Domain-Specific Preferences for Causal Reasoning and Planning

James P. Delgrande
School of Computing Science
Simon Fraser University
Burnaby, B.C.
Canada V5A 1S6
jim@cs.sfu.ca

Torsten Schaub[*]
Institut für Informatik
Universität Potsdam
Postfach 90 03 27
D–14439 Potsdam, Germany
torsten@cs.uni-potsdam.de

Hans Tompits
Institut für Informationssysteme 184/3
Technische Universität Wien
Favoritenstraße 9–11
A–1040 Wien, Austria
tompits@kr.tuwien.ac.at

Abstract

We address the issue of incorporating domain-specific preferences in planning systems, where a preference may be seen as a "soft" constraint that it is desirable, but not necessary, to satisfy. To this end, we identify two types of preferences, *choice preferences* that give a preference over which formulas (typically subgoals) to establish, and *temporal preferences*, which specify a desirable ordering on the establishment of formulas. Preferences may be constructed from actions or fluents but, as we show, this distinction is immaterial. In fact, we allow preferences on arbitrary formulas build from action and fluent names. These preference orderings induce preference ordering on resulting plans, the maximal elements of which yield the preferred plans. We argue that the approach is general and flexible; as well, it handles conditional preferences. Our framework is developed in the context of transition systems; hence, it is applicable to a large number of different action languages, including the well-known language $\mathcal{C}$. Furthermore, our results are applicable to general planning formalisms.

Introduction

In *planning*, the task is to specify a sequence of actions that will achieve a particular goal, given a specification of a (dynamic) domain and an initial situation. Traditional planners have often been based on, or derived from STRIPS (Fikes & Nilsson 1971), or some logical encoding of an action domain, such as the methods by Levesque, Pirri, & Reiter (1998) or Thielscher (1999), although of course there are many others. More recently there has been interest in specifying planning problems in terms of action languages (Gelfond & Lifschitz 1998), based in turn on the notion of a transition system. However, one thing that these approaches have in common is that a successful plan is one that is *executable* (i.e., the various actions can in fact be carried out), and that achieves the given goal; otherwise, it is not a successful plan. However, this fails to allow for desirable but nonessential results, that is to say, "soft constraints" or preferences in the planning process. With the possible exception of some research on incorporating resource constraints in planning, there has been little work in incorporating such preferences.

In realistic domains, preferences are pervasive. Consider the informal example of spending an evening out. The goal is to have a pleasant evening, where a pleasant evening consists of going to a movie, having dinner, etc. You prefer to eat Japanese food to French (etc.); you prefer to eat before the movie. If the theatre is far, you prefer to take public transit to driving. At the theatre, you prefer to find a seat first, then buy popcorn. If you are going on a date, you prefer a romantic comedy to an action flick; if not, you prefer any other film type to a romantic comedy. In each of these cases, a successful plan may be obtained, even though preferences are violated—for example, eating at a French restaurant rather than Japanese does not preclude having a pleasant night out.

In this paper, we consider the general problem of planning in the presence of qualitative, domain-specific preferences, such as in the above example. This is in contrast to *domain-independent preferences*, such as plan length, which do not depend on the meaning of the fluents involved.

In the next section, we consider the general issue of the types of preferences encountered in planning, and suggest a classification. The following section discusses related work. Following this, we present our formal approach; this is done in the context of transition systems, although our results are readily applicable to other approaches. It appears that incorporating preferences in planning unavoidably involves a metalevel (or equivalent) approach, in which one can essentially compare plans. We conclude the section with an extended monkey-and-bananas example. We finish by first considering basic properties of our approach, including complexity issues, and conclude with a brief discussion. Given space limitations, we assume a basic knowledge of planning and action languages; see the aforecited references.

Preferences in Planning

As described, our interests lie with domain-specific qualitative preferences. To begin with, we propose that preferences be classified via two dimensions.

Fluent versus Action Preference

This concerns the object types that a preference applies to: informally, preferences can be expressed between fluents or between actions. In the first case, one might prefer to have

[*] Affiliated with the School of Computing Science at Simon Fraser University, Burnaby, Canada.

white wine to red; in the latter, one might prefer to travel via transit to driving. This distinction is by no means clear-cut however, and a natural-language assertion representing a preference can often be interpreted as being either between actions or between fluents. For example, preferring to have white wine to red seems to be a preference between fluents, whereas preferring to drink white wine over red seems to be a preference on actions. However, we subsequently show that preferences on actions can be equivalently specified as preferences on fluents, and so one can restrict preferences to being on fluents only with no loss of generality. However, unless otherwise stated, we take preferences to be expressed on *formulas*, where a formula is a Boolean expression built from fluent and action names.

Choice versus Temporal Preference

For *choice preference*, one has preferences (typically mutually exclusive) concerning how a subgoal is to be attained. So in our example, the *dinner* subgoal is preferentially satisfied by having Japanese food over French food. For *temporal preference*, one has preferences concerning the order in which subgoals are to be achieved. Thus, the subgoal of having had *dinner* should be preferentially satisfied before that of *movie*. Similarly, choice and temporal preferences involving actions are easily constructed. Thus, we classify a preference as to whether it is a choice preference, or a preference on choosing which formula to establish (expressed by $\leq_c$), or a temporal preference between formulas (expressed by $\leq_t$). These will be specified via two partial preorders. Hence, $\alpha_1 \leq_c \alpha_2$ expresses a choice preference in which α_2 is not less preferred to α_1. One can define the strict counterparts of these orders in the usual way, so that $\alpha_1 <_c \alpha_2$ expresses a choice preference in which α_2 is strictly preferred to α_1. The domains of $\leq_c$ and $\leq_t$ are *formulas*, where a formula is a Boolean expression built from fluent and action names.

Preferences between formulas induce preferences between plans. That is, $\alpha_1 <_c \alpha_2$ expresses the strict preference for a plan in which α_2 is true over one in which α_1 is true (all other things being equal). A temporal preference $\alpha_1 \leq_t \alpha_2$ specifies that we prefer plans (or histories) in which α_1 is achieved prior to α_2 no less than plans where this is not the case. This leads to a third potential distinction on preferences, viz. whether a preference is *relative* to others or *absolute*. In this latter case one might wish to assert that α_1 is (simply) desirable. However this preference can be expressed via choice preference, by $\neg\alpha_1 \leq_c \alpha_1$. Consequently, we do not consider it as an independent classification.

There are other factors to consider. For example, we allow conditional preferences, or preferences applicable in a given context (e.g., in our example, if the distance is great, transit is preferred over driving). We do not address combining preference relations (since this issue is by and large independent of realising preferences in planning). As well, we do not consider domain-independent preferences, for example that short plans are preferred.

Related Work

There has been some work in expressing procedural control knowledge in planning. For example, Bacchus & Kabanza (2000) show how the performance of a forward-chaining planner can be improved by incorporating domain-specific control knowledge; cf. also Doherty & Kvarnstrom (1999) and Nau *et al.* (1999). Son *et al.* (2002) address domain and procedural control knowledge, as well as ordering constraints, in an action language expressed via an extended logic program. Since these are "hard" constraints, requiring (rather than suggesting) that, e.g., some action precedes another, the goal of such work differs from ours.

Work on preferences in planning per se is limited. Wellman & Doyle (1991) suggest that the notion of *goal* is a relatively crude measure for planners to achieve, and rather that a relative preference over possible plan outcomes constitutes (or should constitute) a fundamental objective for planning. They show how to define goals in terms of preferences and, conversely, how to define (incompletely) preferences in terms of sets of goals.

Myers & Lee (1999) assume that there is a set of desiderata, such as affordability or time, whereby successful plans can be ranked. A small number of plans is generated, where the intent is to generate divergent plans. The best plan is then chosen, based on a notion of Euclidean distance between these select attributes. In related work, Haddawy & Hanks (1992) use a utility function to guide a planner.

A interesting approach to preference handling in planning is proposed by Son & Pontelli (2002). This approach extends prioritised default theory (Gelfond & Son 1998) in order to encode actions theories in language $\mathcal{B}$ (Gelfond & Lifschitz 1998). This extension allows for expressing preferences over histories. It is shown how this can be implemented in terms of answer set programming. Their notion of preference, $<$, corresponds to an instance of choice preference on actions: History H_1 is preferred to H_2 if

1. there is an index i and actions b_i in H_1 and a_i in H_2 such that $a_i < b_i$, and
2. for every $j < i$, one has not $b_j < a_j$.

In translating from an action theory to a prioritised default theory, essentially all possible trajectories are encoded in the resulting rules. Consequently, the resulting prioritised theory is very large. As well, they then have essentially a meta-level approach (like ours) since their preferences are on these encoded trajectories. Our approach will generally be more efficient than theirs, since we just consider plans, whereas they encode all action sequences. Further, our approach easily adapts to an anytime algorithm: generate plans selecting the most preferred; at any time the process can be halted and the best-so-far plans returned.

Furthermore, in the definition of a preferred trajectory, comparing action preferences, the compared actions term $prefer(a_i, b_i)$ indicates that the respective actions must occur at the same time point in each trajectory. But this is unrealistic; it may well be for example that it takes more steps to establish the preconditions of action a than those of (preferred) action b; hence in such a case the definition cannot be applied.

Consider where I have an apple and a banana, and I prefer to eat the banana to eat the apple. I can eat the apple immediately, as Action 1. For the banana, I have to peel it (Action 1) and then can eat it (Action 2). However, I cannot compare these trajectories since $eat(apple)$ is at Time Point 1 in the first trajectory and $eat(banana)$ is at Time Point 2 in the second trajectory. (There is a third trajectory where I do a wait at Time Point 1, and $eat(apple)$ at Time Point 2. The second trajectory is preferred to the third trajectory, but this does not solve the problem since the most preferred trajectory should still be trajectory 1.) Last, in preferences among formulae, the preference relation is assumed to be a total order. We allow a partial preorder and are thereby more general.

The latter approach is extended where a declarative language for specifying preferences between trajectories is presented (Son & Pontelli 2004). This language is an extension of action language $\mathcal{B}$ (Gelfond & Lifschitz 1998). Interestingly, it is shown how this preference language can be compiled into logic programs under answer sets semantics. For comparing trajectories, their compilation technique nicely takes advantage of special features for maximising cardinalities provided by the answer set solver Smodels (Niemelä & Simons 1997). The basic notion of preference explored by Son & Pontelli (2004) is based on so-called *desires*, expressed through formulas like ϕ, from which preferences among trajectories are induced in the following way: Given a desire ϕ, a trajectory H is preferred to H' if $H \models \phi$ but $H' \not\models \phi$. (A desire would be expressed in our approach as $\neg\phi <_c \phi$.) This concept is then extended to allow for more complex expressions by means of propositional as well as temporal connectives such as *next*, *until*, etc.

Eiter *et al.* (2003a) describe planning in an answer set framework in which action costs are taken into account. The approach allows the declarative specification of desiderata such as computing the shortest plan, or the cheapest plan, or some combination of these criteria. This is realised by employing weak constraints, which realise filterings of answer sets, and thus of plans, based on *quantitative* criteria.

Transition Systems

We base our formal elaboration on the notion of a *transition system* (Gelfond & Lifschitz 1998), although the approach is readily expressible in standard planning formalisms as well. (Furthermore, in view of the results by **?** (**?**), we can translate our approach into STRIPS.) This subsection recapitulates material taken from Gelfond & Lifschitz (1998).

Definition 1 *An* action signature, *σ, is a triple $\langle V, F, A\rangle$, where V is a set of* value names, *F is a set of* fluent names, *and A is a set of* action *names.*

If $V = \{1, 0\}$, then σ is called propositional. *If V, F, and A are finite, then σ is called* finite.

Definition 2 *A* transition system, *T, over an action signature $\sigma = \langle V, F, A\rangle$ is a triple $\langle S, \nu, R\rangle$, consisting of*

1. *a set S;*
2. *a function $\nu : F \times S \rightarrow V$; and*
3. *a subset R of $S \times A \times S$.*

The elements of S are called *states*. The mapping ν is called a *valuation function*, with $\nu(f, s)$ being the *value* of fluent name f in state s. The states s' such that $\langle s, a, s'\rangle \in R$ are the *possible results of the execution* of action a in state s. If there is at least one triple $\langle s, a, s'\rangle$ contained in R, then a is said to be *executable* in s. We say that action a is *deterministic in s* iff there is at most one such s'. For simplicity, we assume that each action a is deterministic throughout the remainder of this paper.

If σ, S, and R are finite, then T is called a *finite* transition system. For a propositional transition system $T = \langle S, \nu, R\rangle$, fluent f is said to be *true at* state s iff $\nu(f, s) = 1$, otherwise f is *false at s*.

We sometimes use pseudo-first-order notation for specifying fluent names.

Definition 3 *Let $T = \langle S, \nu, R\rangle$ be a transition system over an action signature $\sigma = \langle V, F, A\rangle$. A* history, *$H$, of T is a sequence*

$$(s_0, a_1, s_1, a_2, s_2, \ldots, s_{n-1}, a_n, s_n),$$

where $s_0, \ldots, s_n \in S$, $a_1, \ldots, a_n \in A$, and $n \geq 0$, such that

$$\langle s_{i-1}, a_i, s_i\rangle \in R,$$

for $1 \leq i \leq n$.

We call n the length *of history H. Furthermore, the set of all histories of transition system T is denoted $\mathcal{H}_T$, or, if T is clear from the context, simply $\mathcal{H}$.*

Note that we defined histories in such a way that concurrent actions are excluded, i.e., at each time point, at most one action takes place. This is just for the sake of simplicity and means no restriction of the general case in the sense that our preference framework introduced below works just as well if concurrent actions are allowed. Furthermore, in order to model the flow of time where no actual actions occur (i.e., modelling a “tick of the clock”), it is convenient to allow a vacuous action $do\_nothing$ satisfying $\langle s, do\_nothing, s\rangle$ for all states s.

We assume in the remainder of this paper that action signatures are always propositional. Also, we use the query language $\mathcal{Q}_n$, adapted from Gelfond & Lifschitz (1998), where n is a nonnegative integer which fixes the maximum length of the histories under consideration, introduced next.

Definition 4 *Elements of the query language $\mathcal{Q}_n$ are recursively defined as follows:*

1. *An* atom *is an expression of form $a : i$, where a is an elementary action name and $i < n$, or an expression of form $f : i$, where f is a fluent name and $i \leq n$;*
2. *an* axiom *is an atom possibly preceded by the negation sign $\neg$; and*
3. *a* query *is a Boolean combination of atoms.*

We call the index i the time stamp *of an atom $e : i$ or $f : i$.*

For convenience, if the atoms of a query Q all have the same time stamp, we abbreviate Q by the expression $\phi : i$, where ϕ is the Boolean combination of the action and fluent names comprising Q.

Definition 5 *Let T be a transition system,*

$$H = (s_0, a_1, s_1, a_2, s_2, \ldots, s_{n-1}, a_n, s_n)$$

a history of T of length n, and Q a query over $\mathcal{Q}_n$.

The relation $H \models_T^n Q$ is recursively defined as follows:

1. *If $Q = a : i$, for an action name a, then $H \models_T^n Q$ iff $a = a_{i+1}$;*
2. *if $Q = f : i$, for a fluent name f, then $H \models_T^n Q$ iff f is true at s_i;*
3. *if Q is non-atomic, then $H \models_T^n Q$ is defined as in propositional logic.*

If $H \models_T^n Q$ holds, then H *satisfies* Q. Given a set Γ of axioms, we define $\Gamma \models_T^n Q$ iff every history of T of length n which satisfies all elements of Γ also satisfies Q. For simplicity, if T and n are unambiguously fixed, we usually write $\models$ instead of $\models_T^n$. Given that we deal with deterministic actions, a history H that satisfies query Q can be regarded as a plan that satisfies goal Q.

Incorporating Preferences in Planning

We now describe our approach for dealing with preferences in the context of reasoning about actions. First, we extend the notion of a transition system, taking preference information among fluents and actions into account, and then we describe how this initial preference information induces preferences among histories.

Prioritised Transition Systems

The following concept is central.

Definition 6 *Given a transition system T over action signature $\sigma = \langle V, F, A\rangle$, let $\mathcal{L}$ be the propositional language over atomic sentences in $F \cup A$.*

A prioritised transition system *over σ is a triple*

$$P = \langle T, \leq_c, \leq_t\rangle,$$

where T and σ are as above, and $\leq_c, \leq_t \subseteq \mathcal{L} \times \mathcal{L}$ are partial preorders.

The relation $\leq_c$ is called choice preference *and $\leq_t$ is called* temporal preference.

Recall that a partial preorder is a binary relation which is reflexive and transitive. Using preorders has the advantage that one may distinguish between *indifference* (where both $f \leq g$ and $g \leq f$ hold) and *incomparability* (where neither $f \leq g$ nor $g \leq f$ holds). As usual, given a partial preorder $\leq$ over some set W, we define its strict part, $<$, by $x < y$ iff $x \leq y$ but $y \not\leq x$, for all $x, y \in W$.

The preference relations $\leq_c$ and $\leq_t$ are defined over *formulas*, where a formula is a member of $\mathcal{L}$, i.e., a propositional formula without time stamps. Intuitively, if $\alpha_1 \leq_c \alpha_2$, then we prefer histories in which α_2 is true no less than histories in which α_1 is true. A temporal preference $\alpha_1 \leq_t \alpha_2$, on the other hand, specifies that, if possible, α_1 should become true not later than α_2 becoming true in a history.

In order to talk about a formula being "true in a history", as in the preceding paragraph, we need to extend Definition 5 to deal with the truth of expressions without time stamps.

Definition 7 *Let T be a transition system, as given in Definition 5; let H be a history of T; and let α be a Boolean combination of fluent and action names. Define*

$$H \models_T^n \alpha \textit{ iff } H \models_T^n \alpha : i \textit{ for some } i,\ 0 \leq i \leq n.$$

If $H \models_T^n \alpha$, we say that $\alpha \in \mathcal{L}$ is true in history H, or that H satisfies α. We sometimes write $\models$ for $\models_T^n$ if T and n are clear from the context.

This notation allows, for example, $H \models_T^n \alpha$ and $H \models_T^n \neg\alpha$ to both hold where $\alpha \in \mathcal{L}$, while $H \models_T^n (\alpha \wedge \neg\alpha) : i$ will hold for no index i.

We assume that concepts defined for regular transition systems are similarly defined for prioritised transition systems. For instance, given a prioritised transition system $P = \langle T, \leq_c, \leq_t\rangle$, the histories of P are given by the histories of T.

Choice Preference

We first deal with choice preferences. For binary relation R, define

$$\mathsf{dom}(R) = \{x, y \mid \langle x, y\rangle \in R\},$$

and let R^* be the transitive closure of R.

Given a prioritised transition system $P = \langle T, \leq_c, \leq_t\rangle$ and the set $\mathcal{H}$ of histories of T, we have that there is a partial order $\preceq_c \subseteq \mathcal{H} \times \mathcal{H}$, induced by $\leq_c$, in a manner specified below. The $\preceq_c$-maximal histories will correspond to (choice-) preferred plans. More formally:

Definition 8 *A history H is $\preceq_c$-*preferred *iff it is maximal with respect to relation $\preceq_c$.*

To develop our definition of $\preceq_c$, we first give some preliminary terminology. Note that Definition 8 is formulated without reference to a particular query.

Let $P = \langle T, \leq_c, \leq_t\rangle$ be a prioritised transition system over $\sigma = \langle V, F, A\rangle$, and let H and H' be two histories of T. We define the following set of formulas:

$$\Delta_P(H, H') \quad = \quad \{\alpha \in \mathsf{dom}(\leq_c) \mid H \models \alpha \text{ and } H' \not\models \alpha\}.$$

That is, $\Delta_P(H, H')$ consists of all formulas related by $\leq_c$ which are satisfied at some point in H but never in H'.

Definition 9 *Let $P = \langle T, \leq_c, \leq_t\rangle$ be a prioritised transition system, and let H and H' be two histories of P.*

Then, $H \trianglelefteq_c^P H'$ iff for any formula $\alpha \in \Delta_P(H, H')$, there is some $\alpha' \in \Delta_P(H', H)$ such that $\alpha \leq_c \alpha'$.

If P is unambiguously fixed, we simply write $H \trianglelefteq_c H'$ instead of $H \trianglelefteq_c^P H'$. The reason for using $\Delta_P(H, H')$ in the above definition is that we are only interested in formulas which are not jointly satisfied by the two histories H and H'. A similar construction using "difference sets", though defined on different objects of discourse, was used by Geffner & Pearl (1992).

Clearly, $\trianglelefteq_c$ is reflexive, that is, we have $H \trianglelefteq_c H$ for any history H. However, $\trianglelefteq_c$ is in general not transitive. To see this, consider a prioritised transition system, P, involving three fluents, f, g, and h, such that $f \leq g \leq h$ and $h \leq g$, and assume that we have three histories H, H', H'', such that the following relations hold (by means of suitable actions):

$$\begin{array}{ccc} H \models f, & H \not\models g, & H \models h; \\ H' \not\models f, & H' \models g, & H' \not\models h; \\ H'' \not\models f, & H'' \not\models g, & H'' \models h. \end{array}$$

From this, we get the following sets of differing fluents:

$$\begin{array}{ll} \Delta_P(H,H') = \{f,h\}; & \Delta_P(H',H) = \{g\}; \\ \Delta_P(H',H'') = \{g\}; & \Delta_P(H'',H') = \{h\}; \\ \Delta_P(H,H'') = \{f\}; & \Delta_P(H'',H) = \emptyset. \end{array}$$

Then, it is easy to check that both $H \trianglelefteq_c H'$ and $H' \trianglelefteq_c H''$ hold, but not $H \trianglelefteq_c H''$.

In view of the non-transitivity of $\trianglelefteq_c$, we consider its transitive closure $\trianglelefteq_c^*$, which yields our "official" definition of choice preference over histories as follows.

Definition 10 *For two histories H and H', define*

$$H \preceq_c H' \quad \textit{iff} \quad H \trianglelefteq_c^* H'.$$

So, given the properties of $\trianglelefteq_c^*$, the relation $\preceq_c$ is clearly a partial preorder. Note that $\preceq_c$ may posses non-trivial cycles, i.e., there may exist sequences $H_1, \ldots, H_k$ of histories ($k > 1$) such that, for all $i \in \{1, \ldots, k-1\}$, $H_i \lhd_c H_{i+1}$ and $H_k \lhd_c H_1$. However, by definition, the strict part $\prec_c$ of $\preceq_c$ is always cycle free. If one wants that $\preceq_c$ also does not have any non-trivial cycles, one may replace Definition 10 by setting $H \preceq_c H'$ iff $H \lhd_c^* H'$ or $H = H'$, where $\lhd_c^*$ is the strict part of the transitive closure of $\trianglelefteq_c$.

Any history H satisfying Definition 10 would seem to be undeniably "preferred". However, other definitions are certainly possible (although arguably less compelling). For example, one could rank histories by the number of choice preferences violated. This alternative would make sense where the preferences were *absolute*, i.e., of the form $\neg\alpha <_c \alpha$. Other alternatives can be obtained by using variations of $H \trianglelefteq_c H'$. For instance, instead of the "∀-∃" quantification in Definition 9, one may opt for an "∃-∃" quantification when relating the elements in $\Delta_P(H,H')$ and $\Delta_P(H',H)$, respectively. Many more alternatives are obtainable in such a way within our framework.

Let us consider some basic examples illustrating the choice preference order $\preceq_c$. The first example arguably describes the most obvious case in which two histories are distinguished.

Example 1 *Consider a prioritised transition system, P_1, over fluents f and g such that $f \leq_c g$, and assume histories H and H' satisfying $H \models f \wedge \neg g$ and $H' \models \neg f \wedge g$.*

Then, we have that

$$\Delta_{P_1}(H,H') = \{f\} \quad \textit{and} \quad \Delta_{P_1}(H',H) = \{g\}.$$

Hence, it follows that $H \trianglelefteq_c H'$, and thus $H \preceq_c H'$. In fact, it holds that $H \prec_c H'$.

The next example illustrates our particular interpretation of choice preference.

Example 2 *Let P_2 be a prioritised transition system, comprised again of fluents f and g, and ordered by $f \leq_c g$, and consider histories H and H', where H obeys $H \models f \wedge \neg g$ as before, but H' satisfies $H' \models \neg f \wedge \neg g$.*

We thus obtain

$$\Delta_{P_2}(H,H') = \{f\} \quad \textit{and} \quad \Delta_{P_2}(H',H) = \emptyset.$$

Therefore, we get that $H \ntrianglelefteq_c H'$ and $H' \trianglelefteq_c H$.

Informally, this example illustrates the type of choice preference that we have chosen to implement: A preferred history with respect to P is one that satisfies the choice preferences as much as possible, but disfavouring histories like H' with an empty set $\Delta_P(H',H)$ of distinguishing preference formulas. Observe that no relation among H and H' is obtained in the aforementioned "∃-∃" variant of Definition 9.

Example 3 *Let P_3 be a prioritised transition system defined similarly to the ones in Examples 1 and 2, and H, H' such that $H \models f \wedge g$ and $H' \models \neg f \wedge g$.*

Then,

$$\Delta_{P_3}(H,H') = \{f\} \quad \textit{and} \quad \Delta_{P_3}(H',H) = \emptyset.$$

Both histories agree on (in fact, satisfy) the $\leq_c$-higher fluent, but differ on the $\leq_c$-lesser fluent. The result here is the same as in the previous example.

Given a query, the order $\preceq_c$ can be refined as follows.

Definition 11 *For a query Q and two histories H and H' satisfying Q, define $H' \preceq_c^Q H$ iff $H' \preceq_c H$.*

Note that every $\preceq_c$-maximal history satisfying Q is also $\preceq_c^Q$-maximal, but not vice versa.

Let us consider a more involved example now, based on the well-known monkey-and-bananas scenario (cf., e.g., (Giunchiglia *et al.* 2004)).

Example 4 *A monkey wants a bunch of bananas, hanging from the ceiling, or a coconut, found on the floor; as well, the monkey wants a chocolate bar, found in a drawer.*

In order to get the bananas or coconut, the monkey must push a box to the empty place under the respective item and then climb on top of the box. In order to get the chocolate, the drawer must be opened. Each object is initially at a different location.

We assume that the monkey wants the chocolates, and either the coconuts or the bananas, and he prefers bananas over coconuts. So here, no temporal preferences are specified.

Formally, we use a propositional prioritised transition system $P = \langle T, <_c, \emptyset \rangle$ over an action signature $\sigma = \langle \{0,1\}, F, A \rangle$, specified as follows:

$$\begin{array}{rcl} F & = & \big\{ loc(I, l_i) \mid I \in \{Monkey, Box, Ban, \\ & & \quad Drawer, Coco\}, 1 \leq i \leq 5 \big\} \\ & & \cup \{onBox, hasBan, hasChoc, hasCoco\}; \\ A & = & \{walk(l_i), pushBox(l_i) \mid 1 \leq i \leq 5\} \\ & & \cup \{climbOn, climbOff, graspBan, graspChoc, \\ & & \quad graspCoco, openDrawer\}; \\ \leq_c & = & \{hasCoco \leq_c hasBan\}. \end{array}$$

For this example, we use a concrete transition system $T = \langle S, \nu, R\rangle$ *based on the action language* $\mathcal{C}$ *(Giunchiglia & Lifschitz 1998; Gelfond & Lifschitz 1998); however, we omit the full* (*and straightforward*) *details.*

The query we are interested in is :

$$Q = \mathit{hasChoc} \wedge (\mathit{hasBan} \vee \mathit{hasCoco}) : 7.$$

Initially, the monkey does not have the chocolates, bananas, or coconuts, and each object is at a different location. There are, among others, two histories, H *and* H'*, satisfying* Q*:*[1]

History H	*Action*
STATE 0:	*go to the drawer*
STATE 1:	*open the drawer*
STATE 2:	*grasp the chocolates*
STATE 3:	*walk to the box*
STATE 4:	*push the box to the bananas*
STATE 5:	*climb on the box*
STATE 6:	*grasp the bananas*

History H'	*Action*
STATE 0:	*go to the drawer*
STATE 1:	*open the drawer*
STATE 2:	*grasp the chocolates*
STATE 3:	*walk to the coconuts*
STATE 4:	*grasp the coconuts*

Given the monkey's preference of bananas over coconuts, we expect that H *is preferred over* H'*. This is indeed the case, as it is easy to verify that* H *is* $\preceq_c^Q$*-preferred, but* H' *is not. For this, observe that* $\mathit{hasCoco} \in \Delta_P(H, H')$ *as well as* $\mathit{hasBan} \in \Delta_P(H, H')$*.*

We note that there are of course more histories satisfying the intended goal if we consider histories of length greater than 7. In particular, there are histories satisfying

$$Q' = \mathit{hasChoc} \wedge (\mathit{hasBan} \vee \mathit{hasCoco}) : 8$$

in which the subgoals are achieved in the reverse order as given by H *and* H' (*cf. Example 6 below*)*.*

Temporal Preference

With choice preference, the order $\leq_c$ specifies the relative desirability that a formula be true in a history. Thus $\alpha_2 \leq_c \alpha_1$ implicitly expresses a preference that holds *between* histories (viz., all other things being equal, a history with α_1 true and α_2 not is preferred to a history with α_2 true and α_1 not).

For temporal preferences, the order $\leq_t$ specifies the desired order in which formulas become true *within* a history. Thus, $\alpha_2 \leq_t \alpha_1$ implicitly expresses a preference that should hold *within* a history (viz., that the establishment of α_2 is not later than that of α_1). To this end, for $\leq_t$, it is convenient to be able to refer to the ordering on formulas given by a history.

Definition 12 *For a history* H*, define* $\alpha_1 \leq_H \alpha_2$ *iff*

1. $H \models \alpha_1$ *and* $H \models \alpha_2$*; and*
2. $\min\{j \mid H \models \alpha_1 : j\} \leq \min\{i \mid H \models \alpha_2 : i\}$.

We want to compare histories, say H and H', based on "how well" $\leq_H$ and $\leq_{H'}$ agree with $\leq_t$. Below we give our preferred means of comparing histories. However for the time being, we simply assume that we are given a preorder $\preceq_t \subseteq \mathcal{H} \times \mathcal{H}$, determined from the set of orderings $\{\leq_H\}_{H\in\mathcal{H}}$. That is, we assume that we have an ordering on members of $\mathcal{H}$ that reflects how well their associated ordering $\leq_H$ agrees with $\leq_t$.

The following definition parallels Definition 8:

Definition 13 *A history* H *is* $\preceq_t$-preferred *iff it is maximal with respect to relation* $\preceq_t$*.*

So, a temporally-preferred history H is a history whose associated temporal ordering on formulas $\leq_H$ "best" agrees with $\leq_t$. We next give our preferred definition of temporal preference $\preceq_t$ over histories.

First, it would seem that $H \in \mathcal{H}$ is temporally preferred if $\leq_H$ does not disagree with $\leq_t$; that is if $<_t \cap \leq_H^{-1} = \emptyset$.[2] We extend this to relative preference among histories as follows:

Definition 14 *For histories* H, H'*, define*

$$H' \preceq_t H \;\; \textit{iff} \;\;\; <_t \cap \leq_H^{-1} \subseteq <_t \cap \leq_{H'}^{-1} .$$

That is, H violates fewer preferences in $<_t$ than H' does. Obviously $\preceq_t$ is a partial preorder on $\mathcal{H}$.

Example 5 *Consider a simple example in which we are given* $\alpha_1 <_t \alpha_2$ *only, and where there are three histories,* H_1*,* H_2*, and* H_3*, satisfying a given query such that:*

- H_1 *satisfies the given preference, in that* α_1 *becomes true prior to* α_2*;*
- *in* H_2*, both* α_1 *and* α_2 *become true at the same state; and*
- α_1 *does not become true in* H_3*.*

According to Definition 12, we have

$$\begin{aligned} <_t \cap \leq_{H_1}^{-1} &= \emptyset; \\ <_t \cap \leq_{H_2}^{-1} &= \{\langle \alpha_1, \alpha_2\rangle\}; \quad \textit{and} \\ <_t \cap \leq_{H_3}^{-1} &= \emptyset. \end{aligned}$$

Consequently H_1 *and* H_3 *are temporally preferred histories, since neither violates the preference in* $<_t$*.*

Definition 14 is based on set containment among violations to $\leq_t$. As mentioned, there are alternatives. We could as easily (but less compellingly) base our definition on the cardinality of the set of violations, i.e.,

$$H' \preceq_t H \quad \text{iff} \quad |<_t \cap \leq_H^{-1}| \leq |<_t \cap \leq_{H'}^{-1}| .$$

Clearly, many more alternatives are obtainable by varying the underlying order $\leq_H$ in Definition 12. For instance, instead of using the minimal time point for selecting the earliest state of satisfaction, one may choose the maximal time point for focusing on the latest such states. As well, one may

[1] Observe that, strictly speaking, we have to further assume the tick-of-the-clock action *do_nothing* and apply it twice after State 4 in H' in order that evaluating Q at H' is defined.

[2] For binary relation R, R^{-1} is the relation satisfying $\langle x, y\rangle \in R^{-1}$ iff $\langle y, x\rangle \in R$.

simply require there to be two (arbitrary) time points, so that one formula becomes true before the other. More elaborated orderings could even take into account the number of times a formula is satisfied before another, etc. All this is possible within our framework.

Example 6 *Consider the transition system T and query Q' from Example 4, but where our preferences are now given by*

$$\leq_c = \emptyset \quad \textit{and} \quad \leq_t = \{hasBan \leq_t hasChoc\}.$$

The temporally preferred histories are those in which the bananas are obtained and then chocolate, and those in which coconuts and bananas are obtained. Another way of saying this is that the histories that are not temporally preferred are those violating $hasBan \leq_t hasChoc$.

If we add preference $hasCoco \leq_t hasChoc$, then the preferred histories are those where one of bananas or coconuts are obtained, and then chocolate. If we combine this preference with the choice preference in Example 4, we obtain a history in which both preferences can be satisfied. If the preferences were to conflict, then obviously only one can be satisfied; nonetheless, clearly such a conflict does not prevent us finding a successful plan. As before, there are four histories satisfying Q'. Of these three are temporally preferred:

- *one in which bananas are obtained and then chocolate,*
- *and two others in which coconuts and bananas are obtained.*

Another way of saying this is that there is one history that is not temporally preferred, and that is the history that violates $hasBan \leq_t hasChoc$. If we add preference $hasCoco \leq_t hasChoc$, then there are two preferred histories, where one of bananas or coconuts are obtained, and then chocolate.

Properties of the Preference Approach

In this section, we give some basic properties of our framework, including an analysis of the computational complexity of the main reasoning tasks involved. To begin with, the following result is trivial, but necessary for any approach to planning with preferences.

Theorem 1 *Let T be a transition system, Q a query over $\mathcal{Q}_n$, and $x \in \{c, t\}$.*

If H is a $\preceq_x$-preferred history over $P = \langle T, \leq_c, \leq_t \rangle$ that satisfies Q, then H satisfies Q over T.

In view of this result, observe also that preferences are different from goals. While goals *must* be satisfied, a preference may or may not be satisfied.

As mentioned earlier, we can limit ourselves to just preferences on formulas build from fluents, with no loss of expressive power.

Theorem 2 *Let $P = \langle T, \leq_c, \leq_t \rangle$ be a prioritised transition system over signature $\langle V, F, A \rangle$, and let H be a $\preceq_x$-preferred history satisfying query Q, for $x \in \{c, t\}$.*

Then, there is a translation $\mathcal{T}_f$ such that

$$\mathcal{T}_f(P) = \langle \mathcal{T}_f(T), \mathcal{T}_f(\leq_c), \mathcal{T}_f(\leq_t) \rangle,$$

where

1. *$\mathcal{T}_f(\leq_c), \mathcal{T}_f(\leq_t) \subseteq F \times F$; and*
2. *$\mathcal{T}_f(H)$ is a $\preceq_x$-preferred history satisfying Q.*

The translation is straightforward: For every action one introduces a new fluent that becomes true only as a result of that action; action names are then replaced by these new fluent names in the preferences, to yield an equivalent (with respect to the original language) prioritised transition system. The translation is well-behaved, being *monotonic* as well as requiring *polynomial* (in fact linear) time.

Concerning the computational complexity of our approach, since transition systems are defined in a rather abstract way, leaving the concrete specification details of its constituting elements open, we have to make some additional stipulations in order to derive meaningful decidability results.

First of all, we assume that for a given transition system $T = \langle S, \nu, R \rangle$ over signature $\sigma = \langle V, F, A \rangle$, the involved sets and functions are computable in polynomial time. Hence, computing whether a given fluent is true at a given state is feasible in polynomial time. Likewise, for a prioritised transition system $P = \langle T, \leq_c, \leq_t \rangle$, where T satisfies the above assumptions, the relations $\leq_c$ and $\leq_t$ are computable in polynomial time as well. We say that P is *regular* iff, in addition to these stipulations, T, $\leq_c$, and $\leq_t$ are finite. We then obtain the following result:

Theorem 3 *Let $x \in \{c, t\}$, and let n be a nonnegative integer.*

Deciding whether, for a given regular prioritised transition system $P = \langle T, \leq_c, \leq_t \rangle$, there exists a $\preceq_x$-preferred history satisfying a given query over language $\mathcal{Q}_n$, is Σ_2^P-complete.

A similar reasoning task for ordinary (un-prioritised) transition systems, for example in the action languages $\mathcal{A}$ (Liberatore 1997), $\mathcal{C}$ (Giunchiglia 2000), and $\mathcal{K}$ (Eiter *et al.* 2003c; 2003b)), is NP-complete. Hence, analogous to preference approaches in other declarative knowledge representation methods (like, e.g., for the methods due to Rintanen (1998), Zhang & Foo (1997), and Sakama & Inoue (2000)), we obtain an increase of complexity compared to the host formalism (providing the polynomial hierarchy does not collapse).

Concerning reasoning over preferred histories, we obtain a dual behaviour to the above decision problem.

Theorem 4 *Let $x \in \{c, t\}$, and let n be a nonnegative integer.*

Deciding whether, for a given regular prioritised transition system $P = \langle T, \leq_c, \leq_t \rangle$, a given query over language $\mathcal{Q}_n$ is satisfied in all $\preceq_x$-preferred histories satisfying a given set of axioms, is Π_2^P-complete.

Next, we consider issues dealing with the expressivity of our approach.

First, conditional preferences, or preferences that hold in a specific context, are simply dealt with. A conditional preference may be expressed quasi-formally by $\gamma \supset (\alpha_1 \leq_x \alpha_2)$, for $x \in \{c, t\}$, where the intent is that the preference

$\alpha_1 \leq_x \alpha_2$ is only considered for those histories that satisfy the condition γ. Thus, for example, one may wish to express the preference that

> *if the distance is far, prefer taking a bus to driving*,

or

> *if the monkey has a sweet tooth then he prefers to eat the chocolate before the bananas*.

A conditional preference of form $\gamma \supset (\alpha_1 \leq_x \alpha_2)$ can be expressed in our approach by $(\gamma \wedge \alpha_1) \leq_x (\gamma \wedge \alpha_2)$. This example also shows that preferences on formulas adds expressive power over an approach in which preferences apply only to action and fluent literals.

Second, it might appear that our specification of choice preference, as given by Definitions 7 and 9, is restrictive, in that for choice preference $\alpha_1 \leq_c \alpha_2$ we require that the (implicit) time stamps in α_1 and α_2 be the same across fluents in α_1 and in α_2. Thus, for example, if we have $f \wedge \neg h \leq_c f \wedge g$, our definition of satisfaction for choice preferences requires that f and $\neg h$ occur at the same time point, and similarly for f and g. However, it seems not unreasonable that one may want to prefer a history in which f and g are true, but not necessarily at the same time point.

This could be carried out by a suitable reformulation of the approach. However, the present formalism is already sufficiently general to allow this alternative to be encoded (in much the same fashion that action preferences can be encoded away as in Theorem 2), provided we give an appropriate alternative to Definition 7. To see this, assume first that a fluent name appearing in a choice preference is to be informally interpreted as if it were existentially quantified; to distinguish it from $\leq_c$, call this new choice relation $\leq_c^\exists$. Thus, in our example, now $f \wedge \neg h \leq_c^\exists f \wedge g$, a history in which there is a time point at which f is true and in which there is a time point at which g is true, is to be preferred, all other things being equal, to a history in which f is true at some point and $\neg h$ is true at some point. More formally, we replace Definition 7 by the following.

Definition 15 *Let T be a transition system, as given in Definition 5, and let H be a history of T. Define $H \Vdash_T^n \alpha$ as follows, where α is a Boolean combination of fluent names.*[3]

1. *$H \Vdash_T^n f$, for $f \in F$, iff $H \models_T^n f : i$, for some i where $0 \leq i \leq n$;*
2. *$H \Vdash_T^n \alpha_1 \wedge \alpha_2$ iff $H \Vdash_T^n \alpha_1$ and $H \Vdash_T^n \alpha_2$; and*
3. *$H \Vdash_T^n \neg\alpha$ iff $H \nVdash_T^n \alpha$.*

Thus, $H \Vdash f \wedge \neg h$ just if there are time stamps i and j such that $H \models f : i \wedge \neg h : j$.

We next transform a prioritised transition system to one that has this existential import for choice preferences, as follows:

1. Assume that choice preferences are in negation normal form, so that negations are only directly in front of fluent names.
2. For each unnegated (negated) fluent f occurring in an expression in $\leq_c^\exists$, introduce a new fluent f' ($f'_\neg$) that becomes true as a direct effect of f ($\neg f$) becoming true and subsequently stays true. Let α' be the Boolean expression resulting from so labelling the fluent names in α.
3. A straightforward argument establishes that if T is a prioritised transition system, H a history of T, and α a Boolean combination of fluent names, then
$$H \models_T^n \alpha' \text{ iff } H \Vdash_T^n \alpha. \qquad (1)$$
That is, for $f \in F$, we have that
$$\begin{array}{llll} H \Vdash_T^n f & \text{iff} & H \models_T^n f : i & \text{for some } 0 \leq i \leq n \\ & \text{iff} & H \models_T^n f' : n & \end{array}$$
and
$$\begin{array}{llll} H \Vdash_T^n \neg f & \text{iff} & H \models_T^n \neg f : i & \text{for some } 0 \leq i \leq n \\ & \text{iff} & H \models_T^n f'_\neg : n & \end{array}$$
from which (1) obtains by a straightforward induction.
4. Last, we can define
$$\Delta_P^\exists(H, H') = \{\alpha \in \mathsf{dom}(\leq_c^\exists) \mid H \Vdash \alpha \text{ and } H' \nVdash \alpha\}.$$
Given (1), we are now justified in adopting Definition 9 (adapted for $\Delta_P^\exists(H, H')$) to complete this alternative definition of choice preference.

Lastly, mention should be made of how to combine the preferences given by $\leq_c$ and $\leq_t$. First, these orderings cannot be directly combined, since they are fundamentally different in nature: a history is $\leq_c$-preferred, roughly, if it satisfies the $\leq_c$-greatest formulas compared to other histories; a history is $\leq_t$-preferred, roughly, if it violates fewest (in the sense of $\subseteq$) of the $\leq_t$ preferences. However, the orderings $\leq_c$ and $\leq_t$ generate in turn two preference orderings on histories, and *these* preference orderings on histories may be combined. However, this is the general problem of combining preference orderings, and as such is independent of the problem of preferences in reasoning about action; cf. Wellman & Doyle (1991) for a comprehensive discussion.

Conclusion

We have addressed the notion of preference in planning and reasoning about action and causality. Preferences are “soft” constraints, or desirable (but not required) outcomes that one would like to achieve, all other things being equal. A classification of domain-specific preference types is given, constituting action vs. fluent preferences and choice vs. temporal preferences. The former distinction turns out to not be meaningful (in that preferences can be expressed entirely in terms of fluents). However, we allow preferences to hold over arbitrary formulas built from action and fluent names. Further, the framework allows the expression of conditional preferences, or preferences holding in a given context. as well as absolute preferences, expressing a general desirability that a formula hold in a history. The preference orderings induce preference ordering on the set of plans, the maximal elements of which yield the preferred plans. The approach

[3] Action names are either dispensed with via Theorem 2, or else are handled in the same way as fluent names.

is formulated within the framework of action languages, although our results are applicable to general planning formalisms. We argue that the approach is general and flexible.

The approach relies on generating plans and selecting the most preferred. As such, the approach is readily adaptable to an anytime algorithm, in which one may select the currently-best plan(s), but with the hope of a more-preferred plan being generated. The most obvious topic for future work is to directly generate a preferred plan (rather than selecting from candidate plans); however, this appears to be a significantly difficult problem and at present we have no insights as to its solution. The present approach is currently being implemented as a back end to the action language $\mathcal{C}$.

Acknowledgements

The authors would like to thank Jérôme Lang for providing valuable comments which helped improving the paper. The first author was partially supported by a Canadian NSERC Discovery Grant. The second author was partially supported by the German Science Foundation (DFG) under grant SCHA 550/6, TP C. The collaboration of the second and third author was partially funded by the Information Society Technologies programme of the European Commission, Future and Emerging Technologies under the IST-2001-37004 WASP project. The third author was also partially supported by the IST-2001-33123 CologNeT Network of Excellence.

References

Bacchus, F., and Kabanza, F. 2000. Using temporal logics to express search control knowledge for planning. *Artificial Intelligence* 116(1-2):123–191.

Doherty, P., and Kvarnstrom, J. 1999. TALplanner: An empirical investigation of a temporal logic-based forward chaining planner. In *TIME*, 47–54.

Eiter, T.; Faber, W.; Leone, N.; Pfeifer, G.; and Polleres, A. 2003a. Answer set planning under action costs. *Journal of Artificial Intelligence Research* 19:25–71.

Eiter, T.; Faber, W.; Leone, N.; Pfeifer, G.; and Polleres, A. 2003b. A logic programming approach to knowledge-state planning, II: The $\texttt{dlv}^{\mathcal{K}}$ system. *Artificial Intelligence* 144(1-2):157–211.

Eiter, T.; Faber, W.; Leone, N.; Pfeifer, G.; and Polleres, A. 2003c. A logic programming approach to knowledge-state planning: Semantics and complexity. *ACM Transactions on Computational Logic*. To appear.

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

Geffner, H., and Pearl, J. 1992. Conditional entailment: Bridging two approaches to default reasoning. *Artificial Intelligence* 53(2-3):209–244.

Gelfond, M., and Lifschitz, V. 1998. Action languages. *Electronic Transactions on AI* 3. Available at http://www.ep.liu.se/rs/cis/1998/016/.

Gelfond, M., and Son, T. C. 1998. Reasoning with prioritized defaults. In Dix, J.; Pereira, L. M.; and Przymusinski, T. C., eds., *Proceedings of the 3th International Workshop on Logic Programming and Knowledge Representation (LPKR-97),* volume 1471 of *LNAI*, 164–223. Berlin: Springer.

Giunchiglia, E., and Lifschitz, V. 1998. An action language based on causal explanation: Preliminary report. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (*AAAI'98*), 623–630.

Giunchiglia, E.; Lee, J.; Lifschitz, V.; McCain, N.; and Turner, H. 2004. Nonmonotonic causal theories. *Artificial Intelligence* 153(1-2):49–104.

Giunchiglia, E. 2000. Planning as satisfiability with expressive action languages: Concurrency, constraints and nondeterminism. In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning* (*KR'00*), 657–666.

Haddawy, P., and Hanks, S. 1992. Representations for decision-theoretic planning: Utility functions for deadline goals. In *Proceedings of the Third International Conference on the Principles of Knowledge Representation and Reasoning*, 71–82.

Levesque, H.; Pirri, F.; and Reiter, R. 1998. Foundations for the situation calculus. *Linköping Electronic Articles in Computer and Information Science* 3(18).

Liberatore, P. 1997. The complexity of the language $\mathcal{A}$. *Electronic Transactions on Artificial Intelligence* 1(1–3):13–38. Available at `http://www.ep.liu.se/ej/etai/1997/002/`.

Myers, K., and Lee, T. 1999. Generating qualitatively different plans through metatheoretic biases. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, 570–576.

Nau, D.; Cao, Y.; Lotem, A.; and Muoz-Avila, H. 1999. Shop: Simple hierarchical ordered planner. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 968–973.

Niemelä, I., and Simons, P. 1997. Smodels: An implementation of the stable model and well-founded semantics for normal logic programs. In Dix, J.; Furbach, U.; and Nerode, A., eds., *Proceedings of the Fourth International Conference on Logic Programing and Nonmonotonic Reasoning*, 420–429. Springer-Verlag.

Rintanen, J. 1998. Lexicographic priorities in default logic. *Artificial Intelligence* 106:221–265.

Sakama, C., and Inoue, K. 2000. Prioritized logic programming and its application to commonsense reasoning. *Artificial Intelligence* 123(1-2):185–222.

Son, T., and Pontelli, E. 2002. Reasoning about actions in prioritized default theory. In Flesca, S.; Greco, S.; Leone, N.; and G.Ianni., eds., *Proceedings of the Eighth European Conference on Logics in Artificial Intelligence (JELIA'02)*, volume 2424 of *Lecture Notes in Computer Science*, 369–381. Springer Verlag.

Son, T., and Pontelli, E. 2004. Planning with preferences in logic programming. In Lifschitz, V., and Niemelä, I., eds., *Proceedings of the Seventh International Conference on Logic Programming and Nonmonotonic Reasoning* (*LP-NMR'04*), volume 2923 of *Lecture Notes in Artificial Intelligence*, 247–260. Springer Verlag.

Son, T.; Baral, C.; Nam, T.; and McIlraith, S. 2002. Domain-dependent knowledge in answer set planning. `http://arxiv.org/abs/cs.AI/0207023`.

Thielscher, M. 1999. From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem. *Artificial Intelligence* 111(1–2):277–299.

Wellman, M., and Doyle, J. 1991. Preferential semantics for goals. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, 698–703.

Zhang, Y., and Foo, N. 1997. Answer sets for prioritized logic programs. In Maluszynski, J., ed., *Proceedings of the International Symposium on Logic Programming (ILPS-97)*, 69–84. MIT Press.