The Stochastic Decision Logic

Gavin Rens[1] Thomas Meyer[1] Gerhard Lakemeyer[2]

December 7, 2013

[1]Centre for Artificial Intelligence Research, University of KwaZulu-Natal, and CSIR Meraka, South Africa {grens,tmeyer}@meraka.org.za
[2]RWTH Aachen University, Germany gerhard@cs.rwth-aachen.de

Abstract

We present a decidable logic in which POMDPs can be specified with compact representations, and queries can be posed about (i) the degree of belief in a propositional sentence after an arbitrary finite number of actions and observations and (ii) the utility of a finite sequence of actions after a number of actions and observations. The task of the logic is to check whether a query (stated in the language of the logic) follows from a knowledge base (KB), which is typically a POMDP model specification (also stated in the language of the logic). The main contribution of this work is that the POMDP model specification is allowed to be partial or incomplete with no restriction on the lack of information specified for the model. The KB may even contain information about non-initial beliefs. Essentially, entailment of arbitrary queries (expressible in the language) can be answered. A sound, complete and terminating decision procedure is provided.

Contents

1 Introduction and Motivation

Robots and intelligent agents often have to cope with uncertainty, nondeterminism and noise in their environments. The actuators and sensors of agents and robots are not perfect either; actuators do not always have the exact intended effects and sensors may return erroneous signals, depending on their error profiles and noise in the environment. The robot and its environment form a system. We refer to such stochastic (random) systems as stochastic domains.

One formalism for modelling agents in stochastic domains is the *partially observable Markov decision process* (POMDP) [Aström, 1965, Smallwood and Sondik, 1973, Monahan, 1982, Lovejoy, 1991, Boutilier et al., 1999]. As POMDPs are *decision processes*, there are also procedures to determine 'good' sequences of actions, given a POMDP model and notions of utility. Finding such sequences is called *solving* a POMDP. There are different procedures depending on the kind of POMDP solution sought. The popularity of the POMDP approach is, arguably, due to the relative simplicity and intuitiveness of its model (see the next section) and its general applicability to a wide range of stochastic domains.

Symbolic logic can express information unambiguously and usually compactly. Logic is also powerful in that, given a set of information in the language of a logic, there are usually procedures which can determine whether some other information follows (in a mathematical sense) from the given information. Such 'entailment queries' are useful when an agent wants to make decisions based on its (arbitrary) background knowledge (stored as logical expressions).

Recently, some researchers have investigated formal languages for compactly representing POMDPs [Geffner and Bonet, 1998a, Wang and Schmolze, 2005, Sanner and Kersting, 2010]. They also mention that with a logical language for specifying models, decision-making algorithms can exploit the structure found in these logical specifications.

In this report, we propose the *Stochastic Decision Logic* (SDL), combining the benefits of POMDP theory and logic for posing entailment queries about POMDP models. Full-scale planning will not be considered here. However, projections concerning epistemic situations and expected rewards will be possible. We provide a procedure to determine whether some hypothesised situation follows from a knowledge base of the system and some beliefs corresponding to the real system state or beliefs possibly in conflict with the real system state.

Traditionally, to make any deductions in POMDP theory, a POMDP model must be completely specified. *A major contribution of this work is that it allows the user to determine whether or not a set of sentences is entailed by an arbitrarily precise specification of a POMDP model.* By "arbitrarily precise specification" we mean that the transition function, the perception function, the reward function or the initial belief-state may not be completely defined by the logical specification provided. Moreover, the SDL is *decidable* with respect to entailment. That is, there is a sound, complete and terminating procedure for deciding whether a sentence logically follows from a model.

With SDL, an agent will be able to determine its degree of belief in a state of affairs after any number of actions and associated observations, and the agent will be able to determine the utility of performing any sequence of actions.

Most syntactic and semantic elements of the SDL have their foundations in the

Specification Logic of Actions with Probability [Rens et al., 2013] and the Specification Logic of Actions and Observations with Probability [Rens et al., 2014]. However, one cannot reason about degrees of belief or expected rewards (utility) in those logics.

We now sketch an 'oil-drinking' scenario, where actions and observations may be stochastic. The examples towards the end of the paper are based on this scenario. Imagine a robot that is in need of an oil refill. There is an open can of oil on the floor within reach of its gripper. If there is nothing else in the robot's gripper, it can grab the can (or miss it, or knock it over) and it can drink the oil by lifting the can to its 'mouth' and pouring the contents in (or miss its mouth and spill). The robot may also want to confirm whether there is anything left in the oil-can by weighing its contents with its 'weight' sensor. And once holding the can, the robot may wish to replace it on the floor. There are also rewards and costs involved, which are explained in the examples section.

The domain is (partially) formalized as follows. The robot has the set of (intended) actions $\mathcal{A} = \{\mathtt{grab}, \mathtt{drink}, \mathtt{weigh}, \mathtt{replace}\}$ with expected intuitive meanings. The robot can perceive observations only from the set $\Omega = \{\mathtt{Nil}, \mathtt{Light}, \mathtt{Medium}, \mathtt{Heavy}\}$. Intuitively, when the robot performs a `weigh` action (i.e., it activates its 'weight' sensor) it will perceive either `Light`, `Medium` or `Heavy`; for other actions, it will perceive `Nil`. The robot experiences its world (domain) through two Boolean features: $\mathcal{F} = \{\mathtt{full}, \mathtt{holding}\}$ meaning respectively that the robot believes the oil-can is full and that it is currently holding something in its gripper. $\mathbf{B}\varphi \geq p$ is read 'The degree of belief in φ is greater than or equal to p'. $\mathbf{U}\Lambda > r$ is read 'The utility of performing Λ is greater than r'. Given a complete formalization $\mathcal{K}$ of the scenario sketched here, a robot may have the following queries:

- Is the degree of belief that I'll have the oil-can in my gripper greater than or equal to 0.9, after I attempt grabbing it twice in a row? That is, does

$$\{\mathtt{grab}, \mathtt{obsNil}\}\{\mathtt{grab}, \mathtt{obsNil}\}\mathbf{B}(\mathtt{holding}) \geq 0.9$$

  follow from $\mathcal{K}$?

- After grabbing the can, then perceiving that it has medium weight, is the utility of drinking the contents of the oil-can, then placing it on the floor, more than 6 units? That is, does

$$\{\mathtt{grab}, \mathtt{obsNil}\}\{\mathtt{weigh}, \mathtt{obsMedium}\}\mathbf{U}\{\mathtt{drink}\}\{\mathtt{replace}\} > 6$$

  follow from $\mathcal{K}$?

2 Finite Horizon Partially Observable Markov Decision Processes

In partially observable Markov decision processes (POMDPs), actions have nondeterministic results as in (fully observable) MDPs, but observations are uncertain. In other words, the effect of some chosen action is somewhat unpredictable, yet may be predicted with a probability of occurrence. However, in POMDPs, the world is not directly observable: some data are observable and the agent infers how likely it is that the state of the world is in some specific state. The agent thus believes to some degree—for each possible state—that it is in that state, but it is never certain exactly which state it is in. In fact, the agent maintains a probability distribution over the states reflecting its conviction for being in a state, for each state.

The theory in this paper can be found in [Cassandra et al., 1994], [Kaelbling et al., 1998], [Boutilier et al., 1999] and [Russell and Norvig, 2003].

2.1 Theory

Formally, a POMDP is a tuple $\langle S, A, T, R, Z, O, b^0 \rangle$. The components of the POMDP model are:

- $S = \{s_1, s_2, \ldots, s_n\}$ is a finite set of states of the world (that the agent can be in);
- $A = \{a_1, a_2, \ldots, a_k\}$ is a finite set of actions (that the agent can choose to execute);
- $T : \mathcal{S} \times A \times S \to (\mathbb{R} \cap [0, 1])$ is the *state-transition function*, a probability distribution Π over all (world state, agent action, world state) triples (we write $T(s, a, s')$ to mean the probability of being in s' after performing action a in state s);
- $R : A \times S \to \mathbb{R}$ is the *reward function*, giving the immediate reward gained by the agent, for any state and agent action (we write $R(a, s)$ to mean the reward gained for executing a while in state s);
- $Z = \{z_1, z_2, \ldots, z_m\}$ is a finite set of observations the agent can experience of its world; to be clear, an observation is not a sensing action, it is the information/data the agent has *after* or *provided by* each action with a sensory aspect;
- $O : S \times A \times Z \to (\mathbb{R} \cap [0, 1])$ is the *observation function*, giving for each agent action and the resulting world state, a probability distribution over observations (we write $O(s', a, z)$ to denote the probability of observing z in state s' resulting from performing action a in some other state; O represents the agent's trust in its observations, given the context of each observation;
- b^0 is the initial probability distribution over all world states in S.

Let b be a total function from S into $\mathbb{R}$. Each state s is associated with a probability $b(s) = p \in \mathbb{R}$, such that b is a probability distribution over the set S of all states. b can be called a *belief-state*.

An important function in POMDP theory, is the function that updates the agent's belief-state, or the *state estimation* function SE. $SE(a, z, b) = b_n$ is defined as

$$b_n(s') = \frac{O(s', a, z) \sum_{s \in S} T(s, a, s') b(s)}{Pr(z \mid a, b)}, \tag{1}$$

where $b_n(s')$ is the probability of the agent being in state s' in the 'new' belief-state b_n, relative to a, z and the 'old' belief-state b.

$$Pr(z \mid a, b) = \sum_{s' \in \mathcal{S}} O(s', a, z) \sum_{s \in \mathcal{S}} T(s, a, s') b(s) \tag{2}$$

in the denominator acts as a normalizer here. Equation (1) is derived from the Bayes Rule. Notice that $SE(\cdot)$ requires a belief-state, an action and an observation as inputs to determine the new belief-state.

When the states an agent can be in are *belief*-states (as opposed to objective, single states in S), the reward function R must be lifted to operate over belief-states. The *expected* reward $\rho(a, b)$ for performing an action a in a belief-state b is defined as

$$\rho(a, b) \overset{def}{=} \sum_{s \in S} R(a, s) b(s). \tag{3}$$

Let the *planning horizon* h be the number of steps into the future that the agent will consider each time it selects its next action; h can also be called the *look-ahead depth*; in the recursive equations below, h can be thought of as the number of *steps to go*. Let a *policy* π be a set of h pairs (h', a) where h' is the number of steps to go and a is the action the agent will take at that sage. In other words,

$$\pi = \{(h, a_1), (h-1, a_2), (h-2, a_3), \ldots, (1, a_h)\}.$$

The *value* $V^\pi(b, h)$ of a belief-state b is the expected value of future states, given the actions selected at each step using policy π, until the horizon h is reached. In the equations below, $Pr(z \mid a, b)$ can be read as the probability of reaching belief-state $b_n = SE(a, z, b)$[1]. Future rewards are deemed less important, because the future is less certain than the present; future rewards are thus discounted by a discount factor $0 < \gamma < 1$.

$$\begin{aligned} V^\pi(b, h) &\overset{def}{=} \rho(\pi(h), b) + \gamma \sum_{z \in Z} Pr(z \mid \pi(h), b) V^\pi(SE(\pi(h), z, b), h-1) \\ V^\pi(b, 1) &\overset{def}{=} \rho(\pi(1), b). \end{aligned} \tag{4}$$

The *optimal* value[2] $V^*(b, h)$ of a belief-state b assumes that at each step the action that will maximize the state's value will be selected. The policy is thus implicit and needs not be provided.

$$\begin{aligned} V^*(b, h) &\overset{def}{=} \max_{a \in A} \Big[\rho(a, b) + \gamma \sum_{z \in Z} Pr(z \mid a, b) V^*(SE(a, z, b), h-1) \Big] \\ V^*(b, 1) &\overset{def}{=} \max_{a \in A} \rho(a, b). \end{aligned} \tag{5}$$

While V^* denotes the optimal state-value, the function Q^* denotes the state-action value. The value $Q^*(a, b, h)$ of action a is the value of the current belief-state for executing a, plus the total expected value of belief-states reached thereafter:

$$Q^*(a, b, h) \overset{def}{=} \rho(a, b) + \gamma \sum_{z \in Z} Pr(z \mid a, b) V^*(SE(a, z, b), h-1) \tag{6}$$

[1]So $Pr(z \mid a, b)$ has two readings: as the normalizing factor in the state estimation function and as the probability of reaching the new belief-state.

[2]The value is optimal with respect the a finite horizon. For true optimality, a search should be carried out to a depth equal to the number of action the agent will perform in its lifetime. Optimality in this sense is not applicable here.

Note that $V^*(b, h) = \max_a Q^*(a, b, h)$. If an agent is not in possession of a policy, it can select its next action a^* using Equation 7.

$$a^* = \arg\max_{a \in \mathcal{A}} Q^*(a, b, h) \tag{7}$$

where b is the current belief-state and h is the number of steps to go.

2.2 Making Decisions

We show how to make decisions with POMDPs as related to decision-making in the SDL. We shall assume in this section that a complete POMDP model is given, including the initial belief-state b^0. A decision is made in the belief-state b' reached after a sequence of actions and observations. We consider decisions conditioned on two kinds of statements, (i) the utility or expected reward of a sequence of actions starting in b' and (ii) the probability of being in a particular set of states in b'.

Suppose a sequence of actions and observations is represented as

$$\{a_1 + z_1\}\{a_2 + z_2\} \cdots \{a_y + z_y\}.$$

Determining b' for such a sequence, is determined by finding $b^1 = SE(\alpha_1, \varsigma_1, b^0)$, then $b^2 = SE(\alpha_2, \varsigma_2, b^1)$, then $\ldots$, then $b' = SE(a_y, z_y, b^{y-1})$.

Consider the first kind of statement. Suppose an agent wants to decide which action to execute next, and it is willing or able to reason by projection h steps into the future. Let the h steps be represented as

$$\{a_1\}\{a_2\} \cdots \{a_h\}.$$

The utility of performing each action in the sequence in succession is defined by

$$U(\{a_1\}\{a_2\} \cdots \{a_h\}, b') \stackrel{def}{=} \rho(a_1, b') + \gamma \sum_{z \in Z} Pr(z \mid a_1, b') U(\{a_2\} \cdots \{a_h\}, SE(a_1, z, b')),$$

where $U(\{a_h\}, b') \stackrel{def}{=} \rho(a_h, b')$. Suppose Ξ^h is every possible sequence of actions of length h. Then

$$\max_{\{a_1\}\{a_2\} \cdots \{a_h\} \in \Xi^h} U(\{a_1\}\{a_2\} \cdots \{a_h\}, b') = V^*(b', h).$$

A straight-forward and practical way to use $U(\cdot)$ is to decide which of several possible sequences of actions to execute in b'. Suppose the agent is considering executing either $\{a_1\}\{a_1\}$, $\{a_1\}\{a_2\}$, $\{a_2\}\{a_1\}$ or $\{a_2\}\{a_2\}$. Then it would determine $U(\{a_1\}\{a_1\}, b') = r_{11}$, $U(\{a_1\}\{a_2\}, b') = r_{12}$, $U(\{a_2\}\{a_1\}, b') = r_{21}$ and $U(\{a_2\}\{a_2\}, b') = r_{22}$, and decide to execute the sequence associated with $max\{r_{11}, r_{12}, r_{21}, r_{22}, \}$.

Let $S' \subseteq S$ be a set of states. Making decisions based on the second kind of statement is unusual in POMDP theory. The SDL provides for expressing such statements quite naturally. The probability with which the system/agent is in one of the states in S' is defined by

$$B(S', b) \stackrel{def}{=} \sum_{s \in S'} b(s).$$

One example of how an agent could use such information follows. Suppose the agent knows (or has calculated) that a sequence Ξ of actions will land it in one of the belief-states in set $\mathcal{B}^{\Xi}$, and the agent never wants to be in a state in S' with a probability greater than p. If for any one of $b' \in \mathcal{B}^{\Xi}, B(S', b') > p$, then the agent should not execute Ξ.

2.3 Examples

Let $F = \{f_1, f_2, \ldots, f_m\}$ be a set of Boolean features[3] of interest and $S = \{s_1, s_2, \ldots, s_n\}$ the $2^{|F|}$ set of states induced from F. For instance, if $F = \{f_1, f_2\}$, then $S = \{\{(f_1,0),(f_2,0)\}, \{(f_1,1),(f_2,0)\}, \{(f_1,0),(f_2,1)\}, \{(f_1,1),(f_2,1)\}\}$, where pair $(f,1)$ is in state s if f is a feature of the domain/system when it is in state s, else pair $(f,0)$ is in state s. Let $b' = \{(s_1,p_1),(s_2,p_2),\ldots,(s_n,p_n)\}$ be some belief-state.

Based on the oil-drinking scenario, let f stand for `fill` and let h stand for `holding`. Let g stand for `grab`, d for `drink` and w for `weigh`. Let N stand for `Nil`, L for `Light`, M for `Medium` and H for `Heavy`}. We define the following POMDP model.

- $S = \{\{\}, \{f\}, \{h\}, \{f,h\}\}$;
- $A = \{g, d, w\}$;
- Zero probability transitions are not shown.

$$
\begin{aligned}
&T(\{\}, g, \{\}) = 0.1\\
&T(\{\}, g, \{h\}) = 0.1\\
&T(\{\}, g, \{f,h\}) = 0.8\\
&T(\{f\}, g, \{\}) = 0.1\\
&T(\{f\}, g, \{h\}) = 0.1\\
&T(\{f\}, g, \{f,h\}) = 0.8\\
&T(\{h\}, d, \{\}) = 0.05\\
&T(\{h\}, d, \{h\}) = 0.95\\
&T(\{f,h\}, d, \{\}) = 0.05\\
&T(\{f,h\}, d, \{h\}) = 0.95\\
&T(\{\}, w, \{\}) = 1\\
&T(\{f\}, w, \{f\}) = 1\\
&T(\{h\}, w, \{h\}) = 1\\
&T(\{f,h\}, w, \{f,h\}) = 1
\end{aligned}
$$

- $$
\begin{aligned}
&R(g, \{\}) = -6\\
&R(g, \{f\}) = -1\\
&R(g, \{h\}) = 9
\end{aligned}
$$

[3] Any state which can be described by a set of multi-valued features can be uniquely described by a sufficiently large set of Boolean features.

$$R(g, \{f, h\}) = -1$$
$$R(d, \{\}) = -6$$
$$R(d, \{f\}) = -1$$
$$R(d, \{h\}) = 9$$
$$R(d, \{f, h\}) = -1$$
$$R(w, \{\}) = -5.8$$
$$R(w, \{f\}) = -2$$
$$R(w, \{h\}) = 9.2$$
$$R(w, \{f, h\}) = -2$$

- $Z = \{N, L, M, H\}$
- Zero probability observations are not shown.

$$O(\{\}, g, N) = 1$$
$$O(\{\}, d, N) = 1$$
$$O(\{\}, w, L) = \frac{1}{3}$$
$$O(\{\}, w, M) = \frac{1}{3}$$
$$O(\{\}, w, H) = \frac{1}{3}$$
$$O(\{f\}, g, N) = 1$$
$$O(\{f\}, d, N) = 1$$
$$O(\{f\}, w, L) = \frac{1}{3}$$
$$O(\{f\}, w, M) = \frac{1}{3}$$
$$O(\{f\}, w, H) = \frac{1}{3}$$
$$O(\{h\}, g, N) = 1$$
$$O(\{h\}, d, N) = 1$$
$$O(\{h\}, w, L) = 0.5$$
$$O(\{h\}, w, M) = 0.3$$
$$O(\{h\}, w, H) = 0.2$$
$$O(\{f, h\}, w, L) = 0.1$$
$$O(\{f, h\}, w, M) = 0.2$$
$$O(\{f, h\}, w, H) = 0.7$$

- $b^0 = \{(\{f, h\}, 0.35), (\{f\}, 0.35), (\{h\}, 0.2), (\{\}, 0.1)\}$.

First, we determine whether $B(\{\{f, h\}, \{h\}\}, b^2) > 0.85$, where $b^2 = SE(w, M, b^1)$ and $b^1 = SE(g, N, b^0)$. In other words, we want to determine whether the probability of being in a state where feature `holding` is true is greater than 0.85 after executing the sequence $\{g + N\}\{w + M\}$ in the initial belief-state. It could be that the robot is programmed to decide to grab the oil-can and then weigh it, given a positive result of the query. Using Equation (1), one can calculate that

$$b^1 = \{(\{f,h\}, 0.\overline{81}), (\{f\}, 0), (\{h\}, 0.\overline{09}), (\{\}, 0.\overline{09})\}$$

and

$$b^2 = \{(\{f,h\}, 0.72973), (\{f\}, 0), (\{h\}, 0.12162), (\{\}, 0.14865)\}$$

Finally, we see that

$$0.72973 + 0.12162 > 0.85$$

There is a subtlety in the interpretation of this result: It is not the case that the probability of grabbing the can and then weighing it and perceiving that it is of medium weight and then still holding the can is greater than 0.85. It is the case that IF the robot grabs the oil-can (and perceives nothing/`Nil`) and IF it weighs the can and IF it is perceived to be medium, THEN the probability of still holding the can is greater than 0.85.

Second, we determine whether $U(\{d\}\{d\}, b^1) \leq 7$, where $b^1 = SE(g, N, b^0)$. In other words, we want to determine whether the utility of drinking twice in a row is less than or equal to 7 units after grabbing the oil-can and perceiving nothing (`Nil`) in the initial belief-state. For simplicity, assume $\gamma = 1$.

$$U(\{d\}\{d\}, b^1) = \rho(d, b^1) + \sum_{z \in Z} Pr(z \mid d, b^1) U(\{d\}, SE(d, z, b^1)). \tag{8}$$

Notice that $Pr(L \mid d, b^1) = Pr(M \mid d, b^1) = Pr(H \mid d, b^1) = 0$, because $O(s, d, L) = O(s, d, M) = O(s, d, H) = 0$ for all states s. So (8) becomes

$$\begin{aligned}
U(\{d\}\{d\}, b^1) &= \rho(d, b^1) + Pr(N \mid d, b^1) U(\{d\}, SE(d, z, b^1)) \\
&= \rho(d, b^1) + Pr(N \mid d, b^1) U(\{d\}, b^{2'}) \\
&= \rho(d, b^1) + Pr(N \mid d, b^1) \rho(d, b^{2'}) \\
&= 0.\overline{81}(-1) + 0(-1) + 0.\overline{09}(9) + 0.\overline{09}(-6) + \\
&\qquad 0.\overline{90}(0(-1) + 0(-1) + 0.95(9) + 0.05(-6)) \\
&= 6.95455
\end{aligned}$$

where $Pr(N \mid d, b^1) = 0.\overline{90}$ and $b^{2'} = \{(\{f,h\}, 0), (\{f\}, 0), (\{h\}, 0.95), (\{\}, 0.05)\}$.

Suppose the robot is programmed to never drink twice in a row if the utility of doing so is less than 7 units. Then the robot knows that if it grabs the oil-can (and certainly perceives `Nil`), it will be in a belief-state where it should not drink twice in a row.

3 The Stochastic Decision Logic

First, the syntax of the logic is presented, then its semantics. The last subsection discusses the correspondence between POMDPs and the SDL.

3.1 Syntax

The vocabulary of our language contains six sorts of objects of interest:

1. a finite set of *fluents* $\mathcal{F} = \{f_1, \ldots, f_n\}$,
2. a finite set of names of atomic *actions* $\mathcal{A} = \{\alpha_1, \ldots, \alpha_n\}$,
3. a countable set of *action variables* $V_{\mathcal{A}} = \{v^a_1, v^a_2, \ldots\}$,
4. a finite set of names of atomic *observations* $\Omega = \{\varsigma_1, \ldots, \varsigma_n\}$,
5. a countable set of *observation variables* $V_\Omega = \{v^o_1, v^o_2, \ldots\}$.
6. all *real numbers* $\mathbb{R}$,

From now on, we denote $\mathbb{R} \cap [0, 1]$ as $\mathbb{R}_{[0,1]}$. We refer to elements of $\mathcal{A} \cup \Omega$ as *constants*. We work in a multi-modal setting, in which we have modal operators $[\alpha]$, one for each $\alpha \in \mathcal{A}$. And $\{\alpha + \varsigma\}$ is a *belief update operator* (or *update operator* for short). Intuitively, $\{\alpha + \varsigma\}\Psi$ means 'Ψ holds in the belief-state resulting from performing action α and then perceiving observation ς'. For instance, $\{\alpha_1 + \varsigma_1\}\{\alpha_2 + \varsigma_2\}$ expresses that the agent executes α_1 then perceives ς_1 then executes α_2 then perceives ς_2. $\mathbf{B}$ is a modal operator for belief and $\mathbf{U}$ is a modal operator for utility.

We first define a language $\mathcal{L}$, then a useful sublanguage $\mathcal{L}_{SDL} \subset \mathcal{L}$. The reason why we define $\mathcal{L}$ is because it is easier to define the truth condition for $\mathcal{L}$; the truth conditions for $\mathcal{L}_{SDL}$ then follow directly. First the propositional fragment.

Definition 3.1 $\varphi ::= f \mid \top \mid \neg\varphi \mid \varphi \wedge \varphi$, *where* $f \in \mathcal{F}$.

Then the fragment Φ used in formulae of the form $\varphi \Rightarrow \Phi$ (cf. Def. 3.3).

Definition 3.2 *Let* $\alpha \in (V_{\mathcal{A}} \cup \mathcal{A})$, $v^a \in V_{\mathcal{A}}$, $\varsigma \in (V_\Omega \cup \Omega)$, $v^o \in V_\Omega$, $p \in \mathbb{R}_{[0,1]}$, $r \in \mathbb{R}$ *and* $\bowtie \in \{<, \leq, =, \geq, >\}$.

$$\begin{aligned}\Phi ::= {} & \varphi \mid \alpha = \alpha \mid \varsigma = \varsigma \mid \mathit{Reward}(r) \mid \mathit{Cost}(\alpha, r) \mid [\alpha]\varphi \bowtie p \mid \\ & (\alpha|\varsigma) \bowtie p \mid (\forall v^a)\Phi \mid (\forall v^o)\Phi \mid \neg\Phi \mid \Phi \wedge \Phi.\end{aligned}$$

where φ *is from Definition 3.1.*

$[\alpha]\varphi \bowtie p$ is read 'The probability of reaching a φ-world after executing $\alpha \bowtie p$'. $(\varsigma|\alpha) \bowtie p$ is read 'The probability of perceiving ς, given α was performed $\bowtie p$'.

Definition 3.3 *Let* $\alpha \in (V_{\mathcal{A}} \cup \mathcal{A})$, $v^a \in V_{\mathcal{A}}$, $\varsigma \in (V_\Omega \cup \Omega)$, $v^o \in V_\Omega$, $p \in \mathbb{R}_{[0,1]}$, $r \in \mathbb{R}$ *and* $\bowtie \in \{<, \leq, =, \geq, >\}$. *The language of* $\mathcal{L}$ *is defined as* Θ*:*

$$\begin{aligned}\Lambda ::= {} & \{\alpha\} \mid \Lambda\{\alpha\} \\ \Theta ::= {} & \top \mid \alpha = \alpha \mid \varsigma = \varsigma \mid \mathit{Poss}(\alpha, \varsigma) \mid \mathbf{B}\varphi \bowtie p \mid \\ & \mathbf{U}\Lambda \bowtie r \mid \varphi \Rightarrow \Phi \mid \{\alpha + \varsigma\}\Theta \mid \\ & (\forall v^a)\Theta \mid (\forall v^o)\Theta \mid \neg\Theta \mid \Theta \wedge \Theta \mid \Theta \vee \Theta,\end{aligned}$$

where φ *is from Definition 3.1 and* Φ *is from Definition 3.2.*

The scope of quantifier $(\forall v')$ is determined in the same way as is done in first-order logic. A variable v appearing in a formula Θ is said to be bound by quantifier $(\forall v')$ if and only if v is the same variable as v' and is in the scope of $(\forall v')$. If a variable is not bound by any quantifier, it is free. In $\mathcal{L}$, variables are not allowed to be free; they are always bound.

$Poss(\alpha, \varsigma)$ is read 'It is possible to execute α and then perceive ς'. $\mathbf{B}\varphi \bowtie p$ is read 'The degree of belief in $\varphi \bowtie p$'. Performing $\Lambda = \{\alpha_1\}\{\alpha_2\}\cdots\{\alpha_z\}$ means that $\{\alpha_1\}$ is performed, then $\{\alpha_2\}$ then ... then $\{\alpha_z\}$. $\mathbf{U}\Lambda$ is thus read 'The utility of performing $\Lambda \bowtie r$'. Evaluating Ψ after a sequence of z update operations, means that Ψ will be evaluated after the agent's belief-state has been updated according to the sequence $\{\alpha^1 + \varsigma^1\}\cdots\{\alpha^z + \varsigma^z\}$.

Definition 3.4 *The language of SDL, denoted $\mathcal{L}_{SDL}$, is the subset of formulae of $\mathcal{L}$ excluding formulae containing subformulae of the form $\neg(\varphi \Rightarrow \Phi)$.*

Note that, for instance, $\neg(\varphi \Rightarrow \Phi) \land (\varphi' \Rightarrow \Phi') \land \neg Poss(\alpha, \varsigma) \notin \mathcal{L}_{SDL}$, but $(\varphi \Rightarrow \Phi) \land (\varphi' \Rightarrow \Phi') \land \neg Poss(\alpha, \varsigma) \in \mathcal{L}_{SDL}$. And, for instance, $\neg(\forall v')(\varphi \Rightarrow \Phi) \lor (\varphi' \Rightarrow \Phi') \lor \neg(\forall v'')Poss(\alpha, \varsigma) \notin \mathcal{L}_{SDL}$, but $(\forall v')(\varphi \Rightarrow \Phi) \lor (\varphi' \Rightarrow \Phi') \lor \neg(\forall v'')Poss(\alpha, \varsigma) \in \mathcal{L}_{SDL}$.

$\bot$ abbreviates $\neg\top$, $\theta \rightarrow \theta'$ abbreviates $\neg\theta \lor \theta'$ and $\leftrightarrow$ abbreviates $(\theta \rightarrow \theta') \land (\theta' \rightarrow \theta)$. In grammars φ, Φ and χ, $\theta \land \theta'$ abbreviates $\neg(\theta \lor \theta')$, but in grammars Θ and Ψ, $\lor$ is defined directly. $\rightarrow$ and $\leftrightarrow$ have the weakest bindings, with $\Rightarrow$ just stronger; and $\neg$ the strongest. Parentheses enforce or clarify the scope of operators conventionally.

$c = c$ is an equality literal, $Reward(r)$ is a reward literal, $Cost(\alpha, r)$ is a cost literal, $[\alpha]\varphi \bowtie p$ is a dynamic literal, $(\varsigma|\alpha) \bowtie p$ is a perception literal, and $\varphi \Rightarrow \Phi$ is a law literal. $Poss(\alpha, \varsigma)$ is an executability literal, $\mathbf{B}\varphi \bowtie p$ is a belief literal and $\mathbf{U}\Lambda \bowtie r$ is a utility literal. The negation of all these literals are also literals with the associated names.

Note that formulae with nested modal operators of the form $\mathbf{BB}\varphi$, $\mathbf{BBB}\varphi$, etc., $\mathbf{UU}\Lambda$, $\mathbf{UUU}\Lambda$, etc., $[\alpha][\alpha]\varphi$ and $[\alpha][\alpha][\alpha]\varphi$ et cetera are not in $\mathcal{L}_{SDL}$. However, formulae of the form $\{\alpha+\varsigma\}(\{\alpha'+\varsigma'\}\Psi' \land \{\alpha''+\varsigma''\}\Psi'')$, $\{\alpha+\varsigma\}(\Psi \land \{\alpha'+\varsigma'\}\{\alpha''+\varsigma''\}\Psi')$ for instance, are in $\mathcal{L}_{SDL}$.

3.2 Semantics

Let $w : \mathcal{F} \mapsto \{0, 1\}$ be a total function that assigns a truth value to each fluent. We call w a *world*. Let C be the set of $2^{|\mathcal{F}|}$ *conceivable worlds*, that is, all possible functions w.

Definition 3.5 *An SDL structure is a tuple $\mathcal{D} = \langle R, Q, U \rangle$ such that*

- *$R : \mathcal{A} \mapsto R_\alpha$, where $R_\alpha : (C \times C) \mapsto \mathbb{R}_{[0,1]}$ is a total function from pairs of worlds into the reals; That is, R is a mapping that provides an accessibility relation R_α for each action $\alpha \in \mathcal{A}$; For every $w^- \in C$, it is required that either $\sum_{w^+ \in C} R_\alpha(w^-, w^+) = 1$ or $\sum_{w^+ \in C} R_\alpha(w^-, w^+) = 0$.*
- *$Q : \mathcal{A} \mapsto Q_\alpha$, where $Q_\alpha : (C \times \Omega) \mapsto \mathbb{R}_{[0,1]}$ is a total function from pairs in $C \times \Omega$ into the reals; That is, Q is a mapping that provides a perceivability relation Q_α for each action $\alpha \in \mathcal{A}$; For all $w^+ \in C$, if there exists a $w^- \in C$ such that $R_\alpha(w^-, w^+) > 0$, then $\sum_{\varsigma \in \Omega} Q_\alpha(w^+, \varsigma) = 1$, else $\sum_{\varsigma \in \Omega} Q_\alpha(w^+, \varsigma) = 0$;*

- *U is a pair $\langle Re, Co\rangle$, where $Re : C \mapsto \mathbb{R}$ is a reward function and Co is a mapping that provides a cost function $Co_\alpha : C \mapsto \mathbb{R}$ for each $\alpha \in \mathcal{A}$.*

In POMDPs, the agent does not know in which world $w \in C$ it actually is, but for each w it has a degree of belief that it is in that world. Let $b : C \mapsto \mathbb{R}_{[0,1]}$ be a probability distribution over C, referred to as a *belief-state*. The degree of belief in w is denoted by the probability measure $b(w)$. We refer to all probability mass functions b over C as the set P.

As an agent acts and evolves, what it believes changes, that is, b changes for each agent activity. We assume the agent remains in one physical environment for its lifetime, and we assume that its sets $\mathcal{A}$ and Ω remain the same. The fact that the agent does not change environments has the consequence that the structure modeling the agent and its environment also remain static.

Definition 3.6 *The probability of reaching the next belief-state b', given α and ς is*

$$P_{NB}(\alpha, \varsigma, b) = \sum_{w' \in C} Q_\alpha(\varsigma, w') \sum_{w \in C} R_\alpha(w, w') b(w).$$

$P_{NB}(\cdot)$ has the same intuitive meaning as $Pr(z \mid a, b)$ (Eq. (2)).

Definition 3.7 *We define a* belief update *function $BU(\alpha, o, b) = b'$:*

$$b'(w') = \frac{Q_\alpha(w', \varsigma) \sum_{w \in C} R_\alpha(w, w') b(w)}{P_{NB}(\alpha, \varsigma, b)},$$

for $P_{NB}(\alpha, \varsigma, b) \neq 0$.

$BU(\cdot)$ has the same intuitive meaning as the state estimation function (Eq. (1)).

Given the opportunity to be slightly more clear about the specification of rewards in the SDL, we interpret $R(a, s)$ from the section reviewing POMDPs as $R(s) - C(a, s)$, where $R(s)$ provides the positive reward portion of $R(a, s)$ and $C(a, s)$ provides the punishment or cost portion. By this interpretation, we assume that simply being in a state has an intrinsic reward (independent of an action), however, that punishment is conditional on actions and the states in which they are executed. There are many other ways to interpret $R(a, s)$, and $R(a, s)$ is not even the most general reward function possible; a more general function is $R(s, a, s')$ meaning that rewards depend on a state s, an action executed in s and a state s' reached due to performing a in s. The SDL adopts one of several reasonable approaches. In the semantics of the SDL, we equate a state s with a world w and an action a as $\alpha \in \mathcal{A}$, and interpret $R(a, s)$ as $Re(w) - Co_\alpha(w)$.

We derive a reward function over belief-states for the SDL in a similar fashion as we did with Equation 3, however, including the notion of cost.

$$RC(\alpha, b) = \sum_{w \in C} (Re(w) - Co_\alpha(w)) b(w).$$

Definition 3.8 *Let $\alpha, \alpha' \in \mathcal{A}$, $\varsigma, \varsigma' \in \Omega$, $p \in \mathbb{R}_{[0,1]}$ and $r \in \mathbb{R}$. Let $f \in \mathcal{F}$ and let Θ be any sentence in $\mathcal{L}$. Let $\bowtie \in \{<, \leq, =, \geq, >\}$. We say $\Theta \in \mathcal{L}$ is* satisfied *at world w and belief-state b in SDL structure $\mathcal{D}$ (written $\mathcal{D}bw \models \Theta$) if and only if the following holds:*

$\mathcal{D}bw \models \top$ for all $w \in C$;

$\mathcal{D}bw \models f \iff w(f) = 1$;

$\mathcal{D}bw \models \neg\varphi \iff \mathcal{D}bw \not\models \varphi$;

$\mathcal{D}bw \models \varphi \wedge \varphi' \iff \mathcal{D}bw \models \varphi$ and $\mathcal{D}bw \models \varphi'$;

$\mathcal{D}bw \models \alpha = \alpha' \iff \alpha$ and α' are the same element;

$\mathcal{D}bw \models \varsigma = \varsigma' \iff \varsigma$ and ς' are the same element;

$\mathcal{D}bw \models \mathit{Reward}(r) \iff \mathit{Re}(w) = r$;

$\mathcal{D}bw \models \mathit{Cost}(\alpha, c) \iff \mathit{Co}_\alpha(w) = c$;

$\mathcal{D}bw \models [\alpha]\varphi \bowtie p \iff \sum_{\substack{w' \in C \\ \mathcal{D}bw' \models \varphi}} R_\alpha(w, w') \bowtie p$;

$\mathcal{D}bw \models (\varsigma|\alpha) \bowtie p \iff Q_\alpha(w, \varsigma) \bowtie p$;

$\mathcal{D}bw \models \neg\Phi \iff \mathcal{D}bw \not\models \Phi$;

$\mathcal{D}bw \models \Phi \wedge \Phi' \iff \mathcal{D}bw \models \Phi$ and $\mathcal{D}bw \models \Phi'$;

$\mathcal{D}bw \models \mathit{Poss}(\alpha, \varsigma) \iff P_{NB}(\alpha, \varsigma, b) \neq 0$;

$\mathcal{D}bw \models \mathbf{B}\varphi \bowtie p \iff \sum_{\substack{w' \in C \\ \mathcal{D}bw' \models \varphi}} b(w') \bowtie p$;

$\mathcal{D}bw \models \mathbf{U}\{\alpha\} \bowtie r \iff RC(\alpha, b) \bowtie r$;

$\mathcal{D}bw \models \mathbf{U}\{\alpha\}\Lambda \bowtie r \iff \Big(RC(\alpha, b) + \sum_{\substack{\varsigma \in \Omega \\ b' = BU(\alpha,\varsigma,b) \\ \mathcal{D}b'w \models \mathbf{U}\Lambda = r'}} P_{NB}(\alpha, \varsigma, b) \cdot r'\Big) \bowtie r$;

$\mathcal{D}bw \models \varphi \Rightarrow \Theta \iff$ for all$w' \in C$, $\mathcal{D}bw' \not\models \varphi$ or $\mathcal{D}bw' \models \Theta$;

$\mathcal{D}bw \models \{\alpha + \varsigma\}\Theta \iff P_{NB}(\alpha, \varsigma, b) \neq 0$ and $\mathcal{D}b'w \models \Theta$, where $b' = BU(\alpha, \varsigma, b)$;

$\mathcal{D}bw \models \neg\Theta \iff \mathcal{D}bw \not\models \Theta$;

$\mathcal{D}bw \models \Theta \wedge \Theta' \iff \mathcal{D}bw \models \Theta$ and $\mathcal{D}bw \models \Theta'$;

$\mathcal{D}bw \models \Theta \vee \Theta' \iff \mathcal{D}bw \models \Theta$ or $\mathcal{D}bw \models \Theta'$ or both;

$\mathcal{D}bw \models (\forall v^a)\Upsilon \iff \mathcal{D}bw \models \Upsilon|^{v^a}_{\alpha_1} \wedge \ldots \wedge \Upsilon|^{v^a}_{\alpha_n}$;

$\mathcal{D}bw \models (\forall v^o)\Upsilon \iff \mathcal{D}bw \models \Upsilon|^{v^o}_{\varsigma_1} \wedge \ldots \wedge \Upsilon|^{v^o}_{\varsigma_n}$,

where Υ is either Θ or Ψ, and we write $\Upsilon|^v_c$ to mean the formula Υ with all variables $v \in (V_\mathcal{A} \cup V_\Omega)$ appearing in it replaced by constant $c \in \mathcal{A} \cup \Omega$ of the right sort.

A sentence $\Psi \in \mathcal{L}_{SDL}$ is *satisfiable* if there exists a structure $\mathcal{D}$, a belief-state b and a world w such that $\mathcal{D}bw \models \Psi$, else Ψ is *unsatisfiable*. Let $\mathcal{K} \subset \mathcal{L}_{SDL}$. We say that $\mathcal{K}$ entails Ψ (denoted $\mathcal{K} \models \Psi$) if for all structures $\mathcal{D}$, all belief-states b, all $w \in C$: if $\mathcal{D}bw \models \kappa$ for every $\kappa \in \mathcal{K}$, then $\mathcal{D}bw \models \Psi$. When $\mathcal{K}$ is a finite subset of $\mathcal{L}_{SDL}$, it is easy to show that $\mathcal{K} \models \Psi \iff \bigwedge_{\kappa \in \mathcal{K}} \kappa \wedge \neg\Psi$ is unsatisfiable. The SDL decision procedure for entailment is based on this latter correspondence.

3.3 The Correspondence Between the SDL and POMDPs

In this section we investigate the theoretical and practical correspondence between POMDPs and the SDL.

On the practical side, in the context of the SDL, the domain of interest can be divided into four parts.

(1) The agent's initial belief-state IB, that is, a specification of the worlds the agent should believe it is in when it becomes active, and probabilities associated with those worlds. For instance,

$$\mathbf{B}f < 0.7 \wedge \mathbf{B}(\neg f \wedge h) = 0.2 \wedge \mathbf{B}(\neg f \wedge \neg h) = 0.1$$

is a specification of a particular initial belief-state.

(2) There will likely be facts about the domain that do not change; fixed laws about and constraints on parts of the environment. For instance, 'a roof is above a floor', 'aeroplanes fly', 'the cat's name is Zoë' and so on. Refer to these *static laws* as SL. They have the form $\phi \Rightarrow \varphi$, where ϕ and φ are propositional sentences, and ϕ is the condition under which φ is always satisfied. Such static laws cannot be explicitly stated in POMDPs; the information must be encoded in each relevant state.

(3) The dynamics of the environment or system must be specified. That is, rules about the effects of actions and about conditions for performing the actions. Refer to these *action rules* as AR. Action rules correspond to R of SDL structures and T of POMDP models. For every action α, action rules typically take the form

$$\begin{aligned}
\phi_1 &\Rightarrow [\alpha]\varphi_{11} = p_{11} \wedge \cdots \wedge [\alpha]\varphi_{1n} = p_{1n} \\
\phi_2 &\Rightarrow [\alpha]\varphi_{21} = p_{21} \wedge \cdots \wedge [\alpha]\varphi_{2n} = p_{2n} \\
&\vdots \\
\phi_j &\Rightarrow [\alpha]\varphi_{j1} = p_{j1} \wedge \cdots \wedge [\alpha]\varphi_{jn} = p_{jn},
\end{aligned}$$

where (i) the sum of transition probabilities $p_{i1}, \ldots, p_{in}$ of any rule i must lie in the range $[0, 1]$, (ii) for every i, for any pair of effects φ_{ik} and $\varphi_{ik'}$, $\varphi_{ik} \wedge \varphi_{ik'} \equiv \bot$ and (iii) for any pair of conditions ϕ_i and $\phi_{i'}$, $\phi_i \wedge \phi_{i'} \equiv \bot$.

(4) The observability of the environment must be specified. That is, rules about which observations are perceivable in which situations (worlds). Refer to these *perception rules* as PR. Perception rules correspond to Q of SDL structures and O of POMDP models. For every action α, perception rules typically take the form

$$\begin{aligned}
\phi_1 &\Rightarrow (\varsigma_{11} \mid \alpha) = p_{11} \wedge \cdots \wedge (\varsigma_{1n} \mid \alpha) = p_{1n} \\
\phi_2 &\Rightarrow (\varsigma_{21} \mid \alpha) = p_{21} \wedge \cdots \wedge (\varsigma_{2n} \mid \alpha) = p_{2n} \\
&\vdots \\
\phi_j &\Rightarrow (\varsigma_{j1} \mid \alpha) = p_{j1} \wedge \cdots \wedge (\varsigma_{jn} \mid \alpha) = p_{jn},
\end{aligned}$$

where (i) the sum of perception probabilities $p_{i1}, \ldots, p_{in}$ of any rule i must lie in the range $[0, 1]$ and (ii) for any pair of conditions ϕ_i and $\phi_{i'}$, $\phi_i \wedge \phi_{i'} \equiv \bot$.

(5) The rewards and costs of actions in different states must be specified. Refer to these *utility rules* as UR. Utility rules correspond to U of SDL structures and R of

POMDP models. Utility rules typically take the form

$$\begin{aligned}\phi_1 &\Rightarrow \mathit{Reward}(r_1)\\ \phi_2 &\Rightarrow \mathit{Reward}(r_2)\\ &\vdots\\ \phi_j &\Rightarrow \mathit{Reward}(r_j),\end{aligned}$$

meaning that in all worlds where ϕ_i is satisfied, the agent gets r_i units of reward. And for every action α,

$$\begin{aligned}\phi_1 &\Rightarrow \mathit{Cost}(\alpha, r_1)\\ \phi_2 &\Rightarrow \mathit{Cost}(\alpha, r_2)\\ &\vdots\\ \phi_j &\Rightarrow \mathit{Cost}(\alpha, r_j),\end{aligned}$$

meaning that the cost for performing α in a world where ϕ_i is satisfied is r_i units. The conditions are disjoint as for action and perception rules.

Special rules may be required for dealing with incomplete specifications. A full discussion of domain specification with the SDL is beyond the scope of this paper.

The union of SL, AR, PR and UR is referred to as an agent's *background knowledge* and is denoted BK.

In practical terms, the question to be answered in the SDL is whether $BK \models IB \rightarrow \Theta^-$ holds, where $BK \subset \mathcal{L}_{SDL}$, IB is as described in point (1) above, and $\Theta^- \in \mathcal{L}_{SDL}^{\not\Rightarrow}$ is some sentence of interest, where $\mathcal{L}_{SDL}^{\not\Rightarrow}$ is the subset of formulae of $\mathcal{L}_{SDL}$ excluding formulae containing subformulae of the form $\varphi \Rightarrow \Phi$.

We now make some preliminary remarks to align SDL structures and POMDP models. In much literature on POMDPs, a model or tuple of components is not defined; the components of a POMDP are simply presented outside of any particular structure (e.g., [Geffner and Bonet, 1998b, Smith and Simmons, 2005]). Kaelbling et al. [1998] define a POMDP model as $\langle S, A, T, R, \Omega, O\rangle$ without the initial belief-state, and Virin et al. [2007] define a POMDP model as $\langle S, A, tr, R, \Omega, O, b^0\rangle$, where tr is used instead of T. Pineau et al. [2003] define a POMDP model as $\langle S, A, O, b^0, T, \Omega, R, \gamma\rangle$, even including the discount factor, but switching the use of O and Ω so that for them, O is the set of observations and Ω is the observation function. We use Z for the name of the observation function, as in the book Probabilistic Robotics [Thrun et al., 2005]. Clearly, the convention for describing a POMDP has not yet been settled.

We have chosen to ignore the discount factor γ in the SDL; equivalently, we consider only POMDP models with $\gamma = 1$. This is done for two reasons, (i) we wish to introduce our logic as simply as possible (without being trivial) and (ii) we expect that the SDL will be used for finite horizon reasoning with short horizons; the effect of the discount factor is relatively small over short sequences.

Let δ^w be a complete propositional theory such that $w \models \delta^w$ and for all other $w' \in C$, $w' \not\models \delta^w$. Fix a set of fluents $\mathcal{F}$ and let C be the conceivable worlds induced from $\mathcal{F}$.

Lemma 3.1 *Let $S = \{w_1, w_2, \ldots, w_n\} \subseteq C$, let $b = \{(w_1, p_1), (w_2, p_2), \ldots, (w_n, p_n)\}$, let IB be $\mathbf{B}\delta^{w_1} = p_1 \wedge \mathbf{B}\delta^{w_2} = p_2 \wedge \cdots \wedge \mathbf{B}\delta^{w_n} = p_n$ and let $\Theta^- \in \mathcal{L}_{SDL}^{\not\Rightarrow}$. Let $\mathcal{D}$ be an* *arbitrary SDL structure and w an arbitrary element of C. Then $\mathcal{D}bw \models \Theta^-$ if and only if for all belief-states $b' \in P$, $\mathcal{D}b'w \models IB \rightarrow \Theta^-$.*

Proof:
Clearly, if $\mathcal{D}bw \models IB$ then for all $b' \in P$, if $b' \neq b$, then $\mathcal{D}b'w \not\models IB$. Therefore, $\mathcal{D}bw \models \Theta^-$
$\iff$ for all $b' \in P$, if $b' \neq b$, then $\mathcal{D}b'w \not\models IB$, else $\mathcal{D}b'w \models \Theta^-$
$\iff$ for all $b' \in P$, $\mathcal{D}b'w \not\models IB$ or $\mathcal{D}b'w \models \Theta^-$
$\iff$ for all $b' \in P$, $\mathcal{D}b'w \models IB \rightarrow \Theta^-$. ■

Lemma 3.2 *Let $\langle S, A, T, R, Z, O, b^0 \rangle$ be any POMDP model such that $S \subseteq C$, $S \neq \emptyset$ and $\gamma = 1$. Let $S^\varphi = \{w \in S \mid w \models \varphi\}$, where $\varphi \in \mathcal{L}_{SDL}$ is a propositional sentence.*
For all $b \in P$ and all $w \in C$:
(i) $\mathcal{D}bw \models \mathbf{U}\Lambda = r$ iff $U(\Lambda, b) = r$,
(ii) $\mathcal{D}bw \models \mathbf{B}\varphi = p$ iff $B(S^\varphi, b) = p$ and
(iii) given sequence $\{a^1+z^1\}\cdots\{a^y+z^y\}$, $SE(a^y, z^y, \cdots\ SE(a^1, z^1, b^0)\ \cdots) = BU(a^y, z^y, \cdots\ BU(a^1, z^1, b^0)\ \cdots)$.

Proof:

- Let $\mathcal{A} = A$ and $\Omega = Z$.
- For every $w^\times \in C$, if $w^\times \notin S$, then let $\delta^{w^\times} \Rightarrow \bot \in BK$.
- For all $w, w' \in S$ and all $a \in \mathcal{A}$, let $\delta^w \Rightarrow ([\alpha]\delta^{w'} = p) \in BK$, where $p = T(w, a, w')$.
- For all $w \in S$, all $z \in \Omega$ and all $a \in \mathcal{A}$, let $\delta^w \Rightarrow ((z \mid \alpha) = p) \in BK$, where $p = O(w, a, z)$.
- For all $w \in S$ and all $a \in \mathcal{A}$, let $\delta^w \Rightarrow (\mathit{Reward}(r) \land \mathit{Cost}(a, 0)) \in BK$, where $r = R(a, w)$.
- Let IB be $\mathbf{B}\delta^{w_1} = p_1 \land \mathbf{B}\delta^{w_2} = p_2 \land \cdots \land \mathbf{B}\delta^{w_n} = p_n$, where $b^0 = \{(w_1, p_1), (w_2, p_2), \ldots, (w_n, p_n)\}$ and $S = \{w_1, w_2, \ldots, w_n\}$.

Let $\mathcal{D} = \langle R, Q, U \rangle$ be an SDL structure such that $\mathcal{D}bw \models \bigwedge_{\beta \in BK} \beta$ for all $b \in P$ and all $w \in C$. Due to the above specifications, it must be that for all $w, w' \in S$, all $z \in \Omega$ and all $a \in \mathcal{A}$, $R_a(w, w') = T(w, a, w')$, $Q_a(w, z) = O(w, a, z)$ and $\mathit{Re}(w) - \mathit{Co}_a(w) = R(a, w)$, where $U = \langle \mathit{Re}, \mathit{Co} \rangle$.

Then, for all $a \in \mathcal{A}$, $z \in \Omega$ and $b \in P$, by Definition 3.6, $P_{NB}(a, z, b) = Pr(z \mid a, b)$, by Definition 3.7, $BU(a, z, b) = SE(a, z, b)$ and by definitions of $RC(\cdot)$ and $\rho(\cdot)$, $RC(a, b) = \rho(a, b)$.

Now, given the equations above, the symmetries in the following definitions should be obvious: (i) $\mathcal{D}bw \models \mathbf{U}\Lambda = r$ and $U(\Lambda, b) = r$, (ii) $\mathcal{D}bw \models \mathbf{B}\varphi = p$ and $B(S^\varphi, b) = p$ and (iii) $SE(\cdot)$ and $BU(\cdot)$.

The lemma follows. ■

Theorem 3.1 *Let $\langle S, A, T, R, Z, O, b^0 \rangle$ be any POMDP model such that $S \subseteq C$, $S \neq \emptyset$ and $\gamma = 1$. Let $S^\varphi = \{w \in S \mid w \models \varphi\}$, where $\varphi \in \mathcal{L}_{SDL}$ is a propositional sentence.*

Given a sequence of actions and observations $\{a^1 + z^1\}\cdots\{a^y + z^y\}$, there exists an SDL specification BK and a sentence IB such that: $U(\Lambda, b^y) \bowtie r$ and $B(S^\varphi, b^y) \bowtie' p$, where $b^y = SE(a^y, z^y, \cdots SE(a^1, z^1, b^0) \cdots)$ if and only if $BK \models IB \rightarrow \{a^1+z^1\}\cdots\{a^y+z^y\}(\mathbf{U}\Lambda \bowtie r \land \mathbf{B}\varphi \bowtie' p)$.

Proof:

- Let $\mathcal{A} = A$ and $\Omega = Z$.
- For every $w^\times \in C$, if $w^\times \notin S$, then let $\delta^{w^\times} \Rightarrow \bot \in BK$.
- For all $w, w' \in S$ and all $a \in \mathcal{A}$, let $\delta^w \Rightarrow ([\alpha]\delta^{w'} = p) \in BK$, where $p = T(w, a, w')$.
- For all $w \in S$, all $z \in \Omega$ and all $a \in \mathcal{A}$, let $\delta^w \Rightarrow ((z \mid \alpha) = p) \in BK$, where $p = O(w, a, z)$.
- For all $w \in S$ and all $a \in \mathcal{A}$, let $\delta^w \Rightarrow (\mathit{Reward}(r) \wedge \mathit{Cost}(a, 0)) \in BK$, where $r = R(a, w)$.
- Let IB be $\mathbf{B}\delta^{w_1} = p_1 \wedge \mathbf{B}\delta^{w_2} = p_2 \wedge \cdots \wedge \mathbf{B}\delta^{w_n} = p_n$, where $b^0 = \{(w_1, p_1), (w_2, p_2), \ldots, (w_n, p_n)\}$ and $S = \{w_1, w_2, \ldots, w_n\}$.

Let $\mathcal{D} = \langle R, Q, U\rangle$ be an SDL structure such that $\mathcal{D}bw \models \bigwedge_{\beta\in BK} \beta$ for all $b \in P$ and all $w \in C$.

Let $b^y = SE(a^y, z^y, \cdots SE(a^1, z^1, b^0)\cdots)$.

($\Rightarrow$) Suppose $U(\Lambda, b^y) \bowtie r$ and $B(S^\varphi, b^y) \bowtie' p$. Then by items (i) and (ii) of Lemma 3.2, for all $w \in C$: $\mathcal{D}b^y w \models \mathbf{U}\Lambda \bowtie r$ and $\mathcal{D}b^y w \models \mathbf{B}\varphi \bowtie' p$. Thus, by item (iii) of Lemma 3.2 and the definition of $\{\varsigma + \alpha\}\Phi$, for all $w \in C$: $\mathcal{D}b^0 w \models \{a^1 + z^1\}\cdots\{a^y + z^y\}\mathbf{U}\Lambda = r$ and $\mathcal{D}b^0 w \models \{a^1 + z^1\}\cdots\{a^y + z^y\}\mathbf{B}\varphi = p$. Hence, by construction of $\mathcal{D}$ for all structures $\mathcal{D}'$ and all $w \in C$: if $\mathcal{D}'b^0 w \models \beta$ for every $\beta \in BK$, then $\mathcal{D}'b^0 w \models \{a^1+z^1\}\cdots\{a^y+z^y\}\mathbf{U}\Lambda \bowtie r$ and $\mathcal{D}'b^0 w \models \{a^1+z^1\}\cdots\{a^y+z^y\}\mathbf{B}\varphi \bowtie' p$. By Lemma 3.1, this implies that for all structures $\mathcal{D}'$, all belief-states b and all $w \in C$: if $\mathcal{D}'bw \models \beta$ for every $\beta \in BK$, then $\mathcal{D}'bw \models IB \to \{a^1+z^1\}\cdots\{a^y+z^y\}\mathbf{U}\Lambda \bowtie r$ and $\mathcal{D}'bw \models IB \to \{a^1+z^1\}\cdots\{a^y + z^y\}\mathbf{B}\varphi \bowtie' p$. Referring to item (iii) of Lemma 3.2 and using the definition of $\wedge$, this implies that for all structures $\mathcal{D}'$, all belief-states b, all $w \in C$: if $\mathcal{D}'bw \models \beta$ for every $\beta \in BK$, then $\mathcal{D}'bw \models IB \to \{a^1 + z^1\}\cdots\{a^y + z^y\}(\mathbf{U}\Lambda \bowtie r \wedge \mathbf{B}\varphi \bowtie' p)$. Therefore, due to the definition of entailment, $BK \models IB \to \{a^1+z^1\}\cdots\{a^y+z^y\}(\mathbf{U}\Lambda \bowtie r \wedge \mathbf{B}\varphi \bowtie' p)$.

($\Leftarrow$) On the other hand, suppose $BK \models IB \to \{a^1+z^1\}\cdots\{a^y+z^y\}(\mathbf{U}\Lambda \bowtie r\wedge\mathbf{B}\varphi \bowtie' p)$. Then, due to the definition of entailment, for all structures $\mathcal{D}'$, all belief-states b and all $w \in C$: if $\mathcal{D}'bw \models \beta$ for every $\beta \in BK$, then $\mathcal{D}'bw \models IB \to \{a^1+z^1\}\cdots\{a^y+z^y\}(\mathbf{U}\Lambda \bowtie r\wedge\mathbf{B}\varphi \bowtie' p)$. Thus, by item (iii) of Lemma 3.2 and the definition of $\wedge$, for all structures $\mathcal{D}'$, all belief-states b, all $w \in C$: if $\mathcal{D}'bw \models \beta$ for every $\beta \in BK$, then $\mathcal{D}'bw \models IB \to \{a^1 + z^1\}\cdots\{a^y + z^y\}\mathbf{U}\Lambda \bowtie r$ and $\mathcal{D}'bw \models IB \to \{a^1 + z^1\}\cdots\{a^y + z^y\}\mathbf{B}\varphi \bowtie' p$. Hence, by Lemma 3.1, for all structures $\mathcal{D}'$ and all $w \in C$: if $\mathcal{D}'b^0 w \models \beta$ for every $\beta \in BK$, then $\mathcal{D}'b^0 w \models \{a^1 + z^1\}\cdots\{a^y + z^y\}\mathbf{U}\Lambda \bowtie r$ and $\mathcal{D}'b^0 w \models \{a^1 + z^1\}\cdots\{a^y + z^y\}\mathbf{B}\varphi \bowtie' p$. Due to the way in which $\mathcal{D}$ is constructed, it follows that for all $w \in C$: $\mathcal{D}b^0 w \models \{a^1 + z^1\}\cdots\{a^y + z^y\}\mathbf{U}\Lambda = r$ and $\mathcal{D}b^0 w \models \{a^1 + z^1\}\cdots\{a^y + z^y\}\mathbf{B}\varphi = p$. Using item (iii) of Lemma 3.2 and the definition of $\{\varsigma + \alpha\}\Phi$, the former statement implies that for all $w \in C$: $\mathcal{D}b^y w \models \mathbf{U}\Lambda \bowtie r$ and $\mathcal{D}b^y w \models \mathbf{B}\varphi \bowtie' p$. Therefore, by items (i) and (ii) of Lemma 3.2, it must be the case that $U(\Lambda, b^y) \bowtie r$ and $B(S^\varphi, b^y) \bowtie' p$. ■

4 The Decision Procedure for SDL Entailment

We provide a decision procedure for checking the validity of sentences in $\mathcal{L}_{SDL} \subset \mathcal{L}$. There are two phases in the decision procedure. The first phase uses a tableau approach to (i) catch 'traditional' contradictions, (ii) separate formulae into literals and (iii) prepare the literals for processing in the second phases. We'll call this the *tableau phase*. The second phase creates systems of inequalities, checking their feasibility. We'll call this the *systems of inequalities* (SI) phase.

4.1 The Tableau Phase

The necessary definitions and terminology are given next.

Definition 4.1 *A labeled formula is a pair* (Σ, Ψ)*, where* $\Psi \in \mathcal{L}_{SDL}$ *is any formula and* Σ *is either 0 or a sequence of the form* $0 \xrightarrow{\alpha_1,\varsigma_1} e_1 \xrightarrow{\alpha_2,\varsigma_2} e_2 \cdots \xrightarrow{\alpha_z,\varsigma_z} e_z$ *called an activity sequence. The* e_i *represent belief-states. If* Σ *is* $0 \xrightarrow{\alpha_1,\varsigma_1} e_1 \cdots \xrightarrow{\alpha_z,\varsigma_z} e_z$*, then the concatenation of* Σ *and* $\xrightarrow{\alpha',\varsigma'} e'$*, denoted as* $\Sigma \xrightarrow{\alpha',\varsigma'} e'$ *is the sequence* $0 \xrightarrow{\alpha_1,\varsigma_1} e_1 \cdots \xrightarrow{\alpha_z,\varsigma_z} e_z \xrightarrow{\alpha',\varsigma'} e'$.

A *node* Γ^j_k with superscript j (the *branch* index) and subscript k (the *node* index), is a set of labeled formulae. The initial node, that is, Γ^0_0, to which the tableau rules must be applied, is called the *trunk*.

Definition 4.2 *A* tree T *is a set of nodes. A tree must include* Γ^0_0 *and only nodes resulting from the application of* tableau rules *to the trunk and subsequent nodes. If one has a tree with trunk* $\Gamma^0_0 = \{(0, \Psi)\}$*, we'll say one has* a tree for Ψ.

When we say '...where x is a fresh integer', we mean that x is the smallest positive integer of the right sort (formula label or branch index) not yet used in the node to which the incumbent tableau rule will be applied.

A tableau rule applied to node Γ^j_k creates one or more new nodes; its child(ren). If it creates one child, then it is identified as Γ^j_{k+1}. If Γ^j_k creates a second child, it is identified as $\Gamma^{j'}_0$, where j' is a fresh integer. That is, for every child created beyond the first, a new branch is started.

A node Γ is a *leaf* node of tree T if no tableau rule has been applied to Γ in T. A *branch* is the set of nodes on a path from the trunk to a leaf node. Note that nodes with different branch indexes may be on the some path.

Definition 4.3 Γ *is* higher *on a branch than* Γ' *if and only if* Γ *is an ancestor of* Γ'.

A node Γ is *closed* if $(\Sigma, \bot) \in \Gamma$ for any Σ. It is *open* if it is not closed. A branch is closed if and only if its leaf node is closed. A tree is closed if all of its branches are closed, else it is open.

A preprocessing step occurs, where all (sub)formulae of the form $(\forall v^\alpha)\Psi$ and $(\forall v^\varsigma)\Psi$ are replaced by, respectively, $(\Psi|^{v^\alpha}_{\alpha_1} \land \ldots \land \Psi|^{v^\alpha}_{\alpha_n})$ and $(\Psi|^{v^\varsigma}_{\varsigma_1} \land \ldots \land \Psi|^{v^\varsigma}_{\varsigma_n})$.

Another preprocessing step occurs, where all (sub)formulae of the form $\varphi \Rightarrow \Phi$ are replaced by $\varphi \Rightarrow \Phi'$, where Φ' is the CNF of Φ.

The tableau rules for SLAOP follow. A rule may only be applied to an open leaf node. A rule may not be applied to a formula if it has been applied to that formula higher in the tree, as in Definition 4.3.

Let Γ^j_k be a leaf node.

- rule $\neg$: If Γ^j_k contains a formula (Σ, Ψ) with a double negation somewhere in it, then create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \Psi')\}$, where Ψ' is Ψ with the double negation removed.
- rule $\wedge$: If Γ^j_k contains $(\Sigma, \Psi \wedge \Psi')$ or $(\Sigma, \neg(\Psi \vee \Psi'))$, then create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \Psi), (\Sigma, \Psi')\}$, respectively, $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \neg\Psi), (\Sigma, \neg\Psi')\}$.
- rule $\vee$: If Γ^j_k contains $(\Sigma, \Psi \vee \Psi'))$ or $(\Sigma, \neg(\Psi \wedge \Psi'))$, then create nodes $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \Psi)\}$ and $\Gamma^{j'}_0 = \Gamma^j_k \cup \{(\Sigma, \Psi')\}$, respectively, nodes $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \neg\Psi)\}$ and $\Gamma^{j'}_0 = \Gamma^j_k \cup \{(\Sigma, \neg\Psi')\}$, where j' is a fresh integer.
- rule $=$: If Γ^j_k contains $(\Sigma, c = c')$ or $(\Sigma, \varphi \Rightarrow c = c')$ where $\varphi \not\equiv \bot$ and c and c' are distinct constants, or if Γ^j_k contains $(\Sigma, \neg(c = c'))$ or $(\Sigma, \varphi \Rightarrow \neg(c = c'))$ where $\varphi \not\equiv \bot$ and c and c' are identical constants, then create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \bot)\}$.
- rule $\Rightarrow \wedge$: If Γ^j_k contains $(\Sigma, \varphi \Rightarrow \Phi \wedge \Phi')$, then create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \varphi \Rightarrow \Phi), (\Sigma, \varphi \Rightarrow \Phi'))\}$.
- rule $\delta \Rightarrow$: If Γ^j_k contains $(\Sigma, \varphi \Rightarrow \Phi)$ where Φ is a disjunction of literals, then create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \delta_1 \Rightarrow \Phi), (\Sigma, \delta_2 \Rightarrow \Phi), \ldots, (\Sigma, \delta_n \Rightarrow \Phi)\}$, where $\delta_i \in \mathit{Def}(\varphi)$.
- rule $\Rightarrow \vee$: If Γ^j_k contains $(\Sigma, \varphi \Rightarrow \Phi \vee \Phi')$ where φ is definitive, then create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, (\varphi \Rightarrow \Phi) \vee (\varphi \Rightarrow \Phi'))\}$.
- rule Ξ: If Γ^j_k contains $(\Sigma, \{\alpha + \varsigma\}\Psi)$ then: if Γ^j_k contains (Σ', Ψ') such that $\Sigma' = \Sigma \xrightarrow{\alpha,\varsigma} e$, then create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma', \Psi)\}$, else create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma \xrightarrow{\alpha,\varsigma} e', \Psi)\}$, where e' is a fresh integer.
- rule $\neg\Xi$: If Γ^j_k contains $(\Sigma, \neg\{\alpha+\varsigma\}\Psi)$, then create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \neg \mathit{Poss}(\alpha, \varsigma) \vee \{\alpha + \varsigma\}\neg\Psi)\}$.
- rule $\neg\mathbf{B}$: If Γ^j_k contains $(\Sigma, \neg\mathbf{B}\varphi \bowtie q)$, then
  - if $\bowtie$ is $<$, create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \mathbf{B}\varphi \geq q)\}$.
  - if $\bowtie$ is $\leq$, create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \mathbf{B}\varphi > q)\}$.
  - if $\bowtie$ is $=$, create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \mathbf{B}\varphi < q \vee \mathbf{B}\varphi > q)\}$.
  - if $\bowtie$ is $\geq$, create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \mathbf{B}\varphi < q)\}$.
  - if $\bowtie$ is $>$, create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \mathbf{B}\varphi \leq q)\}$.
- rule $\neg\mathbf{U}$: If Γ^j_k contains $(\Sigma, \neg\mathbf{U}\Lambda \bowtie q)$, then
  - if $\bowtie$ is $<$, create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \mathbf{U}\Lambda \geq q)\}$.

- if $\bowtie$ is $\leq$, create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \mathbf{U}\Lambda > q)\}$.
- if $\bowtie$ is $=$, create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \mathbf{U}\Lambda < q \vee \mathbf{U}\Lambda > q)\}$.
- if $\bowtie$ is $\geq$, create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \mathbf{U}\Lambda < q)\}$.
- if $\bowtie$ is $>$, create node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(\Sigma, \mathbf{U}\Lambda \leq q)\}$.

Definition 4.4 *A branch is* saturated *if and only if every rule that can be applied to its leaf node has been applied.*

4.2 The SI Phase

Let Γ be a leaf node of an open branch of a saturated tree. $SI(\Gamma)$ is the system of inequalities generated from the formulae in Γ. After the tableau phase is completed, the SI phase begins.

> For each leaf node Γ^j_k of an open branch, do the following. If $SI(\Gamma^j_k)$ is infeasible, then create new leaf node $\Gamma^j_{k+1} = \Gamma^j_k \cup \{(0, \bot)\}$.

Definition 4.5 *A tree is called* finished *after the SI phase is completed.*

Definition 4.6 *If a tree for $\neg\Psi$ is closed, we write $\vdash \Psi$. If there is a finished tree for $\neg\Psi$ with an open branch, we write $\not\vdash \Psi$.*

The generation of $SI(\Gamma)$ from the formulae in Γ is explained in the rest of this section. All variables are assumes implicitly non-negative. Let $C^{\#} = \{w_1, w_2, \ldots, w_n\}$ be an ordering of the worlds in C. We may denote an activity sequence as $\Sigma \xrightarrow{\alpha,\varsigma} e$ to refer to the last action α, observation ς and activity-point e in the sequence, where Σ may be the empty sequence. We may also denote an activity sequence as Σe to refer only to the last activity-point in the sequence; if Σ is the empty sequence, then e is the initial activity-point 0.

In the next four subsections, we deal with (i) law literals involving dynamic and perception literals, (ii) activity sequences, (iii) belief literals and (iv) laws involving reward and cost literals, and utility literals.

4.2.1 Action and Perception Laws

For formulae $(\Sigma, \phi \Rightarrow [\alpha]\varphi \bowtie q) \in \Gamma$ and $(\Sigma, \phi \Rightarrow \neg[\alpha]\varphi \bowtie q) \in \Gamma$, for every j such that $w_j \models \phi$ (where j represents the world in which α is executed),

$$c_1 pr^\alpha_{j,1} + c_2 pr^\alpha_{j,2} + \cdots + c_n pr^\alpha_{j,n} \bowtie q, \quad \text{respectively,} \quad c_1 pr^\alpha_{j,1} + c_2 pr^\alpha_{j,2} + \cdots + c_n pr^\alpha_{j,n} \not\bowtie q$$

is in $SI(\Gamma)$, such that $c_k = 1$ if $w_k \models \varphi$, else $c_k = 0$, and the $pr^\alpha_{j,k}$ are variables. Adding equations

$$pr^\alpha_{j,1} + pr^\alpha_{j,2} + \cdots + pr^\alpha_{j,n} = \lceil pr^\alpha_{j,1} + pr^\alpha_{j,2} + \cdots + pr^\alpha_{j,n} \rceil$$

for every j such that $w_j \models \phi$, will ensure that either $\sum_{w' \in W} R_\alpha(w_j, w') = 1$ or $\sum_{w' \in W} R_\alpha(w_j, w') = 0$, for every $w_j \in C$, as stated in Definition 3.5.

Let $m = |\Omega|$. Let $\Omega^{\#} = (\varsigma_1, \varsigma_2, \ldots, \varsigma_m)$ be an ordering of the observations in Ω. With each observation in $\varsigma \in \Omega^{\#}$, we associate a variable pr_j^{ς}, where j represents the world in which ς is perceived. For formulae $(\Sigma, \phi \Rightarrow (\varsigma|\alpha) \bowtie q) \in \Gamma$ and $(\Sigma, \phi \Rightarrow \neg(\varsigma|\alpha) \bowtie q) \in \Gamma$, for every j such that $w_j \models \phi$,

$$pr_j^{\varsigma|\alpha} \bowtie q, \quad \text{respectively,} \quad pr_j^{\varsigma|\alpha} \not\bowtie q$$

is in $SI(\Gamma)$. Adding equations

$$pr_j^{\varsigma_1|\alpha} + pr_j^{\varsigma_2|\alpha} + \cdots + pr_j^{\varsigma_m|\alpha} = \lceil (pr_{1,j}^{\alpha} + pr_{2,j}^{\alpha} + \cdots + pr_{n,j}^{\alpha})/n \rceil$$

for every j such that $w_j \models \phi$, ensures that for all $w_j \in C$, if there exists a $w_i \in C$ such that $R_\alpha(w_i, w_j) > 0$, then $\sum_{\varsigma \in \Omega} Q_\alpha(w_j, \varsigma) = 1$, else $\sum_{\varsigma \in \Omega} Q_\alpha(w_j, \varsigma) = 0$, as stated in Definition 3.5.

4.2.2 Belief Update

Let $BT(e_h, k, \alpha, \varsigma)$ be the abbreviation for the term

$$\frac{pr_k^{\varsigma|\alpha} \sum_{i=1}^{n} pr_{i,k}^{\alpha} \omega_i^{e_h}}{\sum_{j=1}^{n} pr_j^{\varsigma|\alpha} \sum_{i=1}^{n} pr_{i,j}^{\alpha} \omega_i^{e_h}},$$

which is the probability of being in world w_k after performing belief update $\{\alpha + \varsigma\}$ at activity-point e_h, where $n = |C|$. And let $\Pi(e_h, \alpha, \varsigma)$ be the abbreviation for the term

$$\sum_{j=1}^{n} pr_j^{\varsigma|\alpha} \sum_{i=1}^{n} pr_{i,j}^{\alpha} \omega_i^{e_h},$$

which is the probability of reaching the belief-state after performing belief update $\{\alpha+\varsigma\}$ at activity-point e_h.

Suppose Σ is $0 \xrightarrow{\alpha_0,\varsigma_0} e_1 \xrightarrow{\alpha_1,\varsigma_1} e_2 \cdots \xrightarrow{\alpha_{z-1},\varsigma_{z-1}} e_z$ and $\Sigma \neq 0$. For every $(\Sigma, \Psi) \in \Gamma$, the following equations are in $SI(\Gamma)$.

$$\omega_k^{e_{h+1}} = BT(e_h, k, \alpha_h, \varsigma_h) \text{ (for } k = 1, 2, \ldots, n \text{ and } h = 0, 1, \ldots, z-1),$$

$$\Pi(e_h, \alpha_h, \varsigma_h) \neq 0 \text{ (for } h = 0, 1, \ldots, z-1)$$

and

$$\omega_1^{e_h} + \omega_2^{e_h} + \cdots + \omega_n^{e_h} = 1 \text{ (for } h = 0, 1, \ldots, z),$$

where e_0 is 0. Observe that the e_h are integers and $e_i < e_j$ iff $i < j$.

4.2.3 Executability and Belief Literals

For every $(\Sigma e, Poss(\alpha, \varsigma)) \in \Gamma$ or $(\Sigma e, \neg Poss(\alpha, \varsigma)) \in \Gamma$,

$$\Pi(e, \alpha, \varsigma) \neq 0,$$

respectively,

$$\Pi(e, \alpha, \varsigma) = 0$$

is in $SI(\Gamma)$.

For every $(\Sigma e, \mathbf{B}\varphi \bowtie q) \in \Gamma$,

$$c_1\omega_1^e + c_2\omega_2^e + \cdots + c_n\omega_n^e \bowtie q,$$

is in $SI(\Gamma)$, where $c_k = 1$ if $w_k \models \varphi$, else $c_k = 0$.

4.2.4 Rewards, Costs and Utilities

For formulae $(\Sigma, \phi \Rightarrow \mathit{Reward}(r)) \in \Gamma$ and $(\Sigma, \phi \Rightarrow \neg\mathit{Reward}(r)) \in \Gamma$, for every j such that $w_j \models \phi$,

$$R_j = r, \text{ respectively, } R_j \neq r$$

is in $SI(\Gamma)$.

For formulae $(\Sigma, \phi \Rightarrow \mathit{Cost}(\alpha, r)) \in \Gamma$ and $(\Sigma, \phi \Rightarrow \neg\mathit{Cost}(\alpha, r)) \in \Gamma$, for every j such that $w_j \models \phi$,

$$C_j^\alpha = r, \text{ respectively, } C_j^\alpha \neq r$$

is in $SI(\Gamma)$.

Let $RC(\alpha, e) \overset{def}{=} \omega_1^e(R_1 - C_1^\alpha) + \omega_2^e(R_2 - C_2^\alpha) + \cdots + \omega_n^e(R_n - C_n^\alpha)$. For every $(\Sigma e, \mathbf{U}\{\alpha\} \bowtie q) \in \Gamma$,

$$RC(\alpha, e) \bowtie q$$

is in $SI(\Gamma)$.

To keep track of dependencies between variables in inequalities derived from utility literals of the form $(\Sigma, \mathbf{U}\{\alpha\}\Lambda \bowtie q)$, we define a *utility tree*. A set of utility trees is induced from a set Δ which is defined as follows. For every $(\Sigma e, \mathbf{U}\{\alpha\}\Lambda \bowtie q) \in \Gamma$, let $(e \xrightarrow{\alpha,\varsigma} e^\varsigma, \Lambda) \in \Delta$, for every $\varsigma \in \Omega$, where e^ς is a fresh integer. Then, for every $(\xi, \{\alpha\}\Lambda) \in \Delta$, for every $\varsigma \in \Omega$, if $(\xi', \Psi) \in \Delta$ such that $\xi' = \xi \xrightarrow{\alpha,\varsigma} e^\varsigma$, then $(\xi', \Lambda) \in \Delta$, else $(\xi \xrightarrow{\alpha,\varsigma} e^\varsigma, \Lambda) \in \Delta$, where e^ς is a fresh integer. This finishes the definition of Δ.

Suppose $\Omega = \{\varsigma_1, \varsigma_2\}$ and

$(\Sigma \xrightarrow{\alpha',\varsigma'} 13, \mathbf{U}\{\alpha_5\} = 88)$,

$(\Sigma \xrightarrow{\alpha',\varsigma'} 13, \mathbf{U}\{\alpha_1\}\{\alpha_2\} > 61)$,

$(\Sigma \xrightarrow{\alpha',\varsigma'} 13, \mathbf{U}\{\alpha_1\}\{\alpha_3\}\{\alpha_2\} < 62)$,

$(\Sigma \xrightarrow{\alpha',\varsigma'} 13, \mathbf{U}\{\alpha_1\}\{\alpha_4\} = 63)$,

$(\Sigma \xrightarrow{\alpha',\varsigma'} 23, \mathbf{U}\{\alpha_1\}\{\alpha_2\} \geq 64)$ and

$(\Sigma \xrightarrow{\alpha',\varsigma'} 23, \mathbf{U}\{\alpha_2\}\{\alpha_1\} = 65)$

are in Γ'. Then $(\Sigma \xrightarrow{\alpha',\varsigma'} 13, \mathbf{U}\{\alpha_5\} = 88)$ is not involved in the definition of Δ', nevertheless, $RC(\alpha_5, 13) = 88$ is in $SI(\Gamma')$.

With respect to the other utility literals,

$(13 \xrightarrow{\alpha_1,\varsigma_1} 24, \{\alpha_2\})$, $(13 \xrightarrow{\alpha_1,\varsigma_2} 25, \}\{\alpha_2\})$,

$(13 \xrightarrow{\alpha_1,\varsigma_1} 24, \{\alpha_3\}\{\alpha_2\})$, $(13 \xrightarrow{\alpha_1,\varsigma_2} 25, \{\alpha_3\}\{\alpha_2\})$,

$(13 \xrightarrow{\alpha_1,\varsigma_1} 24, \{\alpha_4\})$, $(13 \xrightarrow{\alpha_1,\varsigma_2} 25, \{\alpha_4\})$,

$(23 \xrightarrow{\alpha_1,\varsigma_1} 26, \{\alpha_2\})$, $(23 \xrightarrow{\alpha_1,\varsigma_2} 27, \{\alpha_2\})$,

$(23 \xrightarrow{\alpha_2,\varsigma_1} 28, \{\alpha_1\})$ and $(23 \xrightarrow{\alpha_2,\varsigma_2} 29, \{\alpha_1\})$

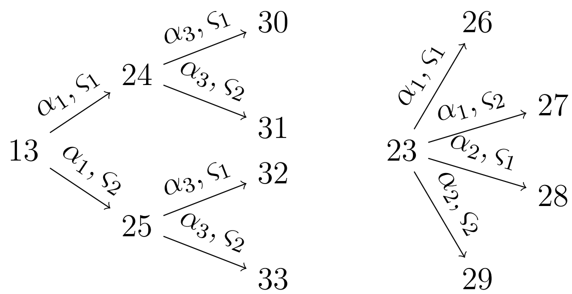

Figure 1: The two utility trees generated from Δ'.

are in Δ'. And due to $(13 \xrightarrow{\alpha_1,\varsigma_1} 24, \{\alpha_3\}\{\alpha_2\}), (13 \xrightarrow{\alpha_1,\varsigma_2} 25, \{\alpha_3\}\{\alpha_2\}) \in \Delta'$, the following are also in Δ'.

$(13 \xrightarrow{\alpha_1,\varsigma_1} 24 \xrightarrow{\alpha_3,\varsigma_1} 30, \{\alpha_2\})$,

$(13 \xrightarrow{\alpha_1,\varsigma_1} 24 \xrightarrow{\alpha_3,\varsigma_2} 31, \{\alpha_2\})$,

$(13 \xrightarrow{\alpha_1,\varsigma_2} 25 \xrightarrow{\alpha_3,\varsigma_1} 32, \{\alpha_2\})$ and

$(13 \xrightarrow{\alpha_1,\varsigma_2} 25 \xrightarrow{\alpha_3,\varsigma_2} 33, \{\alpha_2\})$.

Note how an activity-point is represented by the same integer (for instance, 24) iff it is reached via the same sequence of actions and observations (for instance, $13 \xrightarrow{\alpha_1,\varsigma_1}$).

The set of utility trees is generated from Δ as follows. Δ is partitioned such that $(e \xrightarrow{\alpha,\varsigma} e', \Lambda)$, $(e'' \xrightarrow{\alpha',\varsigma'} e''', \Lambda') \in \Delta$ are in the same partitioning iff $e = e''$. Each partitioning represents a unique utility tree with the first activity-point as the root of the tree. For example, one can generate two utility trees from Δ'; one with root 13 and one with root 23. Each activity sequence of the members of Δ represents a (sub)path starting at the root of its corresponding tree. Figure 1 depicts the two utility trees generated from Δ'. Before considering the general case, we illustrate the method of generating from the utility trees in Figure 1 the required inequalities which must be in $SI(\Gamma)$. Then we show how the required inequalities are generated from utility trees in the general case.

For $(\Sigma \xrightarrow{\alpha',\varsigma'} 13, \mathbf{U}\{\alpha_1\}\{\alpha_2\} > 61) \in \Gamma'$,

$$RC(\alpha_1, 13) + \Pi(13, \alpha_1, \varsigma_1)RC(\alpha_2, 24) + \Pi(13, \alpha_1, \varsigma_2)RC(\alpha_2, 25) > 61.$$

The tree rooted at 13 is used: See that α_1 is executed at activity-point 13, α_2 is executed at activity-point 24 if ς_1 is perceived and α_2 is executed at activity-point 25 if ς_2 is perceived. Moreover, the latter two rewards must be weighted by the probabilities of reaching the respective new belief-states/activity-points.

For $(\Sigma \xrightarrow{\alpha',\varsigma'} 13, \mathbf{U}\{\alpha_1\}\{\alpha_4\} = 63) \in \Gamma'$,

$$RC(\alpha_1, 13) + \Pi(13, \alpha_1, \varsigma_1)RC(\alpha_4, 24) + \Pi(13, \alpha_1, \varsigma_2)RC(\alpha_4, 25) = 63.$$

This time, $\underline{\alpha_4}$ is executed at the activity-points 24 and 25.

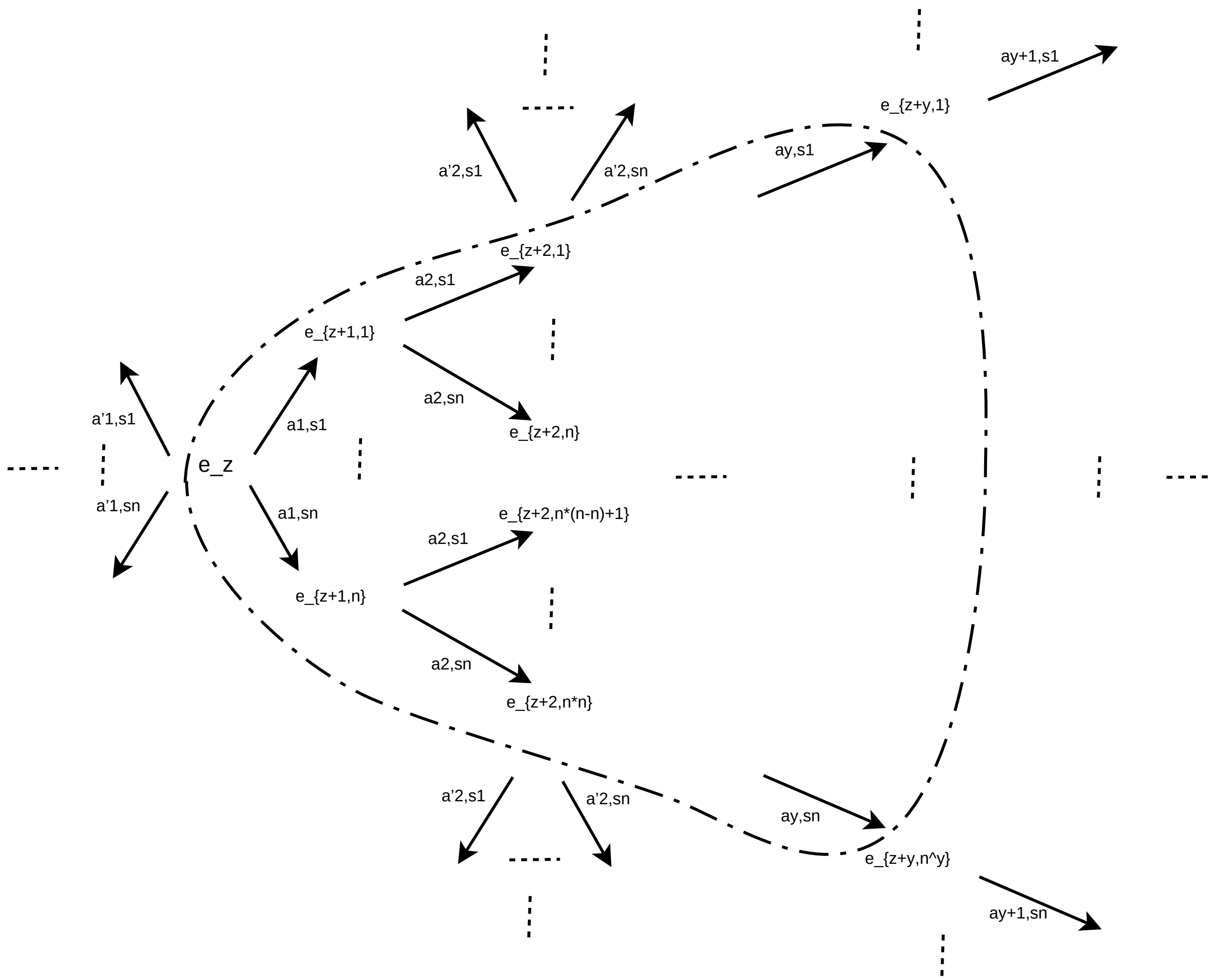

Figure 2: The general form of a utility tree. The enclosed area indicates the subtree corresponding to the general utility literal of (10). The root of the tree, e_z, is situated towards the left of the diagram.

Next, the tree rooted at 23 is used: For $(\Sigma \xrightarrow{\alpha',\varsigma'} 23, \mathbf{U}\{\alpha_1\}\{\alpha_2\} \geq 64) \in \Gamma'$,

$$RC(\alpha_1, 23) + \Pi(23, \alpha_1, \varsigma_1)RC(\alpha_2, 26) + \Pi(23, \alpha_1, \varsigma_2)RC(\alpha_2, 27) \geq 64.$$

For $(\Sigma \xrightarrow{\alpha',\varsigma'} 23, \mathbf{U}\{\alpha_2\}\{\alpha_1\} = 65) \in \Gamma'$,

$$RC(\alpha_2, 23) + \Pi(23, \alpha_2, \varsigma_1)RC(\alpha_1, 28) + \Pi(23, \alpha_1, \varsigma_2)RC(\alpha_1, 29) \geq 64.$$

This time, the second action is not executed at the same activity-points (26 & 27) as in the previous case, because the preceding actions are different in this and the previous case.

For

$$(\Sigma \xrightarrow{\alpha',\varsigma'} 13, \mathbf{U}\{\alpha_1\}\{\alpha_3\}\{\alpha_2\} < 62) \in \Gamma', \tag{9}$$

the inequality is

$$RC(\alpha_1, 13) \begin{array}{l} + \Pi(13, \alpha_1, \varsigma_1)\left(RC(\alpha_3, 24) \begin{array}{ll} + \Pi(24, \alpha_3, \varsigma_1) & RC(\alpha_2, 30) \\ + \Pi(24, \alpha_3, \varsigma_2) & RC(\alpha_2, 31) \end{array} \right) \\ \\ + \Pi(13, \alpha_1, \varsigma_2)\left(RC(\alpha_3, 25) \begin{array}{ll} + \Pi(25, \alpha_3, \varsigma_1) & RC(\alpha_2, 32) \\ + \Pi(25, \alpha_3, \varsigma_2) & RC(\alpha_2, 33) \end{array} \right) \end{array} < 62$$

The size of the tree rooted at 13 is due to (9). Hence, the whole tree is employed to generate the inequality.

In general, a utility tree has the form depicted Figure 2 and a utility literal has the form

$$(\Sigma e_z, \mathbf{U}\{\alpha_1\}\{\alpha_2\}\cdots\{\alpha_y\} \bowtie q). \tag{10}$$

In general, for every utility literal of the form (10) in leaf node Γ, an inequality of the form shown below must be in $SI(\Gamma)$ and can be generated from a utility tree of the form depicted in Figure 2.

$$RC(\alpha_1, e_z) \begin{array}{l} + \Pi(e_z, \alpha_1, \varsigma_1)\left(RC(\alpha_2, e_{z+1,1}) \begin{array}{ll} +\Pi(e_{z+2,1}\alpha_2, \varsigma_1) & \left\{ \cdots \right. \\ +\Pi(e_{z+n,n}, \alpha_2, \varsigma_n) & \left\{ \cdots \right. \end{array} \right. \\ \\ + \Pi(e_z, \alpha_1, \varsigma_n)\left(RC(\alpha_2, e_{z+1,n}) \begin{array}{ll} +\Pi(e_{z+2,n*(n-1)+1}, \alpha_2, \varsigma_1) & \left\{ \cdots \right. \\ +\Pi(e_{z+n,n*n}, \alpha_2, \varsigma_n) & \left\{ \cdots \right. \end{array} \right. \end{array}$$

$$\cdots \begin{array}{l} \left. + \Pi(e_{z+y-2,1}, \alpha_{y-1}, \varsigma_1) RC(\alpha_y, e_{z+y-1,1}) \cdots \right) \Bigg) \\ \\ \left. + \Pi(e_{z+y-2,n^{y-2}}, \alpha_{y-1}, \varsigma_n) RC(\alpha_y, e_{z+y-1,n^{y-1}}) \cdots \right) \Bigg) \end{array} \bowtie q$$

The value to the left of the $\bowtie$ of this general inequality can be written in a compact form as follows. We define the value of a sequence Λ of y actions, starting at activity-point e_z as

$$U(\{\alpha_1\}\{\alpha_2\}\cdots\{\alpha_y\}, e_{z+h-1,x}) \stackrel{def}{=} RC(\{\alpha_1\}, e_{z+h-1,x}) + \sum_{\varsigma_i \in \Omega^\#} \Pi(e_{z+h-1,x}, \alpha_1, \varsigma) U(\{\alpha_2\}\cdots\{\alpha_y\}, e_{z+h,i})$$

$$U(\{\alpha_y\}, e_{z+y-1,x}) \stackrel{def}{=} RC(\alpha_y, e_{z+y-1,x}). \tag{11}$$

The inequality which must be in $SI(\Gamma)$ can thus be written as

$$U(\{\alpha_1\}\{\alpha_2\}\cdots\{\alpha_y\}, e_{z,\_}) \bowtie q,$$

where $e_{z,\_} = e_z$ ($\_$ is a dummy value).

Finally (almost), for every activity-point/node e in every utility tree,

$$\omega_1^e + \cdots + \omega_n^e = 1 \in SI(\Gamma),$$

and for every $e \xrightarrow{\alpha,\varsigma} e'$ in every utility tree,

$$\Pi(e,\alpha,\varsigma) = 0 \mid\mid \Pi(e,\alpha,\varsigma) \neq 0, \omega_1^{e'} = BT(e,1,\alpha,\varsigma), \ldots, \omega_n^{e'} = BT(e,n,\alpha,\varsigma) \in SI(\Gamma) \tag{12}$$

Statement (12) needs explaining:

Until now, whenever it was stated that $\omega_k^{e'} = BT(e,k,\alpha,\varsigma) \in SI(\Gamma)$, the equation matched the occurrence of an update operator $\{\alpha + \varsigma\}$ in a sentence. According to the semantics of $\{\alpha + \varsigma\}$, the new belief-state due to performing α and perceiving ς is reachable; that is, $\Pi(e,\alpha,\varsigma) \neq 0$, where e represents the belief-state before the update. Therefore, matching every $\{\alpha + \varsigma\}$ with $\omega_k^{e'} = BT(e,k,\alpha,\varsigma)$ is appropriate, because the definition of $\omega_k^{e'} = BT(e,k,\alpha,\varsigma)$ implies $\Pi(e,\alpha,\varsigma) \neq 0$. But the occurrence of $e \xrightarrow{\alpha,\varsigma}$ in a utility tree should not necessarily imply the reachability of a belief-state via α, ς from the belief-state represented by e. Nevertheless, if a new belief-state is reachable, then that new belief-state needs to be consistent with the rest of the constraints in the system. Hence, in essence, we are stating that, in the case of utility literals, with respect to the expected utility of the sequence of actions in the literal, IF $\Pi(e,\alpha,\varsigma) \neq 0$ THEN $\omega_k^{e'} = BT(e,k,\alpha,\varsigma) \in SI(\Gamma)$. Unfortunately, one cannot check $\Pi(e,\alpha,\varsigma) \neq 0$ outside the context of the rest of the system of inequalities. So the conditional (12) is made part of the system in the form $\neg A \vee B$, where $||$ represents $\vee$.

Because (12) is not an equation or inequality, it requires a special interpretation to determine the feasibility of the system $SI(\Gamma)$ in which it appears. Let $SI^-(\Gamma)$ be $SI(\Gamma)$ without (12). Quite simply, $SI(\Gamma)$ is feasible if and only if $SI^-(\Gamma) \cup \{\Pi(e,\alpha,\varsigma) = 0\}$ is feasible *or* $SI^-(\Gamma) \cup \{\Pi(e,\alpha,\varsigma) \neq 0, \omega_1^{e'} = BT(e,1,\alpha,\varsigma), \ldots, \omega_n^{e'} = BT(e,n,\alpha,\varsigma)\}$ is feasible.

4.2.5 Decidability of Feasibility of Systems of Inequalities

Lemma 4.1 *Determining whether an SI (as defined in this report) is feasible, is decidable.*

Proof:

Alfred Tarski [Tarski, 1957] defines the first-order logic theory of elementary (real number) algebra as having an infinite number of variables (representing elements of $\mathbb{R}$), algebraic constants 1, 0, -1, two algebraic operation signs + (addition) and $\cdot$ (multiplication), two algebraic relation symbols = (equals) and > (greater than), (logical) sentential connectives $\sim$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), the existential quantifier $\exists$, and a set of axioms defining the theory. "If ξ is any variable, then $(\exists\xi)$ is called a *quantifier expression*.[4] The expression $(\exists\xi)$ is to be read "there exists a ξ such that ."

We show that every equation, disequation and inequality (or *(in)equality* for short) to be included in a system of inequlities as described in this section (the subsections above), can be represented in the language of first-order elementary algebra (FOEA).

[4]He actually uses the symbol E for existential quantification.

First, assuming A and B are in the language of FOEA, note that any (in)equality of the form

$$A < B, A \leq B, A = B, A \geq B \text{ or } A > B$$

is true if and only if, respectively, the FOEA sentence

$$B > A, \sim (A > B), A = B, \sim (B > A) \text{ or } A > B$$

is true. And any (in)equality of the form

$$A \not\bowtie B$$

is true if and only if the FOEA sentence

$$\sim (A \bowtie' B)$$

is true, where $A \bowtie' B$ is the FOEA sentence corresponding to the (in)equality $A \bowtie B$. (In the rest of this section, we denote the FOEA sentence corresponding to the (in)equality $A \bowtie B$ as $A \bowtie' B$.)

Now it is easy to see that (in)equalities of the forms

$$\begin{aligned}
&c_1 pr^{\alpha}_{j,1} + c_2 pr^{\alpha}_{j,2} + \cdots + c_n pr^{\alpha}_{j,n} \bowtie q, \\
&c_1 pr^{\alpha}_{j,1} + c_2 pr^{\alpha}_{j,2} + \cdots + c_n pr^{\alpha}_{j,n} \not\bowtie q, \\
&pr^{\varsigma|\alpha}_{j} \bowtie q, \\
&pr^{\varsigma|\alpha}_{j} \not\bowtie q
\end{aligned}$$

have corresponding FOEA sentence representations

$$\begin{aligned}
&(\exists pr^{\alpha}_{j,1})(\exists pr^{\alpha}_{j,2}) \cdots (\exists pr^{\alpha}_{j,n}) c_1 \cdot pr^{\alpha}_{j,1} + c_2 \cdot pr^{\alpha}_{j,2} + \cdots + c_n \cdot pr^{\alpha}_{j,n} \bowtie' q, \\
&(\exists pr^{\alpha}_{j,1})(\exists pr^{\alpha}_{j,2}) \cdots (\exists pr^{\alpha}_{j,n}) \sim (c_1 \cdot pr^{\alpha}_{j,1} + c_2 \cdot pr^{\alpha}_{j,2} + \cdots + c_n \cdot pr^{\alpha}_{j,n} \bowtie' q), \\
&(\exists pr^{\varsigma|\alpha}_{j}) pr^{\varsigma|\alpha}_{j} \bowtie' q, \\
&(\exists pr^{\varsigma|\alpha}_{j}) \sim (pr^{\varsigma|\alpha}_{j} \bowtie' q),
\end{aligned}$$

where c_k is the constant 1 or 0, and the $pr^{\alpha}_{j,k}$ and $pr^{\varsigma|\alpha}_{j}$ are variables.

Equation

$$pr^{\alpha}_{j,1} + pr^{\alpha}_{j,2} + \cdots + pr^{\alpha}_{j,n} = \lceil pr^{\alpha}_{j,1} + pr^{\alpha}_{j,2} + \cdots + pr^{\alpha}_{j,n} \rceil$$

has the corresponding FOEA sentence representation

$$(\exists pr^{\alpha}_{j,1})(\exists pr^{\alpha}_{j,2}) \cdots (\exists pr^{\alpha}_{j,n})(pr^{\alpha}_{j,1} + pr^{\alpha}_{j,2} + \cdots + pr^{\alpha}_{j,n} = 1 \vee pr^{\alpha}_{j,1} + pr^{\alpha}_{j,2} + \cdots + pr^{\alpha}_{j,n} = 0)$$

and equation

$$pr^{\varsigma_1|\alpha}_{j} + pr^{\varsigma_2|\alpha}_{j} + \cdots + pr^{\varsigma_m|\alpha}_{j} = \lceil (pr^{\alpha}_{1,j} + pr^{\alpha}_{2,j} + \cdots + pr^{\alpha}_{n,j})/n \rceil$$

has the corresponding FOEA sentence representation

$$\begin{aligned}
&(\exists pr^{\alpha}_{1,j})(\exists pr^{\alpha}_{2,j}) \cdots (\exists pr^{\alpha}_{n,j})(\exists pr^{\varsigma_1|\alpha}_{j})(\exists pr^{\varsigma_2|\alpha}_{j}) \cdots (\exists pr^{\varsigma_m|\alpha}_{j}) \\
&(\sim (pr^{\alpha}_{1,j} + pr^{\alpha}_{2,j} + \cdots + pr^{\alpha}_{n,j} > 0) \vee pr^{\varsigma_1|\alpha}_{j} + pr^{\varsigma_2|\alpha}_{j} + \cdots + pr^{\varsigma_m|\alpha}_{j} = 1) \wedge \\
&(\sim (pr^{\alpha}_{1,j} + pr^{\alpha}_{2,j} + \cdots + pr^{\alpha}_{n,j} = 0) \vee pr^{\varsigma_1|\alpha}_{j} + pr^{\varsigma_2|\alpha}_{j} + \cdots + pr^{\varsigma_m|\alpha}_{j} = 0).
\end{aligned}$$

It can also be seen that any equation

$$\sum_{j=1}^{n} pr_j^{\varsigma|\alpha} \sum_{i=1}^{n} pr_{i,j}^{\alpha}\omega_i^{e_h} = 0$$

has the corresponding FOEA sentence representation

$$(\exists pr_1^{\varsigma|\alpha})(\exists pr_2^{\varsigma|\alpha})\cdots(\exists pr_n^{\varsigma|\alpha})(\exists pr_{1,1}^{\alpha})(\exists pr_{1,2}^{\alpha})\cdots(\exists pr_{n,n}^{\alpha})$$
$$\sum_{j=1}^{n} pr_j^{\varsigma|\alpha}\cdot(pr_{1,j}^{\alpha}\cdot\omega_1^{e_h} + pr_{2,j}^{\alpha}\cdot\omega_2^{e_h} + \cdots + pr_{n,j}^{\alpha}\cdot\omega_n^{e_h}) = 0,$$

where the summation symbol $\sum_{j=1}^{n}$ in the FOEA sentence is the obvious abbreviation.

And using the summation symbol to its full, it can also be seen that any equation

$$\omega_k^{e_{h+1}} = \frac{pr_k^{\varsigma|\alpha}\sum_{i=1}^{n} pr_{i,k}^{\alpha}\omega_i^{e_h}}{\sum_{j=1}^{n} pr_j^{\varsigma|\alpha}\sum_{i=1}^{n} pr_{i,j}^{\alpha}\omega_i^{e_h}}$$

has the corresponding FOEA sentence representation

$$(\exists pr_1^{\varsigma|\alpha})(\exists pr_2^{\varsigma|\alpha})\cdots(\exists pr_n^{\varsigma|\alpha})(\exists pr_{1,1}^{\alpha})(\exists pr_{1,2}^{\alpha})\cdots(\exists pr_{n,n}^{\alpha})$$
$$\omega_k^{e_{h+1}}\cdot\sum_{j=1}^{n} pr_j^{\varsigma|\alpha}\cdot\sum_{i=1}^{n} pr_{i,j}^{\alpha}\cdot\omega_i^{e_h} = pr_k^{\varsigma|\alpha}\cdot\sum_{i=1}^{n} pr_{i,k}^{\alpha}\cdot\omega_i^{e_h}$$

where $A\cdot\sum_{j=1}^{n} B_j$ means $A\cdot B_1 + \ldots + A\cdot B_n$.

For any (in)equality not given a FOEA sentence representation above, it should be easy for the reader to derive the FOEA sentence representation.

Tarski provided a finite method which can always decide whether a sentence in the elementary algebra is in the theory [Tarski, 1957]. Hence, feasibility of SIs is decidable. ■

4.3 Examples

To get a better feeling for the expresivity of the SDL and its decision procedure, we work out a few examples. They are all based on the oil-drinking scenario. The set of fluents is $\mathcal{F} = \{\texttt{full}, \texttt{holding}\}$ abbreviated to f and h. Let $w_1 \models f \land h$, $w_2 \models f \land \neg h$, $w_3 \models \neg f \land h$ and $w_4 \models \neg f \land \neg h$. The set of actions is $\mathcal{A} = \{\texttt{grab}, \texttt{drink}, \texttt{weigh}\}$ abbreviated to g, d and w. The set of observations is $\Omega = \{\texttt{Nil}, \texttt{Light}, \texttt{Medium}, \texttt{Heavy}\}$ abbreviated to N, L, M and H.

A full specification of the POMDP model is provided as the basis for the examples in this section. The background knowledge base BK contains the following laws.

Transitions

- $\neg h \Rightarrow [g](f \land h) = 0.8 \land [g](\neg f \land h) = 0.1 \land [g](\neg f \land \neg h) = 0.1$
- $h \Rightarrow [g]\top = 0$
- $h \Rightarrow [d](\neg f \land h) = 0.95 \land [d](\neg f \land \neg h) = 0.05$

- $\neg h \Rightarrow [d]\top = 0$
- $f \wedge h \Rightarrow [w](f \wedge h) = 1$
- $f \wedge \neg h \Rightarrow [w](f \wedge \neg h) = 1$
- $\neg f \wedge h \Rightarrow [w](\neg f \wedge h) = 1$
- $\neg f \wedge \neg h \Rightarrow [w](\neg f \wedge \neg h) = 1$

Perceptions

- $\top \Rightarrow (N \mid g) = 1 \wedge (N \mid d) = 1$
- $f \wedge h \Rightarrow (L \mid w) = 0.1 \wedge (M \mid w) = 0.2 \wedge (H \mid w) = 0.7$
- $\neg f \wedge h \Rightarrow (L \mid w) = 0.5 \wedge (M \mid w) = 0.3 \wedge (H \mid w) = 0.2$
- $\neg h \Rightarrow (\forall v^\varsigma)\neg(v^\varsigma = N) \rightarrow (v^\varsigma \mid w) = \frac{1}{3}$

Utility

- $f \Rightarrow \mathit{Reward}(0)$
- $\neg f \wedge h \Rightarrow \mathit{Reward}(10)$
- $\neg f \wedge \neg h \Rightarrow \mathit{Reward}(-5)$
- $\top \Rightarrow (\forall v^\alpha)(v^\alpha = g \vee v^\alpha = d) \rightarrow \mathit{Cost}(v^\alpha, 1)$
- $f \Rightarrow \mathit{Cost}(w, 2)$
- $\neg f \Rightarrow \mathit{Cost}(w, 0.8)$

4.3.1 First Example

For the first example, we determine whether BK entails

$$\mathbf{B}(f \wedge h) = 0.35 \wedge \mathbf{B}(f \wedge \neg h) = 0.35 \wedge \mathbf{B}(\neg f \wedge h) = 0.2 \wedge \mathbf{B}(\neg f \wedge \neg h) = 0.1 \\ \rightarrow \{g + N\}\{w + M\}\mathbf{B}h > 0.85.$$

Notice that the initial belief-state is fully specified.

For the tableau phase, the trunk is thus $(0, \bigwedge_{\kappa \in BK} \kappa \wedge \mathbf{B}(f \wedge h) = 0.35 \wedge \mathbf{B}(f \wedge \neg h) = 0.35 \wedge \mathbf{B}(\neg f \wedge h) = 0.2 \wedge \mathbf{B}(\neg f \wedge \neg h) = 0.1 \wedge \neg\{g + N\}\{w + M\}\mathbf{B}h > 0.85)$.

Rule $\wedge$ yields

$$(0, \kappa), \ldots, (0, \kappa'), (0, \mathbf{B}(f \wedge h) = 0.35), (0, \mathbf{B}(f \wedge \neg h) = 0.35), \\ (0, \mathbf{B}(\neg f \wedge h) = 0.2), (0, \mathbf{B}(\neg f \wedge \neg h) = 0.1), (0, \neg\{g + N\}\{w + M\}\mathbf{B}h > 0.85) \in \Gamma^0_1,$$

where $\kappa, \ldots, \kappa' \in BK$. Rule $\neg\Xi$ yields

$$(0, \neg \mathit{Poss}(g, N) \vee \{g + N\}\neg\{w + M\}\mathbf{B}h > 0.85) \in \Gamma^0_2.$$

Then rule $\vee$ yields

$$(0, \neg Poss(g, N)) \in \Gamma_0^1 \text{ and } (0, \{g + N\}\neg\{w + M\}\mathbf{B}h > 0.85) \in \Gamma_3^0.$$

We'll deal with the subtree rooted at Γ_3^0 later.

Note that

$$(0, \neg h \Rightarrow [g](f \wedge h) = 0.8 \wedge [g](\neg f \wedge h) = 0.1 \wedge [g](\neg f \wedge \neg h) = 0.1) \in \Gamma_1^0 \subset \Gamma_0^1.$$

Hence, by rule $\Rightarrow \wedge$,

$$(0, \neg h \Rightarrow [g](f \wedge h) = 0.8), (0, \neg h \Rightarrow [g](\neg f \wedge h) = 0.1), (0, \neg h \Rightarrow [g](\neg f \wedge \neg h) = 0.1) \in \Gamma_1^1.$$

Also note that

$$(0, \top \Rightarrow (N \mid g) = 1 \wedge (N \mid d) = 1) \in \Gamma_1^1,$$

and again by rule $\Rightarrow \wedge$,

$$(0, \top \Rightarrow (N \mid g) = 1), (0, \top \Rightarrow (N \mid d) = 1) \in \Gamma_2^1.$$

Assume that the tree saturates and that Γ' is any open leaf node of the (sub)tree rooted at Γ_2^1. Then in the SI phase, due to $(0, \neg h \Rightarrow [g](f \wedge h) = 0.8), (0, \top \Rightarrow (N \mid g) = 1), (0, \neg h \Rightarrow (L \mid w) = \frac{1}{3}), (0, \neg h \Rightarrow (M \mid w) = \frac{1}{3}), (0, \neg h \Rightarrow (H \mid w) = \frac{1}{3}) \in \Gamma'$, the following equations are in $SI(\Gamma')$.

$$\begin{aligned}
pr^g_{2,1} &= 0.8\\
pr^g_{4,1} &= 0.8\\
pr^g_{2,1}+pr^g_{2,2}+pr^g_{2,3}+pr^g_{2,4} &= 1\\
pr^g_{4,1}+pr^g_{4,2}+pr^g_{4,3}+pr^g_{4,4} &= 1\\
pr^{N|g}_1 &= 1\\
pr^{N|g}_2 &= 1\\
pr^{N|g}_3 &= 1\\
pr^{N|g}_4 &= 1\\
pr^{N|g}_1+pr^{L|g}_1+pr^{M|g}_1+pr^{H|g}_1 &= 1\\
pr^{N|g}_2+pr^{L|g}_2+pr^{M|g}_2+pr^{H|g}_2 &= 1\\
pr^{N|g}_3+pr^{L|g}_3+pr^{M|g}_3+pr^{H|g}_3 &= 1\\
pr^{N|g}_4+pr^{L|g}_4+pr^{M|g}_4+pr^{H|g}_4 &= 1\\
pr^{L|w}_1 &= 0.1\\
pr^{L|w}_3 &= 0.5\\
pr^{M|w}_1 &= 0.2\\
pr^{M|w}_3 &= 0.3\\
pr^{H|w}_1 &= 0.7\\
pr^{H|w}_3 &= 0.2\\
pr^{L|w}_2 &= \tfrac{1}{3}\\
pr^{L|w}_4 &= \tfrac{1}{3}\\
pr^{M|w}_2 &= \tfrac{1}{3}\\
pr^{M|w}_4 &= \tfrac{1}{3}\\
pr^{H|w}_2 &= \tfrac{1}{3}\\
pr^{H|w}_4 &= \tfrac{1}{3}\\
pr^{N|w}_1+pr^{L|w}_1+pr^{M|w}_1+pr^{H|w}_1 &= 1\\
pr^{N|w}_2+pr^{L|w}_2+pr^{M|w}_2+pr^{H|w}_2 &= 1\\
pr^{N|w}_3+pr^{L|w}_3+pr^{M|w}_3+pr^{H|w}_3 &= 1\\
pr^{N|w}_4+pr^{L|w}_4+pr^{M|w}_4+pr^{H|w}_4 &= 1
\end{aligned}$$

Also

$$\omega^0_1+\omega^0_2+\omega^0_3+\omega^0_4 = 1 \in SI(\Gamma')$$

and due to $(0, \neg Poss(g, N)), (0, \mathbf{B}(f \wedge h) = 0.35), (0, \mathbf{B}(f \wedge \neg h) = 0.35), (0, \mathbf{B}(\neg f \wedge h) =$

$0.2), (0, \mathbf{B}(\neg f \wedge \neg h) = 0.1) \in \Gamma'$,

$$\sum_{j=1}^{n} pr_j^{N|g} \sum_{i=1}^{n} pr_{i,j}^{g} \omega_i^0 = 0 \tag{13}$$

$$\omega_1^0 = 0.35 \tag{14}$$

$$\omega_2^0 = 0.35 \tag{15}$$

$$\omega_3^0 = 0.2 \tag{16}$$

$$\omega_4^0 = 0.1, \tag{17}$$

respectively, are in $SI(\Gamma')$.

Now $SI(\Gamma')$ is infeasible: No term in (6) may be greater than zero, for example, $pr_1^{N|g} \times pr_{2,1}^{g} \times \omega_2^0$ must equal zero, but $pr_1^{N|g} = 1$ and $pr_{2,1}^{g} = 0.8$ and $\omega_2^0 = 0.35$. Therefore, the subtree rooted at Γ_0^1 is closed.

Coming back to Γ_3^0, due to rule Ξ,

$$(0 \xrightarrow{g,N} 1, \neg\{w + M\}\mathbf{B}h > 0.85) \in \Gamma_4^0.$$

Then by the application of rule $\neg\Xi$ and then rule $\vee$,

$$(0 \xrightarrow{g,N} 1, \neg\{w + M\}) \in \Gamma_0^2 \text{ and } (0 \xrightarrow{g,N} 1, \{w + M\}\neg\mathbf{B}h > 0.85) \in \Gamma_6^0.$$

The case with the subtree rooted at Γ_0^2 is similar to the case above where the subtree is rooted at Γ_0^1. The subtree rooted at Γ_0^2 also closes. Rule Ξ applied to $(0 \xrightarrow{g,N} 1, \{w + M\}\neg\mathbf{B}h > 0.85) \in \Gamma_6^0$ yields

$$(0 \xrightarrow{g,N} 1 \xrightarrow{w,M} 2, \neg\mathbf{B}h > 0.85) \in \Gamma_7^0$$

and then rule $\neg\mathbf{B}$ yields

$$(0 \xrightarrow{g,N} 1 \xrightarrow{w,M} 2, \mathbf{B}h \leq 0.85) \in \Gamma_8^0.$$

Still assuming that the tree saturates, and assume that Γ'' is any open leaf node of the (sub)tree rooted at Γ_8^0. Then in the SI phase, due to $(0, \mathbf{B}(f \wedge h) = 0.35), (0, \mathbf{B}(f \wedge \neg h) = 0.35), (0, \mathbf{B}(\neg f \wedge h) = 0.2), (0, \mathbf{B}(\neg f \wedge \neg h) = 0.1), (0 \xrightarrow{g,N} 1 \xrightarrow{w,M} 2, \mathbf{B}h \leq 0.85) \in \Gamma''$, the following equations are in $SI(\Gamma'')$.

$$\begin{aligned}
\omega_1^0 &= 0.35\\
\omega_2^0 &= 0.35\\
\omega_3^0 &= 0.2\\
\omega_4^0 &= 0.1\\
\omega_1^0 + \omega_2^0 + \omega_3^0 + \omega_4^0 &= 1\\
\omega_1^2 + \omega_3^2 &\leq 0.85\\
\omega_1^1 + \omega_2^1 + \omega_3^1 + \omega_4^1 &= 1\\
\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 &= 1\\
\omega_1^1 &= BT(0, 1, g, N)\\
\omega_2^1 &= BT(0, 2, g, N)\\
\omega_3^1 &= BT(0, 3, g, N)\\
\omega_4^1 &= BT(0, 4, g, N)\\
\omega_1^2 &= BT(1, 1, w, M)\\
\omega_2^2 &= BT(1, 2, w, M)\\
\omega_3^2 &= BT(1, 3, w, M)\\
\omega_4^2 &= BT(1, 4, w, M)
\end{aligned}$$

It turns out that ω_1^2 is constrained to equal 0.72973 and ω_3^2 is constrained to equal 0.12162, and $0.72973 + 0.12162 = 0.85135 > 0.85$. Therefore, the subtree rooted at Γ_0^1 is closed.

Therefore, the whole tree is closed and the initial entailment query holds.

4.3.2 Second Example

For this example, the same background knowledge base is used, but the initial belief-state is not fully specified. We shall determine whether BK entails

$$\begin{aligned}
&\mathbf{B}f = 0.7 \wedge \mathbf{B}(\neg f \wedge h) = 0.2 \wedge \mathbf{B}(\neg f \wedge \neg h) = 0.1\\
&\rightarrow \{g + N\}\{w + M\}\mathbf{B}h > 0.85.
\end{aligned}$$

The tree for $(0, \bigwedge_{\kappa \in BK} \kappa \wedge \mathbf{B}f = 0.7 \wedge \mathbf{B}(\neg f \wedge h) = 0.2 \wedge \mathbf{B}(\neg f \wedge \neg h) = 0.1 \wedge \neg\{g + N\}\{w + M\}\mathbf{B}h > 0.85)$ is very similar to the one generated in the first example. The difference is in the subtree dealing with the belief literals (the subtree rooted at Γ_3^0 in the example above). Assume that the tree saturates and assume that Γ'' is any open leaf node of the subtree under consideration. Now, instead of $\omega_1^0 = 0.35$ and $\omega_2^0 = 0.35$ in $SI(\Gamma'')$, we have $\omega_1^0 + \omega_2^0 = 0.7 \in SI(\Gamma'')$. Observe that this is not a full specification of the belief-state, because ω_1^0 and ω_2^0 may take any values in $\mathbb{R}_{[0,1]}$ as long as $\omega_1^0 + \omega_2^0 = 0.7$.

Let the initial belief-state $b^0 = \{(w_1, x), (w_2, 0.7 - x), (w_3, 0.2), (w_4, 0.1)\}$, where $x \in \mathbb{R}_{[0,0.7]}$. According to our calculations, the belief-state after the updates $\{g + N\}$ and $\{w + M\}$ is $b^2 = \{(w_1, \frac{0.128-0.16x}{0.179-0.223x}), (w_2, 0), (w_3, \frac{0.024-0.03x}{0.179-0.223x}), (w_4, \frac{0.0267-0.0333}{0.179-0.223x})\}$. The system of inequalities will enforce b^2, that is,

$$\omega_1^2 = \frac{0.128 - 0.16x}{0.179 - 0.223x}$$
$$\omega_2^2 = 0$$
$$\omega_3^2 = \frac{0.024 - 0.03x}{0.179 - 0.223x}$$
$$\omega_4^2 = \frac{0.0267 - 0.0333}{0.179 - 0.223x},$$

where $x \in \mathbb{R}_{[0,0.7]}$ will be enforced by the system. Furthermore, $(0 \xrightarrow{g,N} 1 \xrightarrow{w,M} 2, \mathbf{B}h \leq 0.85) \in \Gamma''$ causes $\omega_1^2 + \omega_3^2 \leq 0.85$ to be in $SI(\Gamma'')$. Hence, $\frac{0.128-0.16x}{0.179-0.223x} + \frac{0.024-0.03x}{0.179-0.223x}$ must be less than or equal to 0.85. In other words, it is required that

$$\frac{0.152 - 0.19x}{0.179 - 0.223x} \leq 0.85. \tag{18}$$

But one can determine that there is no value for $x \in \mathbb{R}_{[0,0.7]}$ which will make (18) true. The system $SI(\Gamma'')$ is thus infeasible, the tree closes and the initial entailment query holds.

We draw the reader's attention to the fact that sensible entailments can be queried, even with a partially specified initial belief-state.

4.3.3 Third Example

This time we provide a complete specification of the initial belief-state again, as in the first example, but we under-specify the perception probabilities. Suppose that instead of the law $f \wedge h \Rightarrow (L \mid w) = 0.1 \wedge (M \mid w) = 0.2 \wedge (H \mid w) = 0.7 \in BK$, we have only $f \wedge h \Rightarrow (H \mid w) = 0.7 \in BK'$. (That is, we modify BK to become BK'.)

From a system of inequalities generated from the entailment query and modified background knowledge, one can determine/calculate that the degree of belief in h (`holding`) after the updates $\{g + N\}$ and $\{w + M\}$ is $\frac{0.\overline{81}x+0.0\overline{27}}{0.\overline{81}x+0.\overline{06}}$, where $x \in \mathbb{R}_{[0,0.3]}$ is the probability of perceiving the the oil-can has medium weight at the world where the can is full and the robot is holding it. One can determine that, $\frac{0.\overline{81}x+0.0\overline{27}}{0.\overline{81}x+0.\overline{06}} \leq 0.85$ when $x \leq 0.1976$. The system $SI(\Gamma'')$ is thus feasible, the tree is open and BK' does not entail

$$\mathbf{B}(f \wedge h) = 0.35 \wedge \mathbf{B}(f \wedge \neg h) = 0.35 \wedge \mathbf{B}(\neg f \wedge h) = 0.2 \wedge \mathbf{B}(\neg f \wedge \neg h) = 0.1$$
$$\rightarrow \{g + N\}\{w + M\}\mathbf{B}h > 0.85. \tag{19}$$

Now, because $x \leq 0.1976$, the system

$$\frac{0.\overline{81}x + 0.0\overline{27}}{0.\overline{81}x + 0.\overline{06}} \leq 0.85$$
$$x \geq 0.2$$
$$x \leq 0.3$$

is infeasible. So if BK'' is BK' with the law $f \wedge h \Rightarrow (M \mid w) \geq 0.2$ added to it, the tree closes and BK'' entails (19). Observe that (19) goes from not entailed to entailed by adding a little more information; while the POMDP model remains not completely specified.

4.3.4 Fourth Example

Here we shall determine whether BK entails

$$\mathbf{B}(f \wedge h) = 0.35 \wedge \mathbf{B}(f \wedge \neg h) = 0.35 \wedge \mathbf{B}(\neg f \wedge h) = 0.2 \wedge \mathbf{B}(\neg f \wedge \neg h) = 0.1 \\ \rightarrow \{g + N\}\mathbf{U}\{d\}\{d\} \leq 7.$$

For the tableau phase, the trunk is thus $(0, \bigwedge_{\kappa \in BK} \kappa \wedge \mathbf{B}(f \wedge h) = 0.35 \wedge \mathbf{B}(f \wedge \neg h) = 0.35 \wedge \mathbf{B}(\neg f \wedge h) = 0.2 \wedge \mathbf{B}(\neg f \wedge \neg h) = 0.1 \wedge \neg\{g + N\}\mathbf{U}\{d\}\{d\} \leq 7)$.

Rule $\wedge$ yields

$$(0, \kappa), \ldots, (0, \kappa'), (0, \mathbf{B}(f \wedge h) = 0.35), (0, \mathbf{B}(f \wedge \neg h) = 0.35), \\ (0, \mathbf{B}(\neg f \wedge h) = 0.2), (0, \mathbf{B}(\neg f \wedge \neg h) = 0.1), (0, \neg\{g + N\}\mathbf{U}\{d\}\{d\} \leq 7) \in \Gamma_1^0,$$

where $\kappa, \ldots, \kappa' \in BK$. Rule $\neg\Xi$ yields

$$(0, \neg\{g + N\} \vee \{g + N\}\neg\mathbf{U}\{d\}\{d\} \leq 7) \in \Gamma_2^0.$$

Then rule $\vee$ yields

$$(0, \neg\{g + N\}) \in \Gamma_0^1) \text{ and } (0, \{g + N\}\neg\mathbf{U}\{d\}\{d\} \leq 7) \in \Gamma_3^0.$$

The subtree rooted at Γ_0^1 closes, just like in the first example.

Applying rule Ξ to $(0, \{g + N\}\neg\mathbf{U}\{d\}\{d\} \leq 7)$ yields

$$(0 \xrightarrow{g,N} 1, \neg\mathbf{U}\{d\}\{d\} \leq 7) \in \Gamma_3^0$$

and rule $\neg\mathbf{U}$ yields

$$(0 \xrightarrow{g,N} 1, \mathbf{U}\{d\}\{d\} > 7) \in \Gamma_4^0.$$

The following equations (amongst others) are in $SI(\Gamma'')$.

$$
\begin{aligned}
pr^d_{1,3} &= 0.95 \\
pr^d_{3,3} &= 0.95 \\
pr^d_{1,4} &= 0.05 \\
pr^d_{3,4} &= 0.05 \\
pr^d_{2,1} &= 0 \\
pr^d_{2,2} &= 0 \\
pr^d_{2,3} &= 0 \\
pr^d_{2,4} &= 0 \\
pr^d_{4,1} &= 0 \\
pr^d_{4,2} &= 0 \\
pr^d_{4,3} &= 0 \\
pr^d_{4,4} &= 0 \\
pr^{N|d}_1 &= 1 \\
pr^{N|d}_2 &= 1 \\
pr^{N|d}_3 &= 1 \\
pr^{N|d}_4 &= 1 \\
\omega^0_1 &= 0.35 \\
\omega^0_2 &= 0.35 \\
\omega^0_3 &= 0.2 \\
\omega^0_4 &= 0.1 \\
\omega^0_1 + \omega^0_2 + \omega^0_3 + \omega^0_4 &= 1 \\
\omega^1_1 + \omega^1_2 + \omega^1_3 + \omega^1_4 &= 1 \\
R_1 &= 0 \\
R_2 &= 0 \\
R_3 &= 10 \\
R_4 &= -5 \\
C^d_1 &= 1 \\
C^d_2 &= 1 \\
C^d_3 &= 1 \\
C^d_4 &= 1 \\
\omega^1_1 &= BT(0,1,g,N) \\
\omega^1_2 &= BT(0,2,g,N) \\
\omega^1_3 &= BT(0,3,g,N) \\
\omega^1_4 &= BT(0,4,g,N)
\end{aligned}
$$

According to § 4.2.4, $\Delta = \{(1 \xrightarrow{d,N} 2, \{d\}), (1 \xrightarrow{d,L} 3, \{d\}), (1 \xrightarrow{d,M} 4, \{d\}), (1 \xrightarrow{d,H}$

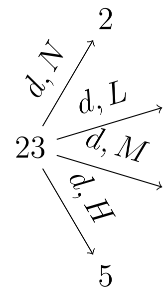

Figure 3: The utility tree generated from $\{(1 \xrightarrow{d,N} 2, \{d\}), (1 \xrightarrow{d,L} 3, \{d\}), (1 \xrightarrow{d,M} 4, \{d\}), (1 \xrightarrow{d,H} 5, \{d\})\}$.

$5, \{d\})\}$. The only utility tree generated from Δ is shown in Figure 3. The following inequality generated for $(0 \xrightarrow{g,N} 1, \mathbf{U}\{d\}\{d\} > 7) \in \Gamma''$ using the utility tree is in $SI(\Gamma'')$.

$$RC(d,1) \begin{array}{c} +\Pi(1,d,N)RC(d,2) \\ +\Pi(1,d,L)RC(d,3) \\ +\Pi(1,d,M)RC(d,4) \\ +\Pi(1,d,H)RC(d,5) \end{array} > 7. \tag{20}$$

Also in $SI(\Gamma'')$ are

$\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 = 1$

$\omega_1^3 + \omega_2^3 + \omega_3^3 + \omega_4^3 = 1$

$\omega_1^4 + \omega_2^4 + \omega_3^4 + \omega_4^4 = 1$

$\omega_1^5 + \omega_2^5 + \omega_3^5 + \omega_4^5 = 1$

$\Pi(1,d,N) = 0 \mid\mid \Pi(1,d,N) \neq 0, \omega_1^2 = BT(1,1,d,N)$

$\Pi(1,d,N) = 0 \mid\mid \Pi(1,d,N) \neq 0, \omega_2^2 = BT(1,2,d,N)$

$\Pi(1,d,N) = 0 \mid\mid \Pi(1,d,N) \neq 0, \omega_3^2 = BT(1,3,d,N)$

$\Pi(1,d,N) = 0 \mid\mid \Pi(1,d,N) \neq 0, \omega_4^2 = BT(1,4,d,N)$

$\Pi(1,d,L) = 0 \mid\mid \Pi(1,d,L) \neq 0, \omega_1^3 = BT(1,1,d,L)$

$\Pi(1,d,L) = 0 \mid\mid \Pi(1,d,L) \neq 0, \omega_2^3 = BT(1,2,d,L)$

$\Pi(1,d,L) = 0 \mid\mid \Pi(1,d,L) \neq 0, \omega_3^3 = BT(1,3,d,L)$

$\Pi(1,d,L) = 0 \mid\mid \Pi(1,d,L) \neq 0, \omega_4^3 = BT(1,4,d,L)$

$\Pi(1,d,M) = 0 \mid\mid \Pi(1,d,M) \neq 0, \omega_1^4 = BT(1,1,d,M)$

$\Pi(1,d,M) = 0 \mid\mid \Pi(1,d,M) \neq 0, \omega_2^4 = BT(1,2,d,M)$

$\Pi(1,d,M) = 0 \mid\mid \Pi(1,d,M) \neq 0, \omega_3^4 = BT(1,3,d,M)$

$\Pi(1,d,M) = 0 \mid\mid \Pi(1,d,M) \neq 0, \omega_4^4 = BT(1,4,d,M)$

$\Pi(1,d,H) = 0 \mid\mid \Pi(1,d,H) \neq 0, \omega_1^5 = BT(1,1,d,H)$

$\Pi(1,d,H) = 0 \mid\mid \Pi(1,d,H) \neq 0, \omega_2^5 = BT(1,2,d,H)$

$\Pi(1,d,H) = 0 \mid\mid \Pi(1,d,H) \neq 0, \omega_3^5 = BT(1,3,d,H)$

$\Pi(1,d,H) = 0 \mid\mid \Pi(1,d,H) \neq 0, \omega_4^5 = BT(1,4,d,H)$

Due to the law $\top \Rightarrow (N \mid g) = 1 \wedge (N \mid d) = 1 \in BK$, the law literal $\top \Rightarrow (N \mid d) = 1$ is in Γ''. Recall that

$$pr_j^{N|d} + pr_j^{L|d} + pr_j^{M|d} + pr_j^{H|d} = \lceil pr_j^{N|d} + pr_j^{L|d} + pr_j^{M|d} + pr_j^{H|d} \rceil$$

is in $SI(\Gamma)$ for each j such that $w_j \models \top$ (in this case, $j = 1,2,3,4$). And because $pr_j^{N|d} = 1 \in SI(\Gamma)$, it follows that $pr_j^{N|d} + pr_j^{L|d} + pr_j^{M|d} + pr_j^{H|d} = 1$. One can thus deduce that

$$pr_j^{L|d} = pr_j^{M|d} = pr_j^{H|d} = 0. \tag{21}$$

Recall that $\Pi(e,\alpha,\varsigma,n) \stackrel{def}{=} \sum_{j=1}^{n} pr_j^{\varsigma|\alpha} \sum_{i=1}^{n} pr_{i,j}^{\alpha} \omega_i^e$. Hence, by Equation (21), for $SI(\Gamma)$ to be feasible, each of $\Pi(1,d,L,4)RC(d,3)$, $\Pi(1,d,M,4)RC(d,4)$ and $\Pi(1,d,H,4)RC(d,5)$ must be equal to zero. Therefore, by Inequality (20), $RC(d,1) + \Pi(1,d,N,4)RC(d,2)$ must be more than 7. That is,

$$\begin{aligned}&\omega_1^1(R_1 - C_1^d) + \omega_2^1(R_2 - C_2^d) + \omega_3^1(R_3 - C_3^d) + \omega_4^1(R_4 - C_4^d) + \\ &\sum_{j=1}^{n} pr_j^{N|d} \sum_{i=1}^{n} pr_{i,j}^{d} \omega_i^1 \big(\omega_1^2(R_1 - C_1^d) + \omega_2^2(R_2 - C_2^d) + \omega_3^2(R_3 - C_3^d) + \omega_4^2(R_4 - C_4^d)\big) > 7.\end{aligned} \tag{22}$$

We have calculated that for $SI(\Gamma'')$ to be feasible, it is required that $\omega_1^1 = 0.\overline{81}, \omega_2^1 = 0, \omega_3^1 = 0.\overline{09}, \omega_4^1 = 0.\overline{09}, \omega_1^2 = 0, \omega_2^2 = 0, \omega_3^2 = 0.95, \omega_4^2 = 0.05$ and $\sum_{j=1}^{n} pr_j^{N|d} \sum_{i=1}^{n} pr_{i,j}^{d} \omega_i^1 = 0.\overline{90}$. Therefore, (16) becomes $6.95455 > 7.$, which is false. So $SI(\Gamma'')$ is infeasible, the tree closes and the entailment query holds.

4.3.5 Fifth Example

We query whether BK entails

$$\begin{aligned}&\mathbf{B}f = 0.7 \wedge \mathbf{B}(\neg f \wedge h) = 0.2 \wedge \mathbf{B}(\neg f \wedge \neg h) = 0.1 \\ &\to \{g + N\}\mathbf{U}\{d\}\{d\} \leq 7.\end{aligned}$$

As in the second example, the initial belief-state is under-specified. The development of the decision proceeds almost exactly as in the previous example. The only difference is that where $\omega_1^0 = 0.35, \omega_2^0 = 0.35 \in SI(\Gamma'')$, now $\omega_1^0 + \omega_2^0 = 0.7 \in SI(\Gamma'')$. As in the previous example, $SI(\Gamma'')$ is feasible if and only if (16) is true. To evaluate (16), one needs the values of $\omega_1^1, \omega_2^1, \omega_3^1, \omega_4^1, \omega_1^2, \omega_2^2, \omega_3^2, \omega_4^2$ and $\sum_{j=1}^{n} pr_j^{N|d} \sum_{i=1}^{n} pr_{i,j}^{d} \omega_i^1$.

Setting the value of ω_1^0 to x and ω_2^0 to $0.7 - x$ as before, we have calculated that these values are constrained to be $\omega_1^1 = \frac{0.64-0.8x}{0.8-x}, \omega_2^1 = 0, \omega_3^1 = \frac{0.08-0.1x}{0.8-x}, \omega_4^1 = \frac{0.08-0.1x}{0.8-x}, \omega_1^2 =$

$0, \omega_2^2 = 0, \omega_3^2 = \frac{0.684-0.855x}{0.72-0.9x}, \omega_4^2 = \frac{0.036-0.045x}{0.72-0.9x}$, and $\sum_{j=1}^{n} pr_j^{N|d} \sum_{i=1}^{n} pr_{i,j}^{d}\omega_i^1 = \frac{0.72-0.9x}{0.8-x}$. So (16) becomes

$$\frac{0.64-0.8x}{0.8-x}(R_1 - C_1^d) + \frac{0.08-0.1x}{0.8-x}(R_3 - C_3^d) + \frac{0.08-0.1x}{0.8-x}(R_4 - C_4^d) \\ + \frac{0.72-0.9x}{0.8-x}\Big(\frac{0.684-0.855x}{0.72-0.9x}(R_3 - C_3^d) + \frac{0.036-0.045x}{0.72-0.9x}(R_4 - C_4^d)\Big) > 7,$$

which is equivalent to

$$\frac{0.64-0.8x}{0.8-x}(R_1 - C_1^d) + \frac{0.08-0.1x}{0.8-x}(R_3 - C_3^d) + \frac{0.08-0.1x}{0.8-x}(R_4 - C_4^d) \\ + \frac{0.684-0.855x}{0.8-x}(R_3 - C_3^d) + \frac{0.036-0.045x}{0.8-x}(R_4 - C_4^d) > 7,$$

which is equivalent to

$$\frac{0.64-0.8x}{0.8-x}(-1) + \frac{0.08-0.1x}{0.8-x}(9) + \frac{0.08-0.1x}{0.8-x}(-6) \\ + \frac{0.684-0.855x}{0.8-x}(9) + \frac{0.036-0.045x}{0.8-x}(-6) > 7,$$

which is equivalent to $x > 0.8$. But we know that $x \leq 0.7$.

So $SI(\Gamma'')$ is infeasible, the tree closes and the entailment query holds. This example shows that non-trivial entailments about the utility of sequences of actions can be confirmed, even without full knowledge about the initial belief-state.

5 Proof Theory

We present the proofs for soundness, completeness and termination of the decision procedure.

5.1 Soundness of the Decision Procedure

Theorem 5.1 (Soundness) *If $\vdash \Psi$ then $\models \Psi$. (Contrapositively, if $\not\models \Psi$ then $\not\vdash \Psi$.)*

Let $\psi = \neg\Psi$. Then $\not\vdash \Psi$ if and only if the tree for ψ is open. And

$$\begin{aligned} \not\models \Psi &\iff \text{not } (\forall \mathcal{D})\ \mathcal{D} \models \Psi \\ &\iff \text{not } (\forall \mathcal{D}, b)\ \mathcal{D}b \models \Psi \\ &\iff \text{not } (\forall \mathcal{D}, b, w)\ \mathcal{D}bw \models \Psi \\ &\iff (\exists \mathcal{D}, b, w)\ \mathcal{D}bw \models \psi. \end{aligned}$$

For the soundness proof, it thus suffices to show that if there exists a structure $\mathcal{D}$, a belief-state b and a worlds w such that $\mathcal{D}bw \models \psi$, then the tree rooted at $\Gamma_0^0 = \{(0, \psi)\}$ is open.

Lemma 5.1 *Let T be a finished tree. For every node Γ in T: If there exists a structure $\mathcal{D}$ such that for all $(\Sigma, \Phi) \in \Gamma$ there exists a belief-state $b \in P$ and a world $w \in C$ such that $\mathcal{D}bw \models \Phi$, then the (sub)tree rooted at Γ is open.*

Proof:
(by induction on the height of the node Γ_k)

Base case:

Height $h = 0; \Gamma_k$ is a leaf. If there exists a structure $\mathcal{D}$ such that for all $(\Sigma, \Phi) \in \Gamma_k$ there exists a b an a w such that $\mathcal{D}bw \models \Phi$, then $(\Sigma', \bot) \notin \Gamma_k$ for all x'. Hence, the sub-tree consisting of Γ_k is open.

Induction step:

If $h > 0$, then some rule was applied to create the child(ren) $\Gamma_{k'}$ of Γ_k. We abbreviate "there exists a structure $\mathcal{D}^j = \langle R^j, Q^j, U^j \rangle$ such that for all $(\Sigma^j, \Phi^j) \in \Gamma_j$ there exists a b^j and a w^j such that $\mathcal{D}^j b^j w^j \models \Phi^j$" as $A(j)$ and we abbreviate "the (sub)tree rooted at Γ_j is open" as $B(j)$.

We must show the following for every rule/phase. IF: If $A(k')$, then $B(k')$, THEN: If $A(k)$, then $B(k)$. We assume the antecedent (induction hypothesis): If $A(k')$, then $B(k')$. To show the consequent, we must assume $A(k)$ and show that $B(k)$ follows.

Note that if the (sub)tree rooted at $\Gamma_{k'}$ is open, then the (sub)tree rooted at Γ_k is open. That is, if $B(k')$ then $B(k)$. So we want to show $B(k')$. But, by the induction hypothesis, $B(k')$ follows from $A(k')$. Therefore, it will suffice, in each case below, to assume $A(k)$, and prove $A(k')$.

- rule $\neg$:
  For the rule to have been applied, $\{(\Sigma, \Psi)\} \subseteq \Gamma_k$ such that Ψ has a double negation somewhere in it, and after its application, $\Gamma_{k'} = \Gamma_k \cup \{(\Sigma, \Psi')\}$, where Ψ' is Ψ with the double negation removed. By assumption, $\mathcal{D}^k b^k w^k \models \Psi$. Hence, $\mathcal{D}^k b^k w^k \models \Psi'$. Thus, $A(k')$.

- rule $\wedge$:
  For the rule to have been applied, $\{(\Sigma, \Psi \wedge \Psi')\} \subseteq \Gamma_k$, and after its application, $\Gamma_{k'} = \Gamma_k \cup \{(\Sigma, \Psi), (\Sigma, \Psi')\}$. By assumption, $\mathcal{D}^k b^k w^k \models \Psi \wedge \Psi'$. Hence, $\mathcal{D}^k b^k w^k \models \Psi$ and $\mathcal{D}^k b^k w^k \models \Psi'$. Thus, $A(k')$.

- rule $\vee$:
  For the rule to have been applied, $\{(\Sigma, \Psi \vee \Psi')\} \subseteq \Gamma_k$, and after its application, either $\Gamma_{k'} = \Gamma_k \cup \{(\Sigma, \Psi)\}$ or $\Gamma_{k''} = \Gamma_k \cup \{(\Sigma, \Psi')\}$. By assumption, $\mathcal{D}^k b^k w^k \models \Psi \vee \Psi'$. Hence, $\mathcal{D}^k b^k w^k \models \Psi$ or $\mathcal{D}^k b^k w^k \models \Psi'$. Thus, $A(k')$ or $A(k'')$. Thus, $B(k')$ or $B(k'')$. Therefore, $B(k)$.

- rule $=$:
  For the rule to have been applied, $(\Sigma, c = c') \subseteq \Gamma_k$ or $(\Sigma, \neg(c = c')) \subseteq \Gamma_k$. The rule is only applied when $(c = c')$, resp., $\neg(c = c')$ is unsatisfiable. Therefore, Γ_k is unsatisfiable. But this contradicts our main assumption $A(k)$. Hence, rule $=$ could not have been applicable to Γ_k.

- rule $\Rightarrow \wedge$:
  For the rule to have been applied, $(\Sigma, \varphi \Rightarrow \Phi \wedge \Phi') \in \Gamma_k$, and after its application, $\Gamma_{k'} = \Gamma_k \cup \{(\Sigma, (\varphi \Rightarrow \Phi) \wedge (\varphi \Rightarrow \Phi'))\}$. By assumption, $\mathcal{D}^k b^k w^k \models \varphi \Rightarrow \Phi \wedge \Phi'$. Hence, $\mathcal{D}^k b^k w^k \models \varphi \Rightarrow \Phi$ and $\mathcal{D}^k b^k w^k \models \varphi \Rightarrow \Phi'$. Thus, $A(k')$.

- rule $\delta \Rightarrow$:
  For the rule to have been applied, $(\Sigma, \varphi \Rightarrow \Phi) \in \Gamma_k$ where Φ is a disjunction of literals, and after its application, $\Gamma_{k'} = \Gamma_k \cup \{(\Sigma, \delta_1 \Rightarrow \Phi), (\Sigma, \delta_2 \Rightarrow \Phi), \ldots, (\Sigma, \delta_n \Rightarrow \Phi)\}$, where $\delta_i \in \mathit{Def}(\varphi)$. By assumption, $\mathcal{D}^k b^k w^k \models \varphi \Rightarrow \Phi$. Hence, $\mathcal{D}^k b^k w^k \models \delta_1 \Rightarrow \Phi$ and $\mathcal{D}^k b^k w^k \models \delta_2 \Rightarrow \Phi$ and ... and $\mathcal{D}^k b^k w^k \models \delta_n \Rightarrow \Phi$. Thus, $A(k')$.

- rule $\Rightarrow \vee$:
  For the rule to have been applied, $(\Sigma, \varphi \Rightarrow \Phi \vee \Phi') \in \Gamma_k$ where φ is definitive, and after its application, $\Gamma_{k'} = \Gamma_k \cup \{(\Sigma, (\varphi \Rightarrow \Phi) \vee (\varphi \Rightarrow \Phi'))\}$. By assumption, $\mathcal{D}^k b^k w^k \models \varphi \Rightarrow \Phi \vee \Phi'$. That is, for all $w \in C$, $\mathcal{D}^k b^k w \not\models \varphi$ or $\mathcal{D}^k b^k w \models \Phi \vee \Phi'$. Let $w' \models \varphi$. Then, because φ is definitive, for all $w \in C$ such that $w \neq w'$, $\mathcal{D}^k b^k w \not\models \varphi$. And necessarily, $\mathcal{D}^k b^k w' \models \Phi \vee \Phi'$, that is, $\mathcal{D}^k b^k w' \models \Phi$ or $\mathcal{D}^k b^k w' \models \Phi'$. Hence, for all $w'' \in C$, $\mathcal{D}^k b^k w'' \not\models \varphi$ or $\mathcal{D}^k b^k w'' \models \Phi$, or for all $w'' \in C$, $\mathcal{D}^k b^k w'' \not\models \varphi$ or $\mathcal{D}^k b^k w'' \models \Phi'$. Therefore, $\mathcal{D}^k b^k w \models \varphi \Rightarrow \Phi$ or $\mathcal{D}^k b^k w \models \varphi \Rightarrow \Phi'$, which implies that $\mathcal{D}^k b^k w \models (\varphi \Rightarrow \Phi) \vee (\varphi \Rightarrow \Phi')$. Thus, $A(k')$.

- rule Ξ:
  For the rule to have been applied, $(\Sigma, \{\alpha + \varsigma\}\Psi) \in \Gamma_k$. And if Γ_k contains (Σ', Ψ') such that $\Sigma' = \Sigma \xrightarrow{\alpha,\varsigma} e$, then $\Gamma_{k'} = \Gamma_k \cup \{(\Sigma', \Psi)\}$, else $\Gamma_{k'} = \Gamma_k \cup \{(\Sigma \xrightarrow{\alpha,\varsigma} e', \Psi)\}$, where e' is a fresh integer. By assumption, $\mathcal{D}^k b^k w^k \models \{\alpha + \varsigma\}\Psi$. Hence, $P_{NB}(\alpha, \varsigma, b) \neq 0$ and $\mathcal{D}^k b' w^k \models \Psi$, where $b' = BU(\alpha, \varsigma, b^k)$. Thus, $A(k')$.

- rule $\neg\Xi$:
  For the rule to have been applied, $(\Sigma, \neg\{\alpha + \varsigma\}\Psi) \in \Gamma_k$, and after its application, $\Gamma_{k'} = \Gamma_k \cup \{(\Sigma, \neg \mathit{Poss}(\alpha, \varsigma) \vee \{\alpha + \varsigma\}\neg\Psi)\}$. By assumption, $\mathcal{D}^k b^k w^k \models \neg\{\alpha + \varsigma\}\Psi$. Then by definition, it's not the case that $P_{NB}(\alpha, \varsigma, b) \neq 0$ and $\mathcal{D}^k b' w^k \models \Psi$, where $b' = BU(\alpha, \varsigma, b^k)$. That is, either $P_{NB}(\alpha, \varsigma, b^k) = 0$ or $\mathcal{D}^k b' w^k \models \neg\Psi$, where $b' = BU(\alpha, \varsigma, b^k)$. That is, either $\mathcal{D}^k b^k w^k \models \neg \mathit{Poss}(\alpha, \varsigma)$ or $\mathcal{D}^k b^k w^k \models \{\alpha + \varsigma\}\neg\Psi$. Hence, $\mathcal{D}^k b^k w^k \models \neg \mathit{Poss}(\alpha, \varsigma) \vee \{\alpha + \varsigma\}\neg\Psi$. Thus, $A(k')$.

- rules $\neg\mathbf{B}$ and $\neg\mathbf{U}$:
  In these cases, the assumption of $A(k)$ leads directly to $A(k')$.

- the SI phase:
  $A(k)$ is assumed. And Γ_k is a leaf node of an open branch of a finished and saturated tree. Let $SI(\Gamma_k)$ be the system of inequalities generated from the formulae in Γ_k according to the instructions for the SI phase. We must show that $SI(\Gamma_k)$ is feasible; then Γ_k will remain a leaf node (without the labeled formula $(0, \bot)$), and $A(k')$ will be trivially true.

  It can be seen from § 4.2 that the generation of equations and inequalities from law literals in Γ_k are direct translations of the semantic definitions of the respective formulae. The directness of their translations is, in part, due to law literals being independent of activity sequences. Given that $A(k)$, the subsystem of inequalities generated from only the law literals must be feasible.

  It may however not be clear how inequalities generated from executability, belief and utility literals in Γ_k affect the feasibility of the whole system $SI(\Gamma_k)$. We analyse this issue next. The inequalities generated from these three kinds of literals are also direct translations of their semantic definitions, except that the variables representing the probabilities of worlds at particular belief-states are carefully chosen: 'World-probability variable' ω_k^e represents the probability of world $w_k \in C^\#$ at activity-point e. An activity sequence

$$0 \xrightarrow{\alpha_0,\varsigma_0} e_1 \xrightarrow{\alpha_1,\varsigma_1} e_2 \cdots e_i \xrightarrow{\alpha_i,\varsigma_i} \cdots \xrightarrow{\alpha_{z-1},\varsigma_{z-1}} e_z.$$

represents the sequence of update operators in a sentence of the form

$$\{\alpha_0,\varsigma_0\}\{\alpha_1,\varsigma_1\}\cdots\{\alpha_i,\varsigma_i\}\cdots\{\alpha_{z-1},\varsigma_{z-1}\}\Psi,$$

where e_i is the activity-point representing the belief-state b^{e_i} reached after updating belief-state b^0 with α_0 and ς_0 to obtain b^{e_1}, then updating b^{e_1} with α_1 and ς_1 to obtain b^{e_2}, then updating b^{e_2} with ... to obtain b^{e_i}. The inequalities are generated in such a way that $\omega_k^{e_i}$ represents $b^{e_i}(w_k)$.

For instance, if $(\Sigma, \neg Poss(\alpha,\varsigma)), (\Sigma \xrightarrow{\alpha,\varsigma} e_z, \Psi) \in \Gamma_k$, then $SI(\Gamma_k)$ should be infeasible. This is because $(\Sigma \xrightarrow{\alpha,\varsigma} e_z, \Psi) \in \Gamma_k$ due to $(\Sigma, \{\alpha+\varsigma\}\Psi) \in \Gamma_k$, which implies that the belief-state obtained after updating the previous belief-state with α and ς (i.e., the belief-state represented by e_z) is reachable, but $(\Sigma, \neg Poss(\alpha,\varsigma)) \in \Gamma_k$ implies that the belief-state represented by e_z is not reachable.

For $SI(\Gamma_k)$ to be infeasible due to this contradiction, requires that the equations generated from the two formulae both refer to the *same variables* $\omega_k^{e_z}$ for $k = 1, 2, \ldots, n$. Then

$$\sum_{j=1}^{n} pr_j^{\varsigma|\alpha} \sum_{i=1}^{n} pr_{i,j}^{\alpha} \omega_i^{e_z} = 0 \tag{23}$$

is generated for $(\Sigma, \neg Poss(\alpha,\varsigma))$ and

$$\omega_1^{e_z} + \omega_2^{e_z} + \cdots + \omega_n^{e_z} = 1 \tag{24}$$

and

$$\omega_k^{e_z} = \frac{pr_k^{\varsigma|\alpha} \sum_{i=1}^{n} pr_{i,k}^{\alpha} \omega_i^{e_{z-1}}}{\sum_{j=1}^{n} pr_j^{\varsigma|\alpha} \sum_{i=1}^{n} pr_{i,j}^{\alpha} \omega_i^{e_{z-1}}}, \text{ for } k = 1, 2, \ldots, n \tag{25}$$

and

$$\sum_{j=1}^{n} pr_j^{\varsigma|\alpha} \sum_{i=1}^{n} pr_{i,j}^{\alpha} \omega_i^{e_{z-1}} \neq 0$$

are generated for $(\Sigma \xrightarrow{\alpha,\varsigma} e_z, \Psi)$. By (24), there exists a variable $\omega_k^{e_z} > 0$. By (25), whenever $\omega_k^{e_z} > 0$, then $pr_k^{\varsigma|\alpha} > 0$ and for some $i \in \{1, 2, \ldots, n\}$, $pr_{i,k}^{\alpha} > 0$ and $\omega_i^{e_{z-1}} > 0$. There exists a term $pr_k^{\varsigma|\alpha} \times pr_{i,k}^{\alpha} \times \omega_k^{e_z}$ of (23) which is greater than zero, which is a contradiction, because then $\sum_{j=1}^{n} pr_j^{\varsigma|\alpha} \sum_{i=1}^{n} pr_{i,j}^{\alpha} \omega_i^{e_z} \neq 0$.

During the generation of activity sequences in rule Ξ, fresh integers are used only when another labeled formula with the same sequence does not exist, else the existing activity-point/integer is used. Rule Ξ is repeated here for convenience: If Γ_k^j contains $(\Sigma, \{\alpha+\varsigma\}\Psi)$ then: if Γ_k^j contains (Σ', Ψ') such that $\Sigma' = \Sigma \xrightarrow{\alpha,\varsigma} e$, then create node $\Gamma_{k+1}^j = \Gamma_k^j \cup \{(\Sigma', \Psi)\}$, else create node $\Gamma_{k+1}^j = \Gamma_k^j \cup \{(\Sigma \xrightarrow{\alpha,\varsigma}$

$e', \Psi)\}$, where e' is a fresh integer. 'World-probability variables' are consistently named according to activity-points in the generation of $SI(\Gamma_k)$. In this way, as in the example above, a chain of constraints is set up such that if $SI(\Gamma_k)$ is infeasible, then the conjunction of formulae involved in the generation of $SI(\Gamma_k)$ is unsatisfiable. In other words, if the conjunction of formulae involved in the generation of $SI(\Gamma_k)$ is satisfiable, then $SI(\Gamma_k)$ is feasible.

In general, $SI(\Gamma_k)$ is infeasible if and only if there is no solution (assignment of values to variables) such that all equations and inequalities in the system are simultaneously true. Assume $SI(\Gamma_k)$ is infeasible. Then there exists at least one equation or inequality that is false. All the kinds of equations and inequalities that can possibly appear in $SI(\Gamma_k)$ are considered in the list below. We now show for each case that if the corresponding equation/inequality is false, then the conjunction of formulae involved in the generation of $SI(\Gamma_k)$ is unsatisfiable. This would contradict the assumption that the conjunction of formulae involved in the generation of $SI(\Gamma_k)$ is satisfiable.

1. $c_1pr^{\alpha}_{j,1} + c_2pr^{\alpha}_{j,2} + \cdots + c_npr^{\alpha}_{j,n} \bowtie q$ such that $c_k = 1$ if $w_k \models \varphi$, else $c_k = 0$—for a formula $(\Sigma, \phi \Rightarrow [\alpha]\varphi \bowtie q) \in \Gamma$, for every j such that $w_j \models \phi$. Simply, if $c_1pr^{\alpha}_{j,1} + c_2pr^{\alpha}_{j,2} + \cdots + c_npr^{\alpha}_{j,n} \bowtie q$ is false, then by the definition of $[\alpha]\varphi \bowtie q$, $\phi \Rightarrow [\alpha]\varphi \bowtie q$ is unsatisfiable.
2. $c_1pr^{\alpha}_{j,1} + c_2pr^{\alpha}_{j,2} + \cdots + c_npr^{\alpha}_{j,n} \not\bowtie q$ such that $c_k = 1$ if $w_k \models \varphi$, else $c_k = 0$—for a formula $(\Sigma, \phi \Rightarrow \neg[\alpha]\varphi \bowtie q) \in \Gamma$, for every j such that $w_j \models \phi$. Symmetrically to the case above, $\phi \Rightarrow \neg[\alpha]\varphi \bowtie q$ is unsatisfiable.
3. $pr^{\alpha}_{j,1} + pr^{\alpha}_{j,2} + \cdots + pr^{\alpha}_{j,n} = \lceil pr^{\alpha}_{j,1} + pr^{\alpha}_{j,2} + \cdots + pr^{\alpha}_{j,n} \rceil$—for a formulae $(\Sigma, \phi \Rightarrow [\alpha]\varphi \bowtie q) \in \Gamma$ or $(\Sigma, \phi \Rightarrow \neg[\alpha]\varphi \bowtie q) \in \Gamma$, for some j such that $w_j \models \phi$. Now, $pr^{\alpha}_{j,1} + pr^{\alpha}_{j,2} + \cdots + pr^{\alpha}_{j,n} = \lceil pr^{\alpha}_{j,1} + pr^{\alpha}_{j,2} + \cdots + pr^{\alpha}_{j,n} \rceil$ being false will ensure that neither $\sum_{w' \in W} R_\alpha(w_j, w') = 1$ nor $\sum_{w' \in W} R_\alpha(w_j, w') = 0$. However, as stated in Definition 3.5, either $\sum_{w' \in W} R_\alpha(w_j, w') = 1$ or $\sum_{w' \in W} R_\alpha(w_j, w') = 0$ for *every* $w_j \in C$. Thus, no SDL structure exists which can satisfy $\phi \Rightarrow [\alpha]\varphi \bowtie q$ or $\phi \Rightarrow \neg[\alpha]\varphi \bowtie q$.
4. $pr^{\varsigma|\alpha}_j \bowtie q$—for a formula $(\Sigma, \phi \Rightarrow (\varsigma|\alpha) \bowtie q) \in \Gamma$, for some j such that $w_j \models \phi$. If $pr^{\varsigma|\alpha}_j \bowtie q$ is false, then by definition of $(\varsigma|\alpha) \bowtie q$, $\phi \Rightarrow (\varsigma|\alpha) \bowtie q$ is unsatisfiable.
5. $pr^{\varsigma|\alpha}_j \not\bowtie q$—for a formula $(\Sigma, \phi \Rightarrow \neg(\varsigma|\alpha) \bowtie q) \in \Gamma$, for some j such that $w_j \models \phi$. Symmetrically to the case above, $\phi \Rightarrow \neg(\varsigma|\alpha) \bowtie q$ is unsatisfiable.
6. $pr^{\varsigma_1|\alpha}_j + pr^{\varsigma_2|\alpha}_j + \cdots + pr^{\varsigma_m|\alpha}_j = \lceil (pr^{\alpha}_{1,j} + pr^{\alpha}_{2,j} + \cdots + pr^{\alpha}_{n,j})/n \rceil$—for a formula $(\Sigma, \phi \Rightarrow (\varsigma|\alpha) \bowtie q) \in \Gamma$ or $(\Sigma, \phi \Rightarrow \neg(\varsigma|\alpha) \bowtie q) \in \Gamma$, for some j such that $w_j \models \phi$. Now, $pr^{\varsigma_1|\alpha}_j + pr^{\varsigma_2|\alpha}_j + \cdots + pr^{\varsigma_m|\alpha}_j = \lceil (pr^{\alpha}_{1,j} + pr^{\alpha}_{2,j} + \cdots + pr^{\alpha}_{n,j})/n \rceil$ being false will ensure that whenever there exists a $w_i \in C$ such that $R_\alpha(w_i, w_j) > 0$, then $\sum_{\varsigma \in \Omega} Q_\alpha(w_j, \varsigma) \neq 1$, and whenever there exists a $w_i \in C$ such that $R_\alpha(w_i, w_j) = 0$, then $\sum_{\varsigma \in \Omega} Q_\alpha(w_j, \varsigma) \neq 0$. However, as stated in Definition 3.5, if there exists a $w_i \in C$ such that $R_\alpha(w_i, w_j) > 0$, then $\sum_{\varsigma \in \Omega} Q_\alpha(w_j, \varsigma) = 1$, else $\sum_{\varsigma \in \Omega} Q_\alpha(w_j, \varsigma) = 0$ for *every* $w_j \in C$. Thus, no SDL structure exists which can satisfy $\phi \Rightarrow (\varsigma|\alpha) \bowtie q$ or $\phi \Rightarrow \neg(\varsigma|\alpha) \bowtie q$.

7. For every $(\Sigma, \Psi) \in \Gamma$, the following equations are in $SI(\Gamma)$.

$$\omega_k^{e_{h+1}} = BT(e_h, k, \alpha_h, \varsigma_h) \text{ (for } k = 1, 2, \ldots, n \text{ and } h = 0, 1, \ldots, z-1), \quad (26)$$

$$\Pi(e_h, \alpha_h, \varsigma_h) \neq 0 \text{ (for } h = 0, 1, \ldots, z-1) \text{ and} \quad (27)$$

$$\omega_1^{e_h} + \omega_2^{e_h} + \cdots + \omega_n^{e_h} = 1 \text{ (for } h = 0, 1, \ldots, z), \quad (28)$$

where Σ is $0 \xrightarrow{\alpha_0, \varsigma_0} e_1 \xrightarrow{\alpha_1, \varsigma_1} e_2 \cdots \xrightarrow{\alpha_{z-1}, \varsigma_{z-1}} e_z$, $\Sigma \neq 0$ and e_0 is 0.
Note that if $(\Sigma, \Psi) \in \Gamma$, then $(0 \xrightarrow{\alpha_0, \varsigma_0} e_1 \xrightarrow{\alpha_1, \varsigma_1} e_2 \cdots \xrightarrow{\alpha_{i-1}, \varsigma_{i-1}} e_i, \{\alpha_i + \varsigma_i\}\Psi')$ must be in Γ, for some $0 < i < z$. And by the semantics, $\mathcal{D}bw \models \{\alpha_i + \varsigma_i\}\Psi'$ if and only if

$$P_{NB}(\alpha_i, \varsigma_i, b) \neq 0 \text{ and } \mathcal{D}b'w \models \Psi', \text{ where } b' = BU(\alpha_i, \varsigma_i, b).$$

If any one of (26), (27) or (28) cannot be made true, then all of them cannot simultaneously be true. Now assume that $\omega_k^{e_{i+1}} \neq BT(e_i, k, \alpha_i, \varsigma_i)$ for some $k \in \{1, 2, \ldots, n\}$ and some $i \in \{0, 1, \ldots, z-1\}$, or $\Pi(e_i, \alpha_i, \varsigma_i) = 0$ for some $i \in \{0, 1, \ldots, z-1\}$, or $\omega_1^{e_{i+1}} + \omega_2^{e_{i+1}} + \cdots + \omega_n^{e_{i+1}} \neq 1$ for some $i \in \{0, 1, \ldots, z\}$. Then, respectively,

$$b^{i+1}(w_k) \neq b'(w_k), \text{ where } b' = BU(\alpha_i, \varsigma_i, b), \text{ i.e., } b^{i+1} \neq b',$$

or

$$P_{NB}(\alpha_i, \varsigma_i, b) = 0,$$

or

$$\sum_{j=1}^{n} b^{i+1}(w_j) \neq 1,$$

where $b^{i+1} = \{(w_1, \omega_1^{e_{i+1}}), (w_2, \omega_2^{e_{i+1}}), \ldots, (w_n, \omega_n^{e_{i+1}})\}$. In any case, $\{\alpha_i + \varsigma_i\}\Psi'$ is unsatisfiable.

8. $\Pi(e, \alpha, \varsigma) \neq 0$—for a formula $(\Sigma e, \mathit{Poss}(\alpha, \varsigma)) \in \Gamma$. If $\Pi(e, \alpha, \varsigma) = 0$, then by definition of $\mathit{Poss}(\alpha, \varsigma)$, there exists no $\mathcal{D}$, $b = \{(w_1, \omega_1^e), (w_2, \omega_2^e), \ldots, (w_n, \omega_n^e)\}$ and $w \in C$ for which $\mathcal{D}bw \models \mathit{Poss}(\alpha, \varsigma)$.

9. $\Pi(e, \alpha, \varsigma) = 0$—for a formula $(\Sigma e, \neg \mathit{Poss}(\alpha, \varsigma)) \in \Gamma$. If $\Pi(e, \alpha, \varsigma) \neq 0$, then by definition of $\mathit{Poss}(\alpha, \varsigma)$, there exists no $\mathcal{D}$ such that $\Pi(e, \alpha, \varsigma) \neq 0$, $b = \{(w_1, \omega_1^e), (w_2, \omega_2^e), \ldots, (w_n, \omega_n^e)\}$ and $w \in C$ for which $\mathcal{D}bw \not\models \mathit{Poss}(\alpha, \varsigma)$.

10. $c_1\omega_1^e + c_2\omega_2^e + \cdots + c_n\omega_n^e \bowtie q$, where $c_k = 1$ if $w_k \models \varphi$, else $c_k = 0$—for a formula $(\Sigma e, \mathbf{B}\varphi \bowtie q) \in \Gamma$. $c_1\omega_1^e + c_2\omega_2^e + \cdots + c_n\omega_n^e \bowtie q$ is false if and only if $\sum_{w_k \in C,\, w_k \models \varphi} b(w_k) \not\bowtie q$, where $b = \{(w_1, \omega_1^e), (w_2, \omega_2^e), \ldots, (w_n, \omega_n^e)\}$ iff $\mathcal{D}bw \models \mathbf{B}\varphi \not\bowtie q$ iff $\mathbf{B}\varphi \bowtie q$ is unsatisfiable when $c_1\omega_1^e + c_2\omega_2^e + \cdots + c_n\omega_n^e \not\bowtie q$.

11. $R_j = r$—for a formula $(\Sigma, \phi \Rightarrow \mathit{Reward}(r)) \in \Gamma$, for some j such that $w_j \models \phi$. If $R_j = r$ must be false, then it's not the case that for all $w' \in C$, $\mathcal{D}bw' \not\models \phi$ or $\mathcal{D}bw' \models \mathit{Reward}(r)$. Therefore, in this case, $\phi \Rightarrow \mathit{Reward}(r)$ is unsatisfiable.

12. $R_j \neq r$—for a formula $(\Sigma, \phi \Rightarrow \neg\mathit{Reward}(r)) \in \Gamma$, for some j such that $w_j \models \phi$. If $R_j \neq r$ must be false, then it's not the case that for all $w' \in C$, $\mathcal{D}bw' \not\models \phi$ or $\mathcal{D}bw' \models \neg\mathit{Reward}(r)$. Therefore, in this case, $\phi \Rightarrow \neg\mathit{Reward}(r)$ is unsatisfiable.

13. $C_j^\alpha = r$—for a formula $(\Sigma, \phi \Rightarrow \mathit{Cost}(\alpha, r)) \in \Gamma$, for some j such that $w_j \models \phi$. If $C_j^\alpha = r$ must be false, then it's not the case that for all $w' \in C$, $\mathcal{D}bw' \not\models \phi$ or $\mathcal{D}bw' \models \mathit{Cost}(\alpha, r)$. Therefore, in this case, $\phi \Rightarrow \mathit{Cost}(\alpha, r)$ is unsatisfiable.

14. $C_j^\alpha \neq r$—for a formula $(\Sigma, \phi \Rightarrow \neg\mathit{Cost}(\alpha, r)) \in \Gamma$, for some j such that $w_j \models \phi$. If $C_j^\alpha \neq r$ must be false, then it's not the case that for all $w' \in C$, $\mathcal{D}bw' \not\models \phi$ or $\mathcal{D}bw' \models \neg\mathit{Cost}(\alpha, r)$. Therefore, in this case, $\phi \Rightarrow \neg\mathit{Cost}(\alpha, r)$ is unsatisfiable.

15. $\omega_1^e(R_1 - C_1^\alpha) + \omega_2^e(R_2 - C_2^\alpha) + \cdots + \omega_n^e(R_n - C_n^\alpha) \bowtie q$ (abbreviated as $RC(\alpha, e) \bowtie q$)—for a formula $(\Sigma e, \mathbf{U}\{\alpha\} \bowtie q) \in \Gamma$. If $RC(\alpha, e) \bowtie q$ must be false, then by definition, $\mathbf{U}\{\alpha\} \bowtie q$ is unsatisfiable at b, where $b = \{(w_1, \omega_1^e), (w_2, \omega_2^e), \ldots, (w_n, \omega_n^e)\}$.

16. $U(\{\alpha_1\}\{\alpha_2\}\cdots\{\alpha_y\}, e_{z,\text{-}}) \bowtie q$—for a utility literal of the form $(\Sigma e_z, \mathbf{U}\{\alpha_1\}\{\alpha_2\} \cdots \{\alpha_y\} \bowtie q)$ for $y \geq 2$. Let e be an arbitrary activity-point representing belief-state $b = \{(w_1, \omega_1^e), (w_2, \omega_2^e), \ldots, (w_n, \omega_n^e)\}$. Then by the generation of $SI(\Gamma)$ and by the definition of a structure $\mathcal{D}$,

$$RC(\alpha, b) = RC(\alpha, e), \tag{29}$$

$$P_{NB}(\alpha, \varsigma, b) = \Pi(e, \alpha, \varsigma) \text{ and} \tag{30}$$

$$b'(w_k) = s\omega_k^{e'} = BT(e, k, \alpha, \varsigma), \text{ where } b' = BU(\alpha, \varsigma, b). \tag{31}$$

Hence, due to equalities (29), (30) and (31),

$$\begin{aligned} & RC(\alpha_1, e_{z,\text{-}}) + \sum_{\varsigma_i \in \Omega^\#} \Pi(e_{z,\text{-}}, \alpha_1, \varsigma_i) U(\Lambda, e_{z+1,i}) = \\ & RC(\alpha_1, e_{z,\text{-}}) + \sum_{\substack{\varsigma_i \in \Omega^\# \\ U(\Lambda, e_{z+1,i}) = r}} \Pi(e_{z,\text{-}}, \alpha_1, \varsigma_i) \cdot r = \\ & RC(\alpha_1, b^z) + \sum_{\substack{\varsigma \in \Omega \\ b' = BU(\alpha_1, \varsigma, b) \\ \mathcal{D}b'w \models \mathbf{U}\Lambda = r}} P_{NB}(\alpha_1, \varsigma, b^z) \cdot r \bowtie q, \end{aligned} \tag{32}$$

where $b^z = \{(w_1, \omega_1^{e_z}), (w_2, \omega_2^{e_z}), \ldots, (w_n, \omega_n^{e_z})\}$ and Λ is $\{\alpha_2\}\cdots\{\alpha_y\}$. Therefore, if $U(\{\alpha_1\}\Lambda, e_{z,\text{-}}) \bowtie q$ must be false, $\mathcal{D}b^z w \models \mathbf{U}\{\alpha_1\}\Lambda \bowtie q$ must be false (for all w; such that $\mathcal{D}$ satisfies equalities (29), (30) and (31)).

17. $\omega_1^{e_i} + \cdots + \omega_n^{e_i} = 1$—for some activity-point/node e_i in some utility tree. If $\omega_1^{e_i} + \cdots + \omega_n^{e_i} = 1$ must be false, then Inequality (32) is undefined, where $b^i = \{(w_1, \omega_1^{e_i}), (w_2, \omega_2^{e_i}), \ldots, (w_n, \omega_n^{e_i})\}$, $b^{i+1} = \{(w_1, \omega_1^{e_{i+1}}), (w_2, \omega_2^{e_{i+1}}), \ldots, (w_n, \omega_n^{e_{i+1}})\}$ and $b^{i+1} = BU(\alpha_i, \varsigma, b^i)$. Therefore, $\mathbf{U}\{\alpha_1\}\Lambda \bowtie q$ cannot be satisfied.

18. Assume

$$\Pi(e, \alpha, \varsigma) = 0 \;||\; \Pi(e, \alpha, \varsigma) \neq 0, \omega_1^{e'} = BT(e, 1, \alpha, \varsigma), \ldots, \omega_n^{e'} = BT(e, n, \alpha, \varsigma) \tag{33}$$

is in $SI(\Gamma)$ for some $e \xrightarrow{\alpha, \varsigma} e'$ in some utility tree.

Let $SI^{-}(\Gamma)$ be $SI(\Gamma)$ without (33). Recall that $SI(\Gamma)$ is feasible if and only if $SI^{-}(\Gamma) \cup \{\Pi(e,\alpha,\varsigma) = 0\}$ is feasible *or* $SI^{-}(\Gamma) \cup \{\Pi(e,\alpha,\varsigma) \neq 0, \omega_1^{e'} = BT(e,1,\alpha,\varsigma), \ldots, \omega_n^{e'} = BT(e,n,\alpha,\varsigma)\}$ is feasible. Thus, if $SI(\Gamma)$ is infeasible due to (33), then (i) $SI^{-}(\Gamma) \cup \{\Pi(e,\alpha,\varsigma) = 0\}$ must be infeasible *and* (ii) $SI^{-}(\Gamma) \cup \{\Pi(e,\alpha,\varsigma) \neq 0, \omega_1^{e'} = BT(e,1,\alpha,\varsigma), \ldots, \omega_n^{e'} = BT(e,n,\alpha,\varsigma)\}$ must be infeasible. We assume $\Pi(e,\alpha,\varsigma) = 0$ must be false in case (i). Therefore, in case (ii), $\Pi(e,\alpha,\varsigma) \neq 0$ must be true and at least one of the equations $\omega_1^{e'} = BT(e,1,\alpha,\varsigma), \ldots, \omega_n^{e'} = BT(e,n,\alpha,\varsigma)$ must be false. Suppose it is $\omega_n^{e'} = BT(e,k,\alpha,\varsigma)$ which is false. Then

$$b'(w_k) \neq b''(w_k), \text{ where } b'' = BU(\alpha,\varsigma,b), \text{ i.e., } b' \neq b'',$$

where $b = \{(w_1,\omega_1^e), (w_2,\omega_2^e), \ldots, (w_n,\omega_n^e)\}$ and $b' = \{(w_1,\omega_1^{e'}), (w_2,\omega_2^{e'}), \ldots, (w_n,\omega_n^{e'})\}$. But by definition of Inequality (32), $b' = b''$. Hence, $\mathbf{U}\{\alpha_1\}\Lambda \bowtie q$ cannot be satisfied.

We have shown for each case that if the corresponding equation/inequality is false, then the conjunction of formulae involved in the generation of $SI(\Gamma_k)$ is unsatisfiable, contradicting the assumption that the conjunction of formulae involved in the generation of $SI(\Gamma_k)$ is satisfiable. $SI(\Gamma_k)$ is thus feasible.

■

5.2 Completeness of the Decision Procedure

Theorem 5.2 (Completeness) *If* $\models \Psi$ *then* $\vdash \Psi$. *(Contrapositively, if* $\not\vdash \Psi$ *then* $\not\models \Psi$.*)*

Let $\psi = \neg\Psi$. Then $\not\vdash \Psi$ means that there is an open branch of a finished tree for ψ. And

$$\begin{aligned} \not\models \Psi &\iff (\exists \mathcal{D})\ \mathcal{D} \not\models \Psi \\ &\iff (\exists \mathcal{D}, b)\ \mathcal{D}b \not\models \Psi \\ &\iff (\exists \mathcal{D}, b, w)\ \mathcal{D}bw \not\models \Psi \\ &\iff (\exists \mathcal{D}, b, w)\ \mathcal{D}bw \models \psi. \end{aligned}$$

For the completeness proof, it thus suffices to construct for some open branch of a finished tree for $\psi \in \mathcal{L}_{SDL}$, a SDL structure $\mathcal{D} = \langle R, Q, U\rangle$ such that there is a belief-state b and a world $w \in C$ such that ψ is satisfied in $\mathcal{D}$ at b at w.

We now start with the description of the construction of a SDL structure, given the leaf node Γ of some open branch of a finished tree.

Definition 5.1 *A solutions for* $SI(\Gamma)$ *is a set of values* $\{s_{i,j}^{\alpha}, s_j^{\varsigma|\alpha}, sR_i, sC_i^{\alpha}, s\omega_i^e \mid \alpha \in \mathcal{A}, \varsigma \in \Omega,\ i,j = 1,2,\ldots,|C|, e = 0$ *or* e *an integer introduced by rule* Ξ *to a formula appearing in* $\Gamma\}$ *such that the assignments of these values to the variables (as follows) solves every equation and inequality in* $SI(\Gamma)$ *simultaneously:* $pr_{i,j}^{\alpha} \leftarrow s_{i,j}^{\alpha}, pr_j^{\varsigma|\alpha} \leftarrow s_j^{\varsigma|\alpha}, R_i \leftarrow sR_i, C_i^{\alpha} \leftarrow sC_i^{\alpha}$ *and* $\omega_i^e \leftarrow s\omega_i^e$ *for all* $\alpha \in \mathcal{A}, \varsigma \in \Omega,\ i,j \in \{1,2,\ldots,|C|\}, e = 0$ *and* e *an integer introduced by rule* Ξ *to a formula appearing in* $\Gamma\}$. *Let* $Z(\Gamma)$ *be the set of solutions for* $SI(\Gamma)$.

$\mathcal{D} = \langle R, Q, U \rangle$ can be constructed as follows. Let sln be a solution in $Z(\Gamma)$.

- For every action $\alpha \in \mathcal{A}$, the accessibility relation R_α can be constructed as follows. For $i, j = 1, 2, \ldots, |C|$, let $R_\alpha(w_i, w_j) = s^\alpha_{i,j}$, where $w_i, w_j \in C^\#$ and $s^\alpha_{i,j} \in sln$.
- For every action $\alpha \in \mathcal{A}$, the perceivability relation Q_α can be constructed as follows. For $j = 1, 2, \ldots, |C|$, let $Q_\alpha(w_j, \varsigma_j) = s^{\varsigma|\alpha}_j$, where $w_j \in C^\#$, $\varsigma \in \Omega$ and $s^{\varsigma|\alpha}_j \in sln$.
- For $i = 1, 2, \ldots, |C|$, let $Re(w_i) = sR_i$, where $w_i \in C^\#$ and $sR_i \in sln$. For every action $\alpha \in \mathcal{A}$ and $i = 1, 2, \ldots, |C|$, let $Co_\alpha(w_i) = sC^\alpha_i$, where $w_i \in C^\#$ and $sC^\alpha_i \in sln$. Let $U = \langle Re, Co \rangle$ such that $Co = \{(\alpha, Co_\alpha) \mid \alpha \in \mathcal{A}\}$.

Lemma 5.2 *$\mathcal{S}$ is an SDL structure.*

Proof:
The components of the structure are well-formed:

- Due to Γ being open (and by the SI phase), we know that $SI(\Gamma)$ is feasible and hence, there exists a solution in $Z(\Gamma)$.

  By construction, R maps each action $\alpha \in \mathcal{A}$ to R_α such that R_α is a relation in $(C \times C) \times \mathbb{R}_{[0,1]}$. Moreover, by the nature of the SI generated from Γ, R_α is a (total) function $R_\alpha : (C \times C) \mapsto \mathbb{R}_{[0,1]}$.

  And by construction, the fact that

  $$pr^\alpha_{j,1} + pr^\alpha_{j,2} + \cdots + pr^\alpha_{j,n} = \lceil pr^\alpha_{j,1} + pr^\alpha_{j,2} + \cdots + pr^\alpha_{j,n} \rceil$$

  is an equation in any SI generated, means that either $\sum_{w' \in W} R_\alpha(w_j, w') = 1$ or $\sum_{w' \in W} R_\alpha(w_j, w') = 0$, for every $w_j \in C$.

- Due to Γ being open (and by the SI phase), we know that $SI(\Gamma)$ is feasible and hence, there exists a solution in $Z(\Gamma)$.

  By construction, Q maps each action $\alpha \in \mathcal{A}$ to Q_α such that Q_α is a relation in $(C \times \Omega) \times \mathbb{R}_{[0,1]}$.

  Moreover, by the nature of the SI generated from Γ, Q_α is a (total) function $Q_\alpha : (C \times \Omega) \mapsto \mathbb{R}_{[0,1]}$.

  And by construction, the fact that in any SI generated, there is an equation

  $$pr^{\varsigma_1|\alpha}_j + pr^{\varsigma_2|\alpha}_j + \cdots + pr^{\varsigma_m|\alpha}_j = \lceil (pr^\alpha_{1,j} + pr^\alpha_{2,j} + \cdots + pr^\alpha_{n,j})/n \rceil$$

  for all $w_j \in C$, if there exists a $w_i \in C$ such that $R_\alpha(w_i, w_j) > 0$, then $\sum_{\varsigma \in \Omega} Q_\alpha(w_j, \varsigma) = 1$, else $\sum_{\varsigma \in \Omega} Q_\alpha(w_j, \varsigma) = 0$.

- By construction, $U = \langle Re, Co \rangle$, where $Re : C \mapsto \mathbb{R}$ and Co is a mapping from $\mathcal{A}$ to a function $Co_\alpha : C \mapsto \mathbb{R}$, for each $\alpha \in \mathcal{A}$.

■

Lemma 5.3 *Let* Γ *be the leaf node of an open branch of a finished tree. We know that* $Z(\Gamma)$ *is not empty. If* $\mathcal{D}$ *is constructed as described above, then for all* $(\Sigma, \Psi) \in \Gamma$*, there exists a* b *and a* w *such that* $\mathcal{D}bw \models \Psi$.

Proof:
The proof will be by induction on the structure of a formula.

The induction step will work as follows. Let $\gamma' \subseteq \Gamma$ be added to Γ due to some rule applied to $\gamma \subseteq \Gamma$. Thus, we need to prove that IF for all $(\Sigma', \Psi') \in \gamma'$, there exists a b' and a w' such that $\mathcal{D}b'w' \models \Psi'$, THEN for all $(\Sigma, \Psi) \in \gamma$, there exists a b and a w such that $\mathcal{D}bw \models \Psi$.

We assume the antecedent (induction hypothesis).

Base case:

- Ψ is $c = c'$. Because $(\Sigma, \bot) \notin \Gamma$ for some Σ, rule $=$ was not applied. Hence, c is identical to c', and $\mathcal{D}bw \models c = c'$ (for all b and w).

- Ψ is $\neg(c = c')$. Because $(\Sigma, \bot) \notin \Gamma$ for some Σ, rule $=$ was not applied. Hence, c is not identical to c', and $\mathcal{S}, w \models \neg(c = c')$ (for all b and w).

- Ψ is $Poss(\alpha, \varsigma)$ (i.e., $(\Sigma e, Poss(\alpha, \varsigma)) \in \Gamma$). By the generation of $SI(\Gamma)$,

$$\Pi(e, \alpha, \varsigma, n) \neq 0 \;\text{ and }\; s\omega_1^e + s\omega_2^e + \cdots + s\omega_n^e = 1$$

  iff

$$\sum_{j=1}^{n} s_j^{\varsigma|\alpha} \sum_{i=1}^{n} s_{i,j}^{\alpha} s\omega_i^e \neq 0 \;\text{ and }\; s\omega_1^e + s\omega_2^e + \cdots + s\omega_n^e = 1.$$

  By construction, this implies that

$$\sum_{j=1,\, w_j \in C\#}^{n} Q_\alpha(\varsigma, w_j) \sum_{i=1,\, w_i \in C\#}^{n} R_\alpha(w_i, w_j) b^e(w_i) \neq 0,$$

  where

$$b^e = \{(w_1, s\omega_1^e), (w_2, s\omega_2^e), \ldots, (w_n, s\omega_n^e)\}$$

  iff

$$P_{NB}(\alpha, \varsigma, b^e) \neq 0$$

  iff

$$\mathcal{D}b^e w \models Poss(\alpha, \varsigma).$$

- Ψ is $\neg Poss(\alpha, \varsigma)$ (i.e., $(\Sigma e, \neg Poss(\alpha, \varsigma)) \in \Gamma$). By the generation of $SI(\Gamma)$,

$$\Pi(e, \alpha, \varsigma, n) = 0 \;\text{ and }\; s\omega_1^e + s\omega_2^e + \cdots + s\omega_n^e = 1$$

  iff

$$\sum_{j=1}^{n} s_j^{\varsigma|\alpha} \sum_{i=1}^{n} s_{i,j}^{\alpha} s\omega_i^e = 0 \;\text{ and }\; s\omega_1^e + s\omega_2^e + \cdots + s\omega_n^e = 1.$$

By construction, this implies that

$$\sum_{j=1,\, w_j \in C\#}^{n} Q_\alpha(\varsigma, w_j) \sum_{i=1,\, w_i \in C\#}^{n} R_\alpha(w_i, w_j) b^e(w_i) = 0,$$

where

$$b^e = \{(w_1, s\omega_1^e), (w_2, s\omega_2^e), \ldots, (w_n, s\omega_n^e)\}$$

iff

$$P_{NB}(\alpha, \varsigma, b^e) = 0$$

iff

$$\mathcal{D} b^e w \models \neg Poss(\alpha, \varsigma).$$

- Ψ is $\mathbf{B}\varphi \bowtie q$ (i.e., $(\Sigma e, \mathbf{B}\varphi \bowtie q) \in \Gamma$). By the generation of $SI(\Gamma)$,

$$c_1 s\omega_1^e + c_2 s\omega_2^e + \cdots + c_n s\omega_n^e \bowtie q,$$

and

$$\omega_1^e + \omega_2^e + \cdots + \omega_n^e = 1,$$

where $c_k = 1$ if $w_k \models \varphi$, else $c_k = 0$.

This implies that

$$\sum_{k=1,\, w_k \in C\#,\, w_k \models \varphi}^{n} b^e(w_k) \ \bowtie \ q,$$

where

$$b^e = \{(w_1, s\omega_1^e), (w_2, s\omega_2^e), \ldots, (w_n, s\omega_n^e)\}$$

iff

$$\mathcal{D} b^e w \models \mathbf{B}\varphi \bowtie q.$$

- Ψ is $\mathbf{U}\Lambda \bowtie q$ (i.e., $(\Sigma e_z, \mathbf{U}\Lambda \bowtie q) \in \Gamma$). When Λ is $\{\alpha\}$, $RC(\alpha, e_z) \bowtie q \in SI(\Gamma)$. That is, by construction,

$$s\omega_1^{e_z}(sR_1 - sC_1^\alpha) + s\omega_2^{e_z}(sR_2 - sC_2^\alpha) + \cdots + s\omega_n^{e_z}(sR_n - sC_n^\alpha) \bowtie q,$$

where $s\omega_k^{e_z}, sR_k, sC_k^\alpha \in sln$, for $k = 1, 2, \ldots, n$. Let $b^z = \{(w_1, s\omega_1^{e_z})$, $(w_2, s\omega_2^{e_z})$, …, $(w_n, s\omega_n^{e_z})\}$. Then,

$$\sum_{k=1, w_k \in C\#}^{n} \big(Re(w_k) - Co_\alpha(w_k)\big) b^z(w_k) \bowtie q$$

iff

$$\mathcal{D} b^z w \models \mathbf{U}\{\alpha\} \bowtie q.$$

Dealing with $(\Sigma e_z, \mathbf{U}\{\alpha_1\}\{\alpha_2\} \cdots \{\alpha_y\} \bowtie q) \in \Gamma$, for $y \geq 2$ is more complicated: The inequality added to $SI(\Gamma)$ is

$$U(\{\alpha_1\}\{\alpha_2\} \cdots \{\alpha_y\}, e_{z,\_}) \bowtie q, \text{ (cf. Eq. (11), § 4.2.4)}$$

where $e_{z,\text{-}} = e_z$. That is,

$$RC(\alpha_1, e_{z,\text{-}}) + \sum_{\varsigma_i \in \Omega^{\#}} \Pi(e_{z,\text{-}}, \alpha_1, \varsigma_i) U(\Lambda, e_{z+1,i}) \bowtie q,$$

where

$$U(\{\alpha_y\}, e_{z+y-1,x}) = RC(\alpha_y, e_{z+y-1,x})$$

and Λ is $\{\alpha_2\} \cdots \{\alpha_y\}$.

Let e be an arbitrary activity-point representing belief-state $b = \{(w_1, s\omega_1^e), (w_2, s\omega_2^e), \ldots, (w_n, s\omega_n^e)\}$. Then by the generation of $SI(\Gamma)$ and by construction of $\mathcal{D}$,

$$RC(\alpha, b) = RC(\alpha, e), \tag{34}$$

$$P_{NB}(\alpha, \varsigma, b) = \Pi(e, \alpha, \varsigma) \text{ and} \tag{35}$$

$$b'(w_k) = s\omega_k^{e'} = BT(e, k, \alpha, \varsigma), \text{ where } b' = BU(\alpha, \varsigma, b). \tag{36}$$

Hence, due to equalities (34), (35) and (36),

$$\begin{aligned} & RC(\alpha_1, e_{z,\text{-}}) + \sum_{\varsigma_i \in \Omega^{\#}} \Pi(e_{z,\text{-}}, \alpha_1, \varsigma_i) U(\Lambda, e_{z+1,i}) \\ & = RC(\alpha_1, e_{z,\text{-}}) + \sum_{\substack{\varsigma_i \in \Omega^{\#} \\ U(\Lambda, e_{z+1,i}) = r}} \Pi(e_{z,\text{-}}, \alpha_1, \varsigma_i) \cdot r \\ & = RC(\alpha_1, b^z) + \sum_{\substack{\varsigma \in \Omega \\ b' = BU(\alpha_1, \varsigma, b) \\ \mathcal{D}b'w \models \mathbf{U}\Lambda = r}} P_{NB}(\alpha_1, \varsigma, b^z) \cdot r \\ & \bowtie q, \end{aligned}$$

where $b^z = \{(w_1, s\omega_1^{e_z}), (w_2, s\omega_2^{e_z}), \ldots, (w_n, s\omega_n^{e_z})\}$. Therefore,

$$\mathcal{D}b^z w \models \mathbf{U}\{\alpha_1\}\Lambda \bowtie q \text{ (for all } w).$$

- Ψ is $\varphi \Rightarrow c = c'$. Because $(\Sigma, \bot) \notin \Gamma$ for some Σ, rule $=$ was not applied. Hence, c is identical to c', and $\mathcal{D}bw \models c = c'$ (for all b and w). Therefore, $\mathcal{D}bw \models \varphi \Rightarrow c = c'$ (for all b and w).

- Ψ is $\varphi \Rightarrow \neg(c = c')$. Because $(\Sigma, \bot) \notin \Gamma$ for some Σ, rule $=$ was not applied. Hence, c is not identical to c', and $\mathcal{S}, w \models \neg(c = c')$ (for all b and w). Therefore, $\mathcal{D}bw \models \varphi \Rightarrow \neg(c = c')$ (for all b and w).

- Ψ is $\varphi \Rightarrow \mathit{Reward}(r)$. By construction, $(w, r) \in \mathit{Re}$ where $w \models \varphi$. Hence, whenever $\mathcal{D}bw \models \varphi$, $\mathcal{D}bw \models \mathit{Reward}(r)$ (for all b). That is, $\mathcal{D}bw' \models \varphi \Rightarrow \mathit{Reward}(r)$ (for all b and w').

- Ψ is $\varphi \Rightarrow \neg\mathit{Reward}(r)$. By construction, $(w, r) \notin \mathit{Re}$ where $w \models \varphi$. Hence, whenever $\mathcal{D}bw \models \varphi$, $\mathcal{D}bw \models \neg\mathit{Reward}(r)$ (for every b). That is, $\mathcal{D}bw' \models \varphi \Rightarrow \neg\mathit{Reward}(r)$ (for all b and w').

- Ψ is $\varphi \Rightarrow \mathit{Cost}(\alpha, r)$. By construction, $(w, r) \in Co_\alpha$ where $w \models \varphi$. Hence, whenever $\mathcal{D}bw \models \varphi$, $\mathcal{D}bw \models \mathit{Cost}(\alpha, r)$ (for all b). That is, $\mathcal{D}bw' \models \varphi \Rightarrow \mathit{Cost}(\alpha, r)$ (for all b and w').

- Ψ is $\varphi \Rightarrow \neg \mathit{Cost}(\alpha, r)$. By construction, $(w, r) \notin Co_\alpha$ where $w \models \varphi$.Hence, whenever $\mathcal{D}bw \models \varphi$, $\mathcal{D}bw \models \neg \mathit{Cost}(\alpha, r)$ (for all b). That is, $\mathcal{D}bw' \models \varphi \Rightarrow \neg \mathit{Cost}(\alpha, r)$ (for all b and w').

- Ψ is $\phi \Rightarrow [\alpha]\varphi \bowtie q$. By the generation of $SI(\Gamma)$,

$$c_1 s^\alpha_{j,1} + c_2 s^\alpha_{j,2} + \cdots + c_n s^\alpha_{j,n} \bowtie q,$$

  where $w_j \models \phi$, and $c_k = 1$ if $w_k \models \varphi$, else $c_k = 0$, and the $s^\alpha_{j,k}$ are in $sln \in Z(\Gamma)$. Then as a direct consequence of the construction of $\mathcal{D}$, whenever $\mathcal{D}bw_j \models \phi$, $\mathcal{D}bw_j \models [\alpha]\varphi \bowtie q$ (for all b and $w_j \in C$). That is, $\mathcal{D}bw' \models \phi \Rightarrow [\alpha]\varphi \bowtie q$ (for all b and w').

- Ψ is $\phi \Rightarrow \neg[\alpha]\varphi \bowtie q$. By the generation of $SI(\Gamma)$,

$$c_1 s^\alpha_{j,1} + c_2 s^\alpha_{j,2} + \cdots + c_n s^\alpha_{j,n} \not\bowtie q,$$

  where $w_j \models \phi$, and $c_k = 1$ if $w_k \models \varphi$, else $c_k = 0$, and the $s^\alpha_{j,k}$ are in $sln \in Z(\Gamma)$. Then as a direct consequence of the construction of $\mathcal{D}$, whenever $\mathcal{D}bw_j \models \phi$, $\mathcal{D}bw_j \models \neg[\alpha]\varphi \bowtie q$ (for all b and $w_j \in C$). That is, $\mathcal{D}bw' \models \phi \Rightarrow \neg[\alpha]\varphi \bowtie q$ (for all b and w').

- Ψ is $\phi \Rightarrow (\varsigma|\alpha) \bowtie q$. By the generation of $SI(\Gamma)$,

$$s^{\varsigma|\alpha}_j \bowtie q,$$

  where $w_j \models \phi$ and $s^{\varsigma|\alpha}_j$ is in $sln \in Z(\Gamma)$. Then as a direct consequence of the construction of $\mathcal{D}$, whenever $\mathcal{D}bw_j \models \phi$, $\mathcal{D}bw_j \models (\varsigma|\alpha) \bowtie q$ (for all b and $w_j \in C$). That is, $\mathcal{D}bw' \models \phi \Rightarrow (\varsigma|\alpha) \bowtie q$ (for all b and w').

- Ψ is $\phi \Rightarrow \neg(\varsigma|\alpha) \bowtie q$. By the generation of $SI(\Gamma)$,

$$s^{\varsigma|\alpha}_j \not\bowtie q,$$

  where $w_j \models \phi$ and $s^{\varsigma|\alpha}_j$ is in $sln \in Z(\Gamma)$. Then as a direct consequence of the construction of $\mathcal{D}$, whenever $\mathcal{D}bw_j \models \phi$, $\mathcal{D}bw_j \models \neg(\varsigma|\alpha) \bowtie q$ (for all b and $w_j \in C$). That is, $\mathcal{D}bw' \models \phi \Rightarrow \neg(\varsigma|\alpha) \bowtie q$ (for all b and w').

Induction step:

- Ψ contains a double negation $\neg\neg$. By rule $\neg$, $(\Sigma, \Psi') \in \Gamma$, where Ψ' is Ψ with the $\neg\neg$ removed. By induction hypothesis, $\mathcal{D}b'w' \models \Psi'$ for some b' and w'. By the definition of $\neg$, $\mathcal{D}b'w' \models \Psi$.

- Ψ is $\psi \wedge \psi'$. By rule $\wedge$, $(\Sigma, \psi), (\Sigma, \psi') \in \Gamma$. By induction hypothesis, $\mathcal{D}b'w' \models \psi$ and $\mathcal{D}b'w' \models \psi'$ for some b' and w'. By the definition of $\wedge$, $\mathcal{D}b'w' \models \psi \wedge \psi'$.

- Ψ is $\neg(\psi \wedge \psi')$. By rule $\vee$, $(\Sigma, \neg\psi) \in \Gamma$ or $(\Sigma, \neg\psi') \in \Gamma$. By induction hypothesis, $\mathcal{D}b'w' \models \neg\psi$ or $\mathcal{D}b'w' \models \neg\psi'$. By the definition of $\vee$, $\mathcal{D}b'w' \models \neg(\psi \wedge \psi')$.

- Ψ is $\psi \vee \psi'$. By rule $\vee$, $(\Sigma, \neg\psi) \in \Gamma$ or $(\Sigma, \neg\psi') \in \Gamma$. By induction hypothesis, $\mathcal{D}b'w' \models \neg\psi$ or $\mathcal{D}b'w' \models \neg\psi'$. By the definition of $\vee$, $\mathcal{D}b'w' \models \neg(\psi \wedge \psi')$.

- Ψ is $\neg\mathbf{B}\varphi \bowtie q$ (i.e., $(\Sigma, \neg\mathbf{B}\varphi \bowtie q) \in \Gamma$). By rule $\neg\mathbf{B}$, $(\Sigma, \mathbf{B}\varphi \not\bowtie q) \in \Gamma$.[5] By induction hypothesis, $\mathcal{D}b'w' \models \mathbf{B}\varphi \not\bowtie q$. Therefore, $\mathcal{D}b'w' \models \neg\mathbf{B}\varphi \bowtie q$.

- Ψ is $\neg\mathbf{U}\Lambda \bowtie q$ (i.e., $(\Sigma, \neg\mathbf{U}\Lambda \bowtie q) \in \Gamma$). (This case is similar to the case when Ψ is $\neg\mathbf{B}\varphi \bowtie q$.) By rule $\neg\mathbf{U}$, $(\Sigma, \mathbf{U}\Lambda \not\bowtie q) \in \Gamma$. By induction hypothesis, $\mathcal{D}b'w' \models \mathbf{U}\Lambda \not\bowtie q$. Therefore, $\mathcal{D}b'w' \models \neg\mathbf{U}\Lambda \bowtie q$.

- Ψ is $\varphi \Rightarrow \Psi'$, where Ψ' is not a literal. Due to the preprocessing step, Ψ' is in CNF. $\varphi \Rightarrow \Psi'$. can be in one of four forms. (i) φ is not definitive and Ψ' has the form $\psi \wedge \psi'$, (ii) φ is not definitive and Ψ' has the form $\psi \vee \psi'$, (iii) φ is definitive and Ψ' has the form $\psi \wedge \psi'$, (iv) φ is definitive and Ψ' has the form $\psi \vee \psi'$.

  (i) By rule $\Rightarrow \wedge$, $(\Sigma, \varphi \Rightarrow \psi \wedge \varphi \Rightarrow \psi') \in \Gamma$. By induction hypothesis, $\mathcal{D}b'w' \models \varphi \Rightarrow \psi \wedge \varphi \Rightarrow \psi'$, which logically implies $\mathcal{D}b'w' \models \varphi \Rightarrow \psi \wedge \psi'$, which implies $\mathcal{D}b'w' \models \varphi \Rightarrow \Psi$.

  (ii) By rule $\delta \Rightarrow$, $(\Sigma, (\delta_1 \Rightarrow \Phi') \wedge (\delta_2 \Rightarrow \Phi') \wedge \cdots \wedge (\delta_n \Rightarrow \Phi'))\}$, where $\delta_i \in \mathit{Def}(\varphi)$. By induction hypothesis, $\mathcal{D}b'w' \models (\delta_1 \Rightarrow \Phi') \wedge (\delta_2 \Rightarrow \Phi') \wedge \cdots \wedge (\delta_n \Rightarrow \Phi')$, which logically implies $\mathcal{D}b'w' \models \varphi \Rightarrow \Psi'$.

  (iii) By rule $\Rightarrow \wedge$, $(\Sigma, \varphi \Rightarrow \psi \wedge \varphi \Rightarrow \psi') \in \Gamma$. By induction hypothesis, $\mathcal{D}b'w' \models \varphi \Rightarrow \psi \wedge \varphi \Rightarrow \psi'$, which logically implies $\mathcal{D}b'w' \models \varphi \Rightarrow \psi \wedge \psi'$, which implies $\mathcal{D}b'w' \models \varphi \Rightarrow \Psi$.

  (iv) By rule $\Rightarrow \vee$, $(\Sigma, \varphi \Rightarrow \psi \vee \varphi \Rightarrow \psi') \in \Gamma$. By induction hypothesis, $\mathcal{D}b'w' \models \varphi \Rightarrow \psi \vee \varphi \Rightarrow \psi'$, which logically implies $\mathcal{D}b'w' \models \varphi \Rightarrow \psi \vee \psi'$, which implies $\mathcal{D}b'w' \models \varphi \Rightarrow \Psi$.

- Ψ is $\{\alpha + \varsigma\}\Psi'$ (i.e., $(\Sigma e, \{\alpha + \varsigma\}\Psi') \in \Gamma$). By rule Ξ, $(\Sigma e \xrightarrow{\alpha,\varsigma} e', \Psi') \in \Gamma$. By induction hypothesis, $\mathcal{D}b'w' \models \Psi'$. By the SI phase, the following are in $SI(\Gamma)$.

$$\omega_k^{e'} = BT(e, k, \alpha, \varsigma, n) \text{ (for } k = 1, 2, \ldots, n),$$

$$\Pi(e, \alpha, \varsigma, n) \neq 0$$

$$\omega_1^e + \omega_2^e + \cdots + \omega_n^e = 1$$

and

$$\omega_1^{e'} + \omega_2^{e'} + \cdots + \omega_n^{e'} = 1.$$

Hence,

$$s\omega_k^{e'} = \frac{s_k^{\varsigma|\alpha} \sum_{i=1}^n s_{i,k}^\alpha s\omega_i^e}{\sum_{j=1}^n s_j^{\varsigma|\alpha} \sum_{i=1}^n s_{i,j}^\alpha s\omega_i^e} \text{ (for } k = 1, 2, \ldots, n), \tag{37}$$

[5] For conciseness, we abuse notation with $\not\bowtie$. In the case of *bowtie* being $=$, Γ contains either $(\Sigma, \mathbf{B}\varphi < q)$ or $(\Sigma, \mathbf{B}\varphi > q)$. By induction hypothesis, $\mathcal{D}b'w' \models \mathbf{B}\varphi < q$, respectively, $\mathcal{D}b'w' \models \mathbf{B}\varphi < q$, which implied $\mathcal{D}b'w' \not\models \mathbf{B}\varphi = q$.

$$\sum_{j=1}^{n} s_j^{\varsigma|\alpha} \sum_{i=1}^{n} s_{i,j}^{\alpha} s\omega_i^e \neq 0 \tag{38}$$

$$s\omega_1^e + s\omega_2^e + \cdots + s\omega_n^e = 1$$

and

$$s\omega_1^{e'} + s\omega_2^{e'} + \cdots + s\omega_n^{e'} = 1.$$

Let $b = \{(w_1, s\omega_1^e), (w_2, s\omega_2^e), \ldots, (w_n, s\omega_n^e)\}$ and let $b' = \{(w_1, s\omega_1^{e'}), (w_2, s\omega_2^{e'}), \ldots, (w_n, s\omega_n^{e'})\}$. Then by the set of equations (37), it must be that $b' = BU(\alpha, \varsigma, b)$. Further, Equation (38) implies $P_{NB}(\alpha, \varsigma, b) \neq 0$. Therefore,

$$P_{NB}(\alpha, \varsigma, b) \neq 0 \text{ and } \mathcal{D}b'w' \models \Psi, \text{ where } b' = BU(\alpha, \varsigma, b)$$

iff

$$\mathcal{D}bw \models \{\alpha + \varsigma\}\Psi \text{ (for all } w).$$

- Ψ is $\neg\{\alpha + \varsigma\}\Psi'$. By rule $\neg\Xi$, $(\Sigma, \neg Poss(\alpha, \varsigma) \vee \{\alpha + \varsigma\}\neg\Psi') \in \Gamma$. By induction hypothesis, $\mathcal{D}b'w' \models \neg Poss(\alpha, \varsigma) \vee \{\alpha + \varsigma\}\neg\Psi'$, that is, $\mathcal{D}b'w' \models \neg Poss(\alpha, \varsigma)$ or $\mathcal{D}b'w' \models \{\alpha + \varsigma\}\neg\Psi'$. Next, we show that both these cases lead to $\mathcal{D}b'w' \models \neg\{\alpha + \varsigma\}\Psi'$.

  By definition,

  $$\mathcal{D}bw \models \{\alpha + \varsigma\}\Psi \iff P_{NB}(\alpha, \varsigma, b) \neq 0 \text{ and } \mathcal{D}b''w \models \Psi, \text{ where } b'' = BU(\alpha, \varsigma, b).$$

  Hence, if $\mathcal{D}b'w' \models \neg Poss(\alpha, \varsigma)$, then $P_{NB}(\alpha, \varsigma, b') = 0$, implying $\mathcal{D}b'w' \not\models \{\alpha + \varsigma\}\Psi'$. And $\mathcal{D}b'w' \models \{\alpha + \varsigma\}\neg\Psi'$ implies $P_{NB}(\alpha, \varsigma, b') \neq 0$ and $\mathcal{D}b''w' \models \neg\Psi'$, where $b'' = BU(\alpha, \varsigma, b')$, which implies $\mathcal{D}b'w' \not\models \{\alpha + \varsigma\}\Psi'$.

■

Corollary 5.1 *By Lemma 5.3, given the leaf node Γ of an open branch of a finished tree, there exists a structure $\mathcal{D}$, belief-state b and world w such that for all $(\Sigma, \Psi) \in \Gamma$, $\mathcal{D}bw \models \Psi$. But $(0, \psi) \in \Gamma$. Thus, if there is a finished open tableau for ψ, then ψ is satisfiable.*

Theorem 5.2 follows directly from Corollary 5.1.

5.3 Termination of the Decision Procedure

Definition 5.2 *Let Ψ' be a strict sub-part of Ψ and let $(\Sigma, \Psi) \in \Gamma$. A tableau rule has the* subformula property *if and only if the new node(s) Γ' created by the application of the rule, contains (Σ', Ψ') or $(\Sigma', \neg\Psi')$ for some Σ', where $(\Sigma', \Psi') \notin \Gamma$, respectively, $(\Sigma', \neg\Psi') \notin \Gamma$.*

Lemma 5.4 *A formula of the form $(\Sigma, \varphi \Rightarrow \Phi)$ can cause only a finite number of tableau rule applications.*

Proof:
Recall that due to a preprocessing step, Φ is in CNF. Let $(\Sigma, \varphi \Rightarrow \Phi_1 \wedge \Phi_2 \wedge \cdots \wedge \Phi_m)$ be in some node Γ, where Φ_i for $i = 1, 2, \ldots, m$ is a disjunction of literals (i.e., $\Phi_1 \wedge \Phi_2 \wedge \cdots \wedge \Phi_m$ is in CNF). After successive applications of rule $\Rightarrow \wedge$, $(\Sigma, \varphi \Rightarrow \Phi_1), (\Sigma, \varphi \Rightarrow \Phi_2), \ldots, (\Sigma, \varphi \Rightarrow \Phi_m) \in \Gamma'$, where Γ' is a descendant of Γ. After successive applications of rule $\delta \Rightarrow$,

$$
\begin{gathered}
(\Sigma, \delta_1 \Rightarrow \Phi_1), (\Sigma, \delta_2 \Rightarrow \Phi_1), \ldots, (\Sigma, \delta_n \Rightarrow \Phi_1), \\
(\Sigma, \delta_1 \Rightarrow \Phi_2), (\Sigma, \delta_2 \Rightarrow \Phi_2), \ldots, (\Sigma, \delta_n \Rightarrow \Phi_2), \\
\vdots \\
(\Sigma, \delta_1 \Rightarrow \Phi_m), (\Sigma, \delta_2 \Rightarrow \Phi_m), \ldots, (\Sigma, \delta_n \Rightarrow \Phi_m) \in \Gamma'',
\end{gathered}
$$

where Γ'' is a descendant of Γ' and $\delta_i \in \mathit{Def}(\varphi)$. If Φ_j for some $j \in \{1, 2, \ldots, m\}$ is a literal, then no tableau rule is applicable to $(\Sigma, \delta_1 \Rightarrow \Phi_j), (\Sigma, \delta_2 \Rightarrow \Phi_j)$ nor $(\Sigma, \delta_n \Rightarrow \Phi_j)$. Else, (for all Φ_j which are not literals, $j \in \{1, 2, \ldots, m\}$), after successive applications of rule $\Rightarrow \vee$ and rule $\vee$, new decendent nodes are created containing formulae of the form $(\Sigma, \delta \Rightarrow \Phi)$, where δ is definitive and Φ is a literal. No tableau rule is applicable to formulae of the form $(\Sigma, \delta \Rightarrow \Phi)$. Clearly, no matter the order in which rules $\Rightarrow \wedge, \delta \Rightarrow, \Rightarrow \vee$ and $\vee$ are applied, the rules can be applied only a finite number of times due to $(\Sigma, \varphi \Rightarrow \Phi_1 \wedge \Phi_2 \wedge \cdots \wedge \Phi_m) \in \Gamma$. ■

Lemma 5.5 *A tree for any formula $\Psi \in \mathcal{L}_{SDL}$ becomes saturated. That is, the tableau phase terminates.*

Proof:
We can divide all the tableau rules into three categories: (i) those which add $(\Sigma, \bot)$ to the new node, (ii) those with the subformula property and (iii) those from the list $\langle \Rightarrow \wedge$, $\delta \Rightarrow$, $\Rightarrow \vee$, $\neg\Xi$, $\neg\mathbf{B}$, $\neg\mathbf{U}\rangle$. Category-(i) rules never cause rules to become applicable later. As a direct consequence of sentences being finite and their subformula property, every category-(ii) rule must eventually become inapplicable.

Rules $\neg\mathbf{B}$ and $\neg\mathbf{U}$ are applied only once to any formula and never to a formula added to the new node.

Rule $\neg\Xi$ adds a formula of the form $(\Sigma, \neg \mathit{Poss}(\alpha, \varsigma) \vee \{\alpha + \varsigma\}\neg\Psi)$ to the new node. After applying rule $\vee$, $(\Sigma, \neg \mathit{Poss}(\alpha, \varsigma))$ causes no rule application and $(\Sigma, \{\alpha + \varsigma\}\neg\Psi)$ is a category-(ii) rule.

Rules $\Rightarrow \wedge$, $\delta \Rightarrow$ and $\Rightarrow \vee$ are only applied to formulae of the form $(\Sigma, \varphi \Rightarrow \Phi)$. By Lemma 5.4, there will be a finite number of rule applications to formulae of the form $(\Sigma, \varphi \Rightarrow \Phi)$.

Therefore, all rules eventually become inapplicable, and it follows that any tree (for any formula) would become saturated. ■

Theorem 5.3 *The decision procedure for SDL terminates.*

Proof:
Due to Lemma 5.5, the tableau phase terminates (with a finite number of branches).

In the SI phase: for each open branch of a tree for some $\Psi \in \mathcal{L}_{SDL}$, the feasibility of an SI is sought once for Γ, where Γ is the leaf node of the branch. Hence, the feasibility of an SI is sought a finite number of times in the SI phase.

By Lemma 4.1, determining the feasibility of an SI terminates and the SI phase thus terminates. ■

Corollary 5.2 *The validity problem for the SDL is decidable.*

Proof:
Because the procedure is sound and complete, it will be decidable if it always terminates, which, by Theorem 5.3, it does. ■

6 Related Work

Amato et al. [2007] solve POMDPs by setting up quadratically constrained linear programs (QCLPs) and then solving these. Their work is similar to our in that they also appeal to a class of nonlinear systems (QCLPs). Their work is different to ours in that they seek to optimize planning in a non-logical setting.

De Weerdt et al. [1999] present a modal logic to deal with imprecision in robot actions and sensors. Their models do not contain an accessibility relation, which makes it hard to understand what it means for an action to be executed. They cannot deal with utilities of actions. No proof system for determining truth of statements in their language is provided.

Bacchus et al. [1999] supply a theory for reasoning with noisy sensors and effectors, with graded belief. They use the situation calculus [McCarthy, 1963] to specify their approach but some elements fall outside the logical language. They don't address utilities of actions. $\mathcal{ESP}$ [Gabaldon and Lakemeyer, 2007] is a construction of Bacchus *et al.*'s approach with the following major differences. (i) Every notion of interest is expressible within the defined language; no second-order logic is needed. (ii) An agent's epistemic state includes sets of probability distributions over both the initial situations and the outcome of stochastic actions after any number of actions. (iii) A probabilistic version of *only-knowing* is defined. (iv) $\mathcal{ESP}$ is founded on $\mathcal{ES}$ [Levesque and Lakemeyer, 2004], which is a fragment of the situation calculus. The semantics of SDL is arguably much simper than that of $\mathcal{ESP}$. To our knowledge, there exists no decidable version or fragment of the situation calculus which can model and reason about POMDPs.

Poole [1998] says this about the Independent Choice Logic using the situation calculus (ICL_{SC}): "The representation in this paper can be seen as a representation for POMDPs". Belief-states can be expressed and belief update can be performed (but maintenance of belief-states is not a necessary component of the system). The ICL_{SC} is a relatively rich framework, with acyclic logic programs which may contain variables, quantification and function symbols. For certain applications, the SDL may be preferred due to its comparative simplicity. And because the SDL's semantics is very close to that of standard POMDP theory, it may be easier to understand by people familiar with POMDPs. Finally, decidability of inferences made in the ICL_{SC} are, in general, not guaranteed.

Iocchi et al. [2009] present a logic called $\mathcal{E}+$ for reasoning about agents with sensing, qualitative nondeterminism and probabilistic uncertainty in action outcomes. Planning with sensing and uncertain actions is also dealt with. Noisy sensing is not dealt with, that is, sensing actions are deterministic. They mention that although they would like to be able to represent action rewards and costs as in POMDPs, $\mathcal{E}+$ does not yet provide the facilities.

7 Concluding Remarks

One advantage of having a logic for specifying POMDPs is that it can be done quite compactly. However, the SDL is not unique in this regard. Although several frameworks have been proposed to deal with stochastic actions, noisy sensing, degree of belief and/or expected future rewards, not one of them can specify and reason about all these notions in a unified decidable logic. The Stochastic Decision Logic (SDL) can deal with all these notions to some degree. In particular, the SDL is for specifying and reasoning about partially observable Markov decision processes (POMDPs).

We showed how POMDP models can be translated into a set of SDL sentences and we also discussed the model-theoretic correspondence between POMDPs and the SDL.

There is no notion of logical entailment in POMDP theory. Logical entailment can be applied to SDL sentences and hence to the POMDP models they represent. *A major contribution of this work is that it allows the user to determine whether or not a set of sentences is entailed by an arbitrarily precise specification of a POMDP model.* As far as we know, this is a novel property of the SDL. Moreover, the procedure for deciding entailment is proved sound, complete and terminating. As a corollary, the entailment question for the SDL is decidable.

Automatic plan generation is highly desirable in cognitive robotics and for autonomous systems modeled as POMDPs. In future work, we would like to take the SDL as the basis for developing a language or framework with which plans can be generated, in the fashion of DTGolog [Boutilier et al., 2000].

References

K. Aström. Optimal control of Markov decision processes with incomplete state estimation. *J. Math. Anal. Appl.*, 10:174–205, 1965.

R. Smallwood and E. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21:1071–1088, 1973.

G. E. Monahan. A survey of partially observable Markov decision processes: Theory, models, and algorithms. *Management Science*, 28(1):1–16, 1982.

W. Lovejoy. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research*, 28:47–66, 1991.

C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *J. Artif. Intell. Res. (JAIR)*, 11:1–94, 1999.

H. Geffner and B. Bonet. High-level planning and control with incomplete information using POMDPs. In *Proc. of Fall AAAI Symposium on Cognitive Robotics*, pages 113–120, Seattle, WA, 1998a. AAAI Press.

C. Wang and J. Schmolze. Planning with POMDPs using a compact, logic-based representation. In *Proc. of 17th IEEE Intl. Conf. on Tools with Artif. Intell. (ICTAI'05)*, pages 523–530, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

S. Sanner and K. Kersting. Symbolic dynamic programming for first-order POMDPs. In *Proc. of 24th Natl. Conf. on Artificial Intelligence (AAAI-10)*, pages 1140–1146. AAAI Press, 2010.

G. Rens, T. Meyer, and G. Lakemeyer. SLAP: Specification logic of actions with probability. *Journal of Applied Logic*, 2013. ISSN 1570-8683. doi: http://dx.doi.org/10.1016/j.jal.2013.09.001. URL `www.sciencedirect.com/science/article/pii/S157086831300075X`. url = http://dx.doi.org/10.1016/j.jal.2013.09.001.

G. Rens, T. Meyer, and G. Lakemeyer. A logic for specifying stochastic actions and observations. In C. Beierle and C. Meghini, editors, *Proc. of 8th Intl. Symposium on Foundations of Information and Knowledge Systems (FoIKS)*, Lecture Notes in Computer Science. Springer-Verlag, 2014. (to appear) url: www.cair.za.net/sites/default/files/outputs/SLAOP-at-FoIKS.pdf.

A. Cassandra, L. Kaelbling, and M. Littman. Acting optimally in partially observable stochastic domains. Technical Report CS-94-20, Brown Universiteit Leuven, Department of Computer Science, Providence, Rhode Island 02912, 1994.

L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998.

S. J. Russell and P. Norvig. *Artificial intelligence: A modern approach*. Prentice Hall, New Jersey, 2nd edition, 2003.

H. Geffner and B. Bonet. Solving large POMDPs using real time dynamic programming. In *In Working Notes of AAAI Fall Symposium on POMDPs*, 1998b.

T. Smith and R. Simmons. Point-based POMDP algorithms: Improved analysis and implementation. In *Proc. of 21st Conf. on Uncertainty in Artificial Intelligence (UAI-05)*, pages 542–549. AUAI Press, 2005.

Y. Virin, G. Shani, S. Shimony, and R. Brafman. Scaling up: Solving pomdps through value based clustering. In *Proc. of 22nd AAAI Conference on Artificial Intelligence*, pages 1290–1295, Menlo Park, CA, July 2007. AAAI Press.

J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *Proc. IJCAI*, pages 1025–1032, 2003.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Massachusetts/England, 2005.

A. Tarski. A decision method for elementary algebra and geometry. Technical report, The RAND Corporation, Santa Monica, Calif., 1957.

C. Amato, D. Bernstein, and S. Zilberstein. Solving POMDPs using quadratically constrained linear programs. In Ramon Lopez de Mantaras, editor, *Proc. of 20th Intl. Joint Conf. on Artif. Intell. (IJCAI-07)*, pages 2418–2424, Menlo Park, CA, January 2007. AAAI Press.

M. De Weerdt, F. De Boer, W. Van der Hoek, and J.-J. Meyer. Imprecise observations of mobile robots specified by a modal logic. In *Proc. of Fifth annual conference of the Advanced School for Computing and Imaging (ASCI-99)*, pages 184–190, 1999.

F. Bacchus, J. Y. Halpern, and H. J. Levesque. Reasoning about noisy sensors and effectors in the situation calculus. *Artificial Intelligence*, 111(1–2):171–208, 1999.

J. McCarthy. Situations, actions and causal laws. Technical report, Stanford University, 1963.

A. Gabaldon and G. Lakemeyer. $\mathcal{ESP}$: A logic of only-knowing, noisy sensing and acting. In *Proc. of 22nd Natl. Conf. on Artificial Intelligence (AAAI-07)*, pages 974–979. AAAI Press, 2007.

H. J. Levesque and G. Lakemeyer. Situations, si! Situation terms no! In *Proc. Principles of Knowledge Representation and Reasoning (KR-04)*, pages 516–526. AAAI Press, 2004.

D. Poole. Decision theory, the situation calculus and conditional plans. *Linköping Electronic Articles in Computer and Information Science*, 8(3), 1998.

L. Iocchi, T. Lukasiewicz, D. Nardi, and R. Rosati. Reasoning about actions with sensing under qualitative and probabilistic uncertainty. *ACM Transactions on Computational Logic*, 10(1):5:1–5:41, 2009.

C. Boutilier, R. Reiter, M. Soutchanski, and S. Thrun. Decision-theoretic, high-level agent programming in the situation calculus. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00) and of the 12th Conference on Innovative Applications of Artificial Intelligence (IAAI-00)*, pages 355–362. AAAI Press, Menlo Park, CA, 2000. ISBN 0-262-51112-6.