Representation of decision-theoretic plans as sets of symbolic decision rules

Niels Peek[1]

Abstract. In recent years, it is increasingly recognised that action planning in real-world domains requires an accurate treatment of uncertainty. Partially-observable Markov decision processes and related decision-theoretic models have been found to provide powerful frameworks for studying this type of planning. Within these frameworks, plans are often expressed as trees or graphs. However, for various reasons it is often more convenient to express plans as collections of decision rules. For instance, domain experts are often able to formulate a number of reliable decision rules that could serve as a starting point in finding an optimal plan. This paper investigates the representation of decision-theoretic plans as sets of symbolic decision rules. It is shown under which conditions such plans are internally consistent, coherent, and complete.

1 INTRODUCTION

Over the last decade, planning research in AI has showed increasing attention for action planning under uncertainty using decision-theoretic principles, or *decision-theoretic planning* for short [1, 2]. Partially-observable Markov decision processes (POMDPs) are generally regarded as the most powerful formal framework for this type of planning, although other formalisms have been suggested as well (e.g. [3]). A notable field of application for decision-theoretic planning is *clinical medicine* [8, 10]: action planning under uncertainty with imperfect information is an important task for the doctor who is treating a patient over a prolonged period of time. This situation occurs for instance in intensive care units where patients are monitored from hour to hour, but also in the management of chronically ill patients who regularly visit their doctor.

Given the specification of a decision-theoretic planning problem, the computational problem is to find a *plan* that maximises the expected value of a predefined utility function. Such a plan prescribes, for any possible sequence of past actions and observations, the optimal decision (action choice); the prescribed choices are thus contingent upon what is known from the past. Popular ways to express such plans are *policy trees* and *plan graphs* [5, 6]. In these trees and graphs, the nodes are labelled with prescribed actions and the arcs represent possible observations. The plan is executed by following the arcs that correspond to the actual observations and choosing the actions along the path. Regardless of the plan representation used, however, the task of computing an optimal decision-theoretic plan is highly complex.

Plan representations such as trees and graphs have a number of drawbacks. First, they do not show how much information is actually needed to make optimal decisions. Sometimes, it is sufficient to take into account only recent observations and neglect all other information, while at other times it is necessary to consider long sequences of past actions and observations to guarantee the best choice [7]. Second, if there exist regularities across a large number of sitations ("always choose action a when you make observation φ") this will also remain obscured in a policy tree. And third, they often do not allow for easy communication with domain experts.

In domains like clinical medicine, where human experts possess specialized knowledge to solve decision problems, it is generally the case that many reliable decision rules can be formulated by these experts. An example rule could be "For this type of patient, try medical therapy, and if no improvements are seen within 3 months, submit the patient to surgery." However, these rules often cover only part of the problem domain, consisting of the frequent and easy problem cases, and not the hard and rare cases. One would like to solve decision-theoretic planning problems by supplementing a given set of such decision rules provided by domain experts, to obtain a plan that covers the entire problem domain. With the incorporated expert knowledge, it should be possible to solve problems of larger size than is currently possible.

In this paper, we investigate the representation of decision-theoretic plans as sets of symbolic decision rules. The formal framework will be multivariate POMDPs; after introducing our notations (Section 2), we define a symbolic language of planning expessions and define this type of plan representation (Section 3). Then, it is shown which types of internal inconsistency and incoherence may occur in such plans (Section 4), and under which conditions our representation formulates a complete plan (Section 5). In the discussion (Section 6), we evaluate the representation using the results that have been obtained, and give a brief comparison with related work.

2 PRELIMINARIES

2.1 Multivariate POMDPs

Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be a set of random variables that jointly describe that state of a stochastic dynamic system. We use $\mathbf{X} = X$, where $X = \{x_1, \ldots, x_m\}$ is a set of values, to denote that $\mathbf{x}_i = x_i$, $i = 1, \ldots, m$. The set X is called a *value set* of $\mathbf{X}$; the set of all value sets of $\mathbf{X}$ is written $\Omega_{\mathbf{X}}$. In a POMDP model over $\mathbf{X}$, the dynamic system described by $\mathbf{X}$ is controlled by a planning agent. To this end, he may choose, at specific time points, actions from a predefined set A. These actions induce changes to the system's state, and provide the agent with observations. At each time point, a reward is received by the planning agent, depending on the current state and action choice.

[1] Dept. of Medical Informatics, University of Amsterdam, P.O. Box 22700, 1100 DE Amsterdam, The Netherlands, e-mail: n.b.peek@amc.uva.nl

Definition 1 (POMDP model) *A* partially-observable Markov decision process model *over* $\mathbf{X}$ *is a 5-tuple* $M = (T, A, p, o, r)$*, where*

- $T = \{0, 1, \ldots, N\}$ *is a set of time points,*
- A *is a set of available* actions*,*
- $p : \Omega_{\mathbf{X}} \times A \times \Omega_{\mathbf{X}} \to [0, 1]$ *is a* transition probability function*,*
- $o : A \to 2^{\mathbf{X}}$ *is an* observation function*, and*
- $r : \Omega_{\mathbf{X}} \times A \to \mathbb{R}$ *is a* reward function*.*

The variables of the set $\mathbf{X}$ jointly describe the dynamic system that is controlled by the planning agent; the time points in T denote moments where the agent is expected to select an action. The action effects are modelled as follows. When value set $X_1 \in \Omega_{\mathbf{X}}$ characterises the system's state at time point $t \in T$, selection of action $a \in A$ will result in a transition to value set $X_2 \in \Omega_{\mathbf{X}}$ at time point $t+1$ with probability $p(X_1, a, X_2)$. Furthermore, the planning agent is able to observe the values of variables from the set $o(a) \subseteq \mathbf{X}$ at time point t.[2] And finally, he receives the reward $r(X_1, a)$. No decision is made at the final time point $t = N$; this moment is included for evaluation of the final state only. The planning agent's objective is to maximise the expected sum of rewards over all time points. As this requires anticipation on the future consequences of his actions, he must formulate a *decision-theoretic plan*. Such a plan prescribes, for any possible sequence of past actions and observations, the action choice that satisfies the maximisation criterion. The task of computing an optimal plan (usually referred to as *solving* the POMDP), is computationally prohobitive when $\Omega_{\mathbf{X}}$, T, or A is large [7].

2.2 Time and change

We now define a symbolic language from the elements of a POMDP model to describe the behaviour of system and planning agent over time. We will refer to a subset $T' \subseteq T$ of subsequent points in T as a *time segment*; we will use $[t_1, t_2]$ as a shorthand notation for the time segment $\{t \in T \mid t_1 \leq t \leq t_2\}$.

Definition 2 (Planning expression) *Let* $\mathbf{X}$ *be a set of random variables and let* $M = (T, A, p, o, r)$ *be a POMDP model over* $\mathbf{X}$*. The set* $\Phi(M)$ *of* planning expressions *for* M *is the Boolean algebra spanned by the set* $S(T) \cup D(T)$*, where*

- $S(T) = \{s_i(t) \mid \mathbf{x}_i \in \mathbf{X}, t \in T\}$ *is a set of* temporal state variables*, where* $s_i(t) \in S(T)$ *takes values from* $\Omega_{\mathbf{x}_i}$*, and*
- $D(T) = \{d(t) \mid t \in T\}$ *is a set of* decision variables*, where each variable* $d(t) \in D(T)$ *takes values from the action set* A*.*

State variables describe the values of random variables from the set $\mathbf{X}$ at different time points. For instance, the expression $s_i(t) = x$ denotes that random variable $\mathbf{x}_i$ has value x at time point t. Decision variables describe the behaviour of the planning agent at different time points. For instance, the expression $d(t) = a$ denotes that the planning agent chooses action a at time point t.

For any subset $V \subseteq S(T) \cup D(T)$, we refer to a conjunction of value assignments to the elements of V as a *configuration* of V. For instance, the conjunction $s_1(t_1) = x_1 \wedge s_2(t_2) = x_2 \wedge d(t_3) = a$ is a configuration of the set $\{s_1(t_1), s_2(t_2), d(t_3)\}$. The set of all configurations of V is denoted by C_V; note that C_V a subset of $\Phi(M)$.

[2] In the standard formulation of POMDPs, observations are always made on the same variable, but its distribution depends on states and actions. In our model formulation, the actions also determine which variables are observed. For instance, physical signs are observed by physical examination, blood gases are observed by laboratory tests, etc.

We conclude this section with some further terminology and notation on planning expressions. For a given time segment $[t_1, t_2]$, we will use $S[t_1, t_2]$ and $D[t_1, t_2]$ instead of $S([t_1, t_2])$ and $D([t_1, t_2])$ as a shorthand notation for sets of state and decision variables. A configuration of $S[t_1, t_2]$ is called a *state sequence* over the time segment, and represents a specific series of states of the system over time. A configuration of $D[t_1, t_2]$ is called an *action sequence*, and represents specific behaviour of the planning agent over time. Finally, a configuration of $S[t_1, t_2] \cup D[t_1, t_2]$, i.e. a conjunction of state and action sequences, is called a *planning history*.

2.3 Decision processes

We now formalise the notion of *decision process* as a probability distribution on all possible planning histories.

Definition 3 (Decision process) *Let* $\mathbf{X}$ *be a set of random variables and let* $M = (T, A, p, o, r)$ *be a POMDP model over* $\mathbf{X}$*. A* decision process P *for* M *is a joint probability distribution on* $S(T) \cup D(T)$*, where for all time points* $t \in T$*,* $t < N$*, all system states* $X_1, X_2 \in \Omega_{\mathbf{X}}$*, and all actions* $a \in A$ *we have*

$$P(S(t+1) = X_2 \mid S(t) = X_1 \wedge d(t) = a) \quad = \quad p(X_1, a, X_2)$$

whenever $P(S(t) = X_1 \wedge d(t) = a) > 0$*.*

A decision process P describes the intertwined reaction over time of the dynamic system to the behaviour of the planning agent and vice versa: it covers the state changes induced by the actions chosen and the agent's responses to his perceptions of those changes. As such, a decision process P comprises both a description of the POMDP and a decision-making strategy: it implements the meta-level perspective from an external observer. We will make extensive use of the fact that all probabilistic expressions pertaining to decision processes take arguments from the language $\Phi(M)$.

Now, let $\mathbb{P}_M$ denote all decision processes for a given POMDP model M. The elements of $\mathbb{P}_M$ differ with respect to the agent's decision-making behaviour. In other words, each of these processes implicitly describes a decision-theoretic plan for model M; let $\rho(P)$ denote this plan. The planning problem associated with a given POMDP model can now be described as follows. Let $U(h)$ be the *utility* (sum of rewards) associated with planning history $h \in C_{S(T) \cup D(T)}$. The *expected utility* of decision process $P \in \mathbb{P}_M$ therefore equals

$$E_P[U] \quad = \quad \sum_{h \in H(T)} P(h) U(h) \tag{1}$$

and the planning agent's task is find a plan $\rho^* = \rho(P^*)$ that maximizes this value. We will use $\mathbb{P}_M^{\text{det}}$ to denote the subset of decision processes where all action choices depend deterministically on past actions and observations.

3 CONTINGENCY PLANNING

3.1 Decision rules and plans

We now turn to expressions that explicitly guide the action choices of the planning agent. An expression that prescribes such a choice for a single specific situation will be called a *decision rule*; a set of decision rules will be called a *contingency plan*. We first introduce the notion of *choice context*, which is the type of expression that may serve as a decision rule's antecedent. Recall that a decision may be based on all decisions and system states in the past and the contemporaneous system state.

Definition 4 (Choice context) *Let $t \in T$ be a decision moment. Any configuration of a subset of $S[0,t] \cup D[0,t-1]$ is called a* choice context *for moment t.*

When $\varphi \vdash s(t_1) = y$, we say that state variable $s(t_1)$ is *covered* by φ, and similarly if $\varphi \vdash d(t_2) = a$, we say that φ *covers* a decision at time point t_2. Note that by definition, $t_1 \leq t$ and $t_2 < t$ if φ is a choice context for moment t. It is possible that a choice context does not cover any state variables and it is also possible that a choice context does not cover any decisions. The (unique) *most general* choice context is the empty conjunction, $\top$; each configuration of $S[0,t] \cup D[0,t-1]$ is a *most specific* context for time point t.

A choice context for time point t does not commit to a decision for that time point; such commitments are expressed in *decision rules*.

Definition 5 (Decision rule) *A* decision rule *for time point $t \in T$ is an expression of the form*

$$\varphi \rightarrow d(t) = a, \tag{2}$$

where φ is a choice context for t. We refer to φ as the antecedent *of the rule and to the expression $d(t) = a$ as its* consequent.

The above decision rule prescribes to choose action a at time point t given the information conveyed by choice context φ. It is said to be *applicable* in all choice contexts ψ for time point t satisfying $\psi \vdash \varphi$. The number of variables that is referred to in φ is the *complexity* of the rule.

Another property of decision rules concerns their potential to be actually executed by a planning agent: this is only possible if the truth of the antecedent of a rule is verifiably by the agent at the time of the decision. This means that if the antecedent covers state variables, then these variables must have been observed by the agent. In this paper, we will pay no further attention to this issue, and refer the reader to [9, Chapter 5] for more details.

Two decision rules $\varphi_1 \rightarrow d(t) = a_1$ and $\varphi_2 \rightarrow d(t) = a_2$ are said to be *potentially conflicting* when their choice contexts are compatible (i.e. $\varphi_1 \wedge \varphi_2 \not\equiv \bot$), but they do prescribe different action choices (i.e. $a_1 \neq a_2$). Potentially conflicting rules induce a contradiction once we start reasoning with evidence that renders both rules applicable: if ψ is a choice context satisfying $\psi \vdash \varphi_1 \wedge \varphi_2$, then

$$\psi \wedge (\varphi_1 \rightarrow d(t) = a_1) \wedge (\varphi_2 \rightarrow d(t) = a_2) \quad \vdash \quad \bot. \tag{3}$$

We now define a set of decision rules for mutually exclusive situations to be a *contingency plan*, or *plan* for short.

Definition 6 (Plan) *A* contingency plan *is a set π of decision rules, where for each pair $\varphi_1 \rightarrow d(t_1) = a_1, \varphi_2 \rightarrow d(t_2) = a_2 \in \pi$ of rules we have either $t_1 \neq t_2$ or $\varphi_1 \wedge \varphi_2 \equiv \bot$.*

A contingency plan π prescribes action choices for a collection of choice contexts in a given planning domain. We require that all rules for a given time point have mutually incompatible antecedents; a plan can therefore not contain potentially conflicting rules or generalisations of its own rules. We note that the size of a plan π may (and often will) exceed the number of decision moments $|T|$, as many decision rules for a single time point can co-exist without being conflicting: these rules then prescribe action choices for different choice contexts.

3.2 Plans and decision processes

We will now relate the planning behaviour that is explicitly described by decision rules to the behaviour that is implicitly present in decision processes.

Definition 7 (Implementation) *Decision process P is said to* implement *decision rule $\varphi \rightarrow d(t) = a$ when $P(\varphi \rightarrow d(t) = a) = 1$; the implementation is* strict *if in addition $P(\varphi) > 0$. The process P is said to implement contingency plan π if it implements all decision rules in π.*

The next proposition serves to sharpen the intuitions for the concept of implementation a bit further.

Proposition 8 *Let P be a decision process. The following statements are equivalent:*

1. *P implements the decision rule $\varphi \rightarrow d(t) = a$;*
2. *$P(\varphi \wedge d(t) = a) = P(\varphi)$; and*
3. *$P(\varphi \wedge \neg d(t) = a) = 0$.*

Proof. See [9]. □

Each of the rules in a plan can be seen as a constraint on the possible behaviours of the planning agent, or equally, on the decision processes that implement it. As such, a plan may exhibit several forms of *overspecification* and *underspecification*. In the next two sections, we further investigate these topics.

4 PLAN CONSISTENCY AND COHERENCE

Implementation of decision rule $\varphi \rightarrow d(t) = a$ in decision process P is a trivial matter when $P(\varphi) = 0$, because then $P(\varphi \wedge \psi) = 0$ for any proposition $\psi \in \Phi(M)$, and any decision rule having φ as its antecedent is thus implemented by P. The notion of strict implementation is therefore more interesting; we then have that $P(\varphi) > 0$ and $P(d(t) = a \mid \varphi) = 1$. Unfortunately, not all contingency plans permit strict implementation, because the flexible nature of contingency plans allows for two forms of internal contradiction in plans. In this section we describe these two forms: *inconsistency* and *incoherence*. For the proofs of both propositions in this section we refer to [9].

Both forms of internal plan contradiction can be traced back to rules making the 'wrong' assumptions about the planning agent's behaviour. We call such assumptions about planning behaviour *presuppositions*.

Definition 9 (Presupposition) *The expression $\varphi_1 \wedge c_{d(t_1)}$, where φ_1 is a choice context for time point t_1, is called a* presupposition *of decision rule $\varphi_2 \rightarrow c_{d(t_2)}$ when $t_1 < t_2$ and $\varphi_2 \vdash \varphi_1 \wedge c_{d(t_1)}$.*

Intuitively, presuppositions of a given decision rule describe planning behaviour that must have been present for the rule to be applicable. When the rule's antecedent does not contain action choices, then the rule does not make any presuppositions. Note, however, that a presupposition of the form $\varphi_1 \wedge c_{d(t_1)}$ does not express that decision $c_{d(t_1)}$ is *necessarily* made in context φ_1. It says that decision $c_{d(t_1)}$ has been made in that context, and the rule making the presupposition implicitly asserts that such is *possible*.

Definition 10 (Exclusion) *Let π be a contingency plan. We say that planning history h is* excluded *by π when there exists a decision rule $\delta \in \pi$ such that $h \wedge \delta \equiv \bot$. The set of all planning histories excluded by plan π is denoted by H_π^-.*

Exclusion of planning history h stems from the fact that it matches with the antecedent φ of some decision rule $\varphi \rightarrow d(t) = a \in \pi$, while contradicting the action choice $d(t) = a$. Then, $h \wedge \delta \equiv \bot$, and it is easily seen that we could never obtain history h by following the plan: the history h is excluded.

Proposition 11 *Let π be a contingency plan and let P be a decision process that implements π. Then, $P(h) = 0$ for all excluded histories $h \in H_\pi^-$.*

More generally, we also say that proposition $\varphi \in \Phi(M)$ is *excluded* by plan π when each $h \in \mathcal{H}(T)$, $h \vdash \varphi$, is excluded by π. It follows that also $P(\varphi) = 0$ when decision process P implements plan π.

We can identify two cases where decision rules make the 'wrong' presuppositions. The first case is where a presupposition is directly contradicted by other rules in the plan through exclusions; we then say that the plan is *inconsistent*.

Definition 12 (Plan consistency) *A contingency plan π is said to be* inconsistent *if one of its rules makes a presupposition that is excluded by the plan itself. Otherwise, the plan is* consistent.

Proposition 13 *An inconsistent plan does not permit strict implementation.*

Note that we cannot restore consistency by adding rules to an inconsistent plan: any extension of an inconsistent plan is also inconsistent.

In the second case, multiple contradictory presuppositions are found in the plan.

Definition 14 (Plan coherence) *Let π be a contingency plan. We say that the plan is* incoherent *if there exist decision rules $\delta_1, \delta_2 \in \pi$ that presuppose $c \wedge d(t) = a_1$ and $c \wedge d(t) = a_2$, respectively, where $c \in C_{S[0,t]}$ is a state sequence over $[0, t]$ and $a_1 \neq a_2$. Otherwise, the plan is said to be* coherent.

Proposition 15 *An incoherent plan does not permit strict implementation by a decision process with deterministic planning behaviour.*

Also the extensions of an incoherent plan are incoherent.

5 PLAN COMPLETENESS

In this section we discuss the *completeness* of plans. The notion of *coverage* provides the basis for characterizing this property. The underlying idea is that a given choice context φ for time point $t \in T$ essentially represents a multitude of more specific contexts – that is, unless φ is a most specific choice context for that time point. Recall that φ is most specific when it is a configuration of the set $S[0, t] \cup D[0, t-1]$; otherwise, more specific contexts are obtained by adding information to φ.

We say that choice context ψ is *covered* by a given plan π, if one of the decision rules in π applies in context ψ, or if we can always find an applicable rule in the plan by adding information to ψ. In the former case, we have that $\varphi \rightarrow d(t) = a \in \pi$ for some action $a \in A$, where $\psi \vdash \varphi$. In the latter case, there exist a collection of mutually incompatible choice contexts $\varphi_1, \ldots, \varphi_m$, each covered by π and more informative than ψ, and such that

$$\psi \quad \equiv \quad \varphi_1 \vee \cdots \vee \varphi_m. \tag{4}$$

It should be noted that the rules associated with $\varphi_1, \ldots, \varphi_m$ may prescribe different actions choices; context ψ is thus not specific enough provide an unambiguous choice, although it is covered by the plan.

The notion of coverage is now formally defined as follows.

Definition 16 (Coverage) *The* coverage *of contingency plan π at time point $t \in T$, written $\mathrm{cover}(\pi, t)$, is the smallest set of choice contexts for time point t such that $\psi \in \mathrm{cover}(\pi, t)$ if*

- *$\psi \rightarrow d(t) = a \in \pi$ for some action $a \in A$,*
- *$\varphi \in \mathrm{cover}(\pi, t)$ and $\psi \vdash \varphi$, or*
- *$\varphi_1, \ldots, \varphi_m \in \mathrm{cover}(\pi, t)$ and $\psi \equiv \varphi_1 \vee \cdots \vee \varphi_m$.*

When $\psi \in \mathrm{cover}(\pi, t)$, we use $\pi_t(\psi) \subseteq A$ to denote the set of actions that may be prescribed by plan π in that context, i.e., $a \in \pi_t(\psi)$ if and only if there exists a rule $\varphi \rightarrow d(t) = a \in \pi$ such that $\psi \wedge \varphi \not\equiv \bot$.

The first and second clause jointly describe the former case identified above, while the third clause describes the latter case.

Given that $\psi \in \mathrm{cover}(\pi, t)$, the plan π is said to be *unequivocal* for context ψ when $|\pi_t(\psi)| = 1$; otherwise, $|\pi_t(\psi)| > 1$, and the plan is called *equivocal* for that context. Notwithstanding the ambivalence in that case, if $a \notin \pi_t(\psi)$ then π will certainly *not* prescribe the action choice $d(t) = a$ in context ψ or any of its specialisations; we say this action choice is not an *option* in context ψ at time point t. Also note that we may have that $a_i = a_j$ for all $i, j = 1, \ldots, m$, rendering π unequivocal for context ψ.

We now say that a contingency plan is *complete* when it covers all possible initial states at time point $t = 0$, and covers all choice contexts for future time points that are compatible with prescribed action choices.

Definition 17 (Plan completeness) *A contingency plan π is* complete *when*

1. *if c is a configuration of $S(0)$, then $c \in \mathrm{cover}(\pi, 0)$, and*
2. *for all $t < N$, if c is a configuration of $S[0, t] \cup D[0, t-1]$ and $c \in \mathrm{cover}(\pi, t)$, then $c \wedge d(t) = a \in \mathrm{cover}(\pi, t+1)$ where $\pi_t(c) = \{a\}$.*

Otherwise, the plan is said to be incomplete.

Intuitively, the property of plan completeness ensures that we always know which action to choose if we have followed the plan at the preceding time points, whatever states the system occupies. Note that the requirement that $c \wedge d(t) = a \in \mathrm{cover}(\pi, t+1)$ does not imply the existence of a rule with that antecedent in π: there will typically be rules that take into account some observation at time point $t+1$ while being less specific with respect to preceding time points. These rules should however jointly cover the case $c \wedge d(t) = a$.

Theorem 1 *Let M be a POMDP model, and let π be a complete contingency plan for M. Then, the decision process $P \in \mathbb{P}_M$ that implements π is unique.*

Proof. (Sketch) Suppose that $P_1, P_2 \in \mathbb{P}_M$ are different decision processes that both implement π. If P_1 and P_2 are different, then there must exist a proposition $\varphi \in \Phi(M)$ for which $P_1(\varphi) \neq P_2(\varphi)$. As we can write φ as a disjunction of mutually incompatible decision-making histories over T, this would imply that there exists a history $h \in \mathcal{H}(T)$ for which $P_1(h) \neq P_2(h)$. It is shown by induction on T that such a history does not exist under the given conditions. □

The decision process $P \in \mathbb{P}_M$ implementing a given plan π is not only unique, but also has deterministic planning behaviour (so we know that $P \in \mathbb{P}_M^{\mathrm{det}}$). Conversely, a given decision process with such behaviour can be shown to implement a consistent and complete plan.

Theorem 2 *Let M be a POMDP model. Each decision process $P \in \mathbb{P}_M^{\mathrm{det}}$ implements a consistent and complete contingency plan for M.*

Proof. (Sketch) We construct the plan π that is implemented by P as follows. Initialise $\pi = \varnothing$. For each $t = 0, 1, \ldots, N$, we enumerate the configurations of the set $S[0,t] \cup D[0,t-1]$; let c denote such a configuration. If $P(c) > 0$, then there exists a unique action a such that

$$P(c \wedge d(t) = a) \quad = \quad P(c), \tag{5}$$

because the planning behaviour in process P is deterministic. We therefore add the decision rule $c \rightarrow d(t) = a$ to π. It is easily verified that the plan thus obtained is consistent and complete, and implemented by P. □

One might be inclined to think that there must exist a bijection between $\mathbb{P}_M^{\det}$ and the set of all consistent and complete plans for a given decision basis. This is not the case as our representation of decision strategies is not unique: multiple, syntactically different plans may describe the same strategy; we refer to [9] for details on this matter.

It should be noted that the above theorems deal with implementation in general, not strict implementation. The possibility to strictly implement complete plans critically depends on the characteristics of the POMDP model: only when the model is strictly positive, i.e. if $p(X_1, a, X_2) > 0$ for all value sets $X_1, X_2 \in \Omega_{\mathbf{X}}$ and all actions $a \in A$, we are sure to find a decision process $P \in \mathbb{P}_M^{\det}$ that strictly implements a complete plan π. Therefore, Theorems 1 and 2 only apply to strict implementation when the control model in question is strictly positive.

6 DISCUSSION

Decision-theoretic planning is a widely accepted and mathematically sound approach to solving planning problems under uncertainty. However, few attempts have been made to integrate this type of planning with the way human experts solve decision problems. Our work attempts to fill this deficiency. We have provided the formal foundations for systems that solve decision-theoretic planning problems by assembling sets of symbolic decision rules.

The following application of our work is foreseen. First, after a given problem domain has been formalised as a POMDP model, a human expert (or panel of experts) is asked to formulate a set of decision rules that is believed to be reliable. This set of decision rules is then checked for potential conflicts, and, if necessary, the expert is asked to refine the rules untill no two rules are applicable in the same situation. Second, the resulting set of rules (which is a partial contingency plan for the planning problem) is checked for consistency and coherence. In general, both inconsisteny and incoherence can be also resolved by refining a number of rules in the plan; the expert can be asked to provide the necessary refinements. The third and final step is to make the plan complete, by adding rules for choice contexts that were previously not covered by the plan.

We are currently designing an algorithm for performing the third step. The algorithm performs Monte Carlo simulations of the decision process; in each simulation, decision rules from the plan are followed whenever they are applicable. For choice contexts *not* covered by the plan, an estimate of the best decision is made after sufficient simulations have been performed. The resulting decision rule is then added to the plan, possibly after consulting the domain expert.

The use of Monte Carlo simulations has two major advantages here. First, it directs the attention towards choice contexts that are relatively probable to occur, but are neglected in the plan provided by the expert. Because these contexts will show up relatively often in the simulations, they will be the first for which new decision rules are proposed. This is valuable because they will also show up relatively often in reality. Second, we avoid putting effort in situations that are excluded by the model or by the decision rules provided by the expert: such situations will simply not occur in the simulations. This is hard to accomplish when a dynamic-programming method is used, because the planning problem is then solved in a backward fashion.

A few existing algorithms for decision-theoretic planning explicitly search in the space of all partial and complete plans. For instance, E. Hansen [4] describes two algorithms for solving POMDPs that represent the plan as a finite-state controller; the algorithms search for an optimal plan by iteratively improving the controller. Also the planning system DRIPS [3] employs a notion of partial plan. These partial plans are expressed as sequences of operators, each of which can be an abstraction of a number of actions. The system iteratively reduces the number of abstractions, at each iteration comparing partial plans on the basis of *plan dominance* relations [11].

In the near future, we intend to finish the design and implement the algorithm for incremental plan construction that was sketch above. The domain of intensive care medicine will serve as a test bed in our investigations.

ACKNOWLEDGEMENTS

The author wishes to thank Peter Lucas, John-Jules Meyer, Silja Renooij, and two anonymous reviewers for their comments on the manuscript.

REFERENCES

[1] C. Boutilier, T. Dean, and S. Hanks, 'Planning under uncertainty: structural assumptions and computational leverage', *Journal of Artificial Intelligence Research*, **11**, 1–94, (1999).

[2] A.R. Cassandra, L.P. Kaelbling, and M.L. Littman, 'Acting optimally and partially observable stochastic domains', in *Proc 12th National Conference on Artificial Intelligence (AAAI–94)*, pp. 1023–1028, (1994).

[3] P. Haddawy, A. Doan, and R. Goodwin, 'Efficient decision-theoretic planning: techniques and empirical analysis', in *Proc 11th Conference on Uncertainty in Artificial Intelligence (UAI–95)*, pp. 229–326, Morgan Kaufmann, (1995).

[4] E. Hansen, 'Solving POMDPs by searching in policy space', in *Proc 14th Conference on Uncertainty in Artificial Intelligence (UAI–98)*, pp. 211–219, Morgan Kaufmann, (1998).

[5] S. Koenig and R.G. Simmons, 'Risk-sensitive planning with probabilistic decision graphs', in *Proc 4th Conference on Knowledge Representation and Reasoning (KR–94)*, (1994).

[6] S.-H. Lin and T. Dean, 'Generating optimal policies for high-level plans with conditional branches and loops', in *New Directions in AI Planning*, eds., M. Ghallab and A. Milani, 187–200, IOS Press, (1996).

[7] M.L. Littman, J. Goldsmith, and M. Mundhenk, 'The computational complexity of probabilistic planning', *Journal of Artificial Intelligence Research*, **9**, 1–36, (1998).

[8] N. Peek, 'Explicit temporal models for decision-theoretic planning of clinical management', *Artificial Intelligence in Medicine*, **15**(2), 135–154, (1999).

[9] N. Peek, *Decision-Theoretic Planning of Clinical Patient Management*, Ph.D. thesis, Institute of Information and Computing Sciences, Utrecht University, (2000).

[10] G. Tusch, 'Optimal sequential decisions in liver transplantation based on a POMDP model', in *ECAI 2000: Proc 14th European Conference on Artificial Intelligence*, ed., W. Horn, pp. 186–190. IOS Press, (2000).

[11] M.P. Wellman, 'Dominance and subsumption in constraint-posting planning', in *Proc 10th International Joint Conference on Artificial Intelligence (IJCAI–87)*, pp. 884–890, (1987).