How to Plan When Being Deliberately Misled

Maurice Pagnucco and **David Rajaratnam** and **Michael Thielscher**
The University of New South Wales, Sydney

Hannes Strass
University of Leipzig, Germany

Abstract

Reasoning agents are often faced with the need to robustly deal with erroneous information. When a robot given the task of returning with the red cup from the kitchen table arrives in the kitchen to find no red cup but instead notices a blue cup and a red plate on the table, what should it do? The best course of action is to attempt to salvage the situation by relying on its preferences to return with one of the objects available. We provide a solution to this problem using the Situation Calculus extended with a notion of belief. We then provide an efficient practical implementation by mapping this formalism into default rules for which we have an implemented solver.

Introduction and Motivating Example

The Robocup@Home (see `www.robocupathome.org`) competition is an international robotics initiative to foster research into domestic robots; robots that can assist humans within a wide range of environments from fetching items within the home through to shopping for groceries. This initiative aims to focus the research in robotics onto developing relevant techniques and technologies.

A necessary requirement for the development of domestic robots is the need to perform tasks in response to user commands. This requires the ability to deal with possibly erroneous information being provided by the human operator (either deliberately or accidentally). For example, the robot may be instructed to fetch an item from another room that is no longer in the place specified by the operator. Furthermore, as this example highlights, not only may the information be incorrect but the robot may not realise this fact until partway through the task.

In keeping with its role as a domestic helper, the robot would be expected to behave in a manner consistent with a human's responses in a similar situation – with *robustness* and *flexibility*. Possible responses could vary from simply reporting the failure back to the operator, through to trying to fulfil a task that most closely satisfies the specified goal, based on background commonsense knowledge.

The requirement to operate under erroneous information is demonstrated in the "General Purpose Service Robot" challenge of the Robocup@Home 2010 Competition.[1] The following scenario is based on an example from this challenge.

[1] `http://www.robocupathome.org/documents/rulebook2010_FINAL_VERSION.pdf`

Scenario 1 *The robot is initially located in the living room of the home. The home has a kitchen with a table in the middle. The robot is told to fetch the red cup from the kitchen table. However, there is no red cup on the kitchen table and the robot only discovers this fact once it arrives in the kitchen and looks for the cup on the table.*

We highlight two separate cases. In the base case there is only a blue cup on the table. In the extended case there is a blue cup and a red plate on the table.

While it is impossible to know the precise intentions of the human operator, the robot can nevertheless apply commonsense knowledge to exhibit natural behaviours. In the first case, faced with no alternatives, it might simply fetch the blue cup. In the second case, the robot might assume that the user is more interested in the type of object than its colour and so would prefer the blue cup over the red plate.

The rest of the paper proceeds as follows. We first provide the technical background to understand the paper. Then we describe a formal specification of the above scenarios in terms of the traditional Situation Calculus. To deal with the problem of erroneous information, we employ a version of the Situation Calculus extended with a notion of belief (Shapiro et al. 2011). Preferences derived from the problem statement and commonsense knowledge are used there to determine alternative courses of action. Next, we present another solution, that is based on prioritised default logic for which we have an implementation. Finally, we show that the two solutions yield the same results, discuss our findings in a broader context and conclude.

Technical Preliminaries

Situation Calculus

The Situation Calculus provides a formal language based on that of classical first-order logic in which to describe dynamic domains (McCarthy 1963; Reiter 2001). Three types of terms are distinguished: *situations* representing a snapshot of the world; *fluents* denoting domain properties that may change as a result of actions; and *actions* that can be performed by the reasoner. We use the predicate $\mathit{Holds}(f, s)$ to specify that a fluent f holds at a particular situation. As a matter of convention a short form is adopted such that for any n-ary fluent $f(x_1, \ldots, x_n)$, writing $f(x_1, \ldots, x_n, s)$ is a short form for $\mathit{Holds}(f(x_1, \ldots, x_n), s)$. A special function $do(a, s)$ represents the situation that results from performing action a at situation s. S_0 denotes the initial situation where no actions have taken place. For each action we need to specify preconditions $Poss(a, s)$ specifying the conditions under which action a is possible in situation s and effect axioms that specify how the value of a fluent changes when an action is performed.[2] For a more comprehensive technical formulation of what is required of a Situation Calculus theory, the reader is referred to (Reiter 2001).

Iterated Belief Revision in the Situation Calculus

A request to an agent to carry out a goal affects its beliefs. For instance, when the agent is asked to collect the red cup from the kitchen table, it is reasonable for the agent to believe that there is in fact a red cup located on the kitchen table. We therefore adopt an extension to the Situation Calculus capable of representing beliefs. Several accounts exist (Shapiro et al. 2011; Demolombe and Pozos-Parra 2006) however we use that of (Shapiro et al. 2011). It is based on the ideas of Moore and extended by (Cohen and Levesque 1990) who introduced knowledge into the Situation Calculus by reifying the accessibility relation in modal semantics for knowledge. These accounts distinguish two types of actions: *physical actions* which alter the world (and hence fluent values) when performed; and, *sensing actions* that are associated with a sensor possessed by the agent and determine the value of a fluent (e.g., a vision system would be used to determine whether a red cup is on a table). Sensing actions are also referred to as *knowledge producing* actions since they inform the reasoner about the value of a fluent but do not alter the state of the world.

(Scherl and Levesque 2003) introduce the relation $B(s', s)$ denoting that if the agent were in situation s, it considers s' to be possible and this is adopted by (Shapiro et al. 2011).[3] The successor state axiom for the B relation is given in the table below as Axiom (1) and states that s'' is possible at the situation resulting from performing action a at situation s whenever the sensing action associated with a agrees on its value at s and s'. $SF(a, s)$ is a predicate that is true whenever the sensing action a returns the sensing value 1 at s and was introduced by (Levesque 1996). The innovation of (Shapiro et al. 2011) is to associate a plausibility with situations. Plausibility values are introduced for initial situations and these values remain the same for all successor situations as expressed in Axiom (2) below. This is critical for preserving introspection properties for belief. The plausibility values themselves are not important, only the ordering over situations that they induce. Axioms (3) and (4) define the situations s' that are most plausible and most plausible situations that are possible (i.e., B-related) at s respectively. In Axiom (5) we define sentence ϕ to be believed in situation s whenever it is true at all the most plausible situations that are possible at s. Finally, Axiom (6) specifies that any situations B-related to an initial situation are also initial situations. The distinguished predicate $Init(s)$ indicates that s is an initial situation.

1. $B(s'', do(a, s)) \equiv \exists s'[B(s', s) \wedge s'' = do(a, s') \wedge SF(a, s') \equiv SF(a, s))]$
2. $pl(do(a, s)) = pl(s)$
3. $MP(s', s) \stackrel{\text{def}}{=} \forall s''.B(s'', s) \supset pl(s') \leq pl(s'')$
4. $MPB(s', s) \stackrel{\text{def}}{=} B(s', s) \wedge MP(s', s)$
5. $Bel(\phi, s) \stackrel{\text{def}}{=} \forall s'.MPB(s', s) \supset \phi[s']$
6. $Init(s) \wedge B(s', s) \supset Init(s')$

[2] In fact, we compile effect axioms into *successor state axioms* (Reiter 2001).

[3] Note the order of the arguments as it differs from that commonly used in modal semantics of knowledge.

We now turn to a formalisation of our scenario by first considering the part of the problem that can be specified by the basic situation calculus before turning to our approach which deals with achieving goals and requires the extensions described in this section.

Formalisation

In the first place the scenario has a number of inherently bivalent properties: the robot is in the living room or kitchen, an object is a cup or a plate, and an object can be red or blue. For simplicity we adopt only one of each pair, with the intuition that the negation of the given property implies that its bivalent pair must hold. For example, if an object is not a cup then it must be a plate.

Objects The Robocup@Home challenge deals with a fixed set of household objects that are determined at the start of the competition. This allows the teams time to train their vision systems to be able to detect and distinguish between these objects. This motivates the following formalisation.

We assume a fixed set I of individual objects, to which we apply a unique names assumption. Intuitively, these names identify the items that the robot is trained to recognise. In our example scenario, these are two cups, one red and one blue, and a red plate: $I = \{C_R, C_B, P_R\}$. We introduce the fluent $Same(x, y)$ to express that two names refer to the same real object, and allow a set of additional names $N = \{O_1, \dots, O_n\}$ ensuring that these names refer only to existing objects in the domain. In our example, we write

$$Same(O_i, C_R, s) \vee Same(O_i, C_B, s) \vee Same(O_i, P_R, s)$$

for $1 \leq i \leq n$. $Same$ is axiomatised by enforcing that identical objects agree on all fluent properties F of the domain:

$$Same(x, y, s) \supset (F(x, s) \equiv F(y, s))$$

Primitive fluents The primitive fluents in our domain and their meanings are as follows. *InKitchen*: the robot is in the kitchen, $Holding(o)$: the robot is holding an object, $OnTable(o)$: the object is on the kitchen table, $Cup(o)$: the object is a cup, $Red(o)$: the object is red.

Primitive actions *SwitchRoom*: if the robot is in the living room then it moves to the kitchen and vice-versa; $PickUp(o)$: pick up an object from the kitchen table.

Sensing In the Robocup@Home challenge the robot is trained to recognise the pre-determined set of objects I. The main sensing task is then to detect whether or not these specific objects are located on the kitchen table. This is encapsulated by the sensing action $SenseOT(o)$ that senses if object $o \in I$ is on the table. The $SF(a, s)$ predicate, introduced in the previous section, is used to axiomatise the act of sensing:

$$SF(PickUp(o), s) \equiv true$$
$$SF(SwitchRoom, s) \equiv true$$
$$InKitchen(s) \supset (SF(SenseOT(o), s) \equiv OnTable(o, s))$$

Initial state In the initial state the robot is in the living room (i.e., not in the kitchen) and is not holding anything: $\neg \textit{InKitchen}(S_0) \wedge (\forall x)(\neg \textit{Holding}(x, S_0))$.

Informing the robot Informing the robot about the operator's belief in the state of the world is formalised outside of the underlying action calculus at the meta-level and is subsequently compiled into the initial state axioms within the appropriate Situation Calculus logic.

Let f be a fluent literal. Then $\textit{Told}(f, S_0)$, which we abbreviate as $\textit{Told}(f)$, represents the act of the operator informing the robot about the operator's understanding of the initial state of the world. In our example this would be $\textit{Told}(\textit{Cup}(O_1))$, $\textit{Told}(\textit{Red}(O_1))$, $\textit{Told}(\textit{OnTable}(O_1))$. A set of operator commands T is *consistent* provided there is no fluent f such that $\textit{Told}(f) \in T$ and $\textit{Told}(\neg f) \in T$.

Precondition axioms The robot can always switch locations: $\textit{Poss}(\textit{SwitchRoom}, s) \equiv true$; the robot can only pick up an item when it is not already holding an object and the item in question is on the kitchen table: $\textit{Poss}(\textit{PickUp}(o), s) \equiv (\forall x)(\neg \textit{Holding}(x, s)) \wedge \textit{InKitchen}(s) \wedge \textit{OnTable}(o, s)$.

Successor state axioms The robot will be in the kitchen as a result of switching rooms if it wasn't already in the kitchen: $\textit{InKitchen}(do(a, s)) \equiv (\neg \textit{InKitchen}(s) \wedge a = \textit{SwitchRoom}) \vee (\textit{InKitchen}(s) \wedge a \neq \textit{SwitchRoom})$; the robot will be holding an object if it picks up the object or was already holding the object: $\textit{Holding}(o, do(a, s)) \equiv a = \textit{PickUp}(o) \vee \textit{Holding}(o, s)$; an item will be on the table only if it was previously on the table and has not been picked up: $\textit{OnTable}(o, do(a, s)) \equiv \textit{OnTable}(o, s) \wedge a \neq \textit{PickUp}(o)$; object type is persistent: $\textit{Cup}(o, do(a, s)) \equiv \textit{Cup}(o, s)$; colour is persistent: $\textit{Red}(o, do(a, s)) \equiv \textit{Red}(o, s)$.

Approach

Our approach can be succinctly summarised as follows. Every planning problem (request to achieve a goal) is considered a new reasoning problem. Two types of statements are used to ascribe initial beliefs and a goal to achieve. They are interpreted at the meta-level and are not part of the object language. The first meta-level statement is $\textit{Told}(f(\vec{x}))$ as described above to establish the agent's initial beliefs. The second is a request of the form $\textit{Goal}(\exists s.\phi(s))$ where $\phi(s)$ is a sentence expressing the goal to be achieved. For example, the request $\textit{Told}(\textit{Cup}(O_1))$, $\textit{Told}(\textit{Red}(O_1))$, $\textit{Told}(\textit{OnTable}(O_1))$, $\textit{Goal}(\exists s.\textit{Holding}(O_1, s))$ asks the agent to collect a red cup from the table. This results in the specification of a reasoning about action problem in the Situation Calculus extended with beliefs. In particular, the request specifies what should be believed in the initial situation S_0 and as such partially restricts the plausibility relation $pl()$. However, our beliefs may be mistaken—there is no red cup on the table—and as a result we need to formulate an alternative course of action to get the best out of the situation at hand. Which alternative course of action to take is determined by preferences that are specified using a meta-level preference relation $<_C$. These preferences place further restrictions on the plausibility of situations $pl()$.

Preferences reflect the robot's commonsense knowledge. In our scenario, for example, the robot may prefer to fetch an object that is of the same type as requested but of a different colour, and most of all prefer to find an object in the room to which it was sent.

$$\textit{OnTable} <_C \textit{Cup} <_C \textit{Red} \tag{1}$$

It is of course possible to conceive of a scenario with several such individual preference relations, which is why our approach assumes a given partial order among fluents. Also it is possible to conceive of a scenario in which the above preference for, say, non-red cups in the kitchen over red non-cups elsewhere is reversed. The operator may be a child building a colour collage and therefore assign greater importance to the colour of the object than its type.

In reality, determining the best set of preferences would be a complex task requiring the robot to combine subtleties of natural language processing with specific knowledge about the operator and the task the operator is trying to achieve. Such considerations are beyond the scope of this paper, and so we just presuppose a given *commonsense preference ordering*, represented by a partial order among fluent names.

Next, we directly compile the $\textit{Told}()$ statements plus an ordering like (1) into a plausibility ordering over all the initial situations. Here, the initial situations encode all possible hypotheses of what the operator might have meant by their commands. The commonsense preference is then used to rank these hypotheses according to their plausibility.

In order to relate this preference ordering to the $\textit{Told}()$ statements we introduce the notation $\langle\cdot\rangle$ to extract the fluent name from a fluent literal (e.g., $\langle \neg \textit{Cup}(O_1) \rangle = \textit{Cup}$).

Definition 1 *Let Σ be a Situation Calculus theory, B the axioms for iterated belief revision in the Situation Calculus, I be the set of domain objects, $N = \{O_1, \ldots, O_n\}$ be a set of additional names, T be a set of consistent operator commands and $<_C$ be a commonsense preference ordering. Then $(\Sigma \cup B, T, <_C)$ is a Situation Calculus theory extended with belief and commonsense preferences such that:*

1. *The initial situations are created by the axioms*

$$(\exists s_\sigma)\left(\textit{Init}(s_\sigma) \wedge \bigwedge_{X \in I} \bigwedge_{i \in \sigma(X)} \textit{Same}(O_i, X, s_\sigma)\right) \tag{2}$$

for all mappings $\sigma : I \to 2^{\{1,\ldots,n\}}$ such that

- *for all $i \in \{1, \ldots, n\}$ exists an $x \in I$ with $i \in \sigma(x)$;*
- *for all $x, y \in I$, $x \neq y$ implies $\sigma(x) \cap \sigma(y) = \emptyset$.*

2. *For every pair $\textit{Init}(s_{\sigma_1})$ and $\textit{Init}(s_{\sigma_2})$ from above:*

$$pl(s_{\sigma_1}) < pl(s_{\sigma_2})$$

iff both

(a) *there is some $\textit{Told}(f(\vec{x})) \in T$ such that $\Sigma \cup \{2\} \models f(\vec{x}, s_{\sigma_1})$ and $\Sigma \cup \{2\} \models \neg f(\vec{x}, s_{\sigma_2})$; and,*

(b) *for every $\textit{Told}(f_1(\vec{x}_1)) \in T$ such that*

$$\Sigma \cup \{2\} \models \neg f_1(\vec{x}_1, s_{\sigma_1}) \text{ and } \Sigma \cup \{2\} \models f_1(\vec{x}_1, s_{\sigma_2})$$

there is a $\textit{Told}(f_2(\vec{x}_2)) \in T$ such that $\langle f_2 \rangle <_C \langle f_1 \rangle$,

$$\Sigma \cup \{2\} \models f_2(\vec{x}_2, s_{\sigma_1}) \text{ and } \Sigma \cup \{2\} \models \neg f_2(\vec{x}_2, s_{\sigma_2})$$

Part 1 creates all the initial situations. Intuitively, the mapping σ says which names are assigned to which real object; so $\sigma_1(C_R) = \{O_1, O_3\}$ means that O_1 and O_3 are considered the same as C_R in situation s_{σ_1}. Naturally, we create only the mappings that constitute a valid way of assigning object names from N to real objects from I.

Part 2 restricts the plausibility relation over initial situations. One initial situation is preferred to another whenever (a) it assigns the value true to some told fluent and the other situation assigns the fluent false; and (b) for every told fluent that holds in the second but not the first situation, there is a *preferred* told fluent for which it is the other way round.

From this formalisation of the scenario, we can establish the fact that the robot will initially believe what it is told.

Proposition 1 *Let Σ be a Situation Calculus theory, B the axioms for iterated belief revision in the Situation Calculus, T be a set of operator commands such that $\Sigma \cup \{f(\vec{x}, S_0) : Told(f(\vec{x})) \in T\}$ is consistent, and $<_C$ be a commonsense preference ordering. Then $(\Sigma \cup B, T, <_C)$ is a Situation Calculus theory extended with belief and commonsense preferences such that for all $Told(f(\vec{x})) \in T$:*

$$(\Sigma \cup B, T, <_C) \models Bel(f(\vec{x}), S_0)$$

Proof (sketch): By construction from Axioms (1)-(5) and the plausibility ordering set up in Definition 1. □

Plan Execution This formalism allows the robot to change its beliefs about what it is told. In this paper we assume that the robot has determined a plan and begun its execution. We can therefore consider the robot's changing beliefs with regards to satisfying its goal of holding object O_1 by considering the situation[4]

$$do([SwitchRoom, SenseOT(C_R), SenseOT(C_B), SenseOT(P_R), PickUp(O_1)], S_0)$$

Initially the robot believes that the object O_1 refers to the red cup C_R. However when the robot arrives in the kitchen it finds that there is only a blue cup on the table. Consequently the robot changes its belief about O_1 so that it now refers to the blue cup C_B. This scenario is visualised by Figure 1 on the next page showing the possible situations based on the robot's beliefs and the plausibility relation.

In the extended example the robot arrives in the kitchen to find both a blue cup and red plate on the table. It therefore has a choice, which it resolves based on its preference for object type over colour (1), consequently modifying its belief about O_1 to again refer to the blue cup.

A Default Logic Approach

The Situation Calculus with beliefs provides an expressive formalism for tackling the problem of agents being deliberately misled and expected to use some basic commonsense reasoning under these circumstances. Next we address the problem of turning the theory into a practical implementation. To this end we adapt a recently developed extension of action logics with default reasoning (Baumann et al. 2010), which can be efficiently implemented using Answer Set Programming (Gelfond 2008). The idea is to treat potentially erroneous information as something that is considered true by default but can always be retracted should the agent make observations to the contrary. We extend the existing approach by *prioritised* defaults that allow us to provide our robot with preferences among different ways of remedying a situation in which it has been deliberately misled.

[4] $do([a_1, \ldots, a_n], s) \stackrel{\text{def}}{=} do(a_n, \ldots, do(a_2, do(a_1, s)) \ldots)$.

Supernormal Defaults To begin with, we instantiate the general framework of (Baumann et al. 2010) to the Situation Calculus and to a restricted form of default rules. Each operator command $Told([\neg]f(\vec{x}), s)$ is translated into a *supernormal* default of the form

$$\frac{: Holds(f(\vec{x}), s)}{Holds(f(\vec{x}), s)} \quad \text{or} \quad \frac{: \neg Holds(f(\vec{x}), s)}{\neg Holds(f(\vec{x}), s)}$$

With these rules the robot will believe, by default, everything it is told. For our running example we thus obtain these three defaults about the initial situation:

$$\delta_{Cup} = \frac{: Holds(Cup(O_1), S_0)}{Holds(Cup(O_1), S_0)} \quad \delta_{Red} = \frac{: Holds(Red(O_1), S_0)}{Holds(Red(O_1), S_0)}$$

$$\delta_{OnTable} = \frac{: Holds(OnTable(O_1), S_0)}{Holds(OnTable(O_1), S_0)}$$

A *Situation Calculus default theory* is a pair (Σ, Δ) where Σ is as above and Δ is a set of default rules.

Priorities In a *prioritised default theory* (Brewka 1994), the default rules are partially ordered by $\prec$, where $\delta_1 \prec \delta_2$ means that the application of default δ_1 is preferred over the application of δ_2. For our purpose, we can map a given commonsense preference ordering among fluent names directly into a partial ordering among the defaults from above. For example, with the ordering given by (1) we obtain $\delta_{OnTable} \prec \delta_{Cup} \prec \delta_{Red}$. A *prioritised Situation Calculus default theory* is a triple $(\Sigma, \Delta, \prec)$ where (Σ, Δ) is as above and $\prec$ is a partial ordering on Δ.

Extensions Reasoning with default theories is based on the concept of so-called *extensions*, which can be seen as a way of assuming as many defaults as possible without creating inconsistencies (Reiter 1980; Brewka 1994).

Definition 2 *Consider a prioritised Situation Calculus default theory $(\Sigma, \Delta, \prec)$. Let E be a set of formulas and define $E_0 := Th(\Sigma)$ and, for $i \geq 0$,*

$$E_{i+1} := Th(E_i \cup \{\gamma \mid \tfrac{: \gamma}{\gamma} \in \Delta, \neg\gamma \notin E\})$$

Then E is an extension *of $(\Sigma, \Delta, \prec)$ iff $E = \bigcup_{i \geq 0} E_i$.*

Let a partial ordering be defined as $E_1 \not\ll E_2$ iff both

(a) there is $\frac{: \gamma}{\gamma}$ in Δ such that $\gamma \in E_1$ but $\gamma \notin E_2$; and,

(b) for every $\frac{: \gamma_1}{\gamma_1}$ such that $\gamma_1 \notin E_1$ but $\gamma_1 \in E_2$ there is $\frac{: \gamma_2}{\gamma_2} \prec \frac{: \gamma_1}{\gamma_1}$ in Δ such that $\gamma_2 \in E_1$ but $\gamma_2 \notin E_2$.

Extension E is a preferred extension *of $(\Sigma, \Delta, \prec)$ iff there is no E' such that $E' \not\ll E$. Entailment $(\Sigma, \Delta, \prec) \approx\!\!| \; \phi$ is defined as ϕ being true in all preferred extensions.*

In our running example, when initially the robot has no information to the contrary it can consistently apply all defaults, resulting in a unique preferred extension that entails

$$Holds(Cup(O_1), S_0) \wedge Holds(Red(O_1), S_0) \wedge Holds(OnTable(O_1), S_0)$$

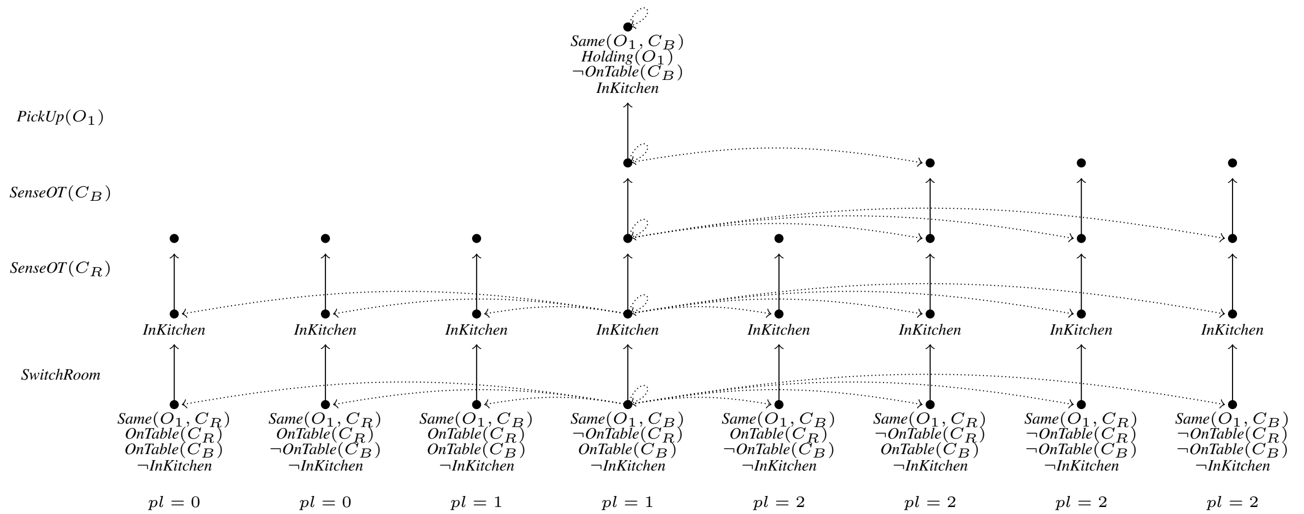

Figure 1: The robot is told to pick up the red cup from the table, but finds only a blue cup. For succinctness, details of the red plate and the status of the persistent fluents *Cup* and *Red* are omitted. Furthermore, only the accessibility relations (dotted lines) for the actual situation (fourth from the left) are shown. The transition of situations based on actions are indicated by the solid vertical lines. Values for the plausibility relation are assigned to the initial situations based on the preferences. The two initial situations in which the robot believes that it is going to pick up the red cup on the table are the most preferred ($pl = 0$). The next preferred are those in which the robot believes that it is going to pick up the blue cup on the table ($pl = 1$). Finally, the least preferred options are situations in which the robot believes that the item to pick up is not on the table ($pl = 2$).

Based on these default conclusions the Situation Calculus axioms entail the same plans for a given goal as those for the Situation Calculus extended with belief and commonsense preferences. But suppose that the robot enters the kitchen and observes what is indicated in Figure 1, that is,

$$\begin{array}{l} \mathit{Same}(O_1, C_R, S) \lor \mathit{Same}(O_1, C_B, S) \\ \mathit{Holds}(\mathit{Cup}(C_R), S) \land \mathit{Holds}(\mathit{Red}(C_R), S) \\ \mathit{Holds}(\mathit{Cup}(C_B), S) \land \neg\mathit{Holds}(\mathit{Red}(C_B), S) \\ \neg\mathit{Holds}(\mathit{OnTable}(C_R), S) \land \mathit{Holds}(\mathit{OnTable}(C_B), S) \end{array}$$

where S is the situation after *SwitchRoom* followed by $\mathit{SenseOT}(C_R)$ and $\mathit{SenseOT}(C_B)$. Disregarding priorities for now, there are two extensions, characterised by

$$\begin{array}{ll} \{\mathit{Same}(O_1, C_R, S), \neg\mathit{Holds}(\mathit{OnTable}(O_1), S)\} & \subseteq E_1 \\ \{\mathit{Same}(O_1, C_B, S), \neg\mathit{Holds}(\mathit{Red}(O_1), S)\} & \subseteq E_2 \end{array}$$

However, given the priorities from above, only E_2 is a preferred extension, triggering the robot to pick up the blue cup.

In the second case of the scenario, the robot further senses that there is also a red plate on the table. In this case there will be a third extension E_3 such that

$$\{\mathit{Same}(O_1, P_B, S), \neg\mathit{Holds}(\mathit{Cup}(O_1), S)\} \subseteq E_3$$

However, as with the first case, E_2 is still the only preferred extension and therefore the robot selects the blue cup.

Implementation Answer Set Programming (ASP) (Gelfond 2008) is well-suited for efficiently implementing nonmonotonic reasoning formalisms like the one developed in the previous section. This is because extended logic programs can be seen as special kinds of default theories (Gelfond and Lifschitz 1991). To put the default logic approach of this paper into practice, we make use of this correspondence and transform a given prioritised Situation Calculus default theory $(\Sigma, \Delta, \prec)$ into an answer set program $P_{\Sigma,\Delta,\prec}$, provided the first-order theory Σ is sufficiently restricted to permit this transformation. Using an off-the-shelf ASP solver,[5] we can then figure out whether a formula is entailed by the default theory via querying the answer set program. There is insufficient space to treat the exact transformation in detail, so we only give the underlying ideas here.

Step 1. We transform the *prioritised* Situation Calculus default theory $(\Sigma, \Delta, \prec)$ into a Situation Calculus default theory $(\Sigma^{\prec}, \Delta^{\prec})$ where the preferences have been encoded at the object-level (Delgrande and Schaub 2000). This is done by explicitly keeping track of default δ's meta-level applicability $\mathsf{ok}(\delta)$ and whether it was applied $(\mathsf{ap}(\delta))$ or blocked $(\mathsf{bl}(\delta))$. For example, δ_{Cup} and δ_{Red} are transformed into

$$\frac{\mathsf{ok}(\delta_{\mathit{Cup}}) : \mathit{Holds}(\mathit{Cup}(O_1), S_0)}{\mathit{Holds}(\mathit{Cup}(O_1), S_0) \land \mathsf{ap}(\delta_{\mathit{Cup}})} \qquad \frac{\mathsf{ok}(\delta_{\mathit{Cup}}) \land \neg\mathit{Holds}(\mathit{Cup}(O_1), S_0) :}{\mathsf{bl}(\delta_{\mathit{Cup}})}$$

$$\frac{\mathsf{ok}(\delta_{\mathit{Red}}) : \mathit{Holds}(\mathit{Red}(O_1), S_0)}{\mathit{Holds}(\mathit{Red}(O_1), S_0) \land \mathsf{ap}(\delta_{\mathit{Red}})} \qquad \frac{\mathsf{ok}(\delta_{\mathit{Red}}) \land \neg\mathit{Holds}(\mathit{Red}(O_1), S_0) :}{\mathsf{bl}(\delta_{\mathit{Red}})}$$

The preference between the defaults is enforced by statements like $(\mathsf{ap}(\delta_{\mathit{Cup}}) \lor \mathsf{bl}(\delta_{\mathit{Cup}})) \supset \mathsf{ok}(\delta_{\mathit{Red}})$, effectively saying that δ_{Red} can only be applied once it is clear whether the more preferred default δ_{Cup} has been "processed".

Step 2. We instantiate the defaults from $\Delta^{\prec}$ and the axioms from $\Sigma^{\prec}$ for the given Situation Calculus signature. This yields a propositional default theory.

Step 3. We rewrite the ground instantiation of $\Sigma^{\prec}$ into a set $P_{\Sigma^{\prec}}$ of extended logic program rules.

[5] Available at `http://potassco.sourceforge.net`.

Step 4. We transform $\Delta^{\prec}$ into a set of logic program rules. A default of the form $\frac{p:q}{r_1 \wedge r_2}$ becomes $\mathtt{r_i} \leftarrow \mathtt{p},\ \mathtt{not}\ -\mathtt{q}$ for $i = 1, 2$; a rule $\frac{p \wedge q:}{r}$ is turned into $\mathtt{r} \leftarrow \mathtt{p},\ \mathtt{q}$. Here `not` is the usual negation as failure of normal logic programs; $-\mathtt{q}$ is a new predicate symbol standing for the (classical) negation of `q` (Gelfond and Lifschitz 1991). The resulting rules together with $P_{\Sigma^{\prec}}$ now form the corresponding answer set program $P_{\Sigma,\Delta,\prec}$ of the initial prioritised Situation Calculus default theory $(\Sigma, \Delta, \prec)$.

Equivalence of the Two Approaches

We are now in a position to state the central result of this paper, which says that our prioritised Situation Calculus default theories are suitable approximations of the Situation Calculus extended with belief and commonsense preferences. Unfortunately, lack of space prevents us from giving a rigorously formal account. Generally speaking, the latter is more expressive for two reasons. First, it allows to infer meta-statements about beliefs, as in $Bel(Bel(\mathit{Red}(O_1), S_0), do(\mathit{SwitchRoom}, S_0))$. Second, all possible situations are ranked according to $pl()$, thus allowing to draw conclusions about their relative ordering, whereas in prioritised default logics the non-preferred extensions are not considered for entailment. However, neither of these two features is relevant for the problem at hand, and we can prove the following.

Theorem 1 *Let Σ be a Situation Calculus theory, B the axioms for iterated belief revision in the Situation Calculus, T a set of consistent operator commands, $<_C$ a commonsense preference ordering, $\Delta, \prec$ a set of default rules and an ordering as explained above, $a_1, \ldots, a_n$ a sequence of actions, and $SF_n := \{[\neg]SF(a_1, S_0), \ldots, [\neg]SF(a_n, do([a_1, \ldots, a_{n-1}], S_0))\}$ a set of literals describing a particular sequence of sensing results. Then*

$$(\Sigma \cup B \cup SF_n, T, <_C) \models Bel(\phi, do([a_1, \ldots, a_n], S_0))$$
$$\text{iff } (\Sigma \cup SF_n, \Delta, \prec) \approx\!\!\!\mid \mathit{Holds}(\phi, do([a_1, \ldots, a_n], S_0))$$

Proof (sketch): By induction on the number of actions n. If $n = 0$, by Proposition 1 the robot believes all operator commands; in a similar way it can be shown that there is a unique preferred extension which entails the exact same statements about S_0 that are true in all most plausible initial situations. For the induction step, if a_{n+1} is a physical action the claim follows from the fact that both axiomatisations share the same basic action theory. If a_{n+1} is a sensing action, then any possible situation in $do([a_1, \ldots, a_n], S_0)$ that contradicts $[\neg]SF(a_{n+1}, do([a_1, \ldots, a_n, a_{n+1}], S_0)$ is no longer possible in $do([a_1, \ldots, a_n, a_{n+1}], S_0)$; likewise, any extension of $(\Sigma \cup SF_n, \Delta, \prec)$ that contradicts this sensing literal is no longer an extension of $(\Sigma \cup SF_{n+1}, \Delta, \prec)$. The claim follows from the structural equivalence of the construction of the plausibility ordering in Definition 1 (Item 2) and the construction of preferred extensions in Definition 2. □

Conclusions

We developed an account of how a reasoner faced with an unachievable goal should nevertheless do its best to salvage the situation by relying on its preferences. We formalised our solution using an extension of the Situation Calculus to handle beliefs. This extension, however, has never been implemented and the naive implementation is computationally infeasible. As a result, the other significant contribution of this paper was a mapping into default rules that allow for an efficient implementation. As a motivating example we used a scenario from the RoboCup@Home rulebook.

An alternative approach would be to consider goal revision (Shapiro, Lespérance, and Levesque 2005). Proposals like this one, however, modify goals at the explicit request of an agent and do not consider that the goals themselves may be unachievable. In our approach, the goal cannot be achieved and we argue that this is more accurately dealt with by reasoning about the robot's beliefs/expectations.

Finally, it should be noted that the formalism proposed here is not simply intended as a theoretical exercise. The authors are actively engaged in translating these results into high-level reasoning modules for the ROS (Quigley et al. 2009) robotics platform.

References

Baumann, R.; Brewka, G.; Strass, H.; Thielscher, M.; and Zaslawski, V. 2010. State defaults and ramifications in the unifying action calculus. In *KR-10*, 435–444.

Brewka, G. 1994. Adding priorities and specificity to default logic. In *JELIA-94*, volume 838 of *LNAI*, 247–260.

Cohen, P., and Levesque, H. 1990. Rational interaction as the basis for communication. In Cohen, P.; Morgan, J.; and Pollack, M., eds., *Intentions in Communication*. 221–256.

Delgrande, J., and Schaub, T. 2000. Expressing Preferences in Default Logic. *AIJ* 123(1–2):41–87.

Demolombe, R., and Pozos-Parra, M. P. 2006. Belief revision in the situation calculus without plausibility levels. In *ISMIS*, volume LNCS 4203, 504–513.

Gelfond, M., and Lifschitz, V. 1991. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing* 9:365–385.

Gelfond, M. 2008. Answer Sets. In *Handbook of Knowledge Representation*. 285–316.

Levesque, H. 1996. What is planning in the presence of sensing? In *AAAI-96*, 1139–1146.

McCarthy, J. 1963. Situations, actions and causal laws. Stanford University Artificial Intelligence Project Memo 2.

Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; and Ng, A. 2009 ROS: an open-source Robot Operating System *ICRA Workshop on OSS*

Reiter, R. 1980. A logic for default reasoning. *AI* 13:81–132.

Reiter, R. 2001. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*.

Scherl, R., and Levesque, H. 2003. Knowledge, action, and the frame problem. *AIJ* 144(1–2):1–39.

Shapiro, S.; Pagnucco, M.; Lespérance, Y.; and Levesque, H. 2011. Iterated belief change in the situation calculus. *AIJ* 175(1):165–192.

Shapiro, S.; Lespérance, Y.; and Levesque, H. 2005. Goal change. In *IJCAI-05*, 582–588.