $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$, we just need to add to $\mathcal{G}$ a rule $\mathbf{G}(\varphi) \leftarrow \mathbf{B}(\varphi)$ for every LTL formula φ appearing in $\mathcal{C}$. Let $\mathcal{C}_{\text{Real}}$ be the resulting agent configuration.

A more or less opposite condition is that an agent should not have a goal that she believes is already the case. In [18] this property is described as goals should be *unachieved* (Un). If we want to impose this restriction in a given agent configuration $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$, we just need to substitute every rule $\mathbf{G}(\varphi) \leftarrow \mathit{body}$ of $\mathcal{G}$ by the rule $\mathbf{G}(\varphi) \leftarrow \mathit{body}, \mathit{not}\ \mathbf{B}(\varphi)$. Let $\mathcal{C}_{\text{Un}}$ be the resulting agent configuration.

Another commonly considered constraint, described in [18] as goals should be *possible* (Poss), is that an agent should not have a goal that he believes to be impossible. This restriction can also be applied to an agent configuration $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$ by substituting every rule $\mathbf{G}(\varphi) \leftarrow \mathit{body}$ of $\mathcal{G}$ by the rule $\mathbf{G}(\varphi) \leftarrow \mathit{body}, \mathit{not}\ \mathbf{B}(\sim\varphi)$. Denote by $\mathcal{C}_{\text{Poss}}$ the resulting agent configuration.

The following proposition states that the above constructions imply that the desired properties hold in the modified agent configuration.

Proposition 13. *Let $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$ be an agent configuration. For every LTL formula φ, we have that*

- $\mathcal{C}_{\text{Real}} \models_{\text{SM}} \mathbf{B}(\varphi) \rightarrow \mathbf{G}(\varphi)$.

If φ is an LTL formula such that $\mathit{head}_g(\mathcal{C}) \setminus \{\varphi\} \not\models_{\text{LTL}} \varphi$, then

- $\mathcal{C}_{\text{Un}} \models_{\text{SM}} \mathbf{G}(\varphi) \rightarrow \neg\mathbf{B}(\varphi)$;
- $\mathcal{C}_{\text{Poss}} \models_{\text{SM}} \mathbf{G}(\varphi) \rightarrow \neg\mathbf{B}(\sim\varphi)$;

where $\mathit{head}_g(\mathcal{C})$ is the set of all LTL formulas occurring in rule heads in $\mathcal{G}$.

The reason why the latter two conditions only hold for formulas φ such that $\mathit{head}_g(\mathcal{C}) \setminus \{\varphi\} \not\models_{\text{LTL}} \varphi$, is that only for such formulas can we guarantee that the rules with head $\mathbf{G}(\varphi)$ are the only responsible for $\mathbf{G}(\varphi)$ being a consequence of $\mathcal{C}$. Otherwise $\mathbf{G}(\varphi)$ could follow from another rule: consider for example $\mathcal{C} = \langle \{\mathbf{B}(p \sqcup q) \leftarrow\}, \{\mathbf{G}(p) \leftarrow\} \rangle$. Then, both $\mathbf{B}(p \sqcup q)$ and $\mathbf{G}(p \sqcup q)$ follow from $\mathcal{C}_{\text{Un}}$.

5 Decidability and implementation

In this section we discuss the decidability and implementation of the following simple reasoning tasks:

- Given an agent configuration $\mathcal{C}$, does the belief $\mathbf{B}(\varphi)$ follow from $\mathcal{C}$?
- Given an agent configuration $\mathcal{C}$, does the goal $\mathbf{G}(\varphi)$ follow from $\mathcal{C}$?

To answer these queries, we need to compute the stable models of $\mathcal{C}$ and then check if they all entail $\mathbf{B}(\varphi)$ and $\mathbf{G}(\varphi)$, respectively. First of all, we prove that for a finite agent configuration, each of the above problems is decidable.

Note that, even if we restrict to a finite set of propositional symbols (for example those that appear in a finite agent configuration), the number of LTL logical theories over this language is infinite. An immediate consequence is that the number of possible stable models of a finite agent configuration is potentially infinite. Interestingly, as we show below, this is not the case and therefore decidability is not compromised. The key idea is the fact that, given a finite agent configuration $\mathcal{C}$, we are able to prove that only those LTL logical theories generated by sets of LTL formulas appearing in $\mathcal{C}$ can be part of a stable model of $\mathcal{C}$, and there is only a finite number of them. To make this precise consider the following sets. Let $\mathit{form}_b(\mathcal{C})$ be the set of LTL formulas that occur in the agent configuration $\mathcal{C}$ in the scope of the belief operator, and $\mathit{head}_b(\mathcal{C}) \subseteq \mathit{form}_b(\mathcal{C})$ the subset of those that occur in the head of a rule. In the same way we can define $\mathit{form}_g(\mathcal{C})$ to be the set of LTL formulas that occur in $\mathcal{C}$ in the scope of the goal operator, and $\mathit{head}_g(\mathcal{C}) \subseteq \mathit{form}_g(\mathcal{C})$ the subset of those that occur in the head of a rule. Of course, if $\mathcal{C}$ is finite then both $\mathit{form}_b(\mathcal{C})$ and $\mathit{form}_b(\mathcal{C})$ are finite.

Theorem 14. *Let $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$ be a finite agent configuration. If $\langle T_b, T_g \rangle$ is a stable model of $\mathcal{C}$, then there exists $A_b \subseteq \mathit{form}_b(\mathcal{C})$ and $A_g \subseteq \mathit{form}_g(\mathcal{C})$ such that $T_b = A_b^{\models_{\text{LTL}}}$ and $T_g = A_g^{\models_{\text{LTL}}}$.*

An immediate consequence of the above theorem is that every finite agent configuration has a finite number of stable models.

Corollary 15. *Every finite agent configuration has finitely many stable models.*

Our aim now is to define an algorithm that modularly combines an LTL reasoner and an ASP solver to compute the answers to the above queries. Recall that the validity problem in LTL is decidable [17]. The advantage of such modular algorithm is that we can leverage existing LTL and ASP reasoners.

Consider a given finite agent configuration $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$. We construct the normal logic program $\mathcal{P}^{\mathcal{C}}$ obtained from $\mathcal{C}$ in which belief and goal atoms containing LTL formulas are encoded as normal logic program atoms:

$$\begin{aligned}\mathcal{P}^{\mathcal{C}} = \mathcal{B} \cup \mathcal{G} \cup \{\, \mathbf{B}(\varphi) \leftarrow\ & \mathbf{B}(\psi_1), \ldots, \mathbf{B}(\psi_n) : \{\psi_1, \ldots, \psi_n\} \subseteq \mathit{head}_b(\mathcal{C}), \\ & \varphi \in \mathit{form}_b(\mathcal{C}) \setminus \{\psi_1, \ldots, \psi_n\}, \{\psi_1, \ldots, \psi_n\} \models_{\text{LTL}} \varphi \,\} \cup \\ \cup \{\, \mathbf{G}(\varphi) \leftarrow\ & \mathbf{G}(\psi_1), \ldots, \mathbf{G}(\psi_n) : \{\psi_1, \ldots, \psi_n\} \subseteq \mathit{head}_g(\mathcal{C}), \\ & \varphi \in \mathit{form}_g(\mathcal{C}) \setminus \{\psi_1, \ldots, \psi_n\}, \{\psi_1, \ldots, \psi_n\} \models_{\text{LTL}} \varphi \,\}\end{aligned}$$

To distinguish the set of stable models of an agent configuration $\mathcal{C}$, which is a set of pairs of LTL theories, from the set of stable models of $\mathcal{P}^{\mathcal{C}}$, which is a subset of the set of belief and goal atoms occurring in $\mathcal{P}^{\mathcal{C}}$, we denote the later by $AS(\mathcal{P}^{\mathcal{C}})$, and by $form(\mathcal{P}^{\mathcal{C}})$ the set of belief and goal atoms appearing in $\mathcal{P}^{\mathcal{C}}$.

The key idea underlying the construction of $\mathcal{P}^{\mathcal{C}}$ is to enrich the original agent configuration with rules that represent the possible interaction occurring between the formulas of the program in order to enforce the interdependency between temporal formulas appearing in the belief and goal atoms. Note that for a given agent configuration $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$, the sets $SM(\mathcal{C})$ and $AS(\mathcal{B} \cup \mathcal{G})$ may not be related, since AS does not take into account the logical interdependency between the formulas appearing in $\mathcal{C}$. As a simple example consider the agent configuration $\mathcal{C} = \langle \emptyset, \{\mathbf{G}(\Box p) \leftarrow;\ \mathbf{G}(q) \leftarrow \mathbf{G}(\Diamond p)\} \rangle$. Then, $SM(\mathcal{C}) = \{\langle \emptyset^{\vDash_{\text{LTL}}}, \{\Box p, \Diamond p, q\}^{\vDash_{\text{LTL}}} \rangle\}$ but $AS(\mathcal{B} \cup \mathcal{G}) = \{\{\mathbf{G}(\Box p)\}\}$. This is the reason why we cannot use an ASP solver directly on the program $\mathcal{B} \cup \mathcal{G}$.

In the case of $\mathcal{P}^{\mathcal{C}}$, we have the following strong relation.

Theorem 16. *Given a finite agent configuration $\mathcal{C} = \langle \mathcal{B}, \mathcal{G} \rangle$, we have that*

1. $\{T_b : \langle T_b, T_g \rangle \in SM(\mathcal{C})\} = \{\{\varphi : \mathbf{B}(\varphi) \in A\}^{\vDash_{\text{LTL}}} : A \in AS(\mathcal{P}^{\mathcal{C}})\}$
2. $\{T_g : \langle T_b, T_g \rangle \in SM(\mathcal{C})\} = \{\{\varphi : \mathbf{G}(\varphi) \in A\}^{\vDash_{\text{LTL}}} : A \in AS(\mathcal{P}^{\mathcal{C}})\}$

The above theorem presents a finite representation of the stable models of an agent configuration: the stable models of the program $\mathcal{P}^{\mathcal{C}}$. An immediate consequence is that the problems of checking if a belief or a goal atom follows from a finite agent configuration are both decidable.

Corollary 17. *Let $\mathcal{C}$ be a finite agent configuration and φ an LTL formula (not necessarily appearing in $\mathcal{C}$). Then, the problems of checking if $\mathcal{C} \vDash_{\text{SM}} \mathbf{B}(\varphi)$ and if $\mathcal{C} \vDash_{\text{SM}} \mathbf{G}(\varphi)$ are both decidable.*

The decidability of entailment for belief and goal atoms can be extended for complex formulas, since these depend only on the atoms appearing in it.

Corollary 18. *Let $\mathcal{C}$ be a finite agent configuration and δ a complex belief or goal formula (not necessarily in $\mathcal{C}$). Then, the problem $\mathcal{C} \vDash_{\text{SM}} \delta$ is decidable.*

6 Related work

There are several approaches in the literature that use temporal logic to model goals [5,12,16,7,10]. The work in [10] uses sets of LTL formulas to define both the belief and the goal bases. These can be easily captured by our non-monotonic framework, as we now show. Formally, in [10] a belief base is a set $\Sigma \subseteq \mathcal{L}_{\text{LTL}}$, a goal base is a set $\Gamma \subseteq \mathcal{L}_{\text{LTL}}$, and a *mental state* is a pair $m = \langle \Sigma, \Gamma \rangle$ such that both Σ and Γ are consistent and $\Gamma \vDash_{\text{LTL}} \Sigma$. The reason for the last condition is to impose the realism principle mentioned in Section 4, i.e., that an agent should have as goals all her beliefs. In fact, the formula $\mathbf{G}(\varphi) \rightarrow \mathbf{B}(\varphi)$ is valid in their

logic. Their language of beliefs and goals is the same as our language L and the satisfaction of formulas of $\mathcal{L}_{BG}$ by a mental state m is defined in a similar way as we do for agent configurations. For a given mental state $m = \langle \Sigma, \Gamma \rangle$ we can consider the corresponding (positive) agent configuration $\mathcal{C}_m = \langle \mathcal{B}_m, \mathcal{G}_m \rangle$ where $\mathcal{B}_m = \{\mathbf{B}(\varphi) \leftarrow : \varphi \in \Sigma\}$ and $\mathcal{G}_m = \{\mathbf{G}(\varphi) \leftarrow : \varphi \in \Gamma\}$. It is immediate to check that the unique stable model of $\mathcal{C}_m$ is precisely $\langle \Sigma^{\models_{\text{LTL}}}, \Gamma^{\models_{\text{LTL}}} \rangle$, and therefore, for every formula δ of $\mathcal{L}_{BG}$, we have that $m \Vdash \delta$ iff $\mathcal{C}_m \models_{\text{SM}} \delta$.

The work in [15] defines a framework for conditional goals using a translation to default logic [14]. Although it does not consider temporal goals, it offers an interesting non-monotonic framework for modeling goals. In what follows we briefly sketch the relation between the work in [15] and ours. Let us start with a very brief presentation of their framework. The language for beliefs and goals is a restriction of our language $\mathcal{L}_{BG}$, in the sense that in the scope of a belief or goal operator only propositional formulas without temporal operators are allowed. Conditional goals are defined through *goal inference rules*, which are of the form: $\beta, \kappa^+, \kappa^- \Rightarrow \phi$, where β, κ^+ and κ^- are sets of propositional formulas. In such a goal inference rule, ϕ represents the goal that can be inferred if the beliefs in β are true, the goals in κ^+ are true and the goals in κ^- are not known to be true. A goal base is a set GI of goal inference rules. A belief base σ is just a set of propositional formulas. For beliefs, the semantics is easily defined by $\langle \sigma, GI \rangle \models_d \mathbf{B}(\varphi)$ if $\sigma \models \varphi$. The semantics of a goal base GI is defined by translating it to a default theory $t(GI)$. This is done by translating each goal inference rule $r = \{\beta_1, \ldots, \beta_k\}, \{\alpha_1, \ldots, \alpha_n\}^+, \{\varphi_1, \ldots, \varphi_m\}^- \Rightarrow \varphi$ in GI such that $\sigma \models \{\beta_1, \ldots, \beta_k\}$ to the default rule $t(r) = \alpha_1 \sqcap \cdots \sqcap \alpha_n : {\sim}\varphi_1, \ldots, {\sim}\varphi_m, \varphi / \varphi$. The (credulous) entailment for goals is then defined as $\langle \sigma, GI \rangle \models_d \mathbf{G}(\varphi)$ if there exists an extension (in the sense of default logic) E of $t(GI)$ such that $E \models \varphi$.

Given a goal base GI and a belief base σ consider the agent configuration $\langle \mathcal{B}_\sigma, \mathcal{G}_{GI} \rangle$ such that $\mathcal{B}_\sigma = \{\mathbf{B}(\varphi) \leftarrow : \varphi \in \sigma\}$ and $\mathcal{G}_{GI}$ is obtained by considering, for each rule $r = \{\beta_1, \ldots, \beta_k\}, \{\alpha_1, \ldots, \alpha_n\}^+, \{\varphi_1, \ldots, \varphi_m\}^- \Rightarrow \varphi$ in GI, the rule $\mathbf{G}(\varphi) \leftarrow \mathbf{B}(\beta_1 \wedge \cdots \wedge \beta_k), \mathbf{G}(\alpha_1 \wedge \cdots \wedge \alpha_n), \mathit{not}\ \mathbf{G}(\varphi_1), \ldots, \mathit{not}\ \mathbf{G}(\varphi_m), \mathit{not}\ \mathbf{G}(\neg\varphi)$. Note that we are just using a fragment of our language to embed both σ and GI. In the case of beliefs, we just need to use facts, and in the case of goals, we do not use temporal formulas nor default negated beliefs. We can then prove that $\langle \sigma, GI \rangle \models_d \mathbf{B}(\varphi)$ iff $\langle \mathcal{B}_\sigma, \mathcal{G}_{GI} \rangle \models_{\text{CSM}} \mathbf{B}(\varphi)$. Also $\langle \sigma, GI \rangle \models_d \mathbf{G}(\varphi)$ iff $\langle \mathcal{B}_\sigma, \mathcal{G}_{GI} \rangle \models_{\text{CSM}} \mathbf{G}(\varphi)$. Moreover, a similar result holds between the skeptical entailment $\models_{dd}$ defined in [15] and our skeptical entailment $\models_{\text{SM}}$. In the case of skeptical entailment, the relation is even stronger since it holds for every formula of $\mathcal{L}_{BG}$. The reason why the above proposition does not follow for any complex formula of $\mathcal{L}_{BG}$ is the fact that the entailment $\models_d$ of [15] is first defined for belief and goal atoms, and only then extended to complex formulas. In this way, in their framework we may have that $\langle \sigma, GI \rangle \models_d \mathbf{G}(p) \wedge \mathbf{G}({\sim}p)$, even for consistent σ and GI, since this means that $\mathbf{G}(p)$ is true in one stable model and $\mathbf{G}({\sim}p)$ is true in a different stable model. This is not the case in our framework. Although this distinguishes our approach from that in [15], a more fundamental difference is that they adopt a particular entailment over the possible extensions (in the

sense of default logic), thus not making use the full richness of the set of stable models. On the contrary, we argue that the set of stable models is fundamental since it can be seen as the set of possible sets of goals an agent can commit to.

Let us now draw some comments on the work of [2]. There, a simple non-monotonic version of temporal logic for specifying goals is defined. This is done by extending the language of temporal logic with two operators to model weak and strong exceptions. Although for lack of space we do not present here the details, it can be shown that the non-monotonic extension N-LTL of LTL presented in [2] can be embedded in our framework. This is not surprising since in our framework the use of default negation allows to model exceptions in a very flexible way.

7 Conclusions and future work

In this paper, we have defined a non-monotonic framework for representing temporal beliefs and goals, along with a stable models like semantics for this expressive language. We have argued that an ASP view of the stable models of an agent configuration can bring a novel perspective on dealing with multiple possible sets of goals. We have proven that some usual properties of beliefs and goals hold in our framework. Moreover, we have shown that the problem of checking the entailment of a formula in a given finite agent configuration is decidable, and we presented an implementation that makes a modular use of an LTL reasoner and an ASP solver. In the end, we have briefly hinted on how existing work on the representation of beliefs and goals can be embedded in our framework.

This work raises several interesting directions for future research. Since the temporal operators can only appear in the scope of belief or goal operators, our approach does not deal with the evolution of the belief and goal bases. An interesting idea to cope with such evolution is to use some extension of dynamic logic programs [1]. Also interesting is to study the connection between our framework and the general approach of parametrized logic programming [9].

Additionally, our work could be integrated with existing agent programming languages/architectures, e.g., 2APL [6] and Jason [3], thus increasing their capabilities to represent and reason about goals.

Finally, we also want to extend our work so that we can consider the currently adopted intentions and how they can influence the stable models that encode the new ones to be adopted. This is an interesting problem but rather complex since we typically would like to keep as many of the current intentions as possible, i.e., the problem might be solvable by treating current intentions as beliefs, but it might require a measure of distance and seek for models that encode minimal change between the current intentions and the goals in possible stable models.

Acknowledgments We would like to thank the anonymous reviewers whose comments helped to improve the paper.

Matthias Knorr, João Leite and Martin Slota were partially supported by FCT under project "ERRO – Efficient Reasoning with Rules and Ontologies" (PTDC/EIA-CCO/121823/2010). Ricardo Gonçalves was supported by FCT

grant SFRH/BPD/47245/2008 and Matthias Knorr was also partially supported by FCT grant SFRH/BPD/86970/2012.

References

1. Alferes, J.J., Banti, F., Brogi, A., Leite, J.A.: The refined extension principle for semantics of dynamic logic programming. Studia Logica 79(1), 7–32 (2005)
2. Baral, C., Zhao, J.: Non-monotonic temporal logics for goal specification. In: Veloso, M.M. (ed.) IJCAI. pp. 236–242 (2007)
3. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming multi-agent systems in AgentSpeak using Jason. Wiley-Interscience (2007)
4. Bratman, M.: Intention, plans, and practical reason. Harvard University Press, Cambridge, MA (1987)
5. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. Artif. Intell. 42(2-3), 213–261 (1990)
6. Dastani, M.: 2APL: a practical agent programming language. Autonomous Agents and Multi-Agent Systems 16(3), 214–248 (2008)
7. Dastani, M., van Riemsdijk, M.B., Winikoff, M.: Rich goal types in agent programming. In: Sonenberg, L., Stone, P., Tumer, K., Yolum, P. (eds.) AAMAS. pp. 405–412. IFAAMAS (2011)
8. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Comput. 9(3-4), 365–385 (1991)
9. Gonçalves, R., Alferes, J.J.: Parametrized logic programming. In: Janhunen, T., Niemelä, I. (eds.) JELIA. Lecture Notes in Computer Science, vol. 6341, pp. 182–194. Springer (2010)
10. Hindriks, K.V., van der Hoek, W., van Riemsdijk, M.B.: Agent programming with temporally extended goals. In: Sierra, C., Castelfranchi, C., Decker, K.S., Sichman, J.S. (eds.) AAMAS (1). pp. 137–144. IFAAMAS (2009)
11. Pnueli, A.: The temporal logic of programs. In: 18th Annual Symposium on Foundations of Computer Science (Providence, R.I., 1977), pp. 46–57. IEEE Comput. Sci., Long Beach, Calif. (1977)
12. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In: Allen, J.F., Fikes, R., Sandewall, E. (eds.) KR. pp. 473–484. Morgan Kaufmann (1991)
13. Rao, A.S., Georgeff, M.P.: BDI agents: From theory to practice. In: Lesser, V.R., Gasser, L. (eds.) ICMAS. pp. 312–319. The MIT Press (1995)
14. Reiter, R.: A logic for default reasoning. Artif. Intell. 13(1-2), 81–132 (1980)
15. van Riemsdijk, M.B., Dastani, M., Meyer, J.J.C.: Goals in conflict: semantic foundations of goals in agent programming. Autonomous Agents and Multi-Agent Systems 18(3), 471–500 (2009)
16. van Riemsdijk, M.B., Dastani, M., Winikoff, M.: Goals in agent systems: a unifying framework. In: Padgham, L., Parkes, D.C., Müller, J.P., Parsons, S. (eds.) AAMAS (2). pp. 713–720. IFAAMAS (2008)
17. Sistla, A.P., Clarke, E.M.: The complexity of propositional linear temporal logics. J. ACM 32(3), 733–749 (Jul 1985)
18. Winikoff, M., Padgham, L., Harland, J., Thangarajah, J.: Declarative & procedural goals in intelligent agent systems. In: Fensel, D., Giunchiglia, F., McGuinness, D.L., Williams, M.A. (eds.) KR. pp. 470–481. Morgan Kaufmann (2002)
19. Wójcicki, R.: Theory of Logical Calculi. Synthese Library, Kluwer Academic Publishers (1988)