A Temporal Logic for Planning under Uncertainty

Manuel Biscaia
Pedro Baltazar
Paulo Mateus
SQIG, Instituto de Telecomunicações,
Dep. Mathematics, IST,
Universidade Técnica de Lisboa, Portugal

Rajagopal Nagarajan
Department of Computer Science,
School of Science and Technology,
Middlesex University, London, UK

February 25, 2013

Abstract

Dealing with uncertainty in the context of planning has been an active research subject in AI. Addressing the case when uncertainty evolves over time can be difficult. In this work, we provide a solution to this problem by proposing a temporal logic to reason about quantities and probability. For this logic, we provide a PSPACE SAT algorithm together with a complete calculus. The algorithm enables us to perform planning under uncertainty via SAT, extending a technique used for classic planning. We can show that any obtained plan will have certain properties (desired or undesired). The calculus can also be used to derive the impossibility of a plan, given a set of specifications.

1 Introduction

One of the most fruitful techniques to solve the planning problem in AI has been by reduction to the satisfiability problem over classical propositional logic [9, 17]. Classical propositional logic does not address the problem of uncertainty and by considering a bounded time horizon one is able to reason and plan efficiently by using propositional symbols representing time. However, the main reason for the efficiency of this type of planning is the advanced SAT algorithms already in existence [17].

In this work, we use a logic that not only allows one to reason about quantities but also about probabilities, named Exogenous Probabilistic Propositional Logic (EPPL) [13, 15]. The term *exogenous* was coined by Kozen [12] to express the fact that the probabilities had an explicit syntax and were not hidden in the propositional symbols or connectives. Here we introduce a temporalization of this logic (μ-calculus extension of EPPL). Although quite expensive computationally, it has a relevant sublogic, Exogenous Probabilistic Linear Time Logic (EPLTL), which, as we show, has a PSPACE SAT algorithm. With these temporal logics at our disposal, and with the efficient algorithms already known to solve the Satisfiability problem of LTL [19], we are able to specify properties in a way relatively close to natural

language. In this way we can produce plans for desired behaviors of agents, taking into account uncertainty. We also present Hilbert calculi for both logics, allowing us to derive that any satisfying plan has certain additional properties.

Temporal logics have a component to reason about states, called the state logic, and temporal modalities to reason about the evolution of states. The probabilistic state logic considered, EPPL, is akin to the logic proposed by [7] with some slight modifications. For the temporal modalites we use the most powerful decidable language, the μ-calculus; and the result is a fixpoint logic whose SAT algorithm allows us to output plans. Other approaches to reasoning about time and probability exist. However, they either have an undecidable SAT problem, such as the probabilistic situation calculus [16], or their SAT problem is not yet solved (for instance PCTL [5, 4]).

There have been three different approaches to planning under uncertainty: full probabilistic [10], contingent probabilistic [2] and conformant probabilistic [3]. The difference between these approaches is the observation model considered. In full probabilistic planning, we assume that the observations are total (but random); in contingent probabilistic, the random observations are partial; and finally, in conformant probabilistic planning, no observations exist (although the underlying behavior is random). The SAT method proposed can be adopted to all the cases, although the specification has to be significantly different for each case. Many algorithms dedicated to each type of probabilistic planning exist, and are in general more restrictive than using an EPLTL specification. However, these are in general much more efficient, in most cases, because intelligent heuristics are used.

This paper is structured as follows. First, we begin by motivating our intended applications by a simple example. Then, we present the syntax, semantics and axiomatisation of the state logic EPPL. After that, we introduce the fixpoint extension MEPL. Then, we address completeness for MEPL and, finally, we revisit the example, formalizing the constraints given in the first section.

The central contribution of this paper is the introduction of a new logic MEPL for planning under uncertainty, which enables us to reason with time explicitly. In particular, we are able to use the fact that MEPL has a sublogic for which the SAT algorithm relies on the SAT problem of LTL, which has been thoroughly studied. The associated complexity results and algorithms are also completely new. The full paper will develop EPPL in significantly more detail than in earlier publications [13, 14], including all the required proofs and presenting case studies.

2 Motivating Example

Assume that a robot *Healer 2.0* has been built and it comes equipped with advanced medical equipment, previously tested in its earlier iteration. This time, however, it is capable of locomotion. Its intended use is in deserted locations, where it can be deployed far from its target, and then rescue its intended target. Unfortunately, the advanced medical equipment occupies a large amount of space, and *Healer 2.0* cannot transport much fuel. The cost of the deployment of the supplies is limited by a threshold; moreover this cost grows quadratically with the distance from the base.

The developers are interested in making a plan containing the robot movements on the grid (requires temporal reasoning), and on how to distribute the supplies along the grid (requires quantitative reasoning).

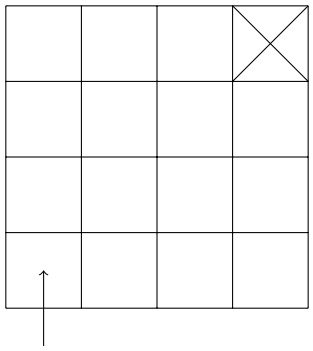

Figure 1: A grid illustrating the possible movements of *Healer 2.0*, with entry point and goal marked

3 A Probabilistic State Logic

The state logic is intended to specify and reason about each instant of time of a given agent. The idea is that, at each instant, certain variables take a probability distribution or an algebraic quantity. We are only able to perform algebraic reasoning about these quantities and probabilities, *i.e.*, addition and multiplication of real terms together with propositional reasoning about the comparison of real terms.

Syntax

The language of EPPL consists of formulae at two levels. The formulae at the first level, *basic formulae*, allow us to reason about state variables, for instance, locations. We can abstract locations as a finite set of propositional symbols Λ. The formulae at the second level, *global formulae*, allow us to perform probabilistic reasoning. We also consider *algebraic real terms*, built over a set of real logic variables R. These terms denote real numbers used for quantitative reasoning at the level of global formulae. The syntax of the language is as follows, where r is any real agebraic number:

$$\begin{aligned}\beta &:= \alpha \,[\!]\, (\neg\beta) \,[\!]\, (\beta \Rightarrow \beta)\\ t &:= x \,[\!]\, r \,[\!]\, \textstyle\int\beta \,[\!]\, (t+t) \,[\!]\, (t \times t)\\ \delta &:= (\mathsf{A}\beta) \,[\!]\, (\beta \bot \beta) \,[\!]\, (t \leq t) \,[\!]\, (\sim\delta) \,[\!]\, (\delta \sqsupset \delta)\end{aligned}$$

Our logic will be able to denote the probability of an event β. Moreover using quantifier elimination over Real Closed Fields [1], we can just add the constants 0 and 1 to our syntax. We will use the usual abbreviations for both global connectives and basic connectives, taking care to distinguish between both.

Semantics

Let $(\Omega, \mathcal{F}, \mathbf{P})$ be a *probability space*, and $\mathbf{X} = (X_\alpha : \Omega \to \mathbf{2})_{\alpha\in\Lambda}$ a *stochastic process* over $(\Omega, \mathcal{F}, \mathbf{P})$ where each X_α is a Bernoulli random variable, *i.e.* X_α ranges over $\mathbf{2} = \{0, 1\}$. The models of EPPL are tuples $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$. Observe that each basic EPPL formula β induces a Bernoulli random variable $X_\beta : \Omega \to \mathbf{2}$, defined as follows: $X_{(\neg\beta)}(\omega) = 1 - X_\beta(\omega)$; $X_{(\beta_1\Rightarrow\beta_2)}(\omega) = \max\{1 - X_{\beta_1}(\omega), X_{\beta_2}(\omega)\}$.

So, each basic formula β represents the measurable subset $\{\omega \in \Omega : X_\beta(\omega) = 1\}$. Moreover, each $\omega \in \Omega$ induces a valuation v_ω over Λ, such that $v_\omega(\alpha) = X_\alpha(\omega)$, for all $\alpha \in \Lambda$. Given an EPPL model $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$, and assignment $\rho : R \to \mathbb{R}$ for the real logical variables, the denotation of algebraic real terms is as follows:

- $[\![x]\!]_{m,\rho} = \rho(x)$;
- $[\![r]\!]_{m,\rho} = r$;
- $[\![t_1 + t_2]\!]_{m,\rho} = [\![t_1]\!]_{m,\rho} + [\![t_2]\!]_{m,\rho}$;
- $[\![t_1 \times t_2]\!]_{m,\rho} = [\![t_1]\!]_{m,\rho} \times [\![t_2]\!]_{m,\rho}$; and
- $[\![\int \beta]\!]_{m,\rho} = \int X_\beta \, d\mathbf{P} = \mathbf{P}(X_\beta = 1)$.

Note that the term $[\![\int \beta]\!]_{m,\rho}$ gives the expected value of X_β. Since X_β is a Bernoulli random variable, the expected value is the same as the probability of observing an outcome ω, such that v_ω satisfies β.

Moreover, the satisfaction of global formulae is given by:

- $m, \rho \Vdash (\mathsf{A}\beta)$ iff $X_\beta(\omega) = 1$ for all $\omega \in \Omega$;
- $m, \rho \Vdash (\beta_1 \perp \beta_2)$ iff $[\![\int \beta_1 \wedge \beta_2]\!]_{m,\rho} = [\![\int \beta_1]\!]_{m,\rho} \times [\![\int \beta_2]\!]_{m,\rho}$;
- $m, \rho \Vdash (t_1 \leq t_2)$ iff $[\![t_1]\!]_{m,\rho} \leq [\![t_2]\!]_{m,\rho}$;
- $m, \rho \Vdash (\sim\delta)$ iff $m, \rho \nVdash \delta$;
- $m, \rho \Vdash (\delta_1 \sqsupset \delta_2)$ iff $m, \rho \Vdash \delta_2$ or $m, \rho \nVdash \delta_1$.

Note that from the semantics just presented, we can see independence formulae as syntactic sugar, by replacing each independence formula $\beta_1 \perp \beta_2$ for $\int \beta_1 \wedge \beta_2 = \int \beta_1 \times \int \beta_2$.

Although our definitions are more general, we intend to reason about probabilities of classical valuations. Each valuation represents an atomic event; a disjunction of two valuations will then represent one of the two atomic events. If they are disjoint, then the probability of satisfying one of the two valuations is the sum of each atomic event. The set of propositional symbols could denote different spatial locations, time moments, or both. We will focus on the spatial perspective, since we will explicitly temporalize EPPL models in forthcoming sections. This will let the spatial reasoning to be probabilistic in nature, by allowing uncertainty of position; while the temporal reasoning can be non-deterministic, allowing many possible evolutions of probability distributions.

Small Model Theorem

Given that we are working towards a complete Hilbert calculus for EPPL through a SAT algorithm, it is important to investigate whether EPPL fulfills a small model theorem. In fact, we will also show that it is enough to consider probability spaces over finite valuations, as any EPPL formula will have a finite number of propositional symbols. Let δ be an EPPL formula; we denote the set of propositional symbols that appear in δ by $prop(\delta)$.

Theorem 3.1 (Small Model Theorem) *If δ is a satisfiable EPPL formula then it has a finite model over valuations using at most $2|\delta| + 1$ algebraic real numbers.*

Decision Algorithm for Satisfaction

The decision algorithm for EPPL satisfaction uses the decidability of the existential theory of real numbers and the small model theorem. Before presenting the algorithm, we introduce some notation. Given an EPPL formula δ, we will denote by

- $iq(\delta)$, the set of all inequalities $(t_1 \leq t_2)$ in δ;
- $bf_{\mathsf{A}}(\delta)$, the set of all universal subformulae $\mathsf{A}\beta$ in δ;
- $at(\delta) = bf_{\mathsf{A}}(\delta) \cup iq(\delta) \cup ip(\delta)$, the set of all global atoms of δ.

From now on, by an *exhaustive conjunction* of literals of $at(\delta)$, we mean a formula ε of the form $\delta_1 \sqcap \ldots \sqcap \delta_k$, where each δ_i is either a global atom, or its negation. Moreover, all global atoms or their negations occur in ε, therefore $k = |at(\delta)|$.

Given a global formula δ, we denote by δ^b the propositional formula obtained by replacing in δ, each global atom δ_i with a fresh propositional symbol α_i, for $i = 1, \ldots, k$. We also replace in δ, the global connectives $\sim$ and $\sqsupset$, by the propositional connectives $\neg$ and $\Rightarrow$, respectively. We denote by v_ε, the valuation over propositional symbols $\alpha_1, \ldots, \alpha_k$, such that $v_\epsilon(\alpha_i) = 1$ iff δ_i occurs positively in ε.

Given an exhaustive conjunction ε of literals of $at(\delta)$, we denote by $lbf_{\mathsf{A}}(\varepsilon)$ the set of basic formulae such that $\beta \in lbf_{\mathsf{A}}(\varepsilon)$ if $\mathsf{A}\beta$ occurs positively in ε (that is, not negated). Similarly, the set of basic formulae that occur nested by a $\sim\mathsf{A}$ in ε is denoted by $lbf_{\mathsf{E}\neg}(\varepsilon)$. Finally, we denote all the inequalities occurring in ε by $liq(\varepsilon)$. This last set contains the new inequalities introduced by the substitution of the independence formulae.

Given a global formula δ in $liq(\varepsilon)$, we denote by δ^a the analytical formula where all terms of the form $\int\beta$ are replaced in δ by $\sum_{v \in V, v \Vdash \beta} x_v$. We assume that x_v is a fresh variable from Var. We can use the PSPACE SAT algorithm of the existential theory of the real numbers [1], that we denote $SatReal$. We assume that this algorithm either returns *no model*, if there is no solution for the input system of inequations, or a *solution array* η, where $\eta(x)$ is the solution for variable x. We denote by $var(\delta)$ the set of real logical variables that occur in δ.

Theorem 3.2 Algorithm 1 decides the satisfiability of an EPPL formula in PSPACE.

Completeness

In general, when presenting a calculus, we are interested in showing that everything that can be proved is true and also that everything that is true can be proved as such. The first property is called *soundness*, while the second property is called *completeness*. Checking soundness ($\vdash \delta \Rightarrow \models \delta$) can be done by just checking the validity of the axioms, one by one. Checking completeness ($\models \delta \Rightarrow \vdash \delta$) is harder, and in some cases it may even be impossible.

In [15] it is shown that a superset of axioms and inference rules presented here is a sound and a (weakly) complete axiomatization of EPPL. Due to the EPPL SAT algorithm, we are able to show that the following calculus is weakly complete:

1. $\vdash_{\mathsf{EPPL}} (\mathsf{A}\beta)$ for each valid propositional formula β,
2. $\vdash_{\mathsf{EPPL}} \delta$ for each instantiation of a propositional tautology δ,
3. $\vdash_{\mathsf{EPPL}} (\mathsf{A}(\beta_1 \Rightarrow \beta_2) \sqcap (\mathsf{A}\beta_1 \sqcap \mathsf{A}\beta_2))$,

Algorithm 1: SatEPPL(δ)

Input: EPPL formula δ
Output: $(V, \mathbf{P})$ (denoting the EPPL model $m = (V, 2^V, \mathbf{P}, \mathbf{X})$) and assignment ρ or *no model*

compute $bf_{\mathsf{A}}(\delta), ip(\delta), iq(\delta)$ *and* $at(\delta)$*;*
foreach exhaustive conjunction ε of literals of $at(\delta)$ such that $v_\varepsilon \Vdash \delta^b$ **do**
  compute $lbf_{\mathsf{A}}(\varepsilon)$*,* $lbf_{\mathsf{E}\neg}(\varepsilon)$ and $liq(\varepsilon)$*;*
  foreach $V \subseteq 2^{prop(\delta)}$ such that $0 < |V| \leq 2|\delta| + 1$, $V \Vdash \wedge lbf_{\mathsf{A}}(\varepsilon)$ and $V \not\Vdash \beta$ for all $\beta \in lbf_{\mathsf{E}\neg}(\varepsilon)$ **do**
    $\kappa \longleftarrow \left(\sum_{v \in V} x_v = 1\right) \sqcap \left(\bigcap_{v \in V} 0 \leq x_v\right)$;
    foreach $\delta \in liq(\varepsilon)$ **do**
      $\kappa \longleftarrow \kappa \cap \delta^a$;
    end
    $\eta \longleftarrow SatReal(\kappa)$;
    if $\eta \neq$ *no model* **then**
      $\mathbf{P}_\eta \longleftarrow \eta_{|\{x_v : v \in V\}}$;
      $\rho_\eta \longleftarrow \eta_{|var(\delta)}$;
      return $(V, \mathbf{P}_\eta)$ and assignment ρ_η;
    end
  end
  return (*no model*);
end

4. $\vdash_{\mathsf{EPPL}} (\mathsf{Af} \equiv \mathbf{F})$,
5. $\vdash_{\mathsf{EPPL}} (\beta_1 \perp \beta_2) \equiv (\int \beta_1 \wedge \beta_2 = \int \beta_1 \times \int \beta_2)$,
6. $\vdash_{\mathsf{EPPL}} (t_1 \leq t_2)$ for each instantiation of a valid analytical inequality,
7. $\vdash_{\mathsf{EPPL}} (\int \mathbf{t} = 1)$,
8. $\vdash_{\mathsf{EPPL}} ((\int(\beta_1 \wedge \beta_2) = 0) \sqsupset (\int(\beta_1 \vee \beta_2) = \int \beta_1 + \int \beta_2))$,
9. $\vdash_{\mathsf{EPPL}} (\mathsf{A}(\beta_1 \Rightarrow \beta_2) \sqsupset (\int \beta_1 \leq \int \beta_2))$.
10. $\delta_1, (\delta_1 \sqsupset \delta_2) \vdash_{\mathsf{EPPL}} \delta_2$.

It is impossible to obtain a strongly complete axiomatization for EPPL (that is, if $\Delta \models \delta$ then $\Delta \vdash \delta$, for infinite Δ) because the logic is not compact [15]. Nevertheless, weak completeness is enough for system verification, since a plan usually just involves a finite number of hypotheses.

For the axiomatization presented, we consider a Hilbert system with a recursive set of axioms and finitary rules. Recall that the axiom schema 6 is decidable due to Tarski's result on the decidability of real closed fields [1]. Thus, the axioms constitute a recursive set.

Theorem 3.3 *The set of rules and axioms is a weakly complete axiomatization of EPPL.*

4 Temporal Extension of Exogenous Probabilistic Logic

We now introduce a μ-calculus extension of EPPL by adopting the fixpoint constructors [11]. We also provide a sound and (weakly-) complete proof system by enriching the μ-calculus proof system [20] with the axioms of EPPL. We will present full MEPL, providing a complete Hilbert calculus and a SAT algorithm.

This temporal extension can be seen as the most general one, subsuming most of the common temporal logics like LTL or CTL. In LTL, time evolves deterministically, as a sequence of instants. In CTL, time branches and behaves like a tree of instants.

Syntax

The syntax of MEPL is formed by enriching the μ-calculus by taking as propositional symbols the global atoms of EPPL. The syntax is as follows, where r is any real algebraic number:

$$\begin{aligned}\beta &:= \alpha \,[\!]\, (\neg\beta) \,[\!]\, (\beta \Rightarrow \beta)\\ t &:= z \,[\!]\, r \,[\!]\, \textstyle\int\beta \,[\!]\, (t+t) \,[\!]\, (t \times t)\\ \phi &:= (\mathsf{A}\beta) \,[\!]\, (t \leq t) \,[\!]\, \xi \,[\!]\, (\sim\phi) \,[\!]\, (\phi \sqsupset \phi) \,[\!]\, \Diamond\phi \,[\!]\, \mu\xi.\phi\end{aligned}$$

where in $\mu\xi.\phi$ any occurrence of ξ in ϕ is under an even number of negations.

The basic formulae and probabilistic terms have the same intuitive meaning as in EPPL. Similarly, the global formulae with fixpoint operators have the same meaning as in the μ-calculus.

Semantics

In order to provide semantics for the logic, we introduce a very simple notion of Kripke structure over EPPL models. An MEPL structure, or an EPPL-Kripke structure, consists of a tuple $M = (S, R, L)$ where S is a non-empty set of states, $R \subseteq S \times S$ is a total relation, and L is a map that assigns an EPPL model (including a variable assignment $\gamma : Z \to \mathbb{R}$) to each state in S. Our models are closely related to the models of probability and knowledge in [6].

The MEPL semantics mimics the semantics of the μ-calculus. In a Kripke structure, L maps each state to a set of propositional symbols (or equivalently to a valuation), that is a propositional model. For our EPPL-Kripke structures, L maps each state to an EPPL model.

We now present the formal semantics of MEPL. For this purpose, we denote by $[\delta]^M_V$ the set of states of M that satisfy δ given valuation $V : \Xi \to 2^S$. The set $[\delta]^M_V$ is defined inductively on the structure of δ as follows:

- $[(\mathsf{A}\beta)]^M_V = \{s \in S : L(s) \Vdash_{\mathsf{EPPL}} (\mathsf{A}\beta)\}$;
- $[t_1 \leq t_2]^M_V = \{s \in S : L(s) \Vdash_{\mathsf{EPPL}} (t_1 \leq t_2)\}$;
- $[\xi]^M_V = V(\xi)$;
- $[(\sim\varphi)]^M_V = S \setminus [\varphi]^M_V$;

- $[(\varphi_1 \sqsupset \varphi_2)]^M_V = [\sim\varphi_1]^M_V \cup [\varphi_2]^M_V$;
- $[(\Diamond\varphi)]^M_V = \{s \in S : \text{ exists } s' \in S, \text{ such that } (s,s') \in R, \text{ and } s' \in [\varphi]^M_V\}$;
- $[\mu\xi.\varphi]^M_V = \bigcap\{S' \subseteq S : [\varphi]^M_{V[\xi \leftarrow S']} \subseteq S'\}$;

  where $V[\xi \leftarrow S']$, as before, denotes the valuation V' that may just differ from V by $V'(\xi) = S'$.

Given a closed formula φ, if $s \in [\varphi]^M_V$ for some valuation V then we write $M, s \Vdash_{\mathsf{MEPL}} \varphi$.

Completeness

In this section we provide a complete calculus for MEPL. Completeness is obtained by using the completeness of the μ-calculus and EPPL. We give the complete axiomatization HC_{MEPL} of MEPL

- all EPPL tautologies,
- all instantiations of μ-calculus tautologies with MEPL formulae,
- $\phi_1, (\phi_1 \sqsupset \phi_2) \vdash_{\mathsf{MEPL}} \phi_2$,
- $(\phi_1[\xi \leftarrow \phi_2] \sqsupset \phi_2) \vdash_{\mathsf{MEPL}} (\mu\xi.(\phi_1 \sqsupset \phi_2))$.

Theorem 4.1 *The axiomatization HC_{MEPL} is weakly complete.*

Satisfaction

Theorem 4.2 *Let ϕ be an MEPL formula. Deciding whether ϕ is satisfiable can be reduced to the satisfiability problem of the μ-calculus.*

The satisfaction problem for μ-calculus is EXPTIME-complete [12]. Furthermore, since the translations used to bring $\phi \in$ MEPL to the μ-calculus realm are exponential on the size of ϕ, one could imagine that the complexity of deciding satisfiability of MEPL formulae is exponentially worse than that for the μ-calculus. However, as we shall see in the following subsection, for some cases we need not apply the SAT algorithm directly.

Exogenous Probabilistic Linear Temporal Logic

An important sub-logic of MEPL is the one obtained by taking just the path connectives next **X** and until **U**, such that $\mathbf{X}\varphi :=_{ab} (\Diamond\varphi)$, and $\varphi_1\mathbf{U}\varphi_2 :=_{ab} (\mu.\xi(\varphi_2 \sqcup (\Diamond(\varphi_1 \sqcap \xi))))$. We call this logic exogenous probabilistic linear temporal logic (EPLTL). We also note that the temporal operators G (meaning always), and F (meaning some time in the future), can be obtained from the until (**U**) operator. We further recall that $\varphi_1\mathbf{U}\varphi_2$ means that φ_1 has to be true until the moment that φ_2 is true, which will assuredly happen.

The algorithms introduced in [18, 8] to solve satisfiability of LTL can be extended to EPLTL.

Theorem 4.3 *The satisfiability problem for EPLTL is in PSPACE.*

5 Applications in planning

We will now show some of the properties MEPL can specify, illustrating also some of its usefulness (although we leave a more complex example for the full paper where we also incorporate uncertainty of positions). We will not use full MEPL, but we will use another important sublogic, a restriction of the μ-calculus to LTL, as it will be enough for our demonstration.

In this 4x4 grid, the robot will have to navigate through the cells until it finds the target, spending fuel to do so, and regaining it at some locations yet to be discovered. The fuel will be carried and delivered in discrete amounts, for instance $\{0, \frac{1}{10}, \ldots, \frac{9}{10}, 1\}$ using k for any of these discrete amounts, and thus we are modeling the fuel level as a probability. We will stipulate that the robot will start its mission with a full tank, and that the developers are willing to spend K extra fuel units to supply the robot.

We will require the following propositional symbols and constants: (i) $\{e_{(i,j)} : i, j \in \{1, \ldots, 4\}\}$, representing the location of the robot; (ii) *fuel*, representing the amount of fuel available; (iii) $\{sup_{(i,j)} : i, j \in \{1, \ldots, 4\}\}$ representing the distribution of the refuel supplies, (iv) $c_{(i,j),(i',j')}$ representing the fuel cost of moving from cell (i, j) to cell (i', j'); (v) K the total amount of fuel the developers are able to deliver for refuel; and (vi) T the threshold cost.

Our specification is as follows:

1. The robot can only be in one place at once:
   $\mathsf{G}((\sqcap_{i,j} \int e_{(i,j)} = 0 \sqcup \int e_{(i,j)} = 1) \sqcap (\sum_{i,j} \int e_{(i,j)} = 1))$
2. The robot can only move to adjacent cells and if it has some fuel left, spending $c_{(i',j')}$ fuel to move from cell $e_{(i,j)}$ to cell $e_{(i',j')}$, where $C_{(i,j)} = \{(i', j') \in \{1, ..., 4\} \times \{1, ..., 4\} : |i - i'| \leq 1, |j - j'| \leq 1, (i, j) \neq (i', j')\}$:
   $$\begin{aligned} &\mathsf{G}((\int \mathit{fuel} = \mathsf{k} \sqcap \mathsf{k} > 0 \wedge \int e_{(i,j)} = 1) \sqsupset \\ &\mathsf{X}((\sqcup_{(i',j') \in C_{(i,j)}} \int e_{(i',j')} = 1) \wedge \\ &((\int \mathit{fuel} = 0 \sqcap \mathsf{k} + K * \int sup_{(i,j)} - c_{(i',j')} < 0) \oplus \\ &(\int \mathit{fuel} = 1 \sqcap \mathsf{k} + K * \int sup_{(i,j)} - c_{(i',j')} > 1) \oplus \\ &(\int \mathit{fuel} = \mathsf{k} + K * \int sup_{(i,j)} - c_{(i,j),(i',j')})))) \end{aligned}$$
3. The probability distribution of supplies is set at the beginning:
   $(\sqcup_{\mathsf{k}}(\int sup_{(i,j)} = \mathsf{k}) \sqsupset \mathsf{G}(\int sup_{(i,j)} = \mathsf{k})) \sqcap \sum_{i,j} \int sup_{(i,j)} = 1$
4. The cost of deployment of supplies is limited by a threshold T; moreover, the cost grows quadratically with the distance:
   $\sum_{i,j} (\int sup_{(i,j)}) \frac{(i+j)^2}{4+4} \leq T$
5. If the robot has no fuel, it will not leave the cell:
   $\sqcup_{i,j} \mathsf{F}((\int \mathit{fuel} = 0 \sqcap \int e_{(i,j)} = 1) \sqsupset \mathsf{G} \int e_{(i,j)} = 1)$
6. The robot begins at cell $(1, 1)$, and its gas tank is full:
   $\int e_{(1,1)} = 1 \sqcap \int \mathit{fuel} = 1$

7. The robot will not revisit any cell:
   $(\sqcup_{i,j}\mathsf{G}(\int e_{(i,j)} = 1 \sqsupset \mathsf{XG}(\int \mathit{fuel} > 0 \sqsupset \int e_{(i,j)} = 0)))$
8. All paths should eventually lead to the target location:
   $\mathsf{F}(\int e_{(4,4)} = 1)$

So, using the SAT algorithm, one is able to decide whether the set of constraints has a model. Note that if we omit the last axiom, the model generated is the model of all possible paths *Healer 2.0* can take. However, if the last axiom is included, the generated model is a trimmed down version of the earlier model, such that all paths are viable and the robot can choose from any of them. Either way, the developers obtain their unknown distribution of the fuel supplies.

6 Conclusions

The proposed logic can be used to specify any type of probabilistic interaction provided that there are a finite number of probability distributions that model the problem. So the example provided in Section 5 can be extended to consider robots with random movements, since we only consider finite amount of fuel. This finitary constraint is necessary for the SAT algorithm to remain decidable, as any attempt to go beyond leads easily to undecidability. Note that one advantage of this approach compared to planning using SAT over propositional logic is that we do not require a bound on time, although both methods require to put a bound on the number of states (in our case in the number of probability distribution). Finally, completeness may help to derive additional properties about the behavior of all solutions.

Acknowledgements

We would like to thank Nick Papanikolaou for suggesting improvements to this paper. Some of this work was carried out when the fourth author was on sabbatical at IST/IT, Lisbon. This work was partially supported by Instituto de Telecomunicações, FCT and EU FEDER through PTDC, namely via the ComFormCrypt project (PTDC/EIA-CCO/113033/2009). The first author was also funded by FCT via PhD grant SFRH/BD/73315/2010 and the second author was funded by a PostDoc grant by FCT.

References

[1] S. Basu, R. Pollack, and R. Marie-Françoise. *Algorithms in Real Algebraic Geometry.* Springer, 2003.

[2] Blai Bonet. Learning depth-first search: A unified approach to heuristic search in deterministic and non-deterministic settings, and its application to mdps. In *ICAPS'06*, pages 3–23, 2006.

[3] Ronen I. Brafman and Jrg Hoffmann. Conformant planning via heuristic forward search: A new approach. In *ICAPS'04*, pages 355–364, 2004.

[4] Tomás Brázdil, Vojtech Forejt, Jan Kretínský, and Antonín Kucera. The satisfiability problem for probabilistic CTL. In *LICS*, pages 391–402, 2008.

[5] Frank Ciesinski and Marcus Größer. On probabilistic computation tree logic. In *Validation of Stochastic Systems*, pages 147–188, 2004.

[6] R. Fagin and J. Y. Halpern. Reasoning about knowledge and probability. *J. ACM*, 41(2):340–367, 1994.

[7] R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Inf. and Comp.*, 87(1/2):78–128, 1990.

[8] Rob Gerth, Doron Peled, Moshe Y. Vardi, and Pierre Wolper. Simple on-the-fly automatic verification of linear temporal logic. In *Proc. XV IFIP WG6.1*, pages 3–18, 1996.

[9] Henry Kautz and Bart Selman. Planning as satisfiability. In *ECAI*, pages 359–363, 1992.

[10] Florent Konigsbuch, Guillaume Infantes, and Ugur Kuter. RFF: a robust, ff-based MDP planning algorithm for generating policies with low probability of failure. In *ICAPS'08*, 2008.

[11] Dexter Kozen. Results on the propositional mu-calculus. *Theor. Comput. Sci.*, 27:333–354, 1983.

[12] Dexter Kozen and Rohit Parikh. A decision procedure for the propositional mu-calculus. In *Proceedings of the Carnegie Mellon Workshop on Logic of Programs*, pages 313–325, London, UK, 1984. Springer-Verlag.

[13] P. Mateus and A. Sernadas. Reasoning about quantum systems. In *JELIA'04*, volume 3229 of *Lecture Notes in AI*, pages 239–251. Springer-Verlag, 2004.

[14] P. Mateus and A. Sernadas. Weakly complete axiomatization of exogenous quantum propositional logic. *Inf. and Comp.*, 204(5):771–794, 2006.

[15] P. Mateus, A. Sernadas, and C. Sernadas. Exogenous semantics approach to enriching logics. In *Essays on the Foundations of Mathematics and Logic*, volume 1, pages 165–194. Polimetrica, 2005.

[16] Paulo Mateus, António Pacheco, Javier Pinto, Amílcar Sernadas, and Cristina Sernadas. Probabilistic situation calculus. *Ann. Math. Artif. Intell.*, 32(1-4):393–431, 2001.

[17] Jussi Rintanen. Planning as satisfiability: Heuristics. *Artif. Intell.*, 193:45–86, 2012.

[18] A. Sistla and E. Clarke. The complexity of propositional linear temporal logics. *J. ACM*, 32(3):733, 1985.

[19] Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification. In *Proc. 1st LICS*, pages 332–344, Cambridge, June 1986.

[20] Igor Walukiewicz. Completeness of Kozen's axiomatisation of the propositional mu-calculus. In *LICS '95*, page 14, Washington, DC, 1995. IEEE Computer Society.