A Non-Monotonic Goal Specification Language for Planning with Preferences

Tran Cao Son* and **Enrico Pontelli*** and **Chitta Baral+** [1]

Abstract. This paper introduces a default logic based approach to defining goal specification languages that can be non-monotonic and allow for the specification of inconsistencies and priorities among goals. The paper starts by presenting a basic goal specification language for planning with preferences. It then defines goal default theories (resp. with priorities) by embedding goal formulae into default logic (resp. prioritizing default logic). It is possible to show that the new language is general, as it can express several features of previously developed goal specification languages. The paper discusses how several other features can be subsumed by extending the basic goal specification language. Finally , we identify features that might be important in goal specification that cannot be expressed by our language.

1 Introduction

An important component of autonomous agent design is *goal specification*. In classical planning, goals deal with reaching one of a particular set of states. Nevertheless, often goals of agents are not just about reaching a particular state; goals are often about satisfying desirable *conditions* imposed on the trajectory. For example, a person can have the following desire in preparing travel plans to conferences:

(*) *I prefer to fly to the conference site (since it is usually too far to drive).*

The user's preference restricts the means that can be used in achieving her goal of reaching the conference site, which leads to the selection of a plan that reaches the conference site by airplane, whenever possible. Ultimately, this affects what actions the person should take in order to achieve the goal.

These observations led to the development of languages for the specification of *soft goals* in planning, e.g., $\mathcal{PP}$ introduced in [14] and modified in [6]. In $\mathcal{PP}$, a *basic desire* is a temporal formula describing desirable properties of a plan. *Atomic* and *general preferences* are particular classes of formulae built over basic desires. A preference formula Φ defines a preference order $\prec_\Phi$ among the trajectories that achieve the *hard goal* of the problem, i.e., for every pair of trajectories α and β, $\alpha \prec_\Phi \beta$ indicates that α is preferable to β. $\prec_\Phi$ is often a partial order and its definition relies on the notion of satisfaction between trajectories and a preference specification. Similar ideas have been considered in the planning community and led to extensions of the planning domain description language PDDL, with features for representing classes of preferences over plans using temporal extended preferences (e.g., [10]).

In [4], the authors argue that a goal specification language should be *non-monotonic* for various reasons, such as elaboration tolerance and simplicity of goal specification. For example, the same traveler with the preference **(*)** would probably not mind driving at most three hours to the conference site if the only flight to the destination requires to travel the day before the conference starts. In this case, her preference becomes:

(**) *Normally, I prefer to fly to the conference site (since it is usually too far to drive). However, if there are no flights on the same day of the conference and the driving time is at most three hours, then I will drive.*

To address this issue, an extension of LTL [11], called N-LTL, has been proposed, allowing weak and strong exceptions to certain rules. A weakness of this language is that it requires the classification of weak and strong exceptions when a goal is specified. In [5], the language ER-LTL is introduced to address this limitation of N-LTL. Similarly to $\mathcal{PP}$, the semantics of N-LTL and ER-LTL relies on the notion of satisfaction between plans and N-LTL or ER-LTL specifications. Observe that the issue of non-monotonicity is dealt within $\mathcal{PP}$ and in the extensions of PDDL by revising the soft goals, which is an approach that N-LTL specifically tries to avoid.

We observe that the focus of the work in [1, 4, 5, 6, 10] is on classical planning, i.e., the planning domains are deterministic and the initial state is complete, while the work in [14] considers non-deterministic domains and only discusses preferences among weak plans. In [2], it is argued that plans for non-deterministic domains should be *policies* (i.e., a partial function from the set of states to the set of actions) and the language π-CTL* is developed for specifying goals in non-deterministic domains. π-CTL* is an extension of CTL* [9] with two modalities A_π and E_π for considering all or some trajectories w.r.t. a given policy. In [3], the language π-CTL* is extended with quantifiers over policies to increase its expressiveness. Policies satisfying a goal specification are viewed as the solutions of a planning problem.

In this paper, we explore an approach based on prioritizing default logic for defining a goal specification language. The new language, called *goal default theories with priorities*, is a variation of prioritizing default logic, in which formulae occurring within a default can be temporal extended preference formulae. We show that the core of the new language subsumes several features from existing goal languages and can be extended to subsume several other features from other goal languages. Finally, we discuss the possible applications of the new language in the study of existing goal languages and the development of new ones.

2 Background

In this section, we briefly review the basic definitions of planning, linear temporal logic (LTL) and its extension for specifying prefer-

[1] *Department of Computer Science New Mexico State University, Las Cruces, New Mexico, USA, email: `tson|epontell@cs.nmsu.edu` and +Department of Computer Science and Engineering, Arizona State University, Tempe, Arizona, USA, email: `chitta@asu.edu`

ences in planning.

2.1 LTL and Temporal Extended Preferences

Let $\mathcal{L}$ be a propositional language. By $\langle p \rangle$ we denote a propositional formula from $\mathcal{L}$. LTL-formulae are defined by the following syntax

$$\begin{array}{lll}\langle f \rangle & ::= & \langle p \rangle \mid \langle f \rangle \wedge \langle f \rangle \mid \langle f \rangle \vee \langle f \rangle \mid \\ & & \neg \langle f \rangle \mid \bigcirc \langle f \rangle \mid \Box \langle f \rangle \mid \Diamond \langle f \rangle \mid \langle f \rangle \mathsf{U} \langle f \rangle \end{array} \quad (1)$$

The semantics of LTL-formulae is defined with respect to sequences of interpretations of $\mathcal{L}$. For later use, we will refer to an interpretation of $\mathcal{L}$ as a *state* and a possibly infinite sequence of interpretations of $\mathcal{L}$, $s_0, s_1, \ldots$, as a *trajectory*. For a trajectory $\sigma = s_0, s_1, \ldots$, by σ_i we denote the suffix $s_i, s_{i+1}, \ldots$ of σ. A trajectory $\sigma = s_0, s_1, \ldots$ *satisfies* an LTL-formula f, denoted by $\sigma \models f$, if $\sigma_0 \models f$ where

- $\sigma_j \models p$ iff $s_j \models p$
- $\sigma_j \models \neg f$ iff $\sigma_j \not\models f$
- $\sigma_j \models f_1 \wedge f_2$ iff $\sigma_j \models f_1$ and $\sigma_j \models f_2$
- $\sigma_j \models f_1 \vee f_2$ iff $\sigma_j \models f_1$ or $\sigma_j \models f_2$
- $\sigma_j \models \bigcirc f$ iff $\sigma_{j+1} \models f$
- $\sigma_j \models \Box f$ iff $\sigma_k \models f$, for all $k \geq j$
- $\sigma_j \models \Diamond f$ iff $\sigma_i \models f$ for some $i \geq j$
- $\sigma_j \models f_1 \mathsf{U} f_2$ iff there exists $k \geq j$ such that $\sigma_k \models f_2$ and for all i, $j \leq i < k$, $\sigma_i \models f_1$.

A finite trajectory $s_0, \ldots, s_n$ satisfies an LTL-formula f if its extension $s_0, \ldots, s_n, s_{n+1}, \ldots$ satisfies f, where $s_k = s_n$ for $k > n$. In order to deal with planning problems, LTL is extended with the following constructs

$$\begin{array}{c}\texttt{at\_end} \langle p \rangle \mid \langle p \rangle \texttt{ sometime\_before } \langle p \rangle \mid \\ \langle p \rangle \texttt{ sometime\_after } \langle p \rangle \end{array} \quad (2)$$

Formulae of the extended LTL are referred to as *Temporal Extended Preferences (TEP)*. Note that the last two are syntactic sugars for LTL formulae. Temporal extended preferences are interpreted over finite trajectories. The notion of satisfaction for standard LTL-formulae is defined as above, while satisfaction of TEP formulae is as follows: given a finite trajectory $\sigma = s_0, \ldots, s_n$:

- $\sigma \models \texttt{at\_end}\, p$ iff $s_n \models p$;
- $\sigma \models p_1$ `sometime_before` p_2 iff for every i, $0 \leq i \leq n$, if $\sigma_i \models p_1$ then $\sigma_j \models p_2$ for some $i \leq j \leq n$; and
- $\sigma_j \models p_1$ `sometime_after` p_2 iff for every i, $0 \leq i \leq n$, if $\sigma_i \models p_1$ then $\sigma_j \models p_2$ for some $0 \leq j < i \leq n$.

2.2 Planning

In this paper, we describe a dynamic domain as a labeled transition system $T = (F, A, S, L)$, where:

- F is a set of fluents (or propositions),
- A is a set of actions,
- S is a set of interpretations (or states) of F, and
- $L \subseteq S \times A \times S$.

Each triple $\langle s_1, a, s_2 \rangle \in L$ indicates that the execution of the action a in the state s_1 might result in the state s_2. T is *deterministic* if for each state s and action a, L contains at most one triple $\langle s, a, s_2 \rangle$; otherwise, T is non-deterministic.

Given a transition system T, a finite or infinite sequence $s_0 a_0 s_1 a_1 \ldots s_n a_n s_{n+1} \ldots$ of alternate states and actions is called a *run* if $\langle s_i, a_i, s_{i+1} \rangle \in L$ for every $i = 0, \ldots$ A *policy* π in a transition system T is a partial function $\pi : S \to A$ from the set of states to the set of actions. A run $s_0 a_0 s_1 a_1 \ldots s_k a_k s_{k+1} \ldots$ is said to be induced by a policy π if $a_i = \pi(s_i)$ for every $i = 0, \ldots, k, \ldots$

Definition 1. *A* planning problem *is a triple* $\langle T, S_i, S_f \rangle$ *where* $T = (F, A, S, L)$ *is a transition system,* $S_i \subseteq S$ *is the set of* initial *states, and* $S_f \subseteq S$ *is the set of* final *states.*

Intuitively, a planning problem asks for a *plan* which transforms the transition system from any state belonging to S_i to some state in S_f. In the rest of the discussion, we assume S_i and S_f to be finite sets. We distinguish two classes of planning problems:

- *Deterministic planning*: in this case, T is deterministic and a *solution* (or *plan*) of $\langle T, S_i, S_f \rangle$ is an action sequence $[a_0; \ldots; a_n]$ such that, for every $s_0 \in S_i$, $s_0 a_0 s_1 a_1 \ldots a_n s_{n+1}$ is a run in T and $s_{n+1} \in S_f$;
- *Non-deterministic planning*: in this case, T is non-deterministic and a solution (or plan) of $\langle T, S_i, S_f \rangle$ is a policy π such that, for every $s_0 \in S_i$ and every run induced by π in T, π is finite and is of the form $s_0 a_0 s_1 a_1 \ldots s_k a_k s_{k+1}$ where $s_{k+1} \in S_f$.

In the following, whenever we refer to a possible plan in a transition system T, we mean a sequence of actions (resp. a policy) if T is deterministic (resp. non-deterministic) that can generate a correct run. Let us illustrate these basic definitions using the following simple example.

Example 1. *Consider a transportation robot. There are different locations, say $l_1, \ldots, l_k$, whose connectivity is given by a graph and there might be different objects at each location. Let O be a set of objects. The robot can travel between two directly connected locations. It can pick up objects at a location, hold them, drop them, and carry them between locations. We assume that, for each pair of connected locations l_i and l_j, the robot has an action $a_{i,j}$ for traveling from l_i to l_j. The robot can hold only one object at a time. The domain can be represented by a transition system $T_1 = (F, A, S, L)$:*[2]

- *F contains the following types of propositions:*
  - *$at(i)$ denotes that the robot is at the location l_i;*
  - *$o\_at(o, i)$ denotes that the object o is at the location l_i;*
  - *$h(o)$ denotes that the robot is holding the object o.*
- *A contains of the following types of actions:*
  - *$a_{i,j}$ the robot moves from l_i to l_j;*
  - *$release(o)$ the robot drops the object o;*
  - *$pickup(o)$ the robot picks up the object o.*
- *S contains the interpretations of F which satisfy the basic constraints, such as the robot is at one location at a time, it holds only one object, etc.*
- *L contains transitions of the form $\langle s, a, s' \rangle$ such that s' is the result of the execution of a in s; for example, if $a = a_{i,j}$ and $at(i) \in s$ then $s' = s \setminus \{at(i)\} \cup \{at(j)\}$.*

T_1 is a deterministic transition system. We will also refer to T_2 as the non-deterministic version of T_1 by defining $T_2 = (F, A, S, L')$ where $L' = L \cup \{\langle s_i, a_{i,j}, s_i \rangle \mid a_{i,j} \in A\}$ and $at(i) \in s$. Intuitively, T_2 encodes the fact that the action $a_{i,j}$ might fail and, when it does, the robot will stay where it was after the execution of $a_{i,j}$.

A planning problem P in this domain is given by specifying the initial location of the robot and of the objects and the final location of the robot and of the objects. It is deterministic (resp. non-deterministic) if T_1 (resp. T_2) is considered.

For example, $P_i = \langle T_i, \{\{at(1)\}\}, S_f \rangle$ where for each $s \in S_f$, $at(k) \in s$ is a planning problem for T_i. A solution for P_1 is a sequence $[a_{1,2}; \ldots; a_{k-1,k}]$. On the other hand, a solution for P_2 is a policy π defined by $\pi(s) = a_{t,t+1}$ iff $at(t) \in s$ for $t < k$.

[2] We simplify the definitions of S and L for readability.

3 A Basic Goal Specification Language for Planning with Preferences

In the literature, a planning problem with preferences is defined as a pair (P, Φ) of a planning problem $P = \langle T, S_i, S_f \rangle$, where $T = (F, A, S, L)$, and a preference formula Φ in a goal specification language. A plan δ of P is called a *preferred plan* if it is a plan for P and satisfies Φ, where the notion of satisfaction of a preference formula by a plan is language dependent.

In general, we can characterize a goal specification language $\mathcal{G}$ over a transition system T by a set of preference formulae $\mathcal{F}$ and a satisfaction relation $\models_{\mathcal{G}}$ between the set of possible plans of T and formulae in $\mathcal{F}$. We will write $\delta \models_{\mathcal{G}} \Phi$ to denote that the plan δ satisfies the formula Φ under the language $\mathcal{G}$.

For later use, we will define a *basic goal specification language* for a transition system $T = (F, A, S, L)$, written as $\mathcal{G}_b = (\mathcal{F}_b, \models_{\mathcal{G}_b})$, as follows:

- the set of preference formulae $\mathcal{F}_b$ is the set of TEP-formulae over $F \cup A$, and
- for a planning problem $P = \langle T, S_i, S_f \rangle$, $\models_{\mathcal{G}_b}$ is defined as follows:
  - if T is deterministic, a plan $\delta = [a_0, \ldots, a_n]$ for a planning problem P is said to satisfy a formula Φ in $\mathcal{F}_b$ if for every $s_0 \in S_i$, $s_0 a_0 s_1 a_1 \ldots a_n s_{n+1}$ is a run in T and $(s_0 \cup \{a_0\}), \ldots, (s_n \cup \{a_n\}), s_{n+1}$ is a trajectory satisfying Φ (in the TEP-language over $F \cup A$);
  - if T is non-deterministic, a solution (policy) π for P is said to satisfy a formula Φ in $\mathcal{F}_b$ if for every $s_0 \in S_i$ and every run $s_0 a_0 s_1 a_1 \ldots s_k a_k s_{k+1}$ in T induced by π, $(s_0 \cup \{a_0\}), \ldots, (s_n \cup \{a_n\}), s_{n+1}$ is a trajectory satisfying Φ (in the TEP-language over $F \cup A$).

In the following, we will assume that any goal specification language $\mathcal{G}$ is a conservative extension of $\mathcal{G}_b$, i.e., (*i*) $\mathcal{G}$ contains all formulae in $\mathcal{G}_b$; and (*ii*) for every planning problem P and a formula Φ in $\mathcal{G}$, if $\Phi \in \mathcal{G}_b$ and $\delta \models_{\mathcal{G}_b} \Phi$ with respect to $\mathcal{G}_b$ then $\delta \models_{\mathcal{G}} \Phi$ with respect to $\mathcal{G}$.

Example 2. *Some preference formulae in $\mathcal{G}_b$ for the transition systems in Ex. 1 are:*

- $\Diamond at(2)$*: the robot should visit the location l_2 during the execution of the plan;*
- $at(1) \wedge \Diamond at(2)$*: the robot must* (i) *start in a state satisfying $at(1)$ (or the robot is at the location l_1 initially); and* (ii) *visit the location l_2 at some point during the execution of the plan;*
- $\Box[at(2) \Rightarrow (\bigvee_{i \neq 2} a_{2i})]$*: whenever the robot visits l_2, it should leave that location immediately by executing an action going to one of its neighbors;*
- $h(o) \Rightarrow \bigcirc\bigcirc \neg h(o)$*: if the robot holds an object o in the initial state then it should release o after the execution of one action;*
- $\Box[h(o) \Rightarrow \bigcirc\bigcirc \neg h(o)]$*: whenever the robot holds an object o it should release o after the execution of an action;*
- $h(o)$ `sometime_before` $at(5)$*: whenever the robot holds the object o, it must visit the location l_5 thereafter before reaching the goal;*
- `at_end` $[\bigwedge_{o \in O} \neg h(o)]$*: at the end, the robot should not hold any object.* □

With a slight abuse of notation, let us view a state s as a formula $\bigwedge_{s \models f} f \wedge \bigwedge_{s \models \neg f} \neg f$. Let S_i and S_f be two sets of states and

$$\Phi = \left[\underbrace{\bigvee_{s \in S_i} s}_{\Phi_1} \wedge \underbrace{\texttt{at\_end} \, [\bigvee_{s \in S_f} s]}_{\Phi_2} \right]$$

It is easy to see that any plan satisfying Φ requires its execution to start from a state satisfying Φ_1, which is one of the states in S_i, and end in a state satisfying Φ_2, which is one of the states in S_f. For this reason, the description of the initial and final states can be folded into a preference formula. We will therefore define planning problems as follows.

Definition 2. *Given a transition system T and a goal specification language $\mathcal{G} = (\mathcal{F}, \models_{\mathcal{G}})$ over T, a goal formula Φ in $\mathcal{F}$ is called a* planning problem. *A solution of* Φ *is a* plan δ *in* T *such that* $\delta \models_{\mathcal{G}} \Phi$.

By Def. 2, a goal formula represents a planning problem. The literature is quite diversified when a user faces two or more goal formulae which are contradictory with each other. For example, the formula $\Diamond at(2)$ is contradictory with $\Box \neg at(2)$; $\Box \neg (\bigwedge_{o \in O} h(o))$ conflicts with $\Diamond h(o_1)$; etc. A possibility is to consider a possible plan as solution if it satisfies some goal formulae. Another possibility is to rank the goal formulae and identify solutions as plans that satisfy the formula with the highest possible ranking. In the following, we will show that a uniform framework for dealing with conflicting goal formulae can be obtained by embedding goal formulae into Reiter's default logic.

4 Goal Default Theories

In this section, we will introduce a new goal specification language, called *goal default theory*. A goal default theory is a variation of Reiter's default theory [12], whose defaults can contain preference formulae. Goal default theories provide a possible treatment of planning with multiple goal formulae.

A goal default theory is defined over a transition system $T = (F, A, S, L)$ and a goal specification language $\mathcal{G} = (\mathcal{F}, \models_{\mathcal{G}})$.

Given a goal specification language $(\mathcal{F}, \models_{\mathcal{G}})$, we say that two formulae φ, ψ in $\mathcal{F}$ are equivalent w.r.t. $\models_{\mathcal{G}}$ if, for each plan δ of T, we have that $\delta \models_{\mathcal{G}} \varphi \Leftrightarrow \psi$.[3] We can easily extend this notion to define the notion of logical consequence w.r.t. $\models_{\mathcal{G}}$—if S is a set of formulae from $\mathcal{F}$ and f is another formula in $\mathcal{F}$, then $S \models_{\mathcal{G}} f$ if for each plan δ of T we have that $\delta \models_{\mathcal{G}} \bigwedge_{\varphi \in S} \varphi$ implies $\delta \models_{\mathcal{G}} f$. Given a set of formulae S, we define $Decl(S) = \{\varphi \mid \varphi \in \mathcal{F}, S \models_{\mathcal{G}} \varphi\}$.

A *preference default* (or *p-default*) d over $\mathcal{G}$ is of the following form

$$\frac{\alpha \, : \, \beta}{\gamma} \tag{3}$$

where α, β, and γ are formulae in $\mathcal{F}$. We call α the *precondition*, β the *justification*, and γ the *consequence* of d, and we denote them with $prec(d)$, $just(d)$, and $cons(d)$, respectively. A default d is said to be

- *Normal* if its justification is equivalent to its conclusion;
- *Prerequisite-free* if its precondition is equivalent to *true*; and
- *Supernormal* if it is normal and prerequisite-free.

Given a set of formulae S from $\mathcal{F}$, a default d is said to be defeated in S if $S \models \neg just(d)$. Some preferences and their representation as p-defaults over $\mathcal{G}_b$ for the domain from Example 1 are given next.

Example 3. *In these examples, o denotes a particular object in the domain.*

[3] $\varphi \Leftrightarrow \psi$ is a shorthand for $(\varphi \wedge \psi) \vee (\neg\varphi \wedge \neg\psi)$.

- *If there is no evidence that the robot is initially at the location l_2, then it should go to l_2:*

$$\frac{\top : \neg at(2)}{\Diamond at(2)} \tag{4}$$

- *Assume that objects might be defective, represented by the proposition $defective$. We can write*

$$\frac{\top \ : \ \Box[\neg \mathtt{defective}(o)]}{\Box[at(2) \Rightarrow h(o)]} \tag{5}$$

to indicate that normally, we would like that the robot holds the object o whenever it is at the location l_2. An exception to this rule is possible if the object o is defective.

- *If the robot is not required to hold the object o in the final state and there is no evidence that it initially holds o, then it should not execute the action of picking up the object o:*

$$\frac{\top \quad : \quad \mathtt{at\_end}\ (\neg h(o)) \wedge \neg h(o)}{\Box[\neg pickup(o)]} \tag{6}$$

- *If there is no evidence that the object o is initially in the wrong place then the robot should not start by executing the action of picking up the object o:*

$$\frac{\mathtt{at\_end}\ (o\_at(o,i)) \ : \ \bigwedge_{i \neq j} \neg o\_at(o,j)}{\neg pickup(o)} \tag{7}$$

- *A stronger version of (7) is*

$$\frac{\mathtt{at\_end}\ (o\_at(o,i)) \ : \ \bigwedge_{i \neq j} \neg o\_at(o,j)}{\Box \neg pickup(o)} \tag{8}$$

indicates that the robot should never pick up the object o if o could already be in the desired final location.

- *If there is the possibility that the robot might reach location l_2, then it must leave the location immediately after its arrival at l_2.*

$$\frac{\top \ : \ \Diamond[at(2)]}{\Box[at(2) \Rightarrow \bigcirc \bigvee_{i \neq 2} a_{2,i}]} \tag{9}$$

- *If there is no evidence that an object o will ever appear in location i then the robot should never go there.*

$$\frac{\top \ : \ \Box[\neg o\_at(o,i)]}{\Box[\bigvee_{j \neq i} \neg a_{j,i}]} \tag{10}$$

In the following, we will refer to the p-defaults in (4)-(9) by $p_1, \ldots, p_6$, respectively. □

We next define the notion of a goal default theory.

Definition 3. *A* goal default theory *over a goal language $\mathcal{G} = (\mathcal{F}, \models_\mathcal{G})$ and a transition system T is a pair $\Sigma = (D, W)$ where D is a set of p-defaults over $\mathcal{G}$ and $W \subseteq \mathcal{F}$.*

Given a set of p-defaults D, we denote with $cons(D)$ the set $cons(D) = \{cons(d) \mid d \in D\}$. A p-default d is *applicable* w.r.t. a set of $\mathcal{F}$ formulae S if $S \models_\mathcal{G} prec(d)$ and $S \not\models_\mathcal{G} \neg just(d)$. Let us denote with $\Pi_D(S)$ the set of p-defaults from D that are applicable w.r.t. S.

Definition 4 (From [12])**.** *Let $\Sigma = (D, W)$ be a goal default theory over $\mathcal{G} = (\mathcal{F}, \models_\mathcal{G})$ and T. An* extension *of Σ is a minimal set $E \subseteq \mathcal{F}$ that satisfies the condition $E = Decl(W \cup Cons(\Pi_D(E)))$. We say that Σ is* consistent *if it has at least one extension.*

From this definition, any default over the propositional language $F \cup A$ is a p-default, and any Reiter's default theory over the language $F \cup A$ is a goal default theory.

Definition 5. *Given a transition system $T = (F, A, S, L)$ and a goal specification language $\mathcal{G} = (\mathcal{F}, \models_\mathcal{G})$ over T, a* planning problem *over T and $\mathcal{G}$ is a goal default theory $\Sigma = (D, W)$ over $\mathcal{G}$ and T.*

The notion of a solution to a planning problem is modified as follows.

Definition 6. *Given a transition system $T = (F, A, S, L)$, a goal specification language $\mathcal{G} = (\mathcal{F}, \models_\mathcal{G})$ over T, and a planning problem Σ over T and $\mathcal{G}$, a* solution *of Σ is a plan δ in T such that $\delta \models_\mathcal{G} E$ for some extension E of Σ.*

Some planning problems over the transition systems in Exp. 1 and the language $\mathcal{G}_b$ are given in the next example.

Example 4 (Continuation of Example 3)**.**
- *Let $\Sigma_1 = (\{p_1\}, \{at(1), \mathtt{at\_end}\ at(5)\})$ where p_1 is the default (4). Intuitively, we have that Σ_1 identifies plans where the robot starts at location l_1, goes through the location l_2, and ends in location l_5.*
- *Let $\Sigma_2 = (\{p_6\}, \{at(1), \mathtt{at\_end}\ at(5)\})$ where p_6 is the default (9). This identifies plans where the robot starts at location l_1, ends in location l_5, and either (*i*) never goes through the location l_2; or (*ii*) never stays in the location l_2 within two consecutive steps.* □

The planning problems in Example 4 are simple, in that they are specified by goal default theories whose set of defaults is a singleton. Let us consider a more complicated example. Assume that we have two temporal formulae Φ and Ψ such that there exists no plan that can satisfy both Φ and Ψ. In this case, the use of goal default theory as a goal formula comes in handy. Indeed, every solution of the planning problem expressed by the goal default theory

$$\Sigma_{\Phi,\Psi} = \left(\left\{ \frac{\top : \neg\Psi}{\Phi}, \frac{\top : \neg\Phi}{\Psi} \right\}, \emptyset \right) \tag{11}$$

satisfies either Φ or Ψ. The following result generalizes this observation.

Proposition 1. *Let $T = (F, A, S, L)$ be a transition system, $\mathcal{G} = (\mathcal{F}, \models_\mathcal{G})$ be a goal specification language, and $\Delta = \{\Phi_1, \ldots, \Phi_n\}$ be a set of preference formulae in $\mathcal{F}$. Furthermore, let*

$$\Sigma_\Delta = \left(\left\{ \ \frac{\top : \Psi}{\Psi} \quad \middle| \quad \Psi \in \Delta \ \right\}, \emptyset \right) \tag{12}$$

- *For every solution δ to the problem Σ_Δ there exists a maximal (w.r.t. $\subseteq$) set of preferences $\Delta_\delta \subseteq \Delta$ such that $\delta \models_\mathcal{G} \bigwedge_{\Psi \in \Delta_\delta} \Psi$;*
- *For every pair of solutions δ and δ' of Σ_Δ, either $\Delta_\delta = \Delta_{\delta'}$ or $\Delta_\delta \not\subseteq \Delta_{\delta'}$ and $\Delta_{\delta'} \not\subseteq \Delta_\delta$.*

5 Goals Default Theories with Priorities

Proposition 1 shows that goal default theories can be used to specify planning problems with multiple preferences which might not be consistent with each other. For instance, consider a traveler from New York to San Francisco who has two preferences: reach the destination as fast as possible (Φ_1) and spend the least amount of money (Φ_2). In general, these two preferences cannot be satisfied at the same time. In this case, it is more reasonable to assume that a plan satisfying one of the criteria is an acceptable solution. Thus, $\Sigma_{\{\Phi_1,\Phi_2\}}$ is a reasonable goal specification if the traveler is impartial about Φ_1 and Φ_2. On the other hand, if the traveler prefers Φ_1 over Φ_2 (or vice versa), we will need to change the goal specification or provide additional ways for the traveler to specify this priority. As it turns out, the literature is rich with approaches for adding priorities to default theories [7, 8] which can be easily adapted to goal default theories. We next define goal default theories with priorities by adapting the work of [7] to goal default theories.

Let us start by introducing static priorities, encoded by a well-ordering relation $\prec$ among p-defaults—i.e., $\prec$ is transitive, irreflexive, and each set of elements admits the least element in the ordering. We denote with $\min_\prec(X)$ the least element of X with respect to $\prec$. We define goal default theory with priorities as follows.

Definition 7. *A* goal default theory with priorities *over a goal language $\mathcal{G} = (\mathcal{F}, \models_\mathcal{G})$ and a transition system T is a triple $(D, W, \prec)$ where D is a set of p-defaults over $\mathcal{G}$, $\prec$ is a well-ordering relation over D, and $W \subseteq \mathcal{F}$.*

Following the general design of prioritizing default theory [7], the notion of preferred extension can be defined by successively simplifying the structure of the defaults.

Let us identify a construction of preferred extension through the application of defaults according to the ordering imposed by $\prec$. Let us introduce the $\mathcal{PR}$ operator which computes the next "preferred" set of goal formulae from an existing one:

- $\mathcal{PR}_\prec(S) = Decl(S \cup \{cons(d)\})$
  if $\Pi^*_D(S) \neq \emptyset \wedge d = \min_\prec(\{x \mid x \in \Pi^*_D(S)\})$;
- $\mathcal{PR}_\prec(S) = S$ if $\Pi^*_D(S) = \emptyset$

where $\Pi^*_D(S) = \{d \mid d \in \Pi_D(S), S \not\models cons(d)\}$. If the elements in D (for a goal default theory (D, W)) are supernormal, then it is possible to use $\mathcal{PR}_\prec$ to produce a monotone sequence of goal formulae, by setting $S_0 = Decl(W)$, $S_{i+1} = \mathcal{PR}_\prec(S_i)$ for any successor ordinal $i + 1$ and $S_i = Decl(\bigcup_{j\leq i} S_j)$ for any limit ordinal i. We will denote the result of this construction as $Pref_\prec(D, W) = \bigcup_{i\geq 0} S_i$.

The process of determining a preferred extension will apply $Pref_\prec$ on a reduced version of the theory, in a style similar to that used in the Gelfond-Lifschitz reduct. Following the model proposed in [7], the reduct of a goal default theory with priorities $(D, W, \prec)$ w.r.t. a set of goal formulae S, denoted $(D^S, W, \prec^S)$, is obtained as follows:

- Determine $D' = \{\frac{\top \,:\, just(d)}{cons(d)} \mid d \in D, S \models_\mathcal{G} prec(d)\}$
- Determine $D^S = \{d \in D' \mid cons(d) \notin S \text{ or } S \not\models_\mathcal{G} \neg just(d)\}$ and $\prec^S$ is such that $d'_1 \prec^S d'_2$ if $d_1 \prec d_2$ and d_1 (d_2) is the $\prec$-least element that introduced d'_1 (d'_2) in D'.

We define preferred extensions as follows.

Definition 8. *Let $(D, W, \prec)$ be a goal default theory with priorities over $\mathcal{G} = (\mathcal{F}, \models_\mathcal{G})$ and T. A* preferred extension *E of $(D, W, \prec)$ is a set of goal formulae in $\mathcal{F}$ such that E is an extension of (D, W) and $E = Pref_{\prec^E}(D^E, W)$.*

Similar to [7], we can generalize the above definitions and define (*i*) a goal default theory with priorities as a triple $(D, W, \prec)$ where (D, W) is a goal default theory and $\prec$ is a partial order among defaults in D; and (*ii*) a set of formulae E is a preferred extension of $(D, W, \prec)$ if it is a preferred extension of some $(D, W, \prec_E)$ for some well-ordering $\prec_E$ which is an extension of $\prec$. For brevity, we omit the precise definitions. Definitions 5 and 6 can be extended in the obvious way: a planning problem is a goal default theory with priorities $(D, W, \prec)$ and its solutions are preferred extensions of $(D, W, \prec)$.

Example 5. *Let us consider the domain in Example 1. Let us assume that, among the objects, there is a very valuable object o_1 and a dangerous object o_2. Furthermore, let us assume that the robot is equipped with actions that can detect the object o_2 whenever the robot is at the same location as o_2. However, the equipment might not be working. We will denote with* working *the fact that the equipment is working properly. Let us consider the two formulae:*

- $\varphi := \Diamond h(o_1)$*: the robot should try to get the object o_1*
- $\psi := \Box[\bigwedge_{i\in\{1,\dots,k\}}(o\_at(o_2, i) \Rightarrow \neg at(i))]$*: the robot should not be at the same place with object o_2 at any time.*

With these formulae, we can define the following p-defaults:

$$g_1 \equiv \frac{\top \;:\; working}{\psi \wedge \varphi} \qquad\qquad g_2 \equiv \frac{\top \;:\; \neg working}{\varphi}$$

g_1 indicates that if the equipment is initially working, then the robot will get o_1 while trying to avoid o_2. g_2 states that if the equipment is not working, then the robot will only worry about getting o_1. The theory $(\{g_1, g_2\}, \emptyset, \{g_1 \prec g_2\})$ states that we prefer that the robot tries to satisfy g_1 before trying to satisfy g_2.

6 Related Work and Discussion

In this section, we relate goal default theories with priorities to existing goal specification languages. We then discuss possible applications of the new language.

- **TEP formulae**: TEP formulae have been implemented in a planner in [1]. Given a set of TEP formulae $\Delta = \{\Phi_1, \dots, \Phi_n\}$, a planning problem is an optimization problem that maximizes the rewards obtained by satisfying the formulae in Δ. Formally, the reward over a plan δ is

$$\Sigma_{\Phi_i\in\Delta,\delta\models\Phi_i} reward(\Phi_i) - \Sigma_{\Phi_i\in\Delta,\delta\not\models\Phi_i} penalty(\Phi_i)$$

  where $reward(\Phi)$ and $penalty(\Phi)$ denote the reward and penalty for satisfying and not satisfying Φ, respectively.
  The planning problem can be expressed by a goal default theory with priorities as follows. Let S be a set of formulae, $S \subseteq \Delta$, and d_S be the default

$$\frac{\top \;:\; \bigwedge_{\Phi\in S}\Phi \wedge \bigwedge_{\Phi\in\Delta\setminus S}\neg\Phi}{\bigwedge_{\Phi\in S}\Phi \wedge \bigwedge_{\Phi\in\Delta\setminus S}\neg\Phi}$$

  Let $D_\Delta = \{d_S \mid S \subseteq \Delta\}$ and $\prec_\Delta$ be the partial order over D_Δ where $d_S \prec_\Delta d_{S'}$ if

$$\begin{array}{l}\Sigma_{\Phi_i\in S} reward(\Phi_i) - \Sigma_{\Phi_i\notin S} penalty(\Phi_i) \geq \\ \Sigma_{\Phi_i\in S'} reward(\Phi_i) - \Sigma_{\Phi_i\notin S'} penalty(\Phi_i).\end{array}$$

  We can show that $(D_\Delta, \emptyset, \prec_\Delta)$ is a goal default theory with priorities representing the given planning problem, i.e., any preferred solution of $(D_\Delta, \emptyset, \prec_\Delta)$ is a solution of the original planning problem and vice versa.
- $\mathcal{PP}$: The language $\mathcal{PP}$ allows the specification of three types of preferences. A *basic desire* φ is a preference over a trajectory and therefore is a part of the basic goal language. An atomic preference is an ordering among basic desires $\Phi = \Phi_1 \lhd \Phi_2 \dots \lhd \Phi_k$ and expresses that the preference Φ_i is more important than Φ_{i+1} for $1 \leq i < k - 1$. An *atomic preference* Φ can be represented by the following goal default theory with priorities

$$\left(\left\{ \frac{\top \,:\, \Phi_i}{\Phi_i} \quad\middle|\quad i = 1, \dots, k \right\}, \emptyset, \prec_\Phi\right)$$

  where $\prec_\Phi$ is defined by $\frac{\top \,:\, \Phi_i}{\Phi_i} \prec_\Phi \frac{\top \,:\, \Phi_j}{\Phi_j}$ for $1 \leq i < j \leq k$.
  A general preference is either an atomic preference or a combination of general preferences, such as $\Phi \& \Psi$, $\Phi|\Psi$, and $!\Phi$, where Φ and Ψ are general preferences. Intuitively, general preferences add finitely many levels to the specification of preferences and

thus cannot be easily represented by goal default theories which assume ceteris paribus over the preferences. Adding priorities allows only an extra layer of comparison between preferences. We view this as a weakness of goal default theories and plan to further investigate this issue.

- **N-LTL** and **ER-LTL**: These two languages allow the specification of weak and strong exceptions within goal formulae represented as LTL-formulae by introducing labels to LTL-formulae. By compiling away the labels as in [4], we can show that $\mathcal{G}_b$ subsumes N-LTL and ER-LTL.

  Observe that the constructs used in N-LTL and ER-LTL are fairly close to default logic. This leads us to believe that interesting collections of N-LTL (ER-LTL) theories can be translated into goal default theories—which would provide a reasonable semantics for N-LTL (ER-LTL) theories with loops that have not been considered so far.

  Finally, we would like to note that $\mathcal{G}_b$ can be easily extended to consider N-LTL (ER-LTL) formulae by

  - extending $\mathcal{F}_b$ with N-LTL (ER-LTL) formulae; and
  - extending $\models_{\mathcal{G}_b}$ to define that $\delta \models_{\mathcal{G}_b} S$ iff $\delta \models_{\mathcal{G}_b} c(S)$ where $c(S)$, a LTL formula, denotes the result of compiling S to an LTL formula as described in [4, 5].

- π-CTL* and P-CTL*: These two languages consider non-deterministic domains and define goals over policies but do not consider preferences among goals. In addition, these languages introduce the operators A, E, A_π, and E_π over paths and the two quantifiers $\mathcal{EP}$ and $\mathcal{AP}$ over state formulae. Nevertheless, we can show that the CTL* part of π-CTL* can be expressed in $\mathcal{G}_b$. Furthermore, $\mathcal{G}_b$ can be extended to allow formulae of π-CTL*. However, the two new state quantifiers are not expressible in our goal language. We observe that as the goal language is parameterized with the satisfaction relation, $\mathcal{G}_b$ can be easily extended with these operators. We strongly believe that these extensions will be sufficient for goal default theories with priorities to capture P-CTL*.

The above discussion highlights features from existing goal languages that can (or cannot) be expressed by our goal language. This also shows that the proposed language can serve as a unified language for evaluating goal languages. The use of default theories as the basic language also provides us with an advantage in the study of computational complexity of goal languages. In this effort, we expect that well-known complexity results on prioritized default theories [13] will be extremely useful. This will provide us with insights for the use of existing goal languages as well as the development of new goal languages.

7 Conclusions and Future Work

In this paper, we describe a default logic based approach to defining non-monotonic goal specification languages. We start with a basic goal specification language and use default logic (or prioritizing default logic) to provide a natural way for dealing with inconsistency and priorities over goals. We show that the new language subsumes some goal languages in the literature and can describe several features from other goal languages. We identify desirable features that cannot be easily expressed by our goal language, among them is the multi-level of preferences between goals, which we intend to investigate in the near future. We also discuss possible applications of the proposed goal language.

Acknowledgments: The first two authors were partially supported by the NSF grants IIS-0812267.

REFERENCES

[1] Jorge A. Baier, Fahiem Bacchus, and Sheila A. McIlraith, 'A heuristic search approach to planning with temporally extended preferences', *Artif. Intell.*, **173**(5-6), 593–618, (2009).

[2] Chitta Baral and Jicheng Zhao, 'Goal specification in presence of non-deterministic actions', in *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004*, eds., Ramon López de Mántaras and Lorenza Saitta, pp. 273–277. IOS Press, (2004).

[3] Chitta Baral and Jicheng Zhao, 'Goal specification, non-determinism and quantifying over policies', in *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*. AAAI Press, (2006).

[4] Chitta Baral and Jicheng Zhao, 'Non-monotonic temporal logics for goal specification', in *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, ed., Manuela M. Veloso, pp. 236–242, (2007).

[5] Chitta Baral and Jicheng Zhao, 'Non-monotonic temporal logics that facilitate elaboration tolerant revision of goals', in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, eds., Dieter Fox and Carla P. Gomes, pp. 406–411. AAAI Press, (2008).

[6] M. Bienvenu, C. Fritz, and S. McIlraith, 'Planning with qualitative temporal preferences', in *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR06)*, pp. 134–144, Lake District, UK, (June 2006).

[7] G. Brewka and T. Eiter, 'Prioritizing default logic', in *Intellectics and Computational Logic*, volume 19 of *Applied Logic Series*, 27–45, Kluwer, (2000).

[8] James Delgrande and Torsten Schaub, 'Expressing preferences in default logic', *Artificial Intelligence*, **123**, 41–87, (2000).

[9] E. A. Emerson, 'Temporal and modal logic', in *Handbook of theoretical computer science: volume B*, ed., J. van Leeuwen, 995–1072, MIT Press, (1990).

[10] M. Fox and D. Long, 'PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains', *Journal of Artificial Intelligence Research*, **20**, 61–124, (2003).

[11] A. Pnueli, 'The temporal logic of programs', in *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 46–57, (1977).

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

[13] Jussi Rintanen, 'Complexity of prioritized default logics', *Journal of Artificial Intelligence Research*, **9**, 423–461, (1998).

[14] Tran Cao Son and Enrico Pontelli, 'Planning with Preferences using Logic Programming', *Theory and Practice of Logic Programming*, **6**, 559–607, (2006).