A Decision-Theoretic Abductive Basis for Planning*

David Poole[†] and Keiji Kanazawa
Department of Computer Science
University of British Columbia
2366 Main Mall
Vancouver, B.C., Canada V6T 1Z4
{poole,kanazawa}@cs.ubc.ca

Abstract

This paper presents a coherent synthesis of logic and decision theory and shows how it can be used. We allow an axiomatization of the world in definite clauses from a set of assumables. These assumables are partitioned into the set of controllable assumables and uncontrollable assumables. The uncontrollable assumables have probabilities associated with them. The logic allows multiple concurrent actions and lets us predict the effects for both the uncontrolled and controlled cases. We show an example of its use and argue that abduction, particularly probabilistic abduction, lays an important groundwork for decision theoretic and probabilistic planning. The main empirical claim is that uncertainty and choices can be represented as independent exogenous events using logic to give the consequences of the events, resulting in a powerful and yet simple representation.

1 Introduction

We propose *decision-theoretic abduction* as a general framework for studying planning. Abduction can simplify the representation of plans and provides an alternative to deduction as a framework for understanding planning. Abductive planning, or assumption-based planning, has a number of other advantages, such as simple semantics, easy implementation, and straightforward extensions to handle probability and decision theory. In the rest of this abstract, we consider probabilistic abduction as applied to planning.

In moving planning into real world applications, issues of uncertainty become increasingly important. In the real world, typical assumptions of complete knowledge and deterministic change, including deterministic effects of actions, are inappropriate and inadequate. To this end, researchers have proposed a variety of methods for handling uncertainty, such as probability and decision theory, in the decision-making process [18, 28, 4, 13, 6, 17, 3].

At the same time, in addressing issues of diagnosis, or explanation in general, researchers also find that uncertainty must be addressed. In determining the cause of an effect, for instance the disease causing some symptom in medical diagnosis, it is rarely possible to determine with certainty the disease from among a number of possible candidates. Researchers have proposed various methods of abduction with probability to address diagnosis with uncertainty (e.g., [11]).

Our aim is to develop a coherent framework that incorporates both planning and diagnosis. Our framework for temporal projection is implemented, and has been pretty well tested for the axiomatization of Section 4. In this paper, we describe the basic representational methodology underlying our research strategy.

2 Review

Abduction can be described as follows. Given a logical theory (the "facts") and a set of assumables, an *explanation* of a goal is a subset of the assumables that is consistent with the facts and which together with the facts logically implies the goal.

2.1 Planning as Abduction

First, we review ideas behind the use of abduction in planning (e.g., [7]). The general idea of planning as abduction is as follows. We assume that we have a logical theory and a set of controllable propositions (i.e., a set of propositions which the agent can set true). Given a goal G that an agent is interested in achieving, a plan

*This work was supported by Institute for Robotics and Intelligent Systems Group B5 and Natural Sciences and Engineering Research Council of Canada Operating Grants OGP0009281 (awarded to Alan Mackworth) and OGPO044121.

[†]Scholar, Canadian Institute for Advanced Research

is a set of propositions which

1. are achievable by the agent and
2. if they were true, the goal would be true.

The set of controllable propositions are typically inconsistent and contain incompatible choices (e.g., "I get up early" and "I sleep in", or "Block $b7$ is put on the table" and "Block $b7$ is thrown on the floor" and "block $b7$ is put on block $b3$"). An achievable set of propositions is a subset of the controllable propositions that is consistent with the logical theory.

There may be a cost or utility associated with a plan reflecting how good it is, but that under this approach all plans are guaranteed to achieve the goal (the goal logically follows from the achievable propositions being true).

2.2 Recognition as Abduction

A more traditional view of abduction [24] is where the assumables are assumptions about what may be true, and the goal is an observation. If the facts are causal rules, then an explanation is a logical theory about what could have caused the observation.

One interesting instance of this is probabilistic Horn abduction [25], where there are probabilities over the assumables. Assumables are grouped into disjoint sets corresponding to random variables. The random variables are probabilistically unconditionally independent (i.e., if α and β are assumables and they are not in the same disjoint set then $p(\alpha \wedge \beta) = p(\alpha) \times p(\beta)$). The logic is definite Horn clauses with no assumables at the heads of clauses (this means that the logic cannot impose any dependencies on the assumables). There is an isomorphism [25] between the propositional version of probabilistic Horn abduction and Bayesian networks [21]. We can represent arbitrary dependence by inventing hypotheses.

2.3 Game/Decision Theory

Decision theory and its multi-agent counterpart, game theory [27, 9], are normative theories of reasoning under uncertainty. The framework below should be seen as representation based on the normalized [27] (or strategic [9]) form of a game, with possible world corresponding to a complete play of a game. What we have added here is that a logic program can be used to give the consequences of the play. This allows us to use a logical representation to represent the world, in particular the consequences of choices. The abductive view means that we only have to consider the relevant parts of the strategies at all times. Below we have only given the one player with nature version of the strategic form, but it can be extended in a reasonably straight forward way to to the multi-agent game theory case. See [23] for details.

3 Decision Theoretic Abductive Planning

The framework we are proposing [23] is one where there are three sets of formulae, namely a set of controllable propositions which the agent can set, a set of uncontrollable propositions with probabilities over them, and a set of facts (definite Horn clauses) which tell us the consequences of these propositions.

Definition 3.1 A decision theoretic abductive planning theory is a triple $\langle R, C, F\rangle$, where

R is a set of sets of ground atomic formulae. An element of R is called a *random variable.* An atomic formula can only appear in only one random variable. There is a probability assigned to each element of each random variable, where sum the probabilities of the elements in each random variable is one.

C is a subset of R. An element of C is called a *controllable set.* The aim is that the agent can control the values of the random variables in C. The probabilities of elements of controllable sets are optional, and if they are present, it means that the agent does not have to control the variable. The probabilities are for the case where the agent does not set the values of the variable.

F is an acyclic [1] set of definite Horn clauses, such that no member of a random variable is at the head of any clause.

Note that when C is empty, then this exactly corresponds to a probabilistic Horn abduction theory.

The semantics is defined in terms of possible worlds. There is a possible world for each selection of one element from each random variable. The atoms which logically follow from F are also true in this possible world.

Formally, the *base* of a possible world w is a set of variables that appear in some random variable, such that for every V in R there is exactly one element of V in the base of w. Two worlds are the same if they have the same base. Atom a is true in world w if it can be derived (using SLDNF) from $F \cup base(w)$, and is false otherwise (in other words w is the stable model [10][1] of $F \cup base(w)$; a stable model semantics enables the use of negation as failure in the theory). The measure of w is the product[2] of the probabilities associated with the elements of the base of w.

[1] As $F \cup base(w)$ is acyclic there is a unique stable model and it can be computed using SLD resolution with negation as failure [1].

[2] When R is infinite, we need a more sophisticated measure as in [25]. That will only complicate this paper, however.

It is easy to see that when we choose an element of each controllable set (making that element have probability one and the other elements of that set have probability zero), this measure over possible worlds satisfies the axioms of probability (as the possible worlds are disjoint and the sum of the probabilities of the possible worlds is one) and so imposes a conditional probability over our language.

We can see the relationship to abduction because to determine the probability of any g we need only consider the minimal explanations of g (with elements of R as assumables). Moreover, if we set up the rules so that the bodies for each rule are disjoint (i.e., if $a \leftarrow b_1$ and $a \leftarrow b_2$ are two rules for a, then there is no possible world in which b_1 and b_2 is true), then the probability of g can be obtained by adding the probabilities of the explanations of g. The probability of an explanation can be obtained by multiplying the probabilities of the elements of the explanation. The rules in our example below are designed so that the rules are disjoint so that the probabilities of the explanations can be added. See [25] for details.

We assume that some of the propositions that are concluded by R are of the form $utility(U)$ meaning that U is a number representing the utility. We assume that R and F are such that there is at most one U such that $utility(U)$ is true in any world. As a convention we assume that there is a base utility of 0 that is the utility of the worlds where $utility(U)$ is not defined for any U (this allows us to have rule bases that are not complete with respect to utility).

A **plan** is a set of elements of controllable sets such that there is at most one element from each controllable set in a plan. The expected utility of plan P is defined by

$$EU(P) = \sum_U p(utility(U)|P) \times U.$$

The aim is to select a plan which maximizes the expected utility.

Note that if we have a goal G, whose probability we want to maximize, we add the clause $utility(1) \leftarrow G$. With no other clauses for utility, then the plan which maximizes the probability that G is achieved is the maximum utility plan.

4 Representing Actions in the Logic

The preceding section presented a semantic framework that is independent of the representation of the actions. In this section we present a temporal representation that can allow for multiple concurrent actions where actions can be seen as controllable propositions.

We adopt a nonreified temporal logic [2][3], with a discrete temporal structure. What is important is the use of propositions parameterized by temporal terms rather than, for example, the discreteness of time. Note that under this representation, the notion of a possible world in the above semantics corresponds to a complete temporal evolution of the world (i.e., it says what happens at times 0, 1, 2 etc).

For expository reasons we develop the representation for an example domain. The domain that we present is a variant of the blocks world that we call the "pesky blocks world". In the pesky blocks world, as in the real world, sensors can be faulty and effectors can make mistakes. Executing actions in the pesky blocks world may result in their intended effects only some of the time. For example, if b is on c in situation 0 (represented $on(b, c, 0)$), and we put b on a in situation 0, (represented $puton(b, a, 0)$), the intended effect, namely $on(b, a, 0 + 1)$, may not be true in all situations. Possible reasons may be that the effector fails to pick up b, drops b, or misses a and ends up toppling over the whole tower that a is in. Thus, there may be uncertainty about what holds true in the state after executing an action, and there is a tradeoff between blindly following a plan and looking at the world to see (or infer) what the true state of the world is. In our framework, we represent an action and its possible effects in terms of hypotheses about what actually happens. A probability distribution over the hypotheses encodes our uncertainty about the actual effects of an action.

Our action ontology is more general than that of STRIPS planners [8]. There are no action preconditions per se in the pesky blocks world. Any action may be attempted at any time, but if the necessary conditions for their intended effects do not hold, then the effects may be completely different. For example, if a target block a is not clear in situation 0 (there is something on top of it), then attempting $puton(b, a, 0)$ may result, among other possibilities, in toppling over the whole tower that a is a part of.

Let us now present in more detail part of our domain theory for the pesky blocks world. The decision theoretic abductive planning language is an extension of pure Prolog with facilities for representing controllable actions and probabilistic hypotheses. The statements in the language are partitioned into random variables R, controllables C, and facts F, as before. Let us first describe controllables.

The controllables are action attempts. These are grouped into disjoint and covering sets, reflecting what we can "logically" attempt (e.g., it doesn't make sense to attempt to put block a both directly on top of block b and directly on top of block c). In the pesky blocks

[3]We do not explicitly adopt a 2-sorted logic, as the only times that will arise are $0, 1, 2, \ldots$. Because our implementation does not use an equality theory, we will just use the times $0, 0+1, 0+1+1, \ldots$.

world, the controllables set C is

$$\{\{puton(X,Z,S), careful(X,Z,S), nothing(X,S) : Z \text{ is a block }\} : X \text{ is a block }, S \text{ is a state }\}.$$

with the following intended interpretations:

$puton(X,Z,S)$: Attempt to put block X on Z in situation S. This is the standard blocks world "puton" action, except that as noted, there are no preconditions in the classical sense, and it may or may not have the desired effect.

$careful(X,Z,S)$: Carefully put block X on Z in situation S. The *careful* action is exactly like *puton* except that it is less prone to error and it takes a longer time to execute.

$nothing(X,S)$: Do nothing to X in situation S.

Random variables model possible states of affairs in the world over which we have no control. These represent the choices that nature makes about how things turn out. A random variable statement has the form

$$random([a_1(\overline{V}) : p_1, a_2(\overline{V}) : p_2, \ldots, a_n(\overline{V}) : p_n])$$

which means that for every tuple $\overline{t}$ of ground terms in our language, $\{a_1(\overline{t}), \ldots, a_n(\overline{t})\} \in R$ and thus the $a_i(\overline{t})$ represent n possible disjoint and covering hypotheses about some possible state of the world , and the p_k represent the prior probability of $a_k(\overline{t})$. The p_k must sum to 1.

As an example, when a $puton(X,Y,S)$ action is executed, it may "succeed" (that is, its intended effects ensue), it may result in X dropping on the table, or it may result in the effector toppling over the whole tower that Y is on. Thus, there are three possible outcomes, which are modeled as follows:

$$\begin{aligned} random([&puton\_success(X,S) : 0.8, \\ &puton\_drop(X,S) : 0.15, \\ &puton\_topple(X,S) : 0.05]). \end{aligned} \tag{1}$$

This says, for example, that the prior probability of success for a puton action is 0.8.

Note that the form of the random variables gives us a form of integrity constraints [15]. For example, in (1), the success of a *puton* action prevents the occurrence of dropping or toppling. The possible outcomes cannot all take place at the same time.

Note that in the semantics, there is a possible world in which $nothing(a,1)$ and $puton\_success(a,1)$ are both true. In other words we are doing nothing with block a at time 1, but if we did put a on some block it would succeed. While this may seem strange, it is important to allow if we want to have independent events. It is also very important when we want to have conditional plans that are adopted by agents without knowing what other agents are doing (here we can see nature as adopting a conditional plan to make the block succeed if it is being put on something — see Section 6 and [23]). While this is true in the semantics, the abductive view lets us consistently ignore whether $puton\_success(a,1)$ is true unless we actually attempt to put a on some block. Thus when reasoning we do not have to consider these incompatible alternatives.

Rules describe the effects of assumptions over the controllables and random variables. Each rule is a definite Horn clause of form

$$H \leftarrow B_1, \ldots, B_n.$$

or

$$H \leftarrow true.$$

The latter is used to describe a rule with an empty body.

Let us now illustrate our framework by presenting some axioms describing the pesky blocks world. First, let us examine the case when a *puton* action is executed. What we wish to know, of course, is the state of the world after the action is executed. The propositions (fluents) of interest are *on* and *clear*[4] ; what blocks are on top of what blocks, and which blocks have nothing on top of them.

If puton is attempted, *on* relations are inferred on the basis of axioms such as:

$$\begin{aligned} on(X,Z,S+1) \leftarrow \; & \\ & puton(X,Z,S) \\ & \wedge puton\_success(X,S) \\ & \wedge puton\_preconds(X,Z,S). \\ on(W,table,S+1) \leftarrow \; & \\ & puton(X,Z,S) \\ & \wedge puton\_topple(X,S) \\ & \wedge puton\_preconds(X,Z,S) \\ & \wedge below(W,Z,S). \\ on(X,table,S+1) \leftarrow \; & \\ & puton(X,Z,S) \wedge puton\_topple(X,S) \\ & \wedge puton\_preconds(X,Z,S). \end{aligned}$$

This axiom states that block X is on block Z in $S+1$, provided that X was attempted to be put on Z in S, the puton succeeded and the preconditions for the puton were true in state S.

Although there are no preconditions required for an attempt of a *puton* action, there are conditions that affect whether or not a *puton* attempt is likely to succeed. We represent these conditions through

[4]As we are using negation as failure we can axiomatize *clear* as

$$\begin{aligned} &clear(X,S) \leftarrow\sim on(Z,X,S) \wedge X \neq table. \\ &clear(table,S) \leftarrow true. \end{aligned}$$

where '$\sim$' means negation as failure [23].

the proposition *puton_preconds*(X, Z, S) that is true whenever the necessary (but not sufficient) conditions for a successful *puton*(X, Z, S) hold true. It can be defined this way:

$$\begin{aligned}
&puton_preconds(X, Z, S) \leftarrow \\
&\quad clear(X, S) \wedge \\
&\quad clear(Z, S) \wedge \\
&\quad X \neq Z \wedge \\
&\quad X \neq table.
\end{aligned}$$

Other *on* relations that are caused by a puton attempt can occur because putting on topples the target tower:

$$\begin{aligned}
&on(W, table, S+1) \leftarrow \\
&\quad puton(X, Z, S) \\
&\quad \wedge puton_topple(X, S) \\
&\quad \wedge puton_preconds(X, Z, S) \\
&\quad \wedge below(W, Z, S). \\
&on(X, table, S+1) \leftarrow \\
&\quad puton(X, Z, S) \wedge puton_topple(X, S) \\
&\quad \wedge puton_preconds(X, Z, S).
\end{aligned}$$

In the case where a *puton* results in dropping the block, the effects are the same as for the case where the action succeeds except that the block ends up on the table rather than on its destination (but only if the destination was a block to begin with):

$$\begin{aligned}
&on(X, table, S+1) \leftarrow \\
&\quad puton(X, Z, S) \wedge puton_drop(X, S) \\
&\quad \wedge puton_preconds(X, Z, S).
\end{aligned}$$

We have now described the axioms for the case where *puton* is attempted and the preconditions are satisfied. Similar axioms can be used for the case when the preconditions do not hold.

The frame axioms can use the notion of clipping:

$$\begin{aligned}
&on(X, Y, S+1) \leftarrow \\
&\quad on(X, Y, S) \wedge \\
&\quad \sim clipped_on(X, Y, S).
\end{aligned}$$

Where clipping is defined as:

$$\begin{aligned}
&clipped_on(X, Y, S) \leftarrow \\
&\quad puton(X, Y, S) \\
&\quad \wedge puton_preconds(X, Y, S). \\
&clipped_on(A, B, S) \leftarrow \\
&\quad puton(X, Z, S) \\
&\quad \wedge puton_preconds(X, Z, S) \\
&\quad \wedge puton_topple(X, S) \\
&\quad \wedge below(A, Z, S).
\end{aligned}$$

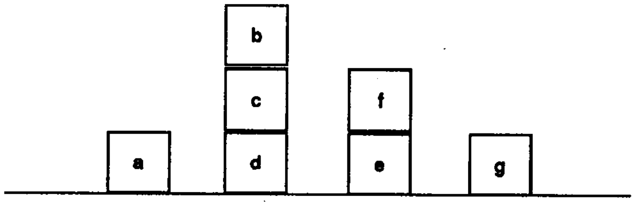

Figure 1: A blocks world situation.

In other words, $on(X, Y)$ is are no longer true are when we have tried to put X on Y or when X is below a block that is toppled.

We also have to consider the case where the preconditions of the actions do not hold. These can be done in an analogous, way. For example, suppose that if the preconditions are false, then either nothing happens, or the whole tower above X is toppled, and those below X are toppled:

$$\begin{aligned}
&random([puton_same(X, S) : 0.8, \\
&\quad puton_messup(X, S) : 0.2]).
\end{aligned}$$

One axiom for messup is the following:

$$\begin{aligned}
&on(A, table, S+1) \leftarrow \\
&\quad puton(X, Z, S) \\
&\quad \wedge \sim puton_preconds(X, Z, S) \\
&\quad \wedge puton_messup(X, S) \\
&\quad \wedge (below(A, X, S) \\
&\quad \vee above(A, B, X, S)).
\end{aligned}$$

In addition to effect axioms, we need to state the initial conditions and desired conditions. For example, these describe the initial situation in figure 1.

$$\begin{aligned}
&on(a, table, 0) \leftarrow true. \\
&on(b, c, 0) \leftarrow true. \\
&on(c, d, 0) \leftarrow true. \\
&on(d, table, 0) \leftarrow true. \\
&clear(a, 0) \leftarrow true. \\
&clear(b, 0) \leftarrow true. \\
&on(e, table, 0) \leftarrow true. \\
&on(f, e, 0) \leftarrow true. \\
&on(g, table, 0) \leftarrow true. \\
&clear(f, 0) \leftarrow true. \\
&clear(g, 0) \leftarrow true.
\end{aligned}$$

As for desired conditions, in general, in a decision theoretic framework, there may not be a goal in the classical sense in planning, because we consider not only what state we want to achieve, but the relative merits of the states that we may get into if our actions don't achieve the goal.

Suppose we have the case that it is best if the goal to be achieved after 4 steps, but if another goal is achieved after 5 steps it is only half as good.

$$goal(S) \leftarrow on(d,c,S) \wedge on(c,b,S)$$
$$\wedge on(b,a,S) \wedge on(a,table,S).$$
$$alt\_goal(S) \leftarrow on(d,b,S) \wedge on(b,table,S).$$
$$utility(10) \leftarrow goal(S) \wedge S \leq 4.$$
$$utility(5) \leftarrow alt\_goal(S) \wedge S \leq 5.$$

Once we have a theory, we can project the consequences of actions (and inaction) using assumption-based reasoning. Our current implementation is a modified version of the probabilistic Horn abduction interpreter [25] with the addition of controllable actions and negation as failure. This interpreter uses logic programming technology to build up proofs of states of the world on the basis of assumptions.

Given our theory, we can, for example, find out the likelihood of success of doing $puton(b,a,0)$. That is, under what assumptions might $on(b,a,0+1)$ be true given $puton(b,a,0)$? Our interpreter finds one explanation $puton\_success(b,0)$, with prior probability 0.8. Thus the probability of $on(b,a,0+1)$ is 0.8. What about the likelihood that b ends up on the table? There are two explanations for $on(b,table,0+1)$, namely $puton\_topple(b,0)$ and $puton\_drop(b,0)$, with prior probability of 0.05 and 0.15, respectively. The probability of b being on the table is the sum of all of its explanations, or 0.2. In a similar fashion, we can also find out the utility of a given plan.

5 Concurrent Actions and Exogenous Events

In our framework, we can represent concurrent actions and exogenous events in a straightforward fashion (cf. [19, 20, 22, 26]). In fact, one way to look at the effect axioms in the previous section is that an effect results from the execution of concurrent actions by an agent (through the controllables) and by nature (through the random variables). It is also possible to model concurrent actions in the sense of two agents undertaking actions at the same time, or an agent undertaking two tasks at the same time.

For example, there is nothing to prevent attempting $puton(b,a,0)$ and $puton(g,f,0)$ at the same time. In fact, this is the reason why the effect axioms for *puton* do not include frame axioms about what does not change as a result of a *puton*. We cannot do that if different *puton* actions could be taking place concurrently.

In the framework that we have presented in this paper, an agent has to make a commitment with regard to all controllables. The agent has to decide for each block whether to move the block or whether to do nothing with the block for each situation. An alternative (which we have also implemented) is to have each controllable as a random variable that may have a default value (or prior distribution) that takes over if no no choice is made for the controllable by a decision maker.

6 Conditional Plans and Influence Diagrams

In this section we show how conditional plans can be embedded within our framework, essentially without adding anything to the basic framework. This will be done first in the context of influence diagrams [14], and then we will show how we can go beyond influence diagrams.

Whereas most work in designing decision theoretic plans has considered what is effectively the extensive form of a game, where the output is in terms of if-then-else rules, we will adopt the form of the strategic form, where a plan (strategy) is an adoption of a function from observables (sensor values and/or partial memory of state) to actions (actuator values) [9]. It turns out that strategic games are more powerful: the strategic form can easily handle the case where there is not total recall (the agent doesn't necessarily remember everything it has chosen), as is the case for most (real and artificial) agents.

We here show how to represent influence diagrams; we extend the representation of Bayesian networks in probabilistic Horn abduction [25] to show to represent decision nodes. Value nodes are just represented as definite Horn clauses that imply a particular utility. Suppose we have decision d with parents $p_1, \cdots, p_m$ (these are either decision nodes or random nodes), then we have to adopt a policy of what for each of the values of $p_1, \cdots, p_m$. Suppose each of the variables are binary with values *yes* and *no*, and we write $d_i = yes$ to mean the decision/action is yes and $p_j = no$ to mean the value of variable p_j is no, etc. This decision can be represented using the single rule

$$d_i = V \leftarrow p_1 = V_1 \wedge \cdots \wedge p_m = V_m \wedge s_{d_i}(V, V_1, \cdots, V_m)$$

With $\{s_{d_i}(yes, V_1, \cdots, V_m), s_{d_i}(no, V_1, \cdots, V_m)\}$ being in C for each value of $V_1, \cdots, V_m$. Thus the agent is to chooses one action for each value of the parents.

Influence diagrams represent one very limited case of our representation; one where we know what decisions will be made at each step and what information will be available, and where the agent remembers all previous observations and decisions. Each decision must encode the stage in the plan where it is; the agent must be able to step through the decisions in the right order (as opposed to the agent being forgetful or even purely reactive). It also does not allow for concurrent actions (these must be coalesced into a large composite decision which offers as alternatives all combinations of all possible actions and non actions).

7 Agents with Limited Abilities

The representation of actions we gave in Section 4 assumes that the agent cannot look at the world, but can remember what step they are in the plan. We can represent programs for various limitations of the capability of agents. For example, a program for purely reactive agent that has no memory and can only react to the values of sensors can be represented in the form

$$do(X,S) \leftarrow val(sensor_1, V_1, S) \wedge \cdots$$
$$\wedge val(sensor_m, V_m, S) \wedge cond\_do(X, V_1, \cdots, V_m)$$

with $\{cond\_do(X, V_1, \cdots, V_m) : X \text{ is possible action }\}$ is in C for every value of $V_1, \cdots, V_m$. The optimal plan corresponds to the optimal purely reactive (stateless) agent with the m sensors given. We would also need axioms stating how the sensor values depend on the state (i.e., rules with heads unifying with $val(sensor_i, V_i, S)$). Again, a proper influence diagram cannot represent this reactive program because a proper influence diagram enforces the *no-forgetting* constraint wherein an agent must remember its past decisions.

For example, following from the example of Section 4, suppose the agent has two sensors one can sense how many blocks are on the table and one can sense how tall the tallest tower is, but the cannot remember what step they are at. This can be represented using rules such as:

$$puton(X,Z,S) \leftarrow num\_ontable(N,S)$$
$$\wedge tallest\_tower(H) \wedge cond\_puton(X,Z,N,H)$$

and similar rules for *careful* and *nothing*. C would contain,
not instances of *puton*, but $\{cond\_puton(X,Z,N,H),$ $cond\_puton(X,Z,N,H),$ $cond\_puton(X,Z,N,H) : Z$ is a block$\}$ for each value of X, N and H. We would also axiomatize $num\_ontable(N,S)$ and $tallest\_tower(H)$, which are straight forward to define in terms of *on*. Note that these relations would define the output of the sensors; these sensors could be faulty. This again is easy to add: we add random noise in the definitions of these relations.

The optimal plan here will be the optimal purely reactive agent with only these two sensors.

8 Related Work

[7, 5] provide abductive characterizations of temporal projection and planning in the event calculus [16]. Our work contrasts with that work foremost by our interest in the representation of uncertainty and preference for use as a decision-theoretic abductive basis for planning.

Haddawy [12] presents a logic for the representation of actions and utilities that is considerably richer than ours. Our research strategy should be seen as one of designing minimalist representations that embody pragmatic compromises between heuristic and epistemological adequacy and testing them by axiomatizing domains and building implementations.

9 Conclusions

The work outlined here is part of an ongoing research project by the authors and others. We have only sketched some ideas here. There is plenty of work to be done in this area but, we believe, also plenty of potential.

We have presented a theory that enables us to make temporal projections abductively. This tells us the consequences of actions that we may perform, including the utility of planning. We have concentrated on providing a specification of what an optimal plan is rather than how one should be computed.

The combination of a Horn causal theory with abducible actions and events, and probabilities over the assumptions provides a simple and, we argue, practical representation that easily handles concurrent events including actions, and that can be extended with utilities to yield a decision theoretic plan representation.

In our framework, when we make an observation, we essentially remove from consideration all possible worlds that are incompatible with the observation. This corresponds to abducing to the causes: the set of minimal explanations of the observation is a succinct description of the set of possible worlds consistent with the observation (the consistent possible worlds are exactly those that extend the minimal explanations of the goal) [24].

Finally, we must be careful to note that adopting an abductive framework for planning is principally a representational commitment. It does not in itself solve many of the classical planning issues such as search. This is one of the (many) problems we have to solve to make this the representation of choice for planning.

References

[1] Krzysztof R. Apt and Martin Bezem. Acyclic programs. *New Generation Computing*, 9(3-4):335–363, 1991.

[2] Fahiem Bacchus, Josh Tenenberg, and Johannes A. Koomen. A non-reified temporal logic. *Artificial Intelligence*, 52(1):87–108, 1991.

[3] Thomas Dean, Leslie Pack Kaelbling, Jak Kirman, and Ann Nicholson. Planning with deadlines in stochastic domains. In *Proceedings of the*

Eleventh National Conference on Artificial Intelligence, pages 574–579. AAAI, 1993.

[4] Thomas Dean and Keiji Kanazawa. Persistence and probabilistic projection. *IEEE Transactions on Systems, Man and Cybernetics*, 19(3):574–585, May/June 1989.

[5] Marc Denecker, Lode Missiaen, and Maurice Bruynooghe. Temporal reasoning with abductive event calculus. In *Proceedings of the TenthEuropean Conference on Artificial Intelligence*, pages 384–388, Vienna, Austria, 1992. ECAI.

[6] Mark Drummond and John Bresina. Anytime synthetic projection: Maximizing the probability of goal satisfaction. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 138–144, Boston, Masssachusetts, 1990. AAAI.

[7] Kave Eshghi. Abductive planning with event calculus. In *Proceedings of the Fifth International Conference on Logic Programming*, pages 562–, 1988.

[8] Richard E. Fikes and Nils J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[9] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, Cambridge Massachusetts, 1992.

[10] M. Gelfond and V Lifschitz. The stable model semantics for logic programming. In R. Kowalski and K. Bowen, editors, *Proceedings of the Fifth Logic Programming Symposium*, pages 1070–1080, Cambridge, Mass., 1988.

[11] Robert Goldman. *A Probabilistic Approach to Language Understanding*. PhD thesis, Department of Computer Science, Brown University, 1990. Available as Technical Report CS-TR-90-34.

[12] Peter Haddawy. *Representing Plans Under Uncertainty: A Logic of Time, Chance, and Action*. PhD thesis, Department of Computer Science, University of Illinois Urbana-Champaign, 1991.

[13] Steven John Hanks. *Projecting Plans for Uncertain Worlds*. PhD thesis, Yale University Department of Computer Science, January 1990.

[14] Ron A. Howard and James E. Matheson. Influence diagrams. In Ron A. Howard and James E. Matheson, editors, *The Principles and Applications of Decision Analysis*. Strategic Decisions Group, Menlo Park, CA 94025, 1984.

[15] Robert Kowalski. *A Logic for Problem Solving*. North-Holland, New York, 1979.

[16] Robert Kowalski and Marek J. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):67–95, 1986.

[17] Nicholas Kushmerick, Steve Hanks, and Daniel Weld. An algorithm for probabilistic planning. Technical Report 93, University of Washington Department of Computer Science, Seattle, Washington, 1993.

[18] Curtis P. Langlotz, Lawrence M. Fagan, Samson W. Tu, and et al. A therapy planning architecture that combines decision theory and artificial intelligence techniques. *Computers and Biomedical Research*, 1987.

[19] Fangzhen Lin and Yoav Shoham. Concurrent actions in the situation calculus. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 590–595, San Jose, California, 1992. AAAI.

[20] Leora Morgenstern and Lynn Andrea Stein. Why things go wrong: A formal theory of causal reasoning. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 518–523, Minneapolis, Minnesota, 1988. AAAI.

[21] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, Los Altos, 1988.

[22] Edwin P. D. Pednault. Formulating multiagent, dynamic-world problems in the classical planning framework. In Michael P. Georgeff and Amy P. Lansky, editors, *Reasoning about Actions and Plans: proceedings of the 1986 Workshop*, pages 47–82. Morgan Kaufmann, Timberline, Oregon, 1987.

[23] David Poole. The independent assumption framework: A game-theory logic and its application to the design of rational agents. Forthcoming.

[24] David Poole. A methodology for using a default and abductive reasoning system. *International Journal of Intelligent Systems*, 5(5):521–548, December 1990.

[25] David Poole. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, 64(1):81–129, 1993.

[26] Murray Shanahan. Prediction is deduction but explanation is abduction. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1140–1145, Detroit, Michigan, 1989. IJCAI.

[27] J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Bahavior*. Princeton University Press, Princeton, third edition, 1953.

[28] Michael P. Wellman. *Formulation of Tradeoffs in Planning Under Uncertainty*. Pitman, London, 1990.