Goals and Rational Action in the Situation Calculus— A Preliminary Report *

Steven Shapiro, Yves Lespérance and Hector J. Levesque[†]
Department of Computer Science
University of Toronto
Toronto, ON, M5S 1A4 Canada
{steven,lesperan,hector}@ai.toronto.edu

Abstract

In this paper, we use an extended version of the situation calculus to formalize goals and rational action. We then use these notions and a definition of ability (Lespérance *et al.* 1995b) to show that an agent that is acting rationally will achieve its goals when it is able to do so.

1 Introduction

This paper describes work on rational action that arose from our efforts to create an explicit representation of the goals of agents in the situation calculus. The utility of an explicit representation of the goals of agents is evident when we consider domains with multiple interacting agents. In domains where agents are communicating and cooperating to perform a task, the ability to specify the knowledge and goals of the agents becomes useful in order to determine that the agents can perform their parts of the task and have the required commitment to see their parts to completion. Designers of agents can use this information to help predict the behaviors of the agents they create. The agents themselves can also use this information to communicate with other agents, and to reason about their behavior.

Our work in this area builds on earlier efforts both to enhance the situation calculus with a richer set of primitives for modelling dynamic worlds ((Reiter 1991), (Scherl & Levesque 1993), (Lin & Reiter 1994), (Lespérance *et al.* 1995b)) and to develop a high-level agent programming language called Golog ((Lespérance *et al.* 1994),(Lespérance *et al.* 1995a)). We are expanding research in both these areas by adding primitives to the extended language to explicitly talk about the goals of agents. Our efforts bring together the work by Scherl and Levesque (Scherl & Levesque 1993) to add an explicit representation of knowledge to the situation calculus with Cohen and Levesque's (Cohen & Levesque 1990) formalization of intentions in terms of modalities for beliefs and goals.

One cannot predict what an agent will do based solely on the specification of its knowledge and goals. It is possible for an "irrational" agent to try to achieve some goal without taking into account what it knows about the world, and it is also possible for it to ignore its goals altogether. We need to model rational action explicitly in order to bridge the gap between the knowledge and goals of agents, and their future actions. Informally, we take an agent to be acting rationally when it is performing actions that it believes[1] will bring about its goals. If agent A knows that agent B has the appropriate goals and knowledge to help A achieve its goals and agent B is acting rationally, then A can rely on B's cooperation, since B will be working to achieve its goals, which coincide with A's goals.

In the next section, we outline previous work on our framework. In section 3, we develop a formalization of the goals of the agent. In section 4, we define what it means for an agent to be acting rationally, and state a theorem that links the abilities and goals of an agent that is acting rationally to what the agent will actually achieve. While this is still work in progress, to our knowledge it is the first attempt to develop a theory of rational action that appeals to the abilities of the agent.

2 Previous work

2.1 Theory of action

Our theory is based on an extended version of the situation calculus (McCarthy & Hayes 1979), a predicate calculus dialect for representing dynamically changing worlds. In this formalism, the world is taken to be in a certain situation. That situation can only change as a result of an agent performing an action. The term $do(a, s)$ represents the situation that results from the agent's executing action a in situation s. For exam-

*This research received financial support from the Information Technology Research Center (Ontario, Canada), the Institute for Robotics and Intelligent Systems (Canada), and the Natural Science and Engineering Research Council (Canada).

[†]Fellow of the Canadian Institute for Advanced Research.

[1]In this paper, we do not distinguish between knowledge and belief. The terms are used interchangeably.

ple, the formula $\text{On}(\text{A}, \text{B}, do(\textsc{putOn}(\text{A}, \text{B}), s))$ could mean that A is on B in the situation resulting from the agent's doing $\textsc{putOn}(\text{A}, \text{B})$ in s. Predicates and function symbols whose value may change from situation to situation (and whose last argument is a situation) are called *fluents*.

An action is specified by first stating the conditions under which it can be performed by means of a *precondition axiom*. For example,[2]

$$\begin{aligned}&Poss(\textsc{pickUp}(x), s) \equiv\\&\quad \forall z \neg \textsc{Holding}(z, s) \wedge \textsc{nextTo}(x, s)\end{aligned}$$

means that it is possible for the agent to pick up an object x in situation s iff it is not holding anything and it is standing next to x in s. Then, one specifies how the action affects the world with *effect axioms*, for example:

$$\begin{aligned}&Poss(\textsc{drop}(x), s) \wedge \textsc{Fragile}(x) \supset\\&\quad \textsc{Broken}(x, do(\textsc{drop}(x), s)).\end{aligned}$$

The above axioms are not sufficient if one wants to reason about change. It is usually necessary to add frame axioms that specify when fluents remain unchanged by actions. The frame problem (McCarthy & Hayes 1979) arises because the number of these frame axioms is of the order of the product of the number of fluents and the number of actions. Our approach incorporates Reiter's treatment of the frame problem (Reiter 1991) (which extends previous proposals: (Pednault 1989), (Schubert 1990) and (Haas 1987)). Reiter describes a procedure which collects all effect axioms about a given fluent. Using the assumption that these axioms specify all the ways the value of the fluent may change, a syntactic transformation is performed to obtain a *successor state axiom* for the fluent, for example:

$$\begin{aligned}&Poss(a, s) \supset [\textsc{Broken}(x, do(a, s)) \equiv\\&\quad (a = \textsc{drop}(x) \wedge \textsc{Fragile}(x)) \vee\\&\qquad (\textsc{Broken}(x, s) \wedge a \neq \textsc{repair}(x))].\end{aligned}$$

This says that x is broken after the agent does action a in situation s iff either the action was to drop x and x is fragile, or x was already broken in s and the action was not to repair it. This treatment avoids the proliferation of axioms, as it only requires a single successor state axiom per fluent and a single precondition axiom per action.[3]

2.2 Knowledge and Perception

Suppose we want to model a world in which there is a safe with a combination lock.[4] If the safe is locked and the correct combination is dialed, then the safe becomes unlocked. However, if the incorrect combination is dialed, the safe explodes. The agent can only dial a combination if the safe is intact, and it is not possible to change the combination of the safe. Here are the axioms for this scenario:

$$\begin{aligned}&Poss(\textsc{dial}(c), s) \equiv \neg\textsc{Exploded}(s)\\&Poss(a, s) \supset [\textsc{Exploded}(do(a, s)) \equiv\\&\quad \exists c(a = \textsc{dial}(c) \wedge \textsc{combOfSafe}(s) \neq c) \vee\\&\qquad \textsc{Exploded}(s)]\\&Poss(a, s) \supset [\textsc{Locked}(do(a, s)) \equiv\\&\quad \forall c(a \neq \textsc{dial}(c) \vee \textsc{combOfSafe}(s) \neq c) \wedge\\&\qquad \textsc{Locked}(s)]\\&Poss(a, s) \supset [\textsc{combOfSafe}(do(a, s)) = c \equiv\\&\quad \textsc{combOfSafe}(s) = c]\end{aligned}$$

In this scenario, the only agents that can definitely unlock the safe are ones that know the combination in advance because if an agent tries a random combination, the safe will likely explode. Suppose the correct combination is written on a piece of paper, and an agent can read the combination from the paper. How can we model the effects on the world of reading the combination? Scherl and Levesque (Scherl & Levesque 1993) call this type of actions (e.g., perception and communication actions) knowledge-producing actions, and they provide an account of how to represent these actions in the situation calculus. Such actions affect the mental state of the agent rather than the state of the external world. For example, after performing the action $\textsc{readCombOfSafe}$, an agent might know the combination of the safe it is trying to open:

$$\begin{aligned}&Poss(\textsc{readCombOfSafe}, s) \supset\\&\quad \exists c\, \mathbf{Know}(\textsc{combOfSafe}(s) = c,\\&\qquad\qquad do(\textsc{readCombOfSafe}, s)).\end{aligned}$$

Knowledge is represented by adapting the possible worlds model to the situation calculus (as first done in (Moore 1985)). $K(s', s)$ represents the fact that in situation s, the agent thinks that it could be in situation s'. We call s' an *alternative situation to s*. $\mathbf{Know}(\phi, s)$ is an abbreviation for the formula $\forall s'(K(s', s) \supset \phi(s'))$.[5]

Scherl and Levesque show how to obtain a successor state axiom for K that completely specifies how knowledge is affected by actions. In our example, the only knowledge-producing action is the $\textsc{readCombOfSafe}$ action. The successor state axiom for K can be specified as follows:

$$\begin{aligned}&Poss(a, s) \supset (K(s^*, do(a, s)) \equiv\\&\quad \exists s'[K(s', s) \wedge s^* = do(a, s') \wedge Poss(a, s') \wedge\\&\qquad (a = \textsc{readCombOfSafe} \supset\\&\qquad\quad \textsc{CombOfSafe}(s') = \textsc{CombOfSafe}(s))]).\end{aligned}$$

First note that for non-knowledge-producing actions (e.g. $\textsc{dial}(c)$), the specification ensures that the only

[2] By convention, unbound variables in a formula are universally quantified.

[3] This discussion ignores the ramification and qualification problems; treatments compatible with our approach were proposed in (Lin & Reiter 1994).

[4] This example is adapted from (Moore 1985).

[5] ϕ is a formula that contains a placeholder now instead of a situation argument, e.g., $\neg\textsc{Locked}(\mathsf{now})$. Where the intended meaning is clear, we suppress the placeholder, e.g., $\neg\textsc{Locked}$. $\phi(s)$ is the formula that results from substituting s for now in ϕ.

change in knowledge that occurs in moving from s to $do(\text{DIAL}(c), s)$ is the fact that the action DIAL has been successfully performed. For the case of a knowledge-producing action such as READCOMBOFSAFE, the idea is that in moving from s to $do(\text{READCOMBOFSAFE}, s)$, the agent not only knows that the action has been performed (as above), but also the value of the associated fluent COMBOFSAFE. Since in this case we require that $\text{COMBOFSAFE}(s') = \text{COMBOFSAFE}(s)$, COMBOFSAFE will have the same value in all s' such that $K(do(\text{READCOMBOFSAFE}, s'), do(\text{READCOMBOFSAFE}, s))$. Observe that for any situation s, $\text{COMBOFSAFE}(do(\text{READCOMBOFSAFE}, s)) = c$ iff $\text{COMBOFSAFE}(s) = c$. Therefore, COMBOFSAFE has the same value in all worlds s^* such that $K(s^*, do(\text{READCOMBOFSAFE}, s))$, and so

$$\exists c\, \mathbf{Know}(\text{COMBOFSAFE}(s) = c, do(\text{READCOMBOFSAFE}, s))$$

holds. This can be extended to an arbitrary number of knowledge-producing actions in a straightforward way.

As a simple example, consider the graph in Figure 1. Situations are nodes in the graph, and the edges are labelled by actions. A subset of the K relation is represented by the boxes around the nodes. If a situation s appears in the same box as another situation s', then $K(s', s)$. The figure illustrates that the agent does not know the combination of the safe in S_0, since the value of COMBOFSAFE is not the same in S_0 and S_0^*. However, K only relates $do(\text{READCOMBOFSAFE}, S_0)$ to itself, therefore in this situation, the agent does know the combination.

2.3 Ability

With the addition of knowledge to the language, it becomes possible to specify what goals the agent *knows how* to achieve. In (Lespérance *et al.* 1995b), $\mathbf{Can}(\phi, s)$ is defined to mean that the agent knows how to achieve ϕ starting in situation s. Intuitively, the definition of **Can** specifies that in order for the agent to be able to achieve ϕ starting in s, it must know in s of some strategy that it can follow to eventually achieve ϕ. A strategy is formalized as a function from situations to actions, which we call an *action selection function* (ASF). As we will see, this way of formalizing strategies is quite expressive. In particular, it allows the agent's choice of action to vary depending on what knowledge it acquires as it acts.

To see how an ASF can be a model for a strategy, consider the safe example of the previous section. One strategy the agent can use to unlock the safe is to find out the combination of the safe by reading it from the paper, and then dialing the combination. Notice that this strategy is not just a pair of actions, since the second action varies according to the actual combination ($\text{DIAL}(c)$ is a different action than $\text{DIAL}(c')$, if $c \neq c'$). The strategy allows the agent to take different actions depending on the knowledge it acquires as it follows the strategy. Let σ be an ASF, i.e., a mapping from situations to actions. Given a starting situation s_0, it is easy to see that σ defines an infinite sequence of situations. $s_1 = do(\sigma(s_0), s_0)$ is the second situation in the sequence. In general, $s_i = do(\sigma(s_{i-1}), s_{i-1})$. We define the predicate **OnPath** to mean that situation s' is in the situation sequence defined by σ and s:[6]

$$\mathbf{OnPath}(\sigma, s, s') \stackrel{\text{def}}{=} s \leq s' \wedge \forall a \forall s^* (s < do(a, s^*) \leq s' \supset \sigma(s^*) = a),$$

The actions that label the transitions between situations in the sequence can be thought of as a possible course of action for the agent to follow if it is in situation s. Also, for every alternative situation s^* that the agent thinks it might be in when it really is in s, σ defines a course of action starting at s^*.

Suppose the agent does not know initially whether the combination is 0 or 1. Thus in situation S_0, there might be two alternative situations S_0 and S_0^*. Let the combination of the safe in situation S_0 be 0, and the combination in S_0^* be 1. An ASF σ_0 that models the strategy outlined earlier has the following mappings: $\sigma_0(S_0) = \text{READCOMBOFSAFE}$, $\sigma_0(S_0^*) = \text{READCOMBOFSAFE}$, $\sigma_0(do(\text{READCOMBOFSAFE}, S_0)) = \text{DIAL}(0)$, and $\sigma_0(do(\text{READCOMBOFSAFE}, S_0^*)) = \text{DIAL}(1)$. This part of σ_0 is illustrated in Figure 1; the mappings are represented by solid edges in the figure. The agent does not know in S_0 whether the combination is 0 or 1. The strategy succeeds because after reading the combination, $do(\text{READCOMBOFSAFE}, S_0^*)$ is not an alternative to $do(\text{READCOMBOFSAFE}, S_0)$, and in the latter situation σ_0 prescribes dialing 0, which is the correct combination.

If an ASF σ maps different alternative situations to different actions, then the agent cannot follow the course of action suggested by σ, since the agent does not know which of the alternative situations it is actually in. We are only interested in ASFs that the agent can follow. We are also only interested in ASFs that define courses of action that are possible to perform. Therefore, in the following discussion we restrict our attention to ASFs that satisfy the following axiom:

$$\mathbf{OnPath}(\sigma, s, s') \supset \exists a\, \mathbf{Know}(\sigma(\mathsf{now}) = a, s') \wedge Poss(\sigma(s'), s').$$

We say that the agent can achieve a goal ϕ in situation s iff there exists an ASF σ such that the agent knows it will get to a state where ϕ holds by following σ:[7]

$$\mathbf{Can}(\phi, s) \stackrel{\text{def}}{=} \exists \sigma \mathbf{Know}(\exists s' (\mathbf{OnPath}(\sigma, \mathsf{now}, s') \wedge \phi(s')), s).$$

[6] $s < s'$ means that there is a sequence of actions that can be performed starting in situation s and which results in situation s'. $s \leq s'$ is an abbreviation for $s < s' \vee s = s'$.

[7] We only require that the agent know that σ will lead the agent to a situation where the goal holds. In (Lespérance *et al.* 1995b), the agent is required to know that σ will lead the agent to a situation where it *knows* the goal holds.

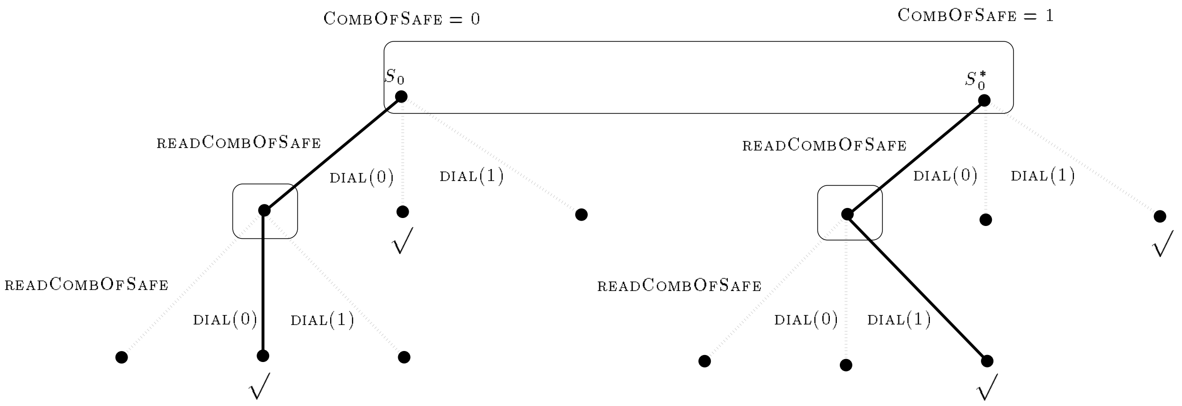

Figure 1: An example of an ASF.

From this definition, it follows that if the agent knows that the safe is intact then it can unlock the safe:

Proposition 1

$$\textbf{Know}(\neg\textsc{Exploded}, S_0) \supset \textbf{Can}(\neg\textsc{Locked}, S_0).$$

The agent can use σ_0 to achieve the goal. It is easy to see from the axioms in Section 2.2 that in the sequences of situations defined by σ_0 starting at S_0 and S_0^*, the safe is eventually unlocked.

An important property of this definition of ability is that it can be used to infer when the agent is not able to achieve a goal. If we modified the example so that the agent was unable to read the combination in situation S_0 (and still did not know the combination in S_0), we could show that $\neg\textbf{Can}(\neg\textsc{Locked}, S_0)$. Note that it is still physically possible for the agent to unlock the safe by dialing the right combination, but since there is no way for the agent to determine the correct combination, it is not able to unlock the safe.

3 Goals

In order to determine rational courses of action it is necessary to first specify the goals of an agent, since one of the requirements of a rational action is that it brings the agent closer to achieving its goals. Following (Cohen & Levesque 1990), we characterize the goals of the agent by specifying the paths (sequences of situations, which we model using ASFs) in which all the goals (both maintenance goals and achievement goals) are achieved. The predicate $H(\sigma, s)$ is used to denote those paths that satisfy the agent's goals. For instance, we extend the safe example by specifying that H is true of those paths where eventually the safe is unlocked and intact:

$$H(\sigma, s) \equiv \\ \textbf{Eventually}(\neg\textsc{Locked} \wedge \neg\textsc{Exploded}, \sigma, s),$$

where $\textbf{Eventually}(\alpha, \sigma, s)$ means that eventually α will hold along the path defined by σ starting at s:[8]

$$\textbf{Eventually}(\alpha, \sigma, s) \stackrel{\text{def}}{=} \\ \exists s^*(\textbf{OnPath}(\sigma, s, s^*) \wedge \alpha(\sigma, s^*)).$$

The path segments in Figure 1 that satisfy H are indicated by a check mark.

Given a specification for H, we can formally state what we mean by a goal. As noted in (Konolige & Pollack 1993), there are some difficulties with using Cohen and Levesque's definition of a goal. They define a goal to be any formula that is true in all goal paths. Suppose that $\alpha \wedge \beta$ is a goal. According to Cohen and Levesque's definition, both α and β are also goals. Therefore, we could imagine a rational agent working to achieve one of the conjuncts as a subgoal. But it is easy to think of circumstances where achieving only one component of a conjunctive goal is undesirable. As Konolige and Pollack did for their intention modality (I), we consider α's that are the agent's "only goals", i.e., α's that are true in all and only the H-paths:

$$\textbf{OGoal}(\alpha, s) \stackrel{\text{def}}{=} \\ \forall\sigma\forall s'[K(s', s) \supset (H(\sigma, s') \equiv \alpha(\sigma, s'))].$$

In the safe example—using the specification of H given at the beginning of this section—the agent initially has the goal to unlock the safe without having it explode:

Proposition 2

$$\textbf{OGoal}(\textbf{Eventually}(\neg\textsc{Locked} \wedge \neg\textsc{Exploded}), s).$$

Notice that for a formula to be an **OGoal**, it only has to be equivalent to H over all paths that start in an alternative situation. Therefore, the goals of an agent can change when the agent acquires knowledge. In Figure 1, the H-paths are the ones in which

[8] We use α to denote a formula with two placeholders sit and asf. Again, we suppress the placeholders where possible. $\alpha(\sigma, s)$ is the formula that results from replacing sit with s and asf with σ.

the agent eventually dials the correct combination. In S_0, the agent does not have the goal to eventually dial 0 because the agent dials 1 in the H-paths that start in the alternative situation S_0^*. However, in $do(\textsc{readCombOfSafe}, S_0)$, the agent has the goal to eventually dial 0 since in all paths that include this situation (and there are no alternative ones), the agent eventually dials 0 iff the path is in H.

4 Rational Action

A definition of the goals of the agent is only useful if the goals somehow constrain the agent's future actions. One way of enforcing such a constraint is to bring in a notion of acting rationally. If an agent is acting rationally, then to the best of its ability it is acting to bring about (and maintain) its goals. In other words, if an agent's actions are rational starting in situation s, then ideally it is following an ASF σ such that $H(\sigma, s)$. But since the agent may be uncertain as to which situation the world is actually in, it ought to be following a course of action that it *knows* will achieve its goals, i.e., $\forall s' K(s', s) \supset H(\sigma, s')$. However, there may not always be such a σ for the agent to follow. We require instead that a rational agent follow a σ such that $H(\sigma, s')$ in a maximal set of s' such that $K(s', s)$. To that end, we define an ordering over ASFs for each situation:

$$\succeq(\sigma_1, \sigma_2, s) \stackrel{\text{def}}{=} \forall s' K(s', s) \wedge H(\sigma_2, s') \supset H(\sigma_1, s').$$

In other words, in situation s, σ_1 is as good as σ_2 (with respect to $\succeq$) iff σ_1 achieves the goals of the agent in all the alternative situations in which σ_2 achieves its goals.

We say that an ASF σ describes a rational course of action in s iff it is maximal in $\succeq$.

$$\textbf{Rational}(\sigma, s) \stackrel{\text{def}}{=} \forall \sigma' (\succeq(\sigma', \sigma, s) \supset \succeq(\sigma, \sigma', s)).$$

The way these notions are defined ensures that the following important principle holds: if an agent has the ability to achieve ϕ, has ϕ as an achievement goal, and is acting rationally, then eventually ϕ will hold.

Theorem 1

$$\begin{array}{l}\forall\sigma\forall s(\textbf{Can}(\phi, s) \wedge \textbf{OGoal}(\textbf{Eventually}(\phi), s) \wedge \\ \quad \textbf{Rational}(\sigma, s) \supset \textbf{Eventually}(\phi, \sigma, s))\end{array}$$

This theorem characterizes the main role that rational action plays in our theory. It connects the knowledge and goals of an agent with its future actions, by guaranteeing that when the agent has sufficient knowledge to bring about an achievement goal, eventually the agent will achieve it.

With this definition of rational action, we can show that if the agent does not know the combination, and it knows that the safe is locked and intact in situation s, then it is rational for the agent to read the combination of the safe and dial it. Let σ_1 be an ASF that prescribes reading the combination initially and then dialing the combination. For any situation s' such that $K(s', s)$, it is easy to see that in the path defined by σ_1 starting at s', eventually the safe will be unlocked and intact. Therefore, $H(\sigma_1, s')$ for any s' such that $K(s', s)$. It follows that for any σ^*, $\succeq(\sigma_1, \sigma^*, s)$. We conclude that $\textbf{Rational}(\sigma_1, s)$.

On the other hand, we can show that it is irrational for the agent to dial a combination without first reading the combination of the safe in s. Let c be a combination and σ_2 be an ASF that maps any situation s'' such that $K(s'', s)$ to the action $\textsc{dial}(c)$. Since the agent does not know the combination in s, there will be a situation s^* such that $K(s^*, s)$ and $\textsc{combOfSafe}(s^*) \neq c$, and therefore $\textsc{Exploded}(do(\textsc{dial}(c), s^*))$. Since the safe always remains exploded once it happens, we can infer that $\neg H(\sigma_2, s^*)$. We showed that $H(\sigma_1, s')$ for any s' such that $K(s', s)$, therefore $\neg \succeq(\sigma_2, \sigma_1, s)$. Since we showed that $\succeq(\sigma_1, \sigma^*, s)$ for any σ^*, it follows that $\neg\textbf{Rational}(\sigma_2, s)$.

Consider a modification to the safe example in which the combination of the safe may be illegible. If the combination is legible, then the agent knows the combination of the safe after reading it, as before. But if the combination of the safe is not legible, then it is not possible for the agent to read the combination. We add a knowledge-producing action $\textsc{senseCombLegible}$, which tells the agent whether the combination is legible. In this case, we can show that it is still rational for the agent to read the combination and dial it should the combination be legible. If the combination is not legible, then the agent can only guess the combination (i.e., dial any combination), and it is rational to do so.

5 Related and Future Work

The framework described above is overly simplistic in various ways. Work is underway to make the paradigm more realistic. In section 3, we saw that it was undesirable to have as goals all the formulae that are true in all goal paths. However, some of these formulae are legitimate subgoals and we should have a way of specifying them. This can be accomplished by altering the framework to allow subgoals to be explicitly defined. The new framework allows incremental specification of subgoals instead of requiring that all the goals of the agent be defined in a single predicate (H).

We saw at the end of Section 4 that it could sometimes be rational for the agent to guess the combination. Although this allows the agent to unlock the safe in some possible worlds, in most, the safe will explode. This might be an overly precarious strategy for the agent to take. The situation can be remedied by making a maintenance goal of never exploding the safe a higher priority goal than the achievement goal of unlocking the safe. Then it will be rational for the agent to dial a combination only if it knows the combination of the safe. We have been investigating the addition of prioritized goals to the framework.

In addition to having an ordering on goals, it would be useful to be able to characterize the plausibility of the alternative situations. One way of doing this would be to have a plausibility ordering on the alternative situations. Then the agent can consider tradeoffs between the priority of goals in a given situation and the likelihood of that situation occurring.

Finally, the current framework only allows for a single agent in the world. We are currently extending the framework to model multiple, interacting agents.

References

Cohen, P. R., and Levesque, H. J. 1990. Intention is choice with commitment. *Artificial Intelligence* 42:213–261.

Haas, A. R. 1987. The case for domain-specific frame axioms. In Brown, F., ed., *The Frame Problem in Artificial Intelligence: Proceedings of the 1987 Workshop*, 343–348. Lawrence, KA: Morgan Kaufmann Publishing.

Konolige, K., and Pollack, M. E. 1993. A representationalist theory of intention. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, 390–395.

Lespérance, Y.; Levesque, H. J.; Lin, F.; Marcu, D.; Reiter, R.; and Scherl, R. B. 1994. A logical approach to high-level robot programming – a progress report. In Kuipers, B., ed., *Control of the Physical World by Intelligent Agents, Papers from the 1994 AAAI Fall Symposium*, 109–119.

Lespérance, Y.; Levesque, H. J.; Lin, F.; Marcu, D.; Reiter, R.; and Scherl, R. B. 1995a. Foundations of a logical approach to agent programming. In *IJCAI-95 Workshop on Agent Theories, Architectures, and Languages.* To appear.

Lespérance, Y.; Levesque, H. J.; Lin, F.; and Scherl, R. B. 1995b. Ability and knowing how in the situation calculus. In preparation.

Lin, F., and Reiter, R. 1994. State constraints revisited. *Journal of Logic and Computation* 4(5):655–678.

McCarthy, J., and Hayes, P. 1979. Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., and Michie, D., eds., *Machine Intelligence*, volume 4. Edinburgh, UK: Edinburgh University Press. 463–502.

Moore, R. C. 1985. A formal theory of knowledge and action. In Hobbs, J. R., and Moore, R. C., eds., *Formal Theories of the Common Sense World.* Norwood, NJ: Ablex Publishing. 319–358.

Pednault, E. P. D. 1989. ADL: Exploring the middle ground between STRIPS and the situation calculus. In Brachman, R.; Levesque, H.; and Reiter, R., eds., *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, 324–332. Toronto, ON: Morgan Kaufmann Publishing.

Reiter, R. 1991. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In Lifschitz, V., ed., *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy.* San Diego, CA: Academic Press. 359–380.

Scherl, R. B., and Levesque, H. J. 1993. The frame problem and knowledge-producing actions. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 689–695. Washington, DC: AAAI Press/The MIT Press.

Schubert, L. 1990. Monotonic solution to the frame problem in the situation calculus: An efficient method for worlds with fully specified actions. In Kyberg, H.; Loui, R.; and Carlson, G., eds., *Knowledge Representation and Defeasible Reasoning.* Boston, MA: Kluwer Academic Press. 23–67.