A Declarative Formalization of STRIPS

J. Sierra-Santibáñez[1]

Abstract. We present a declarative formalization of STRIPS [1] as a reasoning strategy in the *situation calculus* [10]. The idea is to use logic not only to represent planning problems, but also to describe the mental situations, mental actions and reasoning strategy STRIPS uses to solve those problems.

1 INTRODUCTION

We present a declarative formalization of STRIPS in the *situation calculus*. This section outlines its main components and features, at the same time that describes the structure of the paper.

First, we present a *situation calculus* formalization of the STRIPS representation of planning problems. This formalization is proposed in [7], and it is characterized by making the STRIPS database explicit and reifying the formulas that are in the database. This allows handling deletions, and quantifying over those formulas. Then, we describe the situations, actions, and strategy STRIPS uses to reason about those representations of planning problems. Instead of formalizing a reasoning strategy as a meta-theory of the object-theory describing a planning problem [3], we introduce the concepts of *mental situations* and *mental actions* as first order objects of the language. This allows us to formalize in a single theory the dynamics of both: (1) the object situations STRIPS reasons about, which describe a particular planning problem; and (2) the mental situations it constructs to reason about that problem.

The formalization of the reasoning strategy used by STRIPS (*goal stack planning*) is presented in two stages. First, we define the *space of mental situations* on which that reasoning strategy operates. The formalization of mental situations makes the goal stack explicit, and represents subgoals as reified formulas describing the formulas that should and should not be in the database associated with a given situation. The reasoning process performed by STRIPS is explained in terms of a single type of mental action, *Hyp(a)*, which denotes the mental act of hypothesizing the execution of action a at a particular point of the plan currently being constructed. Later on in the paper, we introduce *backtracking* actions which allow retracting wrong hypotheses about actions. Then, we describe the *heuristics* and *search strategy* STRIPS uses to explore the space of mental situations. In particular, we provide axiomatizations of: (1) depth first search (with backtracking); and (2) some of the *heuristics* used by STRIPS for selecting actions and detecting *dead-ends* during the searching process. [15] shows how this simple formalization accounts for the standard STRIPS solution to the blocks world problem known as Sussman's anomaly. The successive mental situations constructed by STRIPS in the process of searching for that solution, and the interaction between such mental situations and the object situations (hypothetical solutions) they reason about are described in detail.

The ideas about the formalization of individual concepts and propositions in [6] are used to distinguish between the fact described by a formula and its truth value. In particular, we use a reified formula to represent the fact described by a formula, and the predicate *True* to assert the fact described by that formula. Thus $True(p)$ is true iff the fact described by the reified formula p is true. We use two subsorts of variables for reified formulas: f for formulas in the *database*; and g for formulas in the *goal stack*.

2 THE REPRESENTATION OF PLANNING PROBLEMS

The main ideas and axioms of the formalization presented in this section are taken from [7]. STRIPS is a planning system that uses a database of logical formulas[2] to represent information about a state. Associated with each situation s is a *database* of logical formulas describing what holds at that situation. The wff[3] $True(DB(f,s))$ is true if formula *f* is in the database associated with situation *s*.

Each action a has a *precondition*, an *add list* and a *delete list*, which are formally characterized by the following atomic formulas: (1) *True(Prec(a,s))* is true provided action *a* can be performed in situation *s* [4]; (2) *Del(f,a,s)* is true if formula f is to be deleted when action a is performed in situation *s*; (3) *Add(f,a,s)* is true if formula f is to be added when action a is performed in situation s.

The function *Result* maps a situation s and an action a into the situation that results when action a is performed in situation s. The following axiom determines what formulas are in the database associated with the situation $Result(a,s)$. If the precondition of action a is met at situation s, then the sentences on the delete list of a are deleted from the database, and the sentences on the add list of a are added to it.

$$\begin{aligned} True(DB(f, Result(a,s))) \leftrightarrow (True(Prec(a,s)) \wedge \\ (Add(f,a,s) \vee (True(DB(f,s)) \wedge \neg Del(f,a,s)))) \vee \\ (\neg True(Prec(a,s)) \wedge True(DB(f,s))) \end{aligned} \tag{1}$$

2.1 Blocks World

As an example, we formalize Sussman's anomaly problem within this framework. The individual variables *x, y* and *z* range over blocks. The constant blocks are *A, B, C* and *T* (for *Table*). The function *On*

[2] The formalization presented applies only to a simplified version of STRIPS in which we allow only atomic formulas in the database, add and delete lists of the actions, and in the stack describing the goal configuration. These conditions can be relaxed as shown in [4].

[3] The function DB maps a reified formula of the sort database f and a situation s into a reified formula of the sort goal stack.

[4] The function *Prec* maps an action a and a situation s into a reified formula of the sort goal stack.

[1] Computer Science Department, Stanford University, CA 94305, USA.

ECAI 98. *13th European Conference on Artificial Intelligence*
Edited by Henri Prade
Published in 1998 by John Wiley & Sons, Ltd.

maps a pair of blocks x and y into the reified formula of the sort database $On(x, y)$ describing the fact that block x is on block y. The function $Move$ maps a pair of blocks x and y into the action $Move(x, y)$ denoting the act of moving block x on top of block y. The *precondition, delete list,* and *add list* of $Move(x, y)$ are described as follows.

Figure 1. Sussman's anomaly.

$$\begin{aligned} &True(Prec(a, s)) \leftrightarrow \\ &\exists xy(a = Move(x, y) \wedge \neg True(DB(On(x, y), s)) \wedge \\ &x \neq T \wedge x \neq y \wedge \forall z \neg True(DB(On(z, x), s)) \wedge \\ &(y \neq T \rightarrow \forall z \neg True(DB(On(z, y), s)))) \end{aligned} \tag{2}$$

$$\begin{aligned} &Del(f, a, s) \leftrightarrow \\ &\exists xyz(f = On(x, z) \wedge a = Move(x, y) \wedge z \neq y) \end{aligned} \tag{3}$$

$$\begin{aligned} &Add(f, a, s) \leftrightarrow \\ &\exists xy(f = On(x, y) \wedge a = Move(x, y)) \end{aligned} \tag{4}$$

The conditions on the initial and goal situations (S_0 and S_g) are respectively described by axioms 5 and 6. The formula $Stack(i, f, s_g)$ is true iff formula f is at location i in the stack describing s_g.[5]

$$\begin{aligned} &True(DB(f, S_0)) \leftrightarrow \\ &\exists xy(f = On(x, y) \wedge ((x = A \wedge y = T) \vee \\ &(x = B \wedge y = T) \vee (x = C \wedge y = A))) \end{aligned} \tag{5}$$

$$\begin{aligned} &Stack(i, f, S_g) \leftrightarrow \\ &\exists xy(f = On(x, y) \wedge ((i = 1 \wedge x = A \wedge y = B) \vee \\ &(i = 2 \wedge x = B \wedge y = C) \vee (i = 3 \wedge x = C \wedge y = T))) \end{aligned} \tag{6}$$

3 THE REASONING STRATEGY

In order to solve a planning problem, STRIPS constructs some datastructures from the representation of the problem, and manipulates such datastructures until it finds a solution or it concludes that one cannot be found [11]. We describe each configuration of those datastructures as a *mental situation*, and the manipulations that STRIPS can perform on them as *mental actions*. Notice that *mental situations* and *mental actions* describe the internal state of a system as it reasons about a problem, whereas *situations* and *actions* describe the particular problem the system reasons about. In order to specify the *reasoning strategy* of a system, we need to describe: (1) the *mental situations* it constructs to reason about a problem; (2) the *mental actions* it can perform to transform those mental situations; and (3) the *heuristics* it uses to select the mental action to perform next during the process of searching for a solution.

[5] The variables i and s_g range over natural numbers and situations, respectively.

Goal Stack Planning The reasoning strategy used by STRIPS is *goal stack planning* [13]. In goal stack planning, the problem solver makes use of a *goal stack* GS that contains both subgoals and actions that have been proposed to satisfy those subgoals, and a database DB of logical formulas describing the *current situation*. The algorithm of goal stack planning can be summarized as follows.

Repeat the cycle below until the goal stack is empty. Otherwise, return P_{DB} –the plan associated with the database describing the current situation– as a solution.

1. Replace the top subgoal of GS by an appropriate action, and add its precondition to the top of GS.
2. Until the top element of GS is a subgoal that cannot be proved from the formulas in DB, do the following.
   (a) If the top element of GS is a subgoal that can be proved from the formulas in DB, pop it from GS.
   (b) If the top element of GS is an action, pop it from GS, apply it to DB, and append it to the end of P_{DB}.
3. If $GS = \emptyset$, check whether the formulas in the stack describing the goal configuration can be proved from DB. Add to GS the formulas that could not be proved from DB.

3.1 Mental Situations

As explained before, we use the concept of a mental situation in order to describe the internal state of a system as it reasons about a problem. A *mental situation* of STRIPS m is characterized by two atomic formulas: (1) $GS(i, g, m)$ is true if subgoal g is at location i in the goal stack associated with m; (2) $Current(m) = s$ is true if s is the current situation described by the database associated with m. Notice that, we do not need an explicit representation of the database associated with a mental situation, because axioms 1 to 5 allow us to compute such a database from the description of the current situation associated with that mental situation.

Axioms 7 and 8 describe how the *initial mental situation* M_0 is constructed from the representation of a planning problem. The database associated with the initial mental situation describes S_0 – the problem's initial situation. The goal stack contains a subgoal[6] of the form $DB(f, S_0)$ for each formula f that is in the stack describing S_g –the problem's goal situation– and is not in the database associated with the current situation (i.e., $\neg True(DB(f, S_0))$). The location numbers of the subgoals in the goal stack are the same as the location numbers of the formulas they refer to in the stack describing the problem's goal situation.

$$Current(M_0) = S_0 \tag{7}$$

$$\begin{aligned} &GS(i, g, M_0) \leftrightarrow \exists f(Stack(i, f, S_g) \wedge \\ &\neg True(DB(f, S_0)) \wedge g = DB(f, S_0)) \end{aligned} \tag{8}$$

[6] The intuition behind the formalization of subgoals as reified formulas of the form $DB(f, S_0)$ is that the problem would be solved if the formulas in the stack describing the problem's goal situation were true in the problem's initial situation.

3.2 Mental Actions

Axioms 9 to 14 describe the effects of the mental action *Hyp(a)* – which summarizes the three-step cycle of the algorithm– on the fluents that characterize a mental situation, *GS(i,f,m)* and *Current(m)*. The expression *Result(Hyp(a),m)* denotes the mental situation resulting from performing $Hyp(a)$ in mental situation m. The predicates GS^1 and GS^2 describe the intermediate goal stacks resulting from steps 1 and 2, respectively.

The effect of step 1 is formalized as follows. A subgoal describing the fact that the precondition of action a should hold at the current situation associated with m is added to the top (location 1) of the goal stack. The location numbers of the subgoals that were in the goal stack associated with m are incremented by 1. Instead of adding action a to the goal stack, we update the situation arguments of the subgoals[7] that were in the goal stack associated with m. The idea is to take into account the fact that adding action a to the top of the goal stack associated with m is equivalent to hypothesizing the execution of a in the current situation associated with m.

$$\begin{aligned} GS^1(i,g,Result(Hyp(a),m)) \leftrightarrow \\ \exists s(Current(m)=s \wedge i=1 \wedge g=Prec(a,s)) \vee \\ \exists g_1(GS(i-1,g_1,m) \wedge g=Update(g_1,a,m)) \end{aligned} \quad (9)$$

The sequences of actions associated with the situation arguments of the subgoals in the goal stack associated with a mental situation describe the plan (or a part of the plan) that is being constructed by STRIPS in the process of searching for a solution. $Hyp(a)$ denotes the mental act of *hypothesizing* that inserting action a after the sequence of actions that have been applied to the database describing the current situation may improve the plan constructed so far. This can be formalized by modifying the situation arguments of the subgoals according to the following update function. The function *Update* maps a subgoal of the form $DB(f,s_1)$ into a subgoal of the form $DB(f,s_2)$, where s_2 is obtained from s_1 as follows. If s is the current situation associated with m and s_1 is the situation resulting of executing the sequence of actions r in s (i.e., $s_1 = Result(r,s)$), then s_2 is the situation resulting of executing the sequence of actions r preceded by action a in s (i.e., $s_2 = Result(\{a|r\},s)$). *Update* maps a subgoal of the form $Prec(b,s_1)$ into a subgoal of the form $Prec(b,s_2)$, where s_2 is obtained from s_1 as explained above.

$$\begin{aligned} g=Update(g_1,a,m) \leftrightarrow \exists s s_1 s_2 r f b((Current(m)=s \wedge \\ s_1=Result(r,s) \wedge s_2=Result(\{a|r\},s)) \wedge \\ ((g_1=DB(f,s_1) \wedge g=DB(f,s_2)) \vee \\ (g_1=Prec(b,s_1) \wedge g=Prec(b,s_2)))) \end{aligned} \quad (10)$$

The effect of step 2 is formalized as follows. True subgoals are popped from GS^1 yielding GS^2.

$$\begin{aligned} GS^2(i,g,Result(Hyp(a),m)) \leftrightarrow \\ GS^1(i,g,Result(Hyp(a),m)) \wedge \neg True(g) \end{aligned} \quad (11)$$

If GS^2 is not empty the situation argument of the top subgoal of GS^2 is taken as the new current situation. If GS^2 is empty, the situation argument of the last subgoal popped from GS^2 is taken as the current situation. Axioms 12 and 13 capture the intuition that a subgoal is popped from the goal stack iff it has been satisfied by the current database at some point during the process of popping off true subgoals.[8] Therefore, the sequence of actions associated with the situation argument of the subgoal must have been applied to DB at that point.

$$\begin{aligned} GS^2(i,g,Result(Hyp(a),m)) \wedge At(g,s_1) \wedge \\ \forall igs(GS^2(i,g,Result(Hyp(a),m)) \wedge At(g,s) \rightarrow \\ s_1 \leq s) \rightarrow Current(Result(Hyp(a),m)) = s_1 \end{aligned} \quad (12)$$

$$\begin{aligned} \neg\exists ig GS^2(i,g,Result(Hyp(a),m)) \wedge \\ GS^1(i,g,Result(Hyp(a),m)) \wedge At(g,s_1) \wedge \\ \forall igs(GS^1(i,g,Result(Hyp(a),m)) \wedge At(g,s) \rightarrow \\ s \leq s_1) \rightarrow Current(Result(Hyp(a),m)) = s_1 \end{aligned} \quad (13)$$

Finally, the effect of step 3 is formalized as follows. If GS^2 is not empty, then the goal stack GS associated with $Result(Hyp(a),m)$ is GS^2. Otherwise, a subgoal of the form $DB(f,s_1)$ is added to GS for each formula f in the stack describing the problem's goal situation that is not true in the current situation.[9]

$$\begin{aligned} GS(i,g,Result(Hyp(a),m)) \leftrightarrow \\ GS^2(i,g,Result(Hyp(a),m)) \vee \\ (\neg\exists ig GS^2(i,g,Result(Hyp(a),m)) \wedge \\ \exists s_1 f(Current(Result(Hyp(a),m)) = s_1 \wedge \\ Stack(i,f,S_g) \wedge \neg True(DB(f,s_1)) \wedge g = DB(f,s_1))) \end{aligned} \quad (14)$$

Appropriateness In order to determine which actions are appropriate to satisfy a subgoal g in the goal stack associated with mental situation m, STRIPS tries to prove the subgoal from the formulas in the database associated with the current situation using resolution. If the subgoal cannot be proved, it selects those actions whose add lists contain the largest number of predicates that can be resolved against the clauses in the incompleted proof of the subgoal. Instead of doing this, we provide an explicit definition of the predicate *Appr* (for *appropriate*) that selects the same actions for the blocks world example considered.

$$\begin{aligned} Appr(Hyp(a),g,m) \leftrightarrow \exists sxyzw(GS(i,g,m) \wedge \\ ((g=DB(On(x,y),s) \wedge a=Move(x,y)) \vee \\ (g=Prec(Move(x,y),s) \wedge a=Move(z,w) \wedge \\ True(DB(On(z,x),s)) \wedge w \neq y \wedge w \neq x \wedge w \neq z) \vee \\ (g=Prec(Move(x,y),s) \wedge a=Move(z,w) \wedge y \neq T \wedge \\ True(DB(On(z,y),s)) \wedge w \neq y \wedge w \neq x \wedge w \neq z))) \end{aligned} \quad (15)$$

3.3 The Space of Mental Situations

The following axioms allow us to define precisely the space of mental situations. We use variables for blocks ($x, y, z, w, \ldots$), for situations

[7] These are the subgoals that would have been below a if the top subgoal of the goal stack had been replaced by action a (step 1 of the algorithm).

[8] The expressions $At(g,s)$ and $s_1 < s_2$ are defined as follows: (1) $At(g,s) \leftrightarrow \exists fa(g = DB(f,s) \vee g = Prec(a,s))$; and (2) $\forall s(\neg s < S_0) \wedge \forall s(\neg s < S_g) \wedge \forall a s s_1(s < Result(a,s_1) \leftrightarrow True(Prec(a,s_1)) \wedge s \leq s_1)$.

[9] In this case, the location numbers of the subgoals in GS are the same as the location numbers of the formulas they refer to in the stack describing S_g.

(s, s_1, …), for actions (a, a_1, …), for mental situations (m, m_1, …), for mental actions (α, α_1, …), for formulas in the database (f, $f_1, \dots$), for subgoals –formulas in the goal stack– ($g, g_1, \dots$), for sequences of actions, ($r, r_1, \dots$), and for sequences of mental actions ($l, l_1, \dots$).

We assume uniqueness of names for every function symbol h, and every pair of distinct function symbols h and g. The symbols h and g are syntactical variables ranging over distinct function symbols; the symbols c_1 to c_4 are syntactical variables ranging over block constants. Axiom 17 defines the expression $m < m_1$ [12], which means that mental situation m_1 can be *reached* from mental situation m by executing a nonempty sequence of *appropriate* mental actions ($m \leq m_1$ is an abbreviation for $m < m_1 \vee m = m_1$). We include domain closure axioms for blocks, mental situations, actions, and mental actions. Axiom 19 of induction allows us to prove that a property holds for all mental situations that are reachable from a given mental situation.

$$h(\vec{x}) \neq g(\vec{y}) \wedge (h(\vec{x}) = h(\vec{y}) \rightarrow \vec{x} = \vec{y}) \quad (16)$$

$$\forall m(\neg m < M_0) \wedge \forall \alpha m m_1(m < Result(\alpha, m_1) \leftrightarrow \exists g Appr(\alpha, g, m_1) \wedge m \leq m_1) \quad (17)$$

$$\forall x(x = A \vee x = B \vee x = C \vee x = T) \wedge \forall m(M_0 \leq m) \quad (18)$$
$$\forall a(\bigvee_{c_1, c_2 \in \{A,B,C,T\}} a = Move(c_1, c_2)) \wedge \forall \alpha(\alpha = Hyp(a))$$

$$\forall P(P(m_1) \wedge \forall m\alpha(m_1 \leq m \wedge P(m) \wedge \exists g Appr(\alpha, g, m) \rightarrow P(Result(\alpha, m))) \rightarrow \forall m(m_1 \leq m \rightarrow P(m))) \quad (19)$$

3.4 The Heuristics

So far, we have described the space of mental situations constructed by STRIPS. This space is, in fact, a *tree* that has been specified by the initial mental situation, the set of appropriate mental actions for a mental situation, and the effects of mental actions on mental situations. We describe now the search strategy according to which STRIPS explores such a tree of mental situations in order to find a solution to a planning problem, and the heuristics it uses for selecting actions and detecting dead-ends during the searching process.

3.4.1 Ordering Mental Actions

Axiom 20 defines a total order $Better$ on the set of appropriate mental actions for a mental situation. The predicate $Better(\alpha, \beta, m)$ is true if mental action α is better than mental action β at mental situation m. Therefore, α will be selected before β at mental situation m in the process of searching for a solution.

The preference relation *Better* is a powerful mechanism for the declarative formalization of heuristics [10] [2] [17] [16]. Here we use only two simple heuristics: (1) actions appropriate for goals in upper levels of the goal stack[10] are preferred to actions appropriate for goals in lower levels; (2) if two actions are appropriate for the same goals, they are selected in alphabetic order $\prec_{alph}$.

[10] A subgoal is above another subgoal in the goal stack if its location number is smaller than the location number of the second subgoal.

$$\begin{aligned} Better(\alpha, \beta, m) \leftrightarrow \exists i g_1(GS(i, g_1, m) \wedge \\ Appr(\alpha, g_1, m) \wedge \neg Appr(\beta, g_1, m) \wedge \forall j g_2(GS(j, g_2, m) \wedge \\ Appr(\beta, g_2, m) \wedge \neg Appr(\alpha, g_2, m) \rightarrow i < j)) \vee \\ (\forall i g(GS(i, g, m) \rightarrow (Appr(\alpha, g, m) \leftrightarrow Appr(\beta, g, m))) \wedge \\ \exists c_1 c_2 c_3 c_4((c_1 \prec_{alph} c_3 \vee (c_1 = c_3 \wedge c_2 \prec_{alph} c_4)) \wedge \\ \alpha = Hyp(Move(c_1, c_2)) \wedge \beta = Hyp(Move(c_3, c_4)))) \end{aligned} \quad (20)$$

3.4.2 The Search Strategy

Given a total order on the set of appropriate mental actions for a mental situation, the relation $Better(m_1, m_2)$ that sorts the set of mental situations according to the order in which they are explored by the depth first search strategy associated with $Better(\alpha, \beta, m)$ is defined by axiom 21 –a variation of axiom (10) in [5].

$$\begin{aligned} Better(m_1, m_2) \leftrightarrow m_1 \leq m_2 \vee \\ \exists \alpha \beta m(Result(\alpha, m) \leq m_1 \wedge Result(\beta, m) \leq m_2 \wedge \\ Better(\alpha, \beta, m)) \end{aligned} \quad (21)$$

Goal Stack Derivation A mental situation m is the *goal stack derivation* of a plan for a planning problem if: (1) there are no formulas in the goal stack associated with m; and (2) it is minimal with respect to the search strategy.

$$\begin{aligned} Goal\text{-}stack\text{-}der(m) \leftrightarrow \neg \exists i g GS(i, g, m) \wedge \\ \forall m_1(\neg \exists i g GS(i, g, m_1) \rightarrow Better(m, m_1)) \end{aligned} \quad (22)$$

Solution The *plan* r returned by STRIPS as a solution to a planning problem is the sequence of actions leading from the problem's initial situation S_0 to the current situation s associated with the *goal stack derivation* m of a plan for it.

$$\begin{aligned} Solution(r) \leftrightarrow \exists m s(Goal\text{-}stack\text{-}der(m) \wedge \\ Current(m) = s \wedge s = Result(r, S_0)) \end{aligned} \quad (23)$$

3.4.3 Backtracking Actions

Backtracking actions are interesting for the study of reasoning strategies. The following axioms allow us to take into account backtracking actions within our formalization. Axiom 24 describes two simple heuristics for determining the appropriateness of backtracking actions, i.e., for detecting *dead-ends* during the searching process. Backtracking actions are taken when: (1) the length[11] of the mental situation explored is equal to some threshold N [12], or (2) its length is smaller than N but there are no appropriate mental actions that can be applied to it.

$$\begin{aligned} Dead\text{-}end(m) \leftrightarrow Length(m) = N \vee \\ (Length(m) < N \wedge \neg \exists m_1(m < m_1)) \end{aligned} \quad (24)$$

[11] We use the following abbreviations $Result(\{\}, m) = m$, and $Result(\{\alpha | l\}, m) = Result(l, Result(\alpha, m))$. The length of a mental situation is defined as follows: (1) $Length(M_0) = 0$; (2) $Length(Result(\{\alpha | l\}, M_0)) = Length(Result(\{l\}, M_0)) + 1$.

[12] The constant N is a natural number describing the threshold of the depth first search strategy used by STRIPS, i.e., the maximum depth to which it explores the tree of mental situations.

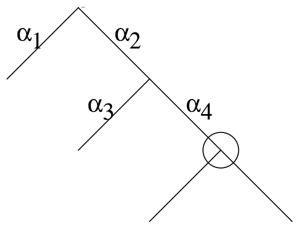

Figure 2. This figure illustrates the definition of *Path(m)* in axiom 26: $Path(m_2) = \{\alpha_1, Back, \alpha_2, \alpha_3\}$, $m_6 = M_0$, $l = \{\alpha_4\}$, and $Path(m) = \{\alpha_1, Back, \alpha_2, \alpha_3, Back, \alpha_4\}$.

It is possible to define more interesting heuristics for the appropriateness of backtracking actions, which detect early on unappropriate choices of the search strategy.

The *path* associated with a mental situation m describes the sequence of mental actions (including backtracking actions of the form *Back*) STRIPS needs to perform in order to find m (see figure 2 for a graphical explanation).

$$\begin{aligned} & m \le m_1 \wedge Dead\text{-}end(m_1) \wedge \forall m_2(Dead\text{-}end(m_2) \rightarrow \\ & Better(m_1, m_2)) \wedge m = Result(l, M_0) \rightarrow Path(m) = l \end{aligned} \quad (25)$$

$$\begin{aligned} & m \le m_1 \wedge Dead\text{-}end(m_1) \wedge \forall m_4(Dead\text{-}end(m_4) \wedge \\ & m \le m_4 \wedge m_4 \ne m_1 \rightarrow Better(m_1, m_4)) \wedge \\ & Dead\text{-}end(m_2) \wedge Better(m_2, m_1) \wedge Path(m_2) = l_2 \wedge \\ & \forall m_3(Dead\text{-}end(m_3) \wedge Better(m_3, m_1) \wedge m_3 \ne m_2 \rightarrow \\ & Better(m_3, m_2)) \wedge m = Result(l, m_5) \wedge m_5 \le m_2 \wedge \\ & \forall m_6(m_6 \le m \wedge m_6 \le m_2 \rightarrow m_6 \le m_5) \rightarrow \\ & Path(m) = \{l_2, Back, l\} \end{aligned} \quad (26)$$

Taking into account backtracking actions (i.e., including axioms 24, 25 and 26) requires modifying the definition of *goal stack derivation* (axiom 22) as follows.

$$\begin{aligned} & Goal\text{-}stack\text{-}der(m) \leftrightarrow \neg\exists i g GS(i, g, m) \wedge Length(m) \le N \wedge \\ & \forall m_1(\neg\exists i g GS(i, g, m_1) \wedge Length(m_1) \le N \rightarrow Better(m, m_1)) \end{aligned}$$

4 CONCLUSIONS

We have presented a formalization of STRIPS as a reasoning strategy in the *situation calculus*. In doing so, we have proposed and illustrated a conceptual framework for the declarative formalization of reasoning strategies consisting of mental situations, mental actions, and heuristics. Declarative formalizations of reasoning strategies have a number of advantages. One of them is that they can be improved by simple additions of better heuristics [17] [16]. For example, axioms 15, 20 and 24 describe the heuristics used for determining which actions are appropriate for achieving a subgoal, ordering actions, and detecting dead-ends during the searching process. We can enhance the reasoning strategy formalized in the paper by simply adding new disjuncts describing better heuristics to the right hand side of any of those axioms.

ACKNOWLEDGEMENTS

The author would like to thank John McCarthy for introducing her into the idea of the *mental situation calculus*, suggesting the problem of the declarative formalization of reasoning strategies, and providing valuable ideas, comments and help whenever they were needed. Luc Steels encouraged and enriched the writing of this paper with valuable ideas. Mike Genesereth read carefully a previous version of the paper, and contributed with valuable comments. John McCarthy, Tom Costello, Eyal Amir and Aarati Parmar participate in useful discussions that helped to shape the final version of the paper. This work was supported partly by the Navy Grant N00014-94-1-0775, and partly by the Air Force Grant F49620-97-1-0207.

REFERENCES

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

[2] M. R. Genesereth and J.Y. Hsu, 'Partial programs', Technical Report Logic Group 89-20, Department of Computer Science, Stanford University, (1989).

[3] F. Harmelen and J.R. Balder, '*(ML)*^{2}: a formal language for KADS models of expertise', *Knowledge Acquisition*, **4**(1), 127–161, (March 1992). Special issue: *The KADS approach to knowledge engineering.*

[4] V. Lifschitz, 'On the semantics of STRIPS', in *Proceedings of the Workshop on Reasoning about Actions and Plans*, pp. 1–9, (1986).

[5] F. Lin, 'Applications of the situation calculus to formalizing control and strategic information: The prolog cut operator', in *Proceedings of IJCAI-97*, pp. 1412–1418, (1997).

[6] J. McCarthy, 'First order theories of individual concepts and propositions', *Machine Intelligence*, **9**, (1979).

[7] J. McCarthy, 'Formalization of STRIPS in situation calculus', Technical Report Formal Reasoning Group, Department of Computer Science, Stanford University, (1985).

[8] J. McCarthy, *Formalizing Common Sense: papers by John McCarthy*, Edited by Vladimir Lifschitz, Ablex, Norwood, NJ, 1990.

[9] J. McCarthy, 'Making robots conscious of their mental states', Technical Report Formal Reasoning Group, Department of Computer Science, Stanford University, (1995).

[10] J. McCarthy and P. Hayes, 'Some philosophical problems from the standpoint of Artificial Intelligence', *Machine Intelligence*, **4**, 463–502, (1969).

[11] N.J. Nilsson, *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann Publishers, Inc. San Mateo, California, 1998.

[12] R. Reiter, 'Proving properties of states in the situation calculus', *Artificial Intelligence*, **64**, 337–351, (1993).

[13] E. Rich and K. Knight, *Artificial Intelligence*, Second edition, McGraw-Hill, Inc., 1991.

[14] J. Sierra-Santibáñez, *Software agents require formal Knowledge Level models*, Ph.D. dissertation, Free University of Brussels, 1996.

[15] J. Sierra-Santibáñez, 'Declarative formalization of reasoning strategies', Technical Report Formal Reasoning Group, Department of Computer Science, Stanford University, (1997).

[16] J. Sierra-Santibáñez, 'Declarative formalization of heuristics', in *International Workshop Validation and Verification of Knowledge Based Systems*, (1998).

[17] J. Sierra-Santibáñez, 'Declarative formalization of strategies for action selection', in *Seventh International Workshop on Non-monotonic Reasoning*, (1998).

[18] L. Steels, 'The spontaneous self-organization of an adaptive language', *Machine Intelligence*, **15**, (1996).

[19] L. Steels and P. Vogt, 'Grounding adaptive language games in robotic agents', in *Proceedings of the* 4^{th} *European Conference on Artificial Life*. Cambridge, MA. The MIT Press., (1997).