# On the Undecidability of Probabilistic Planning and Infinite-Horizon Partially Observable Markov Decision Problems

**Omid Madani**
Dept. of Comp. Sci. and Eng.
University of Washington, Box 352350
Seattle, WA 98195-2350 USA
madani@cs.washington.edu

**Steve Hanks**
Adaptive Systems Group
Harlequin Inc.
1201 Third Avenue, Suite 2380
Seattle, WA 98105 USA
hanks@harlequin.com

**Anne Condon**
Computer Sciences Deptartment
University of Wisconsin
Madison, WI 53706 USA
condon@cs.wisc.edu

**Abstract**

We investigate the computability of problems in probabilistic planning and partially observable infinite-horizon Markov decision processes. The undecidability of the *string-existence* problem for probabilistic finite automata is adapted to show that the following problem of plan existence in probabilistic planning is undecidable: *given a probabilistic planning problem, determine whether there exists a plan with success probability exceeding a desirable threshold.* Analogous policy-existence problems for partially observable infinite-horizon Markov decision processes under discounted and undiscounted total reward models, average-reward models, and state-avoidance models are all shown to be undecidable. The results apply to corresponding approximation problems as well.

## 1 Introduction

We show that problems in probabilistic planning (Kushmerick, Hanks, & Weld 1995; Boutilier, Dean, & Hanks 1999) and infinite-horizon partially observable Markov decision processes (POMDPs) (Lovejoy 1991; White 1993) are uncomputable. These models are central to the study of decision-theoretic planning and stochastic control problems, and no computability results have previously been established for probabilistic planning. The undecidability of finding an optimal policy for an infinite-horizon POMDP has been a matter of conjecture (Papadimitriou & Tsitsiklis 1987), (Littman 1996), (Blondel & Tsitsiklis 1998). Our results settle these open problems and complement the research on the computational complexity of finite-horizon POMDP problems (Papadimitriou & Tsitsiklis 1987; Littman 1996; Mundhenk, Goldsmith, & Allender 1997; Littman, Goldsmith, & Mundhenk 1998).

We show that the following basic *plan-existence* problem in probabilistic planning is undecidable:

Given a probabilistic planning problem:

- a set of states
- a probability distribution over the value of the initial state
- a set of goal states
- a set of operators that effect stochastic state transitions
- a rational threshold $\tau$ on the probability of plan success

determine whether there is a sequence of operators that will leave the system in a goal state with probability at least $\tau$.

The probabilistic planning problem can be recast as an infinite-horizon undiscounted total reward POMDP problem, the problem being to determine whether there is a policy for the process with expected value at least $\tau$ (Boutilier, Dean, & Hanks 1999). Undecidability results for probabilistic planning thus have consequences for at least some POMDP problems as well. In this paper we demonstrate the undecidability of POMDPs for a variety of optimality criteria: total undiscounted and discounted reward, average reward per stage, and a state-oriented negative criterion discussed in (Puterman 1994). We also show the undecidability of several related approximation problems. An interesting consequence of our results on the impossibility of finding approximately optimal plans is that if the length of a candidate solution plan is bounded in size—even by an exponential function of the input description length—the solution found can be arbitrarily suboptimal.

Our analysis assumes incomplete information about the system state (partial observability), but does not set any *a priori* bound on the length of the solution plan. Even so, the undecidability result holds whether the set of admissible plans have finite length, infinite length, or either. Previous research had addressed either other models such as the fully observable case (Littman 1997), and bounded-length plans and finite-horizon POMDPs (see (Goldsmith & Mundhenk 1998) for a survey), or special cases, for example establishing decidability and computational complexity of goal-state reachability with either nonzero probability

or probability one (Alur, Couroubetis, & Yannakakis 1995; Littman 1996).

Our undecidability results for the probabilistic planning problems are based on the *string-existence* or *emptiness* problem for probabilistic finite-state automata (PFAs). The undecidability of this problem was first established in (Paz 1971). However we use the reduction in (Condon & Lipton 1989) for our work, since properties of the reduction help establish results for several additional open problems, including the threshold-isolation problem also raised in (Paz 1971). The work in (Condon & Lipton 1989) in turn is based on an investigation of *Interactive Proof Systems* introduced in (Goldwasser, Micali, & Rackoff 1985), and an elegant technique developed in (Freivalds 1981) to show the power of randomization in two-way PFAs.

The paper is organized as follows. The next section defines PFAs and the string-existence problem, and sketches the reduction of (Condon & Lipton 1989), highlighting aspects used in subsequent proofs. The remainder of the section establishes the undecidability of related approximation problems and the threshold-isolation problem.

The following section makes the connection between PFAs and probabilistic planning, proving the undecidability of the latter problem. POMDPs are addressed next: several optimality criteria are introduced, then the policy-existence problem is defined and shown to be undecidable regardless of which optimality criterion is adopted.

## 2 PFAs and the String-Existence Problem

A probabilistic finite-state automaton $M$ is defined by a quintuple $M = (Q, \Sigma, T, s, f)$ where $Q$ is a finite set of states, $\Sigma$ is the input alphabet, $T$ is a set of $|Q| \times |Q|$ row-stochastic transition matrices[1], one for each symbol in $\Sigma$, $s \in Q$ is the initial state of the PFA, and $f \in Q$ is an *accepting* state. The automaton occupies one state from $Q$ at any point in time, and at each stage transitions stochastically from one state to another. The state transition is determined as follows:

1. The current input symbol $a$ determines a transition matrix $M_a$.
2. The current state $s$ determines the row $M_a[s]$, a probability distribution over the possible next states.
3. The state changes according to the probability distribution $M_a[s]$.

The automaton halts when it transitions to the accepting state $f$. In this paper, we restrict attention to pfa's in which the accepting state $f$ is absorbing: $M_a[f, f] = 1.0, \forall a \in \Sigma$.

[1]Throughout the paper, we make the standard assumption that all the numbers (e.g. transition probabilities) are rational.

We say the automaton *accepts* the string $w \in \Sigma^*$ ($\Sigma^*$ denotes the set of all finite strings on $\Sigma$) if the automaton ends in the accepting state upon reading the string $w$, otherwise we say it *rejects* the string. We denote by $p^M(w)$ the acceptance probability of string $w$ by PFA $M$. The acceptance probability $p^M(w)$ for an infinite string $w$ is defined naturally as the limit $\lim_{i\to\infty} p^M(w_i)$, where $w_i$ denotes the length $i$ prefix of $w$.

**Definition 1** *The* string-existence *problem for PFAs is the problem of deciding whether or not there is some input string $w \in \Sigma^*$ that the given PFA accepts with probability exceeding an input threshold $\tau$.*

Both (Paz 1971) and (Condon & Lipton 1989) establish the undecidability of this problem, also known as the *emptiness problem*:

**Theorem 2.1** *(Paz 1971)(Condon & Lipton 1989) The string-existence problem for PFAs is undecidable.*

In the next subsection, we describe the properties of the reduction developed in (Condon & Lipton 1989), followed by a more detailed explanation of the proof. The details of the proof are used to develop corollaries related to probabilistic planning and POMDP problems, most notably Lemma 4.3, which establishes the undecidability of optimal policy construction for discounted-total-reward infinite-horizon POMDPs.

### 2.1 Properties of the Reduction

In (Condon & Lipton 1989), the (undecidable) question of whether a Turing Machine (TM) accepts the empty string is reduced to the question of whether a PFA accepts any string with probability exceeding a threshold. The PFA constructed by the reduction tests whether its input is a concatenation of *accepting sequences*. An accepting sequence is a legal sequence of TM configurations beginning at the initial configuration and terminating in an accepting configuration.

The reduction has the property that if the TM is accepting, *i.e.* it accepts the empty string, then the PFA accepts sufficiently long concatenations of accepting sequences with high probability. But if the TM is *not* accepting, the PFA accepts all strings with low probability. We next formalize these properties and use them in subsequent undecidability results. The following section explains how the PFA generated by the reduction has these properties.

**Theorem 2.2** *There exists an algorithm which, given a two counter TM as input and any rational $\epsilon > 0$ and integer $K \geq 1$, outputs a PFA $M$ satisfying the following:*

1. *If the TM does not accept the empty string, the PFA $M$ accepts no string with probability exceeding $\epsilon$.*
2. *If the TM is accepting, then let string $w$ represent the accepting sequence, and let $w^n$ denote $w$ concatenated $n$ times. We have $\lim_{n\to\infty} p^M(w^n) = 1 - (1/2)^K$, and $\forall n, p^M(w^n) < 1 - (1/2)^K$.*

We conclude this section making two additional points about the string-existence problem.

- Due to the separation between the acceptance probability of the PFA in the two cases of the TM accepting the empty string or otherwise, the string-existence problem remains undecidable if the strict inequality in the description of the existence problem is replaced by a weak equality ($\geq$) relation.
- Although the problem is posed in terms of the existence of finite strings, the result holds even if the strings have infinite length.

## 2.2 Details of the Reduction

The class of TMs used in the reduction in (Condon & Lipton 1989) are *two-counter* TMs, which are as powerful as general TMs. The constructed PFA is supposed to detect whether a sequence of computations represents a valid accepting computation (accepting sequence) of the TM. This task reduces to the problem of checking the legality of each transition from one configuration of the TM to the next, which amounts to verifying that

- the first configuration has the machine in the start state
- the last configuration has the machine in the accepting state
- each transition is legal according to the TM's transition rules.

All these checks can be carried out by a deterministic finite state automaton, except the check as to whether the TM's counter contents remain valid across consecutive configurations. The PFA rejects immediately if any of the easily verifiable transition rules are violated, which leaves only the problem of validating the counters' contents across each transition.

On each computation step taken by a two-counter TM the counters' contents either stay the same, get incremented by 1, or get decremented by 1. Assuming without loss of generality that the counter contents are represented in unary, this problem reduces to checking whether two strings have the same length: given a string $a^n b^m$, does $n = m$?

Although this question cannot be answered exactly by any PFA, a *weak equality* test developed in (Condon & Lipton 1989) and inspired by (Freivalds 1981) can answer it in a strict and limited sense which is nonetheless sufficient to allow the reduction. The weak equality test works as follows. The PFA scans its input string $a^n b^m$, and with high probability enters an *indecision* state (or equivalently we say the outcome of the test is indecision). With some low probability the PFA enters a one of two "decisive" states. If the substrings have equal length the PFA either enters a *correct* state or a *suspect* state. It enters these two states equiprobably. However, suppose that the PFA enters a decisive state but the input string is composed of *unequal*-length substrings ($m \neq n$). In this case the *suspect* outcome is $k$ times more likely than the *correct* outcome, where the *discrimination factor* $k$ can be made as large as desired by increasing the size of the PFA.

The PFA of the reduction carries out a *global* test of its own on a candidate accepting sequence for the TM, using the weak-equality test to check for counter increments or decrements on consecutive configurations. Given a candidate accepting sequence, if the outcome of *all* the tests are decisive and *correct*, the PFA accepts the input. If the outcome of *all* the tests are *suspect*, the PFA rejects the input. Otherwise, the PFA remains in the *global-indecision* state until it detects the start of the next candidate accepting sequence (start configuration of the TM), or until it reaches the end of the input. If it is in the global-indecision state at the end of the input, it rejects.

If the original TM accepts the empty string, observe that the probability that the PFA accepts can approach the upper limit 1/2 on an input string consisting of a concatenation of sufficiently many accepting sequences. If the TM does not accept the empty string, it follows from the properties of the weak-equality test that the probability that the PFA accepts any string is no larger than $1/k$.

By making a minor adjustment to the PFA, the acceptance probability of the PFA when the TM accepts the empty string can be made arbitrarily close to 1: Instead of rejecting or accepting if it sees an all *suspect* or an all *correct* outcome on a single candidate accepting sequence, the PFA can instead increment an *all-decisive* counter with a finite upper limit $K$. The PFA accepts its input if and only if the the all-decisive counter reaches $K$, *and* it has seen an all *correct* on a candidate sequence. Hence, if the TM is accepting, the PFA accepts concatenation of sufficiently many accepting sequences with probability arbitrarily close to $1 - (1/2)^K$. In addition, for the cases when the TM is not accepting, the acceptance probability of the PFA can be made as small as desired for a given counter upper limit $K$, by choosing the discrimination factor $k$ of the weak-equality test to be large.

## 2.3 Undecidability of Approximations

The question of approximability is an important one, especially when computing an optimal answer is impossible. Unfortunately, it follows from the next corollary that approximations, such as computing a string which the PFA accepts with probability within an additive constant or multiplicative factor $\epsilon < 1$ of the maximum acceptance probability of the PFA[2] are also uncomputable.

**Corollary 2.3** *For* any *fixed* $\epsilon, 0 < \epsilon < 1$, *the following problem is undecidable: Given is a PFA M for*

[2]The maximum acceptance probability is taken as the supremum over the acceptance probability over all strings.

*which one of the two cases hold:*

- *The PFA accepts some string with probability greater than* $1 - \epsilon$.
- *The PFA accepts no string with probability greater than* $\epsilon$.

*Decide whether case 1 holds.*

**Proof.** The corollary is an immediate consequence of the properties outlined in Theorem 2.2, and the fact that $\epsilon$ in the reduction can be made as small as desired. □

### 2.4 Undecidability of the Threshold-Isolation Problem

There might be some hope for decidability of the string-existence problem for special cases: those for which the given threshold (also called a *cutpoint*) is *isolated* for the PFA:

**Definition 2** *(Rabin 1963) Let $M$ be a PFA. The threshold $\tau$ is $\epsilon$-isolated with respect to $M$ if $|p^M(x) - \tau| \geq \epsilon$ for all $x \in \Sigma^*$, for some $\epsilon > 0$.*

**Definition 3** *The threshold-isolation problem is, given a PFA $M$ and a threshold $\tau$, decide whether, for some $\epsilon > 0$, the threshold $\tau$ is $\epsilon$-isolated for the PFA $M$.*

Isolated thresholds are interesting because PFAs with isolated thresholds have less expressive power than general PFAs, thus the corresponding decision problems are easier. The language accepted by a PFA $M$ given a threshold $\tau$, denoted by $L(M, \tau)$, is the set of all strings that take the PFA to the accepting state with probability greater than $\tau$:

$$L(M, \tau) = \{w \in \Sigma^* : p^M(w) > \tau\}.$$

General PFAs are powerful enough to accept even non-context-free languages (see (Paz 1971) for an example). However, Rabin in (Rabin 1963) showed that PFA with *isolated* thresholds accept regular languages. A natural question then is: given a PFA and a threshold, whether the threshold is isolated for the PFA. If we can compute the answer and it is positive, then we can presumably compute the regular language accepted by the PFA, and see whether it is empty or not. That would afford at least the opportunity to recognize and solve a special case of the general string-existence problem.

The decidability of the isolation problem was raised in (Paz 1971), and was heretofore an open question to the best of our knowledge. The reduction in this paper shows that recognizing an isolated threshold is hard as well:

**Corollary 2.4** *The threshold-isolation problem is undecidable.*

**Proof.** As stated in Theorem 2.2, we can design the reduction with $\epsilon = 1/3$, and $K = 1$. It follows that if the TM is not accepting, then there is no string that the PFA accepts with probability greater than 1/3, while if the TM is accepting, there are (finite) strings that the PFA accepts with probability arbitrarily close to 1/2. In other words, the threshold 1/2 is isolated iff the TM is not accepting. □

## 3 Undecidable Problems in Probabilistic Planning

This work was originally motivated by questions about the computability of probabilistic planning problems, e.g. the problems introduced in (Kushmerick, Hanks, & Weld 1995; Boutilier, Dean, & Hanks 1999).

The probabilistic planning problem, studied in (Kushmerick, Hanks, & Weld 1995) for example, involves a finite set of states, a finite set of actions effecting stochastic state transitions, a start state (or probability distribution over states), a goal region of the state space, and a threshold $\tau$ on the probability of plan success. The problem is to find *any* sequence of actions that would move the system from the start state to a goal state with probability at least $\tau$.

While it had been well established that restricted versions of this problem were decidable, though intractable as a practical matter (Papadimitriou & Tsitsiklis 1987; Bylander 1994; Littman, Goldsmith, & Mundhenk 1998), the complexity of the general probabilistic planning problem (i.e. without restrictions on the nature of the transitions or the length of solution plan considered) had not been determined.

The results of the previous section establish the uncomputability of such problems in the general case—when there is no restriction imposed on the length of solution plans considered. Uncomputability follows when it is established that a sufficiently powerful probabilistic planning language can model any given PFA, so that any question about a PFA can be reformulated as a probabilistic planning problem.

This is the case for the probabilistic planning model investigated in (Kushmerick, Hanks, & Weld 1995). This model is based on STRIPS propositional planning (Fikes & Nilsson 1971) with uncertainty in the form of (conditional) probability distributions added to the action effects. It is established in (Boutilier, Dean, & Hanks 1999) that the propositional encoding of states is sufficient to represent any finite state space, and the extended probabilistic STRIPS action representation is sufficient to represent any stochastic transition matrix. Thus the string-existence problem ("is there any input string that moves the automaton from the start state to an accepting state with probability at least $\tau$?') can be directly reformulated in the planning context ("is there any sequence of actions that moves the system from a start state to a goal state with probability at least $\tau$?'). An algorithm that solved the planning problem would answer the question of whether or not such a plan exists by either generating the plan or terminating having failed to do so, thus solving the equivalent string-existence problem. Thus, as a corollary of

the undecidability of the string-existence problem for PFAs we obtain:

**Theorem 3.1** *The plan-existence problem is undecidable.*

We also note that due to the tight correspondence between PFA's and probabilistic planning problems, the other undecidability results from the previous section apply as well:

- "Approximately satisficing planning," generating a plan that is within some additive or multiplicative factor of the threshold is undecidable.
- Deciding whether the threshold for a particular planning problem represents an isolated threshold for that problem is undecidable.

Having established a connection between PFAs and probabilistic planning, we next explore the connection between PFAs (and probabilistic planning) and POMDPs.

## 4 Undecidable Problems for POMDPs

Markov decision processes and their partially observable variants provide a general model of control for stochastic processes (Puterman 1994). In a partially observable Markov decision process (POMDP) problem, a decision maker is faced with a dynamic system $S$ modeled by a tuple $S = (Q, \Sigma, T, R, O, s)$, a generalization of our PFA definition with similar semantics: $Q$ and $\Sigma$ are sets of $n$ states and $m$ actions respectively, $T$ is a set of $n \times n$ row-stochastic transition matrices, one for each action in $\Sigma$.

The POMDP model generalizes the PFA/Planning model in two ways: a more general model of observability, and a more general model of reward and optimality.

In the Planning/PFA model, it is assumed that the decision-making agent will not be able to observe the world as it executes its plan, thus is limited to precomputing then blindly executing its solution. This can be viewed as a limiting case of the POMDP model: the unobservable MDP or UMDP.

In the POMDP generalization, the agent receives an *observation* from the world after every stage of execution, which might provide some information about the prevailing world state. Observation information is specified through the parameter $O$, which supplies probabilities of the form $P(o|s, a, s')$: the probability that observation $o$ would have been received, given that the system was in state $s$, action $a$ was performed, which effected a transition to state $s'$. The agent maintains a probability distribution over the prevailing world state, then updates that information every time it takes an action $a$ and receives an observation $o$. The solution to a POMDP problem is a *policy*: a mapping from the actions so far taken and the observations so far received to an action. The term *plan* is often used to refer to a policy in the unobservable case, where there are no observations; thus a policy consists of a sequence of actions.

Unlike the Planning/PFA model which strives to find *any* plan that exceeds the threshold, the MDP model computes a policy that maximizes an *objective function*; a variety of objective functions are explored in the literature.

Most objective functions are based on the idea of a *reward function*, the function $R(s, a)$ which associates a reward or penalty for taking an action $a$ while in a state $s$. Additional aspects of the objective function are:

- The *horizon*. The horizon determines how many actions are to be executed. Typically considered are *finite-horizon* problems where the policy is executed for a fixed number of steps, and *infinite-horizon* problems where the policy is executed for an indeterminate number of steps.
- The *discount factor*. In a *discounted* objective function, rewards gathered in earlier stages of execution are valued more highly than rewards gathered in later stages. A *discount factor* $0 \leq \beta < 1$ is provided, and the reward gathered at stage $i$ is actually $\beta^i R(s_i, a_i)$. The *undiscounted* case—$\beta = 1$—provides the same reward for an $(s, a)$ pair regardless of the stage at which it occurs.
- Total versus average reward. In the former case the objective is to maximize the sum of all (possibly discounted) rewards over the (possibly infinite) horizon. In the latter case the objective is to maximize the total reward divided by the number of stages (taken as a limit in the infinite-horizon case).

We will refer to the choice of a horizon, a discount factor and an aggregation operator as an *optimality criterion*. The criteria most often studied in the literature are:

- Maximizing total discounted reward over a finite or infinite horizon.
- Maximizing average reward over a finite or infinite horizon.
- Maximizing total undiscounted reward over a finite horizon.
- Maximizing total undiscounted reward over an infinite horizon under restrictions on the reward function and system dynamics that bound the total reward possible.

In this paper we are primarily interested in infinite-horizon problems, as (1) complexity results for finite-horizon problems are well established (Goldsmith & Mundhenk 1998) (Mundhenk, Goldsmith, & Allender 1997), and (2) the Planning/PFA problem maps to an infinite-horizon POMDP, but not to a finite-horizon model.

We are now in a position to define the *policy-existence* problem for POMDPs, under a given optimality criterion. The space of policies considered in the following definitions is an important consideration. All

of the lemmas hold when the space of policies includes any one or more of the following sets: finite action sequences of indefinite length, infinite sequences, or algorithms that create such finite or infinite sequences.

**Definition 4** *The policy-existence problem (with respect to an optimality criterion) is, given a POMDP and a threshold, whether there exists a policy with expected value greater than the threshold.*

### 4.1 Undecidability for Positive-Bounded Models under Total Undiscounted Reward

The most direct result involves a special case of infinite-horizon undiscounted total-reward models called *positive bounded* (Puterman 1994). The essential feature of this model is that the reward structure and system dynamics for a problem must ensure that the total reward gathered is bounded both above and below, even over an infinite horizon.

The planning problem can easily be posed as a positive-bounded POMDP:

- the same observation $o$ is received regardless of the state and action (non-observability)
- unit reward is gathered on the execution of *any* action on the goal state (Figure 1a)
- the execution of *any* action at the goal state leads to an absorbing state: the system stays in that state and gathers no additional rewards (Figure 1a)
- all other states and actions incur no reward.

From this equivalence we can immediately establish the following lemma:

**Theorem 4.1** *The policy-existence problem for positive-bounded problems under the infinite-horizon total reward criterion is undecidable.*

**Proof.** Since any planning problem can be posed as a positive-bounded POMDP, we can easily verify that an effective algorithm for that problem could be used to solve the plan-existence problem, and by Corollary 3.1 such an algorithm cannot exist. To see this, note that a plan, say a finite sequence of actions, exists for the planning (PFA) problem with probability of reaching the goal (success probability) exceeding $\tau$, if and only if a finite sequence of actions exist with value exceeding $\tau$ for the corresponding UMDP model (as outlined above and in Figure 1a): Let $p$ denote the success probability of a finite sequence of actions $w$ in the planning problem. Then $p$ is the expected total reward of action sequence $wa$ ($w$ followed by any action $a$) in the corresponding UMDP model. Conversely, if $v$ is the value of a sequence $w$ in the UMDP model, then $v$ is the success probability of sequence $w$ in the planning problem.

A similar equivalence holds for infinite action sequences. □

### 4.2 Undecidability under the Average Reward Criterion

The indirect connection to PFAs allows extension of the previous result to *all* undiscounted total-reward models, and to average-reward models as well.

**Theorem 4.2** *The policy-existence problem under the infinite-horizon average reward criterion is undecidable.*

**Proof.** The proof is complete once we observe that questions on acceptance probability of strings for a given PFA can be readily turned to questions on the value of similar strings in a related UMDP model. This transformation is achieved by modeling the probability of reaching the accepting state $f$ using rewards (Figure 1b). It can be verified that there is a string accepted by the PFA $M$ with probability exceeding $\tau$ if and only if there is a string with average reward greater than $\tau$ for the corresponding UMDP model. To see this, assume for some string $w$, $p^M(w) > \tau$, and denote by $v(w)$ the average reward of $w$ under the corresponding UMDP model. We must have, for any action $a$, $p^M(w)\frac{k+1}{k+|w|} \leq v(wa^k)$ where $wa^k$ denotes $w$ concatenated with $k$ repetitions of action $a$. The inequality follows from writing $v(w)$ in terms of the probability of reaching the goal state on each prefix of $w$. We thus have $\lim_{k\to\infty} v(wa^k) \geq p^M(w)$. Hence for some $k$, $v(wa^k) > \tau$. Conversely, we can verify that for any string $w$, $v(w) \leq p^M(w)$, so if $v(w) > \tau$, $p^M(w) > \tau$.

A similar equivalence holds for infinite strings. □

### 4.3 Undecidability of the Discounted-Reward Model

We turn now to the most commonly studied model: maximizing total expected discounted reward over an infinite horizon. Here, as in the proof of Lemma 4.2, we make a small change in the PFA constructed in the emptiness reduction.

**Theorem 4.3** *The policy-existence problem under the infinite-horizon discounted criterion is undecidable.*

**Proof.** Let us take the PFA constructed in the reduction of Section 2.2 and change it to a *leaky* PFA as follows. Let $d$ be any rational value such that $0 < d < 1$. The leaky PFA, upon reading an input symbol, continues as the original PFA with probability $1-d$. Otherwise, we say that it *leaks*, and in this case it makes a transition to either an absorbing rejection state or the absorbing accepting state, each with equal overall probability $d/2$. It is not hard to verify that maximizing the probability of reaching the accepting state in such a leaky PFA corresponds to maximizing the expected total discounted reward in a UMDP with a reward structure as described in the proof of Lemma 4.2 and Figure 1(a), where the discount factor is $\beta = 1-\frac{d}{2}$. We show that if the TM is accepting (see Section 2.1), then the leaky PFA accepts some strings with probability greater than 1/2, while if the TM is not accepting,

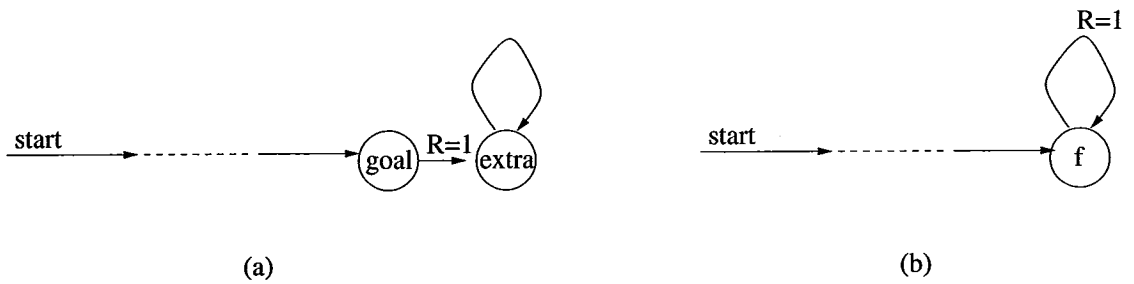


Figure 1: (a) The criterion of maximizing the probability of reaching a goal state in probabilistic planning modeled as a total reward criterion. The old absorbing goal state (labeled goal) now, on any action, has a transition to an extra absorbing state with reward of 1.0. All other rewards are zero. (b) Similarly, a PFA string-existence problem modeled as an average reward problem.

every finite string is accepted with probability less than 1/2.

Assume the TM is accepting, and let $w$ be an accepting sequence of the TM. Assume the original PFA accepts only after $K \geq 2$ decisive outcomes (the all-decisive counter limit $K$ is explained in the end of the reduction subsection).

Let $q$ denote the probability that the PFA "halts" (i.e. goes into one of the absorbing states) on reading $w^j$. Let $p$ denote the probability that the PFA has leaked *given* that it halts, i.e. it halts due to the leak and not due to the remaining possibility of having $K$ decisive outcomes (which has $1-p$ probability). Hence, given that the PFA halts on $w^j$, the probability of acceptance is:

$$1/2p + (1-p)(1-1/2^K) = 1/2p + (1-p)(1/2+\epsilon),$$

for some $\epsilon > 0$, and the overall probability of acceptance is $q[1/2p + (1-p)(1-1/2^K)]$. As $q$ approaches 1 with increasing $j$ in $w^j$, we need only argue that $p$ is bounded above by a constant strictly less than 1, for sufficiently large $j$, to show that acceptance probability exceeds 1/2 for some $j$. We note that $p = 1$, when $j < K$. With $j \geq K$, the probability that the PFA leaks can be no larger than $1 - p(e)$, where $p(e)$ denotes the probability of event $e$, the event that the PFA does not leak on $w^j$, but halts (upon reading the last symbol, so that it has made $K$ decisions), hence $p(e) > 0$.

Assume the TM is not accepting. A candidate (accepting) sequence refers to a sequence of TM configurations where the first one is the initial TM configuration and the last is an accepting configuration. Any input string $s$ can be viewed as a concatenation of $j \geq 0$ candidate sequences appended with a possibly empty string $u$ where none of the prefixes of $u$ is a candidate sequence: $s = w_1 w_2 \cdots w_j u, j \geq 0$. If $j \leq K$, then probability of acceptance of the leaky PFA is $qp/2$, where $q < 1$ is the probability of halting on $s$ and $p = 1$ is the probability of leaking given that the PFA halts. Here, $p = 1$ because there is no other possibility for halting, but note that $q < 1$. If $j > K$, then probability of acceptance is: $q(p/2 + (1-p)(1/2-\epsilon))$, where $q < 1$ is the probability of halting on $s$ and $p < 1$ is the probability of leaking given that the PFA halts. Given that the PFA halts and does not leak, the probability of acceptance is strictly less than 1/2, as the PFA is keeping a counter, and the probability of $K$ suspect outcomes is more than 1/2 (for appropriately small $K > 1$, such as $K = 2$).

A similar conversion to the one in the proof of Lemma 4.1 reduces the string-existence problem for the leaky PFA to the question of policy-existence in a UMDP under the discounted criterion, thus completing the proof. □

We note that an inapproximability result similar to the one for PFAs also holds for POMDPs under the total undiscounted reward and the average reward criteria. However, under the discounted criterion, the optimal value is approximable to within any $\epsilon > 0$, due to the presence of the discount factor.

### 4.4 Undecidability under a Negative Model

The optimality criteria studied to this point involve maximizing the expected benefits of executing a policy. An alternative goal would be to choose a policy likely to avoid distaster. In these cases (*state-oriented negative* models) the objective is to minimize the probability of entering one or more designated negative states over the infinite horizon. We use the reduction in the previous proof to establish the undecidability of this particular negative model; the technique should be applicable to other negative models as well.

**Theorem 4.4** *Policy existence under the state-oriented negative model is undecidable.*

**Proof.** We reduce the string-existence question for the leaky PFA in reduction of Lemma 4.3 to this problem. Note that in the string-existence reduction for the leaky PFA, if the TM is accepting, there exist infinite (and therefore finite) sequences of symbols on

which the probability of acceptance of the leaky PFA exceeds 1/2. If the TM is rejecting, the probability of acceptance of no infinite sequence is over 1/2 (an infinite sequences with acceptance equals 1/2 may exist.). Take the rejecting absorbing state of the leaky PFA to be the state to avoid and the (undecidable) question would be whether there is an infinite sequence that avoids the rejecting state with probability exceeding 1/2. □

## 5 Summary

This paper investigated the computability of plan existence for probabilistic planning, and policy existence for a variety of infinite-horizon POMDPs. A correspondence was established between probabilistic (non-observable) planning and probabilistic finite-state automata, and the reduction of (Condon & Lipton 1989) was exploited to show that many natural questions in this domain are undecidable. The PFA and planning problems were then viewed as a special case of infinite-horizon POMDPs, thus providing undecidability results for a variety of POMDP models, both discounted and undiscounted.

It is now well established that optimal planning without full observability is prohibitively difficult both in theory and practice (Papadimitriou & Tsitsiklis 1987; Littman 1996; Mundhenk, Goldsmith, & Allender 1997). These results suggest that it may be more promising to explore alternative problem formulations, including restrictions on the system dynamics and the agent's sensing and effecting powers that are useful for realistic problem domains yet are more amenable to exact or approximate solution algorithms.

## Acknowledgements

Thanks to Bill Rounds, who first pointed out to us the connection between PFAs and probabilistic planning. Hanks and Madani were supported in part by ARPA/Rome Labs Grant F30602-95-1-0024, and in part by NSF grant IRI-9523649. Anne Condon was supported by NSF grants CCR-92-57241 HRD-627241.

## References

Alur, R.; Couroubetis, C.; and Yannakakis, M. 1995. Distinguishing tests for nondeterministic and probabilistic machines. In *Proc. of 27th STOC*, 363–372.

Blondel, V. D., and Tsitsiklis, J. N. 1998. A survey of computational complexity results in systems and control. Submitted to Automatica, 1998.

Boutilier, C.; Dean, T.; and Hanks, S. 1999. Decision theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research* 157–171. To appear.

Bylander, T. 1994. The computational complexity of propositional STRIPS planning. *Artificial Intelligence* 69:161–204.

Condon, A., and Lipton, R. 1989. On the complexity of space bounded interactive proofs. In *30th Annual Symposium on Foundations of Computer Science.*

Fikes, R. E., and Nilsson, N. J. 1971. STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2(3-4):189–208.

Freivalds, R. 1981. Probabilistic two way machines. In *Proc. International Symposium on Mathematical Foundations of Computer Science*, volume 118, 33–45. Springer-Verlag.

Goldsmith, J., and Mundhenk, M. 1998. Complexity issues in markov decision processes. In *Proc. IEEE conference on Computational Complexity.*

Goldwasser, S.; Micali, S.; and Rackoff, C. 1985. The knowledge complexity of interactive protocols. In *Proc. of 17th STOC*, 291–304.

Kushmerick, N.; Hanks, S.; and Weld, D. S. 1995. An algorithm for probabilistic planning. *Artificial Intelligence* 76(1-2):239–286.

Littman, M. L.; Goldsmith, J.; and Mundhenk, M. 1998. The computational complexity of probabilistic planning. *Artificial Intelligence Research.*

Littman, M. 1996. *Algorithms for Sequential Decision Making.* Ph.D. Dissertation, Brown.

Littman, M. L. 1997. Probabilistic propositional planning: Representaions and complexity. In *Proceedings of the 14th National Conference on AI.* AAAI Press.

Lovejoy, W. 1991. A survey of algorithmic methods for partially observable Markov decision processes. *Annals of Operations Research* 47–66.

Mundhenk, M.; Goldsmith, J.; and Allender, E. 1997. The complexity of policy existence problem for partially-observable finite-horizon Markov decision processes. In *Mathematical Foundations of Computer Science*, 129–38.

Papadimitriou, C. H., and Tsitsiklis, J. N. 1987. The complexity of Markov decision processes. *Mathematics of operations research* 12(3):441–450.

Paz, A. 1971. *Introduction to Probabilistic Automata.* Academic Press.

Puterman, M. L. 1994. *Markov Decision Processes.* Wiley Inter-science.

Rabin, M. O. 1963. Probabilistic automata. *Information and Control* 230–245.

White, D. 1993. *Markov Decision Processes.* Wiley.