Validating Domains and Plans for Temporal Planning via Encoding into Infinite-State Linear Temporal Logic: Additional Material

Omitted for blind review

Proofs

In this section we prove all the theorems appearing in the paper titled "Validating Domains and Plans for Temporal Planning via Encoding into Infinite-State Linear Temporal Logic".

Proof of Theorem 1

Definition 1. *Given a planning problem $\mathcal{P}$ and a plan π the trace of π (written σ^π) is a trace on the variables $V^\mathcal{P}$ defined as follows.*

Let $\chi : \mathbb{N} \to \mathbb{R}^+$ be the function ordering the times of the event in the execution of π:

$$\chi(i) \doteq \begin{cases} 0 & \textit{if } i = 0 \\ min(S \cup \{t, d \mid \langle t, a, d\rangle \in \pi\}) & \textit{if } S \neq \emptyset \\ \chi(i-1) & \textit{otherwise} \end{cases}$$

with

$$S \doteq \{t \mid \langle t, f, v, t_0, d_0\rangle \in CH(\mathcal{P}, \pi), t > \chi(i-1)\}$$

For each $i \in \mathbb{N}$ and each $f \in F$, we define $\sigma^\pi(i)(\overline{f}) \doteq \tau^\pi_f(\chi(i))$. For each $i \in \mathbb{N}$ and each $a \in A$, we define:

- $\sigma^\pi(i)(\overline{a}) \doteq \top$ *iff* $\chi(i) \in \bigcup_{\langle t,a,d\rangle \in \pi} [t, t+d]$*;*
- $\sigma^\pi(i)(s_a) \doteq min(\{t \mid \chi(i) \leq t, \langle t, a, d\rangle \in \pi\})$*;*
- $\sigma^\pi(i)(\delta_a) \doteq d^a_i$ *with* $d^a_i \doteq d$ *s.t.* $\langle \sigma^\pi(i)(s_a), a, d\rangle \in \pi$.

Finally, we define $\sigma^\pi(i)(t) \doteq \chi(i)$ and
$\sigma^\pi(i)(\omega) \doteq max(\{\chi(i) \mid i \in \mathbb{N}\} \cup \{[\![e_2]\!]^{\tau^\pi}_{0,ms_\pi}(0) \mid \langle [e_1, e_2] \, e_3\rangle \in G\}) + 1$.

Theorem 1. *Let $\mathcal{P}$ be a planning problem admitting a solution, then $\epsilon^d_\mathcal{P}$ is satisfiable.*

Proof. Let π be a solution to $\mathcal{P} \doteq \langle F, T, A, G\rangle$. We show that $\sigma^\pi, 0 \models \epsilon^d_\mathcal{P}$ by showing that each constraint of $\epsilon^d_\mathcal{P}$ is satisfied.

1. $t = 0$ is satisfied, because $\chi(0)$ is 0
2. $\mathbf{G}\ (\omega = \omega')$ is satisfied because ω is assigned to a constant in all the states.
3. $\mathbf{G}\ ((t < \omega) \to (t < t'))$ holds because χ is monotonically increasing while its value is lower than $\sigma^\pi(0)(\omega)$.
4. $\mathbf{G}\ ((t \geq \omega) \to ((\bigwedge_{f \in V^\mathcal{P}} \overline{f} = \overline{f}') \wedge (\bigwedge_{a \in A} \neg \overline{a}) \wedge (t = t'))$holds because the chronicle τ^π cannot possibly change any fluent value after ω and no actions are executed after ω, hence all the $\overline{a}$ are set to $\bot$.
5. $\bigwedge_{a \in A} \mathbf{G}\ (\overline{a} \to (\delta'_a = \delta_a \wedge s'_a = s_a))$ holds under the assumption that π contains no pair of self-concurrent actions. This is the only assumption in the encoding. The constraint holds because the values of s_a and of δ_a do not change in the steps in which a is executing.
6. $\bigwedge_{a \in A} \mathbf{G}\ ((\neg\overline{a} \wedge \mathbf{X}\, \overline{a}) \to \mathbf{X}\, (\overline{a}\, \mathbf{U}\, (\mathbf{X}\, \neg\overline{a} \wedge t = \delta_a + s_a)))$ holds because:
   - $\overline{a}$ is true during all the time points in which a is executing
   - Eventually, there will be a step with $t = \delta_a + s_a$ because of the definition of χ.
7. $\mathbf{G}\ ((\neg\overline{a} \wedge \mathbf{X}\, \overline{a}) \to \mathbf{X}\, (\overline{a}\, \mathbf{U}\, (t = (\!|e_1|\!)_a \wedge \overline{f} = (\!|e_2|\!)_a)))$ holds because there is a step for each effect due to the definition of χ and because τ^π is the induced chronicle of a valid plan, hence it respects all the effects.
8. Analogously, $\mathbf{G}\ (((\mathbf{X}\, \overline{a} \wedge t < (\!|e_1|\!)'_a \wedge \mathbf{X}\, (t \geq (\!|e_1|\!)_a)) \to (\!|e_3|\!)_a) \wedge \quad ((\overline{a} \wedge t \geq (\!|e_1|\!)_a \wedge t < (\!|e_2|\!)_a) \to (\!|e_3|\!)_a))$ and $\mathbf{G}\ (((\mathbf{X}\ \overline{a} \wedge t \leq (\!|e_1|\!)'_a \wedge \mathbf{X}\ (t > (\!|e_1|\!)_a)) \to (\!|e_3|\!)_a) \wedge \quad ((\overline{a} \wedge t > (\!|e_1|\!)_a \wedge t < (\!|e_2|\!)_a) \to (\!|e_3|\!)_a))$ hold because τ^π is the induced chronicle of a valid plan, hence all condition of each action is satisfied.

$\square$

Proof of Theorem 2

Theorem 2. *$\epsilon^d_\mathcal{P} \wedge \epsilon^p_\mathcal{P} \wedge \phi^p_\mathcal{P}$ is satisfiable if and only if the planning problem $\mathcal{P}$ admits a solution.*

Proof. We first prove that if $\mathcal{P}$ admits a solution then $\epsilon^d_\mathcal{P} \wedge \epsilon^p_\mathcal{P} \wedge \phi^p_\mathcal{P}$ is satisfiable by showing that $\sigma^\pi, 0 \models \epsilon^d_\mathcal{P} \wedge \epsilon^p_\mathcal{P} \wedge \phi^p_\mathcal{P}$.

Due to the proof of theorem 1, we know that $\sigma^\pi, 0 \models \epsilon^d_\mathcal{P}$.
$\sigma^\pi, 0 \models \epsilon^p_\mathcal{P}$ because:

1. $\tau_\mathcal{P}$ holds because ω is greater than the timing of any change in $CH(\mathcal{P}, \pi)$ that comprises all the TILs and because σ^π has a step at the time of each change respecting the change itself.
2. $\bigwedge_{[e_1,e_2] e_3 \in G} \omega \geq (\!|e_2|\!)_\emptyset$ holds because ω is greater than the timing of any goal condition.
3. $\mathbf{G}(((t < (\!|e_1|\!)_\emptyset \wedge \mathbf{X}(t \geq (\!|e_1|\!)_\emptyset)) \to (\!|e_3|\!)_\emptyset) \wedge ((t \geq (\!|e_1|\!)_\emptyset \wedge t < (\!|e_2|\!)_\emptyset) \to (\!|e_3|\!)_\emptyset) \wedge ((t = (\!|e_1|\!)_\emptyset \wedge t = 0) \to (\!|e_3|\!)_\emptyset))$ for each goal $\langle e_1, e_2] \, e_3\rangle$ holds because π is a valid plan for $\mathcal{P}$ and σ^π is constructed using τ^π.
4. $\bigwedge_{(e_1,e_2] e_3 \in G} \mathbf{G}\ (((t \leq (\!|e_1|\!)_\emptyset \wedge \mathbf{X}\ (t > (\!|e_1|\!)_\emptyset)) \to (\!|e_3|\!)_\emptyset) \wedge ((t > (\!|e_1|\!)_\emptyset \wedge t < (\!|e_2|\!)_\emptyset) \to (\!|e_3|\!)_\emptyset))$ for each

goal $\langle [e_1, e_2]\ e_3\rangle$ holds because π is a valid plan for $\mathcal{P}$ and σ^π is constructed using τ^π.

Finally, $\sigma^\pi, 0 \models \phi^p_{\mathcal{P}}$ because π is a valid plan and then no two changes in $CH(\mathcal{P}, \pi)$ can happen at the same time in its execution. Moreover, due to the ANML semantics, fluents do not change value without a change causing the change.

We now prove that if $\sigma, 0 \models \epsilon^d_{\mathcal{P}} \wedge \epsilon^p_{\mathcal{P}} \wedge \phi^p_{\mathcal{P}}$, then $\mathcal{P}$ admits a solution.

Let $\pi^\sigma \doteq \{\langle \sigma(i)(s_a), a, \sigma(i)(\delta_a)\rangle | \sigma(i)(\overline{a}) = \top, a \in A, i \in \mathbb{N}\}$, we show that π^σ is a valid plan for $\mathcal{P}$.

For the sake of contradiction, let us assume that π^σ is invalid, then either:

1. there exists $t \in \mathbb{R}$ and $f \in F$ s.t. $|\langle [\![v]\!]^{\tau^{\pi^\sigma}}_{t_0,d_0}(t_0) \mid \langle t, f, v, t_0, d_0\rangle \in CH(\mathcal{P}, \pi)\rangle| > 1$. This is impossible due to the constraints in $\phi^p_{\mathcal{P}}$ that must be respected by σ.
2. There exists $\langle t, a, d\rangle \in \pi$ with $a \doteq \langle C, E\rangle$, and there exists $c \in C$ (with $c = \langle [\![e_1, e_2]\!]\ e_3\rangle$) and $x \in \Omega_{\tau^\pi}(c, t, d)$, s.t. $[\![e_3]\!]^{\tau^\pi}_{t,d}(x)$ does not hold. This is impossible due to constraints 8 and 9 of $\epsilon^d_{\mathcal{P}}$.
3. There exists goal condition $c = \langle [\![e_1, e_2]\!]\ e_3\rangle \in G$ and $x \in \Omega_{\tau^\pi}(c, 0, ms_\pi)$ s.t. $[\![e_3]\!]^{\tau^\pi}_{0,ms_\pi}(x)$ does not hold. This is impossible because $\sigma, 0 \models \gamma_{\mathcal{P}}$.

$\square$

Reducing PDDL 2.1 to ANML

It is possible to reduce any planning problem expressed in the temporal fragment of PDDL 2.1 into ANML. In this section we briefly describe how this reduction is done.

What makes the reduction non-trivial is that the ANML semantics does not impose a minimum time quantum ϵ separating an effects and conditions on the same variable. In fact, ANML only requires a positive amount of time to pass, while PDDL 2.1 requires that at least ϵ time passes. One might think that it suffices to move all the effects from time t to time $t + \epsilon$, but this does not allow for an effect or a condition at time $t + \epsilon$ that is permitted in PDDL.

In order to solve this issue, we exploit the possibility of predicating on time points: for each predicate p in PDDL we record the time of the last effect on p in a rational variable t_p, and we require that all other conditions and effects happening at time t respect the constraint $t - t_p \geq \epsilon$.

More precisely, the encoding works as follows. We assume a positive rational value ϵ is given, and we describe how to syntactically translate a grounded problem specification I expressed in PDDL 2.1 level 3 (the temporal fragment of PDDL 2.1) into an ANML problem instance $\mathcal{P} \doteq \langle F, T, A, G\rangle$. (We do not detail the syntax and semantics of PDDL 2.1 that is described in (Fox and Long 2003), but we assume a general understanding of the language at an informal level.)

- For each predicate p in in I, we create a Boolean fluent p and a rational fluent t_p in $\mathcal{P}$ (i.e. $p \in F_{\mathbb{B}}$ and $t_p \in F_{\mathbb{R}}$).
- For each predicate p assigned to $\top$ in the initial state, we add the following TIL to $\mathcal{P}$: $\langle [0]\, p := \top\rangle$. For all the predicates p unassigned in the initial state, we add $\langle [0]\, p := \bot\rangle$ to T.
- For each action a in I, we create an action $\langle C, E\rangle$ in $\mathcal{P}$ such that:
  - For each `at start` condition literal l in a (with $l = p$ or $l = \neg p$), we create a condition $\langle [\mathrm{START}]\, l \wedge \mathrm{START} - t_p \geq \epsilon\rangle$ in C.
  - For each `at end` condition literal l in a (with $l = p$ or $l = \neg p$), we create a condition $\langle [\mathrm{END}]\, l \wedge \mathrm{END} - t_p \geq \epsilon\rangle$ in C.
  - For each `over all` condition literal l in a (with $l = p$ or $l = \neg p$), we create a condition $\langle (\mathrm{START}, \mathrm{END})\ l \wedge \mathrm{START} - t_p \geq \epsilon \wedge \mathrm{END} - t_p \geq \epsilon\rangle$ in C.
  - For each `at start` effect positive literal p in a, we create two effects in E: $\langle [\mathrm{START}]\ p := \top\rangle$ and $\langle [\mathrm{START}]\ t_p := \mathrm{START}\rangle$. Moreover, we create a condition $\langle [\mathrm{START}]\ \mathrm{START} - t_p \geq \epsilon\rangle$ in C.
  - For each `at start` effect negative literal $\neg p$ in a, we create two effects in E: $\langle [\mathrm{START}]\ p := \bot\rangle$ and $\langle [\mathrm{START}]\ t_p := \mathrm{START}\rangle$. Moreover, we create a condition $\langle [\mathrm{START}]\ \mathrm{START} - t_p \geq \epsilon\rangle$ in C.
  - For each `at end` effect positive literal p in a, we create two effects in E: $\langle [\mathrm{END}]\ p := \top\rangle$ and $\langle [\mathrm{END}]\ t_p := \mathrm{START}\rangle$. Moreover, we create a condition $\langle [\mathrm{END}]\, \mathrm{END} - t_p \geq \epsilon\rangle$ in C.
  - For each `at end` effect negative literal $\neg p$ in a, we create two effects in E: $\langle [\mathrm{END}]\ p := \bot\rangle$ and $\langle [\mathrm{END}]\ t_p := \mathrm{START}\rangle$. Moreover, we create a condition $\langle [\mathrm{END}]\ \mathrm{END} - t_p \geq \epsilon\rangle$ in C.
- For each timed-initial-literal at time t over positive literal p in I, we add a TIL $\langle [t]\, p := \top\rangle$ to T and a goal condition $\langle [t]\ t - t_p \geq \epsilon\rangle$ in G.
- For each timed-initial-literal at time t over negative literal $\neg p$ in I, we add a TIL $\langle [t]\ p := \bot\rangle$ to T and a goal condition $\langle [t]\ t - t_p \geq \epsilon\rangle$ in G.
- For each goal literal l in I, we add a goal $\langle [\mathrm{END}]\, l \wedge \mathrm{END} - t_p \geq \epsilon\rangle$ in G.

In this way, we imposed an artificial ϵ-separation between two effects on the same fluent and between a condition and an effect on the same fluent. This separation is fully compatible with the PDDL 2.1 semantics. We can thus employ our encoding technique on any PDDL 2.1 specification (fully considering the formal semantics of PDDL) by rewriting it first into ANML. Finally, note that this rewriting is parametric in the ϵ value: given a PDDL specification and an ϵ it produces an ANML specification.