On the Expressive Power of Intermediate and Conditional Effects in Temporal Planning

Nicola Gigante[1] , **Andrea Micheli**[2] , **Enrico Scala**[3]
[1]Free University of Bozen-Bolzano, Italy
[2]Fondazione Bruno Kessler, Trento, Italy
[3]University of Brescia, Italy
nicola.gigante@unibz.it, amicheli@fbk.eu, enrico.scala@unibs.it

Abstract

Automated planning is the task of finding a sequence of *actions* that reach a desired goal, given a description of their applicability and their effects on the world. In *temporal* planning, actions have a duration and can overlap in time. In modern temporal planning formalisms, two features have been introduced which are very useful from a modeling perspective, but are not yet thoroughly understood: *intermediate conditions and effects* (ICE) and *conditional effects*. The expressive power of such constructs is yet not well comprehended, especially when time is dense, and no minimum separation is required between mutex events. This paper reveals that both ICE and conditional effects do *not* add expressive power with regards to common temporal planning formalisms. In particular, we show how they can be compiled away using a polynomial-size encoding that makes no assumptions on the time domain. This encoding advances our understanding of these features, and allow for their use with simple temporal planners that lack their support. Moreover, it provides a constructive proof showing that temporal planning with ICE and conditional effects remains PSPACE-complete.

1 Introduction

Automated planning is the task of finding a course of *actions* that, when executed starting from a known initial state, reaches a state satisfying a given goal condition. In *temporal planning*, actions have a non-zero duration, and can overlap in time. When multiple instances of the same action are allowed to overlap with themselves, temporal planning has been proved to be EXPSPACE-complete (Rintanen 2007) and *undecidable* (Gigante et al. 2020; Gigante et al. 2022), respectively on discrete and dense time domains, while it becomes PSPACE-complete, in both cases, without self-overlap of actions.

Different temporal planning formalisms have been introduced. PDDL 2.1 (Fox and Long 2003) extends the de-facto standard PDDL language with durative actions. The ANML language (Smith, Frank, and Cushing 2008), introduced in the context of space exploration, mixes features from action-based STRIPS-like formalisms such as PDDL with concepts taken from timeline-based planning languages (Frank and Jónsson 2003; Cesta et al. 2009; Gigante et al. 2017).

A few features of these formalisms exist that are very useful in practice but not yet well understood: *intermediate conditions and effects* (ICE), and *conditional effects*.

With ICE, an action can affect or be conditioned in time points different than the start and the end of its execution time interval. For example, an action might specify an effect to happen 5 time steps after its execution started, and at the same time require a given condition to hold in between 2 time steps after its starting and 5 time steps before its ending. This distinctive feature of the ANML language is not syntactically supported by PDDL 2.1 and, depending on some semantic assumptions about the underlying time domain, it is not clear if such a feature makes ANML more expressive.

With *conditional effects*, the effects of actions can be specified to apply or not depending on some state-dependent condition. For example, an action might be specified to set the truth of a proposition p at its end, only if a certain condition holds at its start. This feature, as formalized in PDDL 2.1, poses some challenges. Neither the polynomial nor the exponential encoding of classical conditional effects (Nebel 2000) work in the temporal case. In temporal planning the conditions under which a given effect is applied depend on the entire execution of the action (*e.g.*, an effect at the *start* of an action that depends on a condition evaluated *over all* the execution of the action), and this poses a significant challenge for the support of this feature. Conditional effects are available in PDDL 2.1, but are not supported by ANML; again, it is unclear whether PDDL 2.1 without such a feature is less expressive.

In this paper, we settle these questions, showing that both ICE and conditional effects *do not* add expressive power. We do so by providing a polynomial-size construction that encodes ICE and conditional effects into simple durative actions. This construction, which can be applied to both PDDL 2.1 and ANML problems, only assumes that self-overlap of actions is forbidden, and is independent from any assumption about the discrete or dense nature of the underlying time model, and from any assumption on the separation between mutex events (ϵ-*separation* or *non-zero separation* semantics). We are not aware of any compilations for conditional effects in temporal planning, and the only known compilation schemata for ICE only applies under ϵ-separation semantics (Fox, Long, and Halsey 2004; Smith 2003), which corresponds to assuming discrete time (Rintanen 2007).

Our construction employs *simultaneity constraints*, a

novel modeling technique that allows the endpoints of some actions, or whole actions, to always happen simultaneously. How to encode such constraints without particular assumptions on the time domain was not generally known before. Thanks to the terse encoding of these constraints, our compilation of ICE and conditional effects is quite lightweight, requiring only a handful of additional actions and fluents in the worst case.

Our results have a number of implications:

1. ICE and conditional effects do not increase the expressive power of temporal planning formalisms;
2. such features can be used with simple temporal planners that do not support them;
3. the homogeneous treatment of both features in terms of simultaneity constraints highlights a relationship between the two that was never noted before;
4. the temporal planning problem with both features, without self-overlap of actions, is still PSPACE-complete both on discrete and dense time.

The paper is structured as follows. After giving the necessary background information in Section 2, we introduce ICE and conditional effects in Section 3. Then, we introduce simultaneity constraints in Section 4, showing how to encode such constraints into temporal planning problems. Then, Section 5 uses simultaneity constraints to show how to encode ICE, while Section 6 turns its attention to conditional effects. Section 7 discusses related work, and Section 8 concludes with final remarks and mentions future work.

2 Background

Following Fox and Long (2003), this section introduces the temporal planning problem we are interested in. Given a set P of *propositions*, let $\mathbb{B}_P$ be the set of *Boolean formulas* over P.

Definition 1 (Temporal Planning Problem)**.** *A temporal planning problem is a tuple $\mathcal{P} = \langle P, A, I, G\rangle$, where P is a set of propositions, A is a set of durative actions, $I \subseteq P$ is the initial state, and $G \in \mathbb{B}_P$ is the goal condition. A snap (instantaneous) action is a tuple $h = \langle \mathrm{pre}(h), \mathrm{eff}^+(h), \mathrm{eff}^-(h)\rangle$, where $\mathrm{pre}(h) \in \mathbb{B}_P$ is the pre-condition and $\mathrm{eff}^+(h), \mathrm{eff}^-(h) \subseteq P$ are two disjoint sets of propositions, called the* positive *and* negative *effects of h, respectively. We write $\mathrm{eff}(h)$ for $\mathrm{eff}^+(h) \cup \mathrm{eff}^-(h)$. A durative action $a \in A$ is a tuple $\langle a_\vdash, a_\dashv, \mathrm{pre}^{\leftrightarrow}(a), [L_a, U_a]\rangle$, where $a_\vdash$ and $a_\dashv$ are the* start *and* end *snap actions, respectively, $\mathrm{pre}^{\leftrightarrow}(a) \in \mathbb{B}_P$ is the* over-all condition*, and $L_a \in \mathbb{Q}_{>0}$ and $U_a \in \mathbb{Q}_{\geq 0} \cup \{\infty\}$ are the bounds on the action duration.*

The above definition corresponds to temporal planning as defined by PDDL 2.1 (without numeric variables). The problem is defined using a set-theoretic representation, where states are subsets of P, which is the universe of facts over which one can determine the status of things, and the possible transitions that may take place in the system. When interpreted as a state, a subset s of P lists those atoms which hold true in it and implicitly asserts false those which are not part of it (closed-world assumption). Conditions are Boolean formulas over P. The initial state I specifies what holds at the beginning, before execution, while action tuples in A specify the dynamics of the system, that is, how a state can change, and under which propositional and temporal conditions such changes may happen. An action tuple delegates the specification of the state transition to two *snap actions*: one is relative to when the durative action starts, and one to when the durative action ends. As classical planning actions, these instantaneous transitions are represented by a pair, which encodes the applicability of the transition ($\mathrm{pre}(h)$), and the effects that the transition has on the state when applied ($\mathrm{eff}^+(h)$ and $\mathrm{eff}^-(h)$). Unlike in classical planning problems, however, in temporal planning actions last for a certain time (they are *durative*), and their duration has to satisfy the given lower and upper time limits, that is, $[L_a, U_a]$. Moreover, since the state can change while the action is under execution, we can further require that all intermediate states satisfy a given invariant condition ($\mathrm{pre}^{\leftrightarrow}(a)$). Note that PDDL 2.1 allows for instantaneous classical actions to be defined alongside durative actions. Here, for simplicity we do not allow for this feature, but it would be straightforward to do so. Finally, the Boolean formula G determines what needs to be achieved in order for the problem to be solved.

A collection of action tuples from A, together with a starting time and a duration, is called a plan.

Definition 2 (Plan)**.** *Let $\mathcal{P} = \langle P, A, I, G\rangle$ be a planning problem. A plan for $\mathcal{P}$ is a set of tuples $\pi = \{\langle a_1, t_1, d_1\rangle, \cdots, \langle a_n, t_n, d_n\rangle\}$, where, for each $1 \leq i \leq n$, $a_i \in A$ is a durative action, $t_i \in \mathbb{Q}_{\geq 0}$ is its start time, and $d_i \in \mathbb{Q}_{>0}$ is its duration.*

A plan can be understood as a set of timed decisions the agent can take over time. Indeed, each tuple of a plan defines what action needs to start (a_n), when it must be initiated (t_n), and how long it has to last (d_n). Note that this semantics assumes a *dense* time model, but all the constructions in this paper apply to a *discrete* time model as well.

In order to precisely state whether a given plan is valid with respect to the temporal planning problem it represents a candidate solution for, in the following we recall and summarize the state-transition model interpretation of a temporal plan given by Fox and Long (2003).

Definition 3 (Set of timed snap actions)**.** *A timed snap action (TSA) is a pair $\langle t, h\rangle$, where $t \in \mathbb{Q}_{\geq 0}$ and h is a snap action. Given a plan $\pi = \{\langle a_1, t_1, d_1\rangle, \cdots, \langle a_n, t_n, d_n\rangle\}$, the set of TSAs of π is defined as:*

$$H(\pi) = \{\langle t, a_\vdash\rangle, \langle t + d, a_\dashv\rangle \mid \langle a, t, d\rangle \in \pi\}$$

Given a set of timed snap actions, we define the induced parallel plan as the sequence of sets of timed snap actions sharing the same time index. As we will see, the validity of plans can be stated by defining constraints over the induced parallel plan that can be extracted from the timed snap actions of a plan.

Definition 4 (Induced parallel plan)**.** *Let π be a plan and let $H(\pi) = \{\langle t'_1, h_1\rangle, \cdots, \langle t'_m, h_m\rangle\}$ be the set of TSAs of π. The induced parallel plan for π is the sequence*

$\pi^{ind} = \langle\langle t''_1, \{h \mid \langle t''_1, h\rangle \in H(\pi)\}\rangle, \cdots, \langle t''_k, \{h \mid \langle t''_k, h\rangle \in H(\pi)\}\rangle\rangle$*, which is ordered and grouped with respect to the time index, that is,* $\forall i, j \in \{1, \cdots, k\}$*,* $i < j$ *if and only if* $t''_i < t''_j$*, and if* $t''_i = t''_j$*, then* $i = j$*. Given* $c_i = \langle a_i, t_i, d_i\rangle \in \pi$*, we denote by* $\pi^{ind}_{\vdash}(c_i) = x$*, with* $t''_x = t_i$*, and* $\pi^{ind}_{\dashv}(c_i) = y$*, with* $t''_y = t_i + d_i$*, the indexes of the pairs in* π^{ind} *containing, in the right hand side, the snap actions* $a_{i\vdash}$ *and* $a_{i\dashv}$ *relative to* c_i*, respectively.*

In order to define the validity of plans, we need to decide what happens to events that affect the same variables. We say that two snap actions are *mutex* whenever one interferes with at least one effect or precondition of the other. The concept is formally defined as follows.

Definition 5 (Mutex snap actions)**.** *Given a set of propositions* P *and a Boolean formula* $\phi \in \mathbb{B}_P$*, let* $P(\phi)$ *be the set of propositions mentioned by* ϕ*. Two snap actions* h *and* z *are* mutually exclusive *(mutex), denoted by* $\mathrm{mutex}(h, z)$*, if either* $P(\mathrm{pre}(h)) \cap \mathrm{eff}(z) \neq \varnothing$*, or* $P(\mathrm{pre}(z)) \cap \mathrm{eff}(h) \neq \varnothing$*, or* $\mathrm{eff}^+(h) \cap \mathrm{eff}^-(z) \neq \varnothing$*, or* $\mathrm{eff}^+(z) \cap \mathrm{eff}^-(h) \neq \varnothing$*.*

We are now ready to define the notion of plan validity. Intuitively, a plan is said to be valid if (i) the induced plan is a classical goal-reaching execution where all overall and duration constraints are satisfied, and (ii) mutex snap actions do not appear at the same time in a plan.

Given a set of propositions $s \subseteq P$, called *state*, and a condition $\phi \in \mathbb{B}_P$, we write $s \models \phi$ if s satisfies ϕ under the classical semantics of Boolean formulas.

Definition 6 (Plan validity)**.** *Given a temporal planning problem* $\mathcal{P} = \langle P, A, I, G\rangle$ *and a plan* $\pi = \{\langle a_1, t_1, d_1\rangle, \cdots, \langle a_n, t_n, d_n\rangle\}$ *for* $\mathcal{P}$*, and* $\pi^{ind} = \langle\langle t'_1, B_1\rangle, \cdots, \langle t'_m, B_m\rangle\rangle$ *be its induced plan. Then,* π *is valid if the following statements hold:*

1. $\forall i \in \{1, \cdots, n\}\ L_{a_i} \leq d_i \leq U_{a_i}$,
2. *there are no* $h, z \in B_i$*, with* $h \neq z$*, for some* $i \in \{1, \cdots, m\}$*, such that* $\mathrm{mutex}(h, z)$*,*
3. *given* $s_0 = I$*, for all* $i \in \{1, \ldots, m\}$*, it holds that:*
   (a) $s_i = (s_{i-1} \setminus \bigcup_{h \in B_i} \mathrm{eff}^-(h)) \cup \bigcup_{h \in B_i} \mathrm{eff}^+(h)$*;*
   (b) $s_i \models \bigwedge_{h \in B_i} \mathrm{pre}(h)$*;*
   (c) $s_m \models G$,
4. *for all* $c = \langle a, t, d\rangle \in \pi$ *and all* $\pi^{ind}_{\vdash}(c) \leq k < \pi^{ind}_{\dashv}(c)$*, we have* $s_k \models \mathrm{pre}^{\leftrightarrow}(a)$*.*
5. *actions do not self-overlap,* i.e.*, there are no* $1 \leq i, j \leq n$*, with* $i \neq j$*, such that* $a = a_i = a_j$ *and* $t_i \leq t_j \leq t_i + d_i$

Note that Item 2 of Definition 6 only assumes a non-zero time separation between mutex events (*non-zero separation* semantics), without assuming a concrete minimum amount of time (ϵ*-separation* semantics). The constructions in this paper do not depend on this assumption though, and work with ϵ-separation semantics as well. Note also that Item 5 explicitly excludes self-overlap of actions. This assumption is quite natural given that self-overlap leads to *undecidability* in dense time domains (Gigante et al. 2022).

3 ICE and Conditional Effects

In this section, we introduce the features that are the subject of this paper, that is, *intermediate conditions and effects* (ICE), and *conditional effects*.

ICE allow the modeler to state that something happens or has to happen in time points different from the start or the end of the action execution interval. An effect, for example, can be placed at k time steps after the start of the action, and a condition can be required to hold from k steps after the start to l steps before the end, and so on. Formally, planning problems with ICE can be defined as follows.

Definition 7 (Intermediate conditions and effects)**.** *Given a positive rational number* $k \in \mathbb{Q}_{\geq 0}$*, an* ICE term *is either a term* $\mathsf{START} + k$ *or* $\mathsf{END} - k$*.*

An ICE effect *is a tuple* $e = \langle \tau, \mathrm{eff}^+, \mathrm{eff}^-\rangle$ *where* τ *is an ICE term and* eff^+ *and* eff^- *are sets of propositions.*

An ICE condition *is a pair* $c = \langle \tau_1, \tau_2, \phi\rangle$ *where* τ_1 *and* τ_2 *are ICE terms and* ϕ *is a Boolean formula, such that:*

1. *if* $\tau_1 \equiv \mathsf{START} + k$ *and* $\tau_2 \equiv \mathsf{START} + l$*, then* $k < l$*;*
2. *if* $\tau_1 \equiv \mathsf{END} - k$ *and* $\tau_2 \equiv \mathsf{END} - l$*, then* $k > l$*;*

Definition 8 (Planning problem with ICE)**.** *A* durative action with ICE *is a tuple* $a = \langle a_\vdash, a_\dashv, \mathrm{pre}^{\leftrightarrow}(a), [L_a, U_a], E_a, C_a\rangle$ *where* $a_\vdash$*,* $a_\dashv$*,* $\mathrm{pre}^{\leftrightarrow}(a)$*,* L_a*, and* U_a *are defined as in Definition 1,* E_a *is a set of ICE effects, and* C_a *is a set of ICE conditions. A* planning problem with ICE *is a planning problem as in Definition 1 but made of durative actions with ICE.*

A thorough formal description of the semantics of ICE conditions and effects is beyond the scope of this paper, but can be found in the literature (Valentini, Micheli, and Cimatti 2020). Intuitively, an ICE effect $e = (\tau, \mathrm{eff}^+, \mathrm{eff}^-)$ applies the positive effects eff^+ and the negative effects eff^- to the state at the time specified by τ: if $\tau = \mathsf{START} + k$, the effects are applied k time steps after the start of the action. Similarly, if $\tau = \mathsf{END} - k$, the effects are applied k time steps before the end of the action. An ICE condition $c = \langle \tau_1, \tau_2, \phi\rangle$ requires that ϕ holds during all the time points between τ_1 and τ_2 (extrema included).

Figure 1 shows an example usage of ICE. The setting is the operation of a production plant. A production machine makes some treatment over a part of the final product, which is done in 50 seconds. However, to avoid the part to oxidize and become useless, it has to be picked and stored within other 50 seconds. This can be modeled by making the action make-treatment 100 seconds long, with an intermediate effect at $t = 50$ that sets the $done$ fluent, signaling the end of the treatment. Then, the pick action has to be executed within the next 50 seconds, because the end of make-treatment requires the $picked$ fluent to be true.

We now turn our attention to conditional effects. In the context of temporal planning, they are defined by Fox and Long (2003) in their definition of PDDL 2.1. Intuitively, with conditional effects on durative actions, one can make the application of an effect depend on a combination of conditions evaluated at the start of, at the end of, and over all the interval of execution of the action. For example, one may say that at the start of an action the effect p is applied only if the formula θ holds at the end, and ψ holds true over

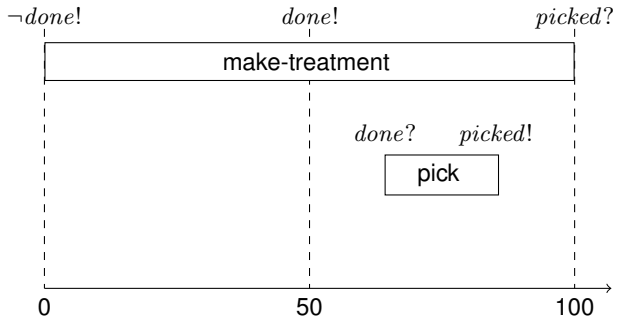

Figure 1: Example use case of ICE. Question marks denote conditions, exclamation marks denote effects.

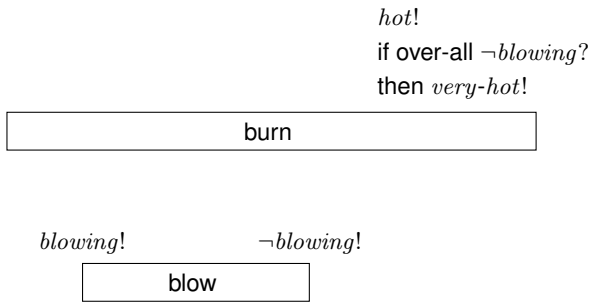

Figure 2: Example use case of conditional effects. Question marks denote conditions, exclamation marks denote effects.

the entire execution. Formally, we can define conditional effects as follows.

Definition 9 (Conditional effects)**.** *Let P be a set of propositions. A* conditional effect *is a tuple $e = \langle \phi, \psi, \theta, \text{eff}^+_\vdash, \text{eff}^-_\vdash, \text{eff}^+_\dashv, \text{eff}^-_\dashv \rangle$, where $\phi \in \mathbb{B}_P$ is the* at-start *condition, $\psi \in \mathbb{B}_P$ is the* over-all *condition, $\theta \in \mathbb{B}_P$ is the* at-end *condition, $\text{eff}^+_\vdash \subseteq P$ and $\text{eff}^-_\vdash \subseteq P$ are the positive and negative* at-start *effects, and $\text{eff}^+_\dashv \subseteq P$ and $\text{eff}^-_\dashv \subseteq P$ are the positive and negative* at-end *effects.*

Definition 10 (Problem with conditional effects)**.** *A* durative action *with conditional effects is a tuple $a = \langle a_\vdash, a_\dashv, \text{pre}^\leftrightarrow(a), [L_a, U_a], E_a \rangle$, where $a_\vdash$, $a_\dashv$, $\text{pre}^\leftrightarrow(a)$, L_a, and U_a are defined as in Definition 1, and E_a is a set of* conditional effects*. A* planning problem with conditional effects *is a problem $\mathcal{P} = \langle P, A, I, G \rangle$ where P, I and G are defined as in Definition 1, and A is a set of durative actions with conditional effects.*

Intuitively, if an action a is equipped with a conditional effect $e = \langle \phi, \psi, \theta, \text{eff}^+_\vdash, \text{eff}^-_\vdash, \text{eff}^+_\dashv, \text{eff}^-_\dashv \rangle$, and the action is executed, the positive and negative *at-start* and *at-end* effects $\text{eff}^+_\vdash$, $\text{eff}^-_\vdash$, $\text{eff}^+_\dashv$, $\text{eff}^-_\dashv$ are applied if and only if the *at-start* condition ϕ holds at the start of the execution, the *over-all* condition ψ holds during all the execution of the action (extrema excluded), and the *at-end* condition θ holds at the end of the execution. As in the case of ICE in the previous section, we do not delve into all the formal details of the semantics of planning problems with conditional effects, redirecting the reader to Fox and Long (2003).

The kind of situations that can be expressed with conditional effects in temporal planning is qualitatively different than what can be done in classical planning, because a condition can involve events that are very distant in the plan. For example, one may say that the effect of setting a fluent at the end of an action is applied only if some condition has held over all the execution of the action. This is what we use to express the use case depicted in Figure 2. An action burn is used to switch on a burner that has the purpose of increasing the temperature. The burner has only a single mode of operation and can be switched on only for a fixed amount of time, but the problem requires different temperature levels. For this reason, some air can be blown to balance the temperature (the blow action). Hence, if the burn action is executed alone, both the hot and $very\text{-}hot$ fluents are set. If the blow action is executed in the middle, only the hot fluent is set.

Conditional effects emerge in various compilations from planning with state trajectory constraints (*e.g.*, (Torres and Baier 2015; Bonassi et al. 2021)) and conformant planning (*e.g.*, (Palacios and Geffner 2007; Grastien and Scala 2020; Scala and Grastien 2021) to classical planning. In the case of state-trajectory constraints, conditional effects are used to keep track of the side effect of each agent's decision on a monitoring automaton that represents the satisfiability status of the constraints. For example, Torres and Baier (2015) use a NFA automaton that accepts all plans that satisfy the provided trajectory constraints. In conformant planning problems, for example, Grastien and Scala (2020) use conditional effects to link the action effects with some assumption over the possible worlds in the agent's initial belief. In the resulting classical planning problem, the agent is required to find a plan that works for each such assumption. The support of conditional effects that can be conditioned throughout the entire execution of an action has the potential to extend such approaches to temporal planning problems as well.

4 Simultaneity Constraints

In order to show how to encode ICE and conditional effects into a temporal planning problem, we use the concept of *simultaneity constraint*.

We need some terminology first.

Definition 11 (Plan projection)**.** *Let $\mathcal{P} = \langle P, A, I, G \rangle$ be a temporal planning problem and let A' be a set of actions such that $A' \cap A \neq \varnothing$. Given a plan π for $\mathcal{P}$, the* projection *of π over A', denoted $\pi|_{A'}$, is the plan obtained from π by removing any action $a \in A \setminus A'$.*

Intuitively, simultaneity constraints impose some snap actions to always happen at the same time. They are formally defined as follows.

Definition 12 (Simultaneity constraint)**.** *Let $\mathcal{P} = \langle P, A, I, G \rangle$ be a temporal planning problem, and let a_{*_a} and b_{*_b} be two snap actions, where $*_a, *_b \in \{\vdash, \dashv\}$. A problem $\mathcal{P}' = \langle P', A', I', G' \rangle$* augments *$\mathcal{P}$ with the* simultaneity constraint *$a_{*_a} \;/\!/\; b_{*_b}$ if any plan π is a valid plan for $\mathcal{P}'$ if and only if $\pi|_A$ is a valid plan for $\mathcal{P}$ and, for each $t \in \mathbb{Q}_{\geq 0}$*

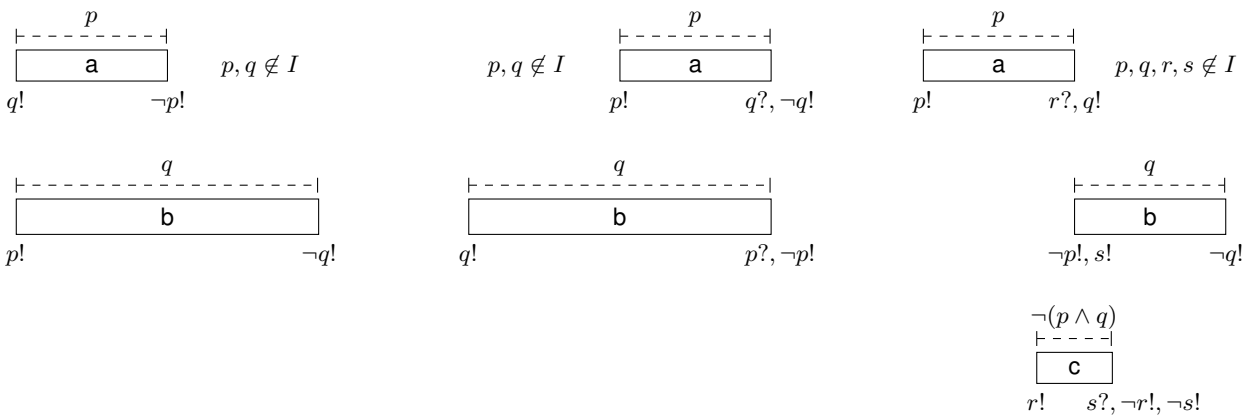

Figure 3: Schematics for the construction of $a_{\vdash} \;/\!/\; b_{\vdash}$ (left), $a_{\dashv} \;/\!/\; b_{\dashv}$ (center), and $a_{\dashv} \;/\!/\; b_{\vdash}$ (right). Question marks denote conditions, exclamation marks denote effects, dashed lines denote overall conditions.

such that $(t, a_{*_a}) \in H(\pi)$*, we have* $(t, b_{*_b}) \in H(\pi)$*, and* vice versa*, for each* $t \in \mathbb{Q}_{\geq 0}$ *such that* $(t, b_{*_b}) \in H(\pi)$*, we have* $(t, a_{*_a}) \in H(\pi)$*.*

As an example, given two durative actions a and b, if we augment $\mathcal{P}$ with the simultaneity constraint $a_{\dashv} \;/\!/\; b_{\vdash}$ we restrict valid plans for π to those where for each instance of a there is an instance of b where the end of a and the start of b are simultaneous, and *vice versa*.

We show now how to add a simultaneity constraint to a temporal planning problem. The encoding is different for the three cases of $a_{\vdash} \;/\!/\; b_{\vdash}$ (both starting snap actions), $a_{\dashv} \;/\!/\; b_{\dashv}$ (both ending snap actions), and $a_{\dashv} \;/\!/\; b_{\vdash}$ (an ending and a starting snap action). The key ideas of how to encode simultaneity constraints in the three cases are shown in Figure 3. Let us understand how the encoding works intuitively before delving into the formal details. In the case of $a_{\vdash} \;/\!/\; b_{\vdash}$, two fresh propositions p and q are introduced, which are initially set to false. The overall conditions of a and b are enriched by asking respectively p and q, while the starting effect of a sets q and the starting effect of b sets p. Now, a cannot start before the start of b because it needs p to hold, and b cannot start before the start of a because it needs q to hold. The result is that the two actions have to start together. The end effects of a and b then reset p and q to *false* to recover the initial state, allowing the construction to repeat if needed. A similar reasoning holds for $a_{\dashv} \;/\!/\; b_{\dashv}$.

The case for $a_{\dashv} \;/\!/\; b_{\vdash}$ is different, as it involves a *clip construction*. A third fresh action c is introduced, which is used to hold together a and b, alongside fresh propositions p, q, r, and s. The propositions r and s are used to force c to start before the end of a and end after the start of b, enclosing the two snap actions that we want to glue together. Then, a has overall condition p, while b has overall condition q, but C has overall condition $\neg(p \wedge q)$, thus a and b cannot overlap. Moreover, the end of a sets q, so the overall condition of C risks to be invalidated as soon as a ends. However, the start of b sets $\neg p$, hence the only way for the overall condition of C to stay satisfied is for $a_{\dashv}$ and $b_{\vdash}$ to happen at the same time. We can now formally define what we described above. Let us start from the case of $a_{\vdash} \;/\!/\; b_{\vdash}$.

Definition 13 (Encoding of $a_{\vdash} \;/\!/\; b_{\vdash}$)**.** *Given a temporal planning problem* $\mathcal{P} = \langle P, A, I, G \rangle$*, two actions* $a, b \in A$*, and two propositions* p *and* q *(not necessarily in* P*), we denote as* $\mathrm{enc}(a_{\vdash} \;/\!/\; b_{\vdash}, \mathcal{P}, p, q)$ *the temporal planning problem* $\mathcal{P}' = \langle P', A', I', G' \rangle$ *defined as follows:*

1. $P' = P \cup \{p, q\}$*;*
2. $I' = I$ (i.e.*,* p *and* q *start false) and* $G' = G$*;*
3. $A' = A \setminus \{a, b\} \cup \{a', b'\}$ *where* a' *and* b' *are like* a *and* b *excepting for what follows:*
   - *(a)* $\mathrm{eff}^{+}(a'_{\vdash}) = \mathrm{eff}^{+}(a_{\vdash}) \cup \{q\}$*;*
   - *(b)* $\mathrm{eff}^{-}(a'_{\dashv}) = \mathrm{eff}^{-}(a_{\dashv}) \cup \{p\}$*;*
   - *(c)* $\mathrm{pre}^{\leftrightarrow}(a') = \mathrm{pre}^{\leftrightarrow}(a) \wedge p$*;*
   - *(d)* $\mathrm{eff}^{+}(b'_{\vdash}) = \mathrm{eff}^{+}(b_{\vdash}) \cup \{p\}$*;*
   - *(e)* $\mathrm{eff}^{-}(b'_{\dashv}) = \mathrm{eff}^{-}(b_{\dashv}) \cup \{q\}$*;*
   - *(f)* $\mathrm{pre}^{\leftrightarrow}(b') = \mathrm{pre}^{\leftrightarrow}(b) \wedge q$*;*

For the case of $a_{\dashv} \;/\!/\; b_{\dashv}$ the construction is as follows.

Definition 14 (Encoding of $a_{\dashv} \;/\!/\; b_{\dashv}$)**.** *Given a temporal planning problem* $\mathcal{P} = \langle P, A, I, G \rangle$*, two actions* $a, b \in A$*, and two propositions* p *and* q *(not necessarily in* P*), we denote as* $\mathrm{enc}(a_{\dashv} \;/\!/\; b_{\dashv}, \mathcal{P}, p, q)$ *the temporal planning problem* $\mathcal{P}' = \langle P', A', I', G' \rangle$ *defined as follows:*

1. $P' = P \cup \{p, q\}$*;*
2. $I' = I$ (i.e.*,* p *and* q *start false) and* $G' = G$*;*
3. $A' = A \setminus \{a, b\} \cup \{a', b'\}$ *where* a' *and* b' *are like* a *and* b *excepting for what follows:*
   - *(a)* $\mathrm{eff}^{+}(a'_{\vdash}) = \mathrm{eff}^{+}(a_{\vdash}) \cup \{p\}$*;*
   - *(b)* $\mathrm{eff}^{-}(a'_{\dashv}) = \mathrm{eff}^{-}(a_{\dashv}) \cup \{q\}$*;*
   - *(c)* $\mathrm{pre}(a'_{\dashv}) = \mathrm{pre}(a_{\dashv}) \wedge q$*;*
   - *(d)* $\mathrm{pre}^{\leftrightarrow}(a') = \mathrm{pre}^{\leftrightarrow}(a) \wedge p$*;*
   - *(e)* $\mathrm{eff}^{+}(b'_{\vdash}) = \mathrm{eff}^{+}(b_{\vdash}) \cup \{q\}$*;*
   - *(f)* $\mathrm{eff}^{-}(b'_{\dashv}) = \mathrm{eff}^{-}(b_{\dashv}) \cup \{p\}$*;*
   - *(g)* $\mathrm{pre}(b'_{\dashv}) = \mathrm{pre}(b_{\dashv}) \wedge p$*;*
   - *(h)* $\mathrm{pre}^{\leftrightarrow}(b') = \mathrm{pre}^{\leftrightarrow}(b) \wedge q$*;*

The case of $a_{\dashv} \;/\!/\; b_{\vdash}$ is encoded as follows.

Definition 15 (Encoding of $a_\dashv \mathbin{//} b_\vdash$)**.** *Given a temporal planning problem $\mathcal{P} = \langle P, A, I, G\rangle$, two actions $a, b \in A$, and three propositions p, q, and r (not necessarily in P), we denote as $\mathrm{enc}(a_\dashv \mathbin{//} b_\vdash, \mathcal{P}, p, q, r)$ the temporal planning problem $\mathcal{P}' = \langle P', A', I', G'\rangle$ defined as follows:*

1. $P' = P \cup \{p, q\}$;
2. $I' = I$ (i.e., p, q, r, and s start false) and $G' = G$;
3. $A' = A \setminus \{a, b\} \cup \{a', b', c\}$, *where a' and b' are like a and b excepting for what follows:*
   - (a) $\mathrm{eff}^+(a'_\vdash) = \mathrm{eff}^+ a_\vdash \cup \{p\}$;
   - (b) $\mathrm{eff}^+(a'_\dashv) = \mathrm{eff}^+ a_\dashv \cup \{q\}$;
   - (c) $\mathrm{pre}(a'_\dashv) = \mathrm{pre}(a_\dashv) \wedge r$;
   - (d) $\mathrm{pre}^\leftrightarrow(a') = \mathrm{pre}^\leftrightarrow(a) \wedge p$;
   - (e) $\mathrm{eff}^-(b'_\vdash) = \mathrm{eff}^-(b_\vdash) \cup \{p\}$;
   - (f) $\mathrm{eff}^+(b'_\vdash) = \mathrm{eff}^+(b_\vdash) \cup \{s\}$;
   - (g) $\mathrm{eff}^-(b'_\dashv) = \mathrm{eff}^-(b_\dashv) \cup \{q\}$;
   - (h) $\mathrm{pre}^\leftrightarrow(b') = \mathrm{pre}^\leftrightarrow(b) \wedge q$;

   and $c = \langle c_\vdash, c_\dashv, \mathrm{pre}^\leftrightarrow(c), [L_c, U_c]\rangle$ is defined as follows:
   - (a) $\mathrm{eff}^+(c_\vdash) = \{r\}$, $\mathrm{eff}^-(c_\vdash) = \varnothing$;
   - (b) $\mathrm{eff}^+(c_\dashv) = \varnothing$, $\mathrm{eff}^-(c_\dashv) = \{r, s\}$;
   - (c) $\mathrm{pre}(c_\vdash) = \top$, $\mathrm{pre}(c_\dashv) = s$;
   - (d) $\mathrm{pre}^\leftrightarrow(c) = \neg(p \wedge q)$;
   - (e) $L_a = U_a = \min\{L_a, L_b\}$.

One can check that the constructions in Definitions 13, 14 and 15 work as intended.

Lemma 1 (Soundness of simultaneity constraints encoding)**.** *Given a temporal planning problem $\mathcal{P} = \langle P, A, I, G\rangle$, two actions $a, b \in A$, and three propositions $p, q, r \notin P$, then:*

1. *the planning problem $\mathrm{enc}(a_\vdash \mathbin{//} b_\vdash, \mathcal{P}, p, q)$ augments $\mathcal{P}$ with the simultaneity constraint $a_\vdash \mathbin{//} b_\vdash$;*
2. *the planning problem $\mathrm{enc}(a_\dashv \mathbin{//} b_\dashv, \mathcal{P}, p, q)$ augments $\mathcal{P}$ with the simultaneity constraint $a_\dashv \mathbin{//} b_\dashv$;*
3. *the planning problem $\mathrm{enc}(a_\dashv \mathbin{//} b_\vdash, \mathcal{P}, p, q, r)$ augments $\mathcal{P}$ with the simultaneity constraint $a_\dashv \mathbin{//} b_\vdash$.*

Another kind of simultaneity constraint will come useful later: a constraint that imposes a durative action to be simultaneous with one of a set of other actions.

Definition 16 (Simultaneity constraint between actions)**.** *Let $\mathcal{P} = \langle P, A, I, G\rangle$ be a temporal planning problem, and let $a \in A$ and $b_0, \ldots, b_n \in A$ be durative actions. A problem $\mathcal{P}' = \langle P', A', I', G'\rangle$* augments *$\mathcal{P}$ with the simultaneity constraint $a \mathbin{//} \{b_0, \ldots, b_n\}$ if any plan π is valid for $\mathcal{P}'$ if and only if $\pi|_A$ is valid for $\mathcal{P}$ and, for each instance of a executing in $\pi|_A$, there is an instance of one of $b_0, \ldots, b_n$ executing in $\pi|_A$ starting and ending exactly at the same time.*

Intuitively, if a is an action and $b_0, \ldots, b_n$ are actions, when we augment a problem $\mathcal{P}$ with a constraint $a \mathbin{//} \{b_0, \ldots, b_n\}$, we impose that for each instance of a, at least one of $b_0, \ldots, b_n$ is executed exactly at the same time of a. To encode this constraint, we can at first observe that augmenting a problem with $a \mathbin{//} \{b\}$ is equivalent to augmenting it with the two simultaneity constraints $a_\vdash \mathbin{//} b_\vdash$ and $a_\dashv \mathbin{//} b_\dashv$. Then, by reusing some auxiliary propositions, we can encode the disjunction between $b_0, \ldots, b_n$.

Definition 17 (Encoding of $a \mathbin{//} \{b_0, \ldots, b_n\}$)**.** *Given a temporal planning problem $\mathcal{P} = \langle P, A, I, G\rangle$ and actions $a, b_0, \ldots, b_n \in A$, $\mathrm{enc}(a \mathbin{//} \{b_0, \ldots, b_n\}, \mathcal{P})$ denotes the problem recursively defined as follows:*

$$\begin{aligned}\mathrm{enc}(a \mathbin{//} \varnothing, \mathcal{P}) &= \mathcal{P}\\ \mathrm{enc}(a \mathbin{//} \{b_0, \ldots, b_n\}, \mathcal{P}) &= \mathrm{enc}(a_\vdash \mathbin{//} b_{0\vdash}, \mathcal{P}'', p_1, q_1)\\ \text{where } \mathcal{P}'' &= \mathrm{enc}(a_\dashv \mathbin{//} b_{0\dashv}, \mathcal{P}', p_2, q_2)\\ \mathcal{P}' &= \mathrm{enc}(a \mathbin{//} \{b_0, \ldots, b_{n-1}\})\end{aligned}$$

where $p_1, p_2, q_1, q_2 \notin P$ are four fresh propositions.

The recursive definition of $\mathrm{enc}(a \mathbin{//} \{b_0, \ldots, b_n\}, \mathcal{P})$ in Definition 17 augments the original problem $\mathcal{P}$, once for each action $\{b_0, \ldots, b_n\}$, with two simultaneity constraints that bind together $a_\vdash$ with $b_{i\vdash}$ and $a_\dashv$ with $b_{i\dashv}$. However, this construction does more than the simple combination of $a_\vdash \mathbin{//} b_{i\vdash}$ and $a_\dashv \mathbin{//} b_{i\dashv}$: by reusing the same propositions p_1, p_2, q_1, q_2 to encode the simultaneity constraints between the snap actions of a and $b_0, \ldots, b_n$, we make sure that the constructions shown in Figure 3 are satisfied by any one of the actions $b_0, \ldots, b_n$. One can check the following result.

Lemma 2 (Soundness of Definition 17)**.** *Given a temporal planning problem $\mathcal{P} = \langle P, A, I, G\rangle$, and durative actions $a, b_0, \ldots, b_n \in A$, the temporal planning problem $\mathrm{enc}(a \mathbin{//} \{b_0, \ldots, b_n\}, \mathcal{P})$ augments $\mathcal{P}$ with the constraint $a \mathbin{//} \{b_0, \ldots, b_n\}$.*

5 Intermediate Conditions and Effects

Thanks to the simultaneity constraints developed in Section 4, we can easily encode planning problem with ICE into a plain planning problem.

Theorem 1 (ICE encoding)**.** *Given a planning problem with ICE $\mathcal{P} = \langle P, A, I, G\rangle$, there exists a planning problem $\mathcal{P}' = \langle P', A', I', G'\rangle$ such that any plan π is valid for $\mathcal{P}'$ if and only if $\pi|_A$ is valid for $\mathcal{P}$.*

Proof. Let $a = \langle a_\vdash, a_\dashv, \mathrm{pre}^\leftrightarrow(a), [L_a, U_a], E_a, C_a\rangle$ be a durative action with ICE from $\mathcal{P}$. Then, a can be simply replaced by the corresponding plain durative action, after augmenting the problem with some auxiliary actions and suitable simultaneity constraints. How exactly this is done depends on the kind of ICE involved (see Figure 4):

1. For an ICE effect $\langle \mathsf{START} + k, \mathrm{eff}^+, \mathrm{eff}^-\rangle$, we add an auxiliary action e with $\mathrm{eff}^+(e_\dashv) = \mathrm{eff}^+$ and $\mathrm{eff}^-(e_\dashv) = \mathrm{eff}^-$, $L_e = U_e = k$, and we augment the problem with the simultaneity constraint $a_\vdash \mathbin{//} e_\vdash$.
2. Similarly, for an ICE effect $\langle \mathsf{END} - k, \mathrm{eff}^+, \mathrm{eff}^-\rangle$, we add an auxiliary action e with $\mathrm{eff}^+(e_\vdash) = \mathrm{eff}^+$ and $\mathrm{eff}^-(e_\vdash) = \mathrm{eff}^-$, $L_e = U_e = k$, and we augment the problem with the simultaneity constraint $a_\dashv \mathbin{//} e_\dashv$.
3. For an ICE condition $\langle \mathsf{START} + k, \mathsf{START} + l, \phi\rangle$ we add two auxiliary actions e_1 and e_2. We set $\mathrm{pre}(e_{2\vdash}) = \mathrm{pre}(e_{2\dashv}) = \mathrm{pre}^\leftrightarrow(e_2) = \phi$, $L_{e_1} = U_{e_1} = k$, $L_{e_2} = U_{e_2} = l - k$, and we augment the problem with the constraints $a_\vdash \mathbin{//} e_{1\vdash}$ and $e_{1\dashv} \mathbin{//} e_{2\vdash}$.

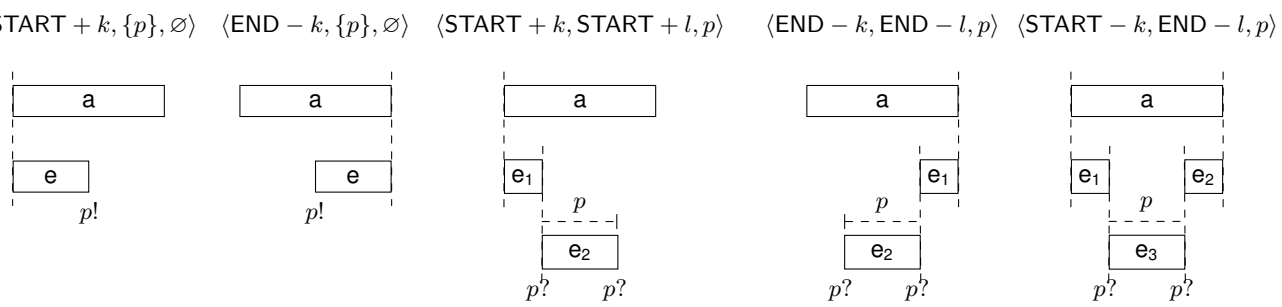

Figure 4: Constructions to encode ICE using simultaneity constraints. Question marks denote conditions, exclamation marks denote effects, horizontal dashed lines denote overall conditions, vertical dashed lines denote simultaneity constraints.

4. Similarly, for an ICE condition $\langle \mathsf{END}-k, \mathsf{END}-l, \phi\rangle$ we add two auxiliary actions e_1 and e_2. We set $\mathrm{pre}(e_{2\vdash}) = \mathrm{pre}(e_{2\dashv}) = \mathrm{pre}^{\leftrightarrow}(e_2) = \phi$, $L_{e_1} = U_{e_1} = l$, $L_{e_2} = U_{e_2} = k - l$, and we augment the problem with the constraints $a_\dashv \mathbin{/\!/} e_{1\dashv}$ and $e_{1\vdash} \mathbin{/\!/} e_{2\dashv}$.
5. For an ICE condition $\langle \mathsf{START}+k, \mathsf{END}-k, \phi\rangle$, we add three auxiliary actions e_1, e_2, and e_3. We set $\mathrm{pre}(e_{3\vdash}) = \mathrm{pre}(e_{3\dashv}) = \mathrm{pre}^{\leftrightarrow}(e_3) = \phi$, $L_{e_1} = U_{e_1} = k$, $L_{e_2} = U_{e_2} = l$, and we augment the problem with the constraints $a_\vdash \mathbin{/\!/} e_{1\vdash}$, $e_{1\dashv} \mathbin{/\!/} e_{3\vdash}$, $e_{3\dashv} \mathbin{/\!/} e_{2\vdash}$, and $e_{2\dashv} \mathbin{/\!/} a_\dashv$.

It can be checked that any plan π is valid for $\mathcal{P}'$ if and only if $\pi|_A$ is valid for $\mathcal{P}$. □

By applying this compilation of ICE to the example problem of Figure 1, we replace make-treatment with a corresponding plain durative action that has no intermediate effect. Then, we add action e with duration 50 that sets $done$ true at its end, and is forced to start together with make-treatment by means of the simultaneity constraint $a_\vdash \mathbin{/\!/} e_\vdash$ with a = make-treatment. The rest of the problem is left untouched; it is easy to see that any plan starting the make-treatment action at time t needs to start e at the same time, causing the positive effect on $done$ to happen at time $t + 50$, hence capturing the ICE semantics. Moreover, the plan where e is projected away is a valid plan for the original problem with ICE.

Note that the constructions employed in Theorem 1 produce only a few additional actions. In particular, one additional action is required for ICE effects, while five additional actions (including the actions needed for the encoding of simultaneity constraints) are needed to encode the most complex kind of ICE condition (*i.e.*, $\langle \mathsf{START}+k, \mathsf{END}-l, \phi\rangle$). In any case, the growth in size of the resulting problem is only polynomial. This fact, together with the complexity results by Gigante et al. (2022), implies the following.

Corollary 1 (Complexity of planning with ICE)**.** *Finding whether a valid plan exists for a planning problem with ICE is PSPACE-complete.*

6 Conditional Effects

We focus now on how problems with conditional effects can be polynomially encoded into plain planning problems.

Theorem 2 (Conditional effects encoding)**.** *Given a planning problem with conditional effects $\mathcal{P} = \langle P, A, I, G\rangle$, there exists a planning problem $\mathcal{P}' = \langle P', A', I', G'\rangle$ such that any plan π is valid for $\mathcal{P}'$ iff $\pi|_A$ is valid for $\mathcal{P}$.*

Proof. Figure 5 shows the construction for an example conditional effect. For a durative action a with conditional effect $e = \langle \phi, \psi, \theta, \mathrm{eff}^+_\vdash, \mathrm{eff}^-_\vdash, \mathrm{eff}^+_\dashv, \mathrm{eff}^-_\dashv\rangle$, we introduce five auxiliary actions b_0, b_1, b_2, b_3, and b'_3, and we augment the problem with the simultaneity constraint $a \mathbin{/\!/} \{b_0, b_1, b_2, b_3\}$ and $b_{3\vdash} \mathbin{/\!/} b'_{3\vdash}$. In this way, we create the situation depicted in the figure, with one of b_0, b_1, b_2, b_3 forced to happen simultaneous to a, and b'_3 forced to start at the start of b_3. Now, we set conditions and effects of b_0 to mimic the conditional effect, that is:

$$\mathrm{pre}(b_{0\vdash}) = \phi \qquad \mathrm{pre}^{\leftrightarrow}(b_0) = \psi \qquad \mathrm{pre}(b_{0\dashv}) = \theta$$
$$\mathrm{eff}^+(b_{0\vdash}) = \mathrm{eff}^+_\vdash \quad \mathrm{eff}^-(b_{0\vdash}) = \mathrm{eff}^-_\vdash \quad \mathrm{eff}^+(b_{0\dashv}) = \mathrm{eff}^+_\dashv$$
$$\mathrm{eff}^-(b_{0\dashv}) = \mathrm{eff}^-_\dashv$$

Hence, b_0 is executed (simultaneously to a), only if all the conditions of the conditional effect are satisfied, applying the corresponding effects. We then handle the case where the conditions of the conditional effect are *not* satisfied. To this end, we set the conditions of b_1, b_2, and b'_3 accordingly:

$$\mathrm{pre}(b_{1\vdash}) = \neg\phi \quad \mathrm{pre}(b_{2\dashv}) = \neg\theta \quad \mathrm{pre}(b'_{3\dashv}) = \neg\psi$$

In this way, b_1 is executed (simultaneously to a) when the *at-start* condition of the conditional effect is not satisfied, b_2 is executed when the *at-end* condition of the conditional effect is not satisfied, and b_3 is executed when the *over-all* condition of the conditional effect is not satisfied, *i.e.*, when there exists at least one time point (the end of b'_3) during the execution of the action where $\neg\psi$ holds. As the last detail, an auxiliary proposition x is introduced to force b'_3 to end before b_3, by setting $\mathrm{eff}^+(b'_{3\vdash})$, $\mathrm{eff}^-(b'_{3\dashv}) = \{x\}$ and $\mathrm{pre}(b_{3\vdash}) = \neg x$. Note that, as noted in the figure, b_1, b_2, and b_3 are mutually exclusive with b_0 due to their conditions. However, b_1, b_2 and b_3 may potentially execute together with no harm.

One can check that the construction described above indeed encodes the given conditional effect. □

Consider the example problem of Figure 2: to compile away the conditional effect at the end of action burn we

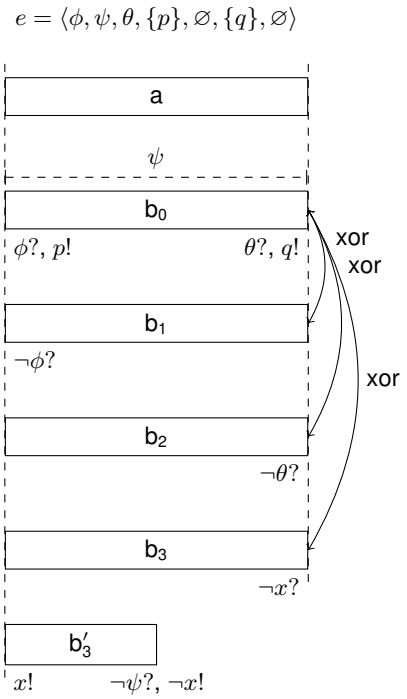

Figure 5: Constructions to encode conditional effects of a durative action using simultaneity constraints. Question marks denote conditions, exclamation marks denote effects, horizontal dashed lines denote overall conditions, vertical dashed lines denote simultaneity constraints, arrows denote mutual exclusion between actions.

only need the actions b_0, b_3 and b'_3 of the above construction as the condition only involves an over-all constraint. The action burn in the resulting problem will only have the unconditional, end effect setting hot true, and either action b_0 or b_3 is required to be in parallel with burn. The action b_0 has an over-all condition $\neg blowing$ and an end effect setting $very\text{-}hot$ true, encoding the execution of the conditional effect. Instead, the action b_3 requires action b'_3 to start at the same time of b_3 and by means of the end condition of b'_3 checks if there exists a point within the duration of burn where $blowing$ is true, hence encoding the non-execution of the conditional effect.

Note that the cost of the encoding employed in Theorem 2 is only five actions for each conditional effect.

Corollary 2 (Complexity of planning with conditional effects)**.** *Finding whether a valid plan exists for a planning problem with conditional effects is PSPACE-complete.*

7 Related Work

Intermediate Conditions and Effects (ICE) as defined in this paper follow the abstract syntax of the Action Notation Modeling Language (ANML) (Smith, Frank, and Cushing 2008). The NDL planning language (Rintanen 2015) supports delayed effects with timing relative to the starting of the action. ICE is useful in several contexts, for example for modeling schedules and events (Valentini, Micheli, and Cimatti 2020) and to compile away duration uncertainty in temporal planning (Cimatti et al. 2018). Some planners natively support ICE (Bit-Monnot et al. 2020; Micheli and Scala 2019; Valentini, Micheli, and Cimatti 2020), but the majority of temporal planners do not.

Under the assumption of ϵ-separation semantics, two compilations for ICE have been presented in (Cimatti et al. 2018). These compilations build on the clip-action construction by Fox, Long, and Halsey (2004), and on the container-action construction by Smith (2003), respectively. The former construction introduces an additional action with duration $2 \times \epsilon$, forcing two snap-actions to happen at the same time by means of additional propositions. This effectively encodes a simultaneity constraint that can be used to encode ICE. The latter, instead, uses an envelope action that forces a fixed time distance between two or more actions that are constrained to happen one after the other. Note that the container-action compilation is less general than the clip, because it assumes that the actions have fixed durations.

Our compilation significantly extends the scope achievable by either of these existing compilations. First of all, we do not assume the ϵ-separation semantics: our compilation works both when the minimum time quantum must be specified by the user and when it is computed automatically by the planner. Moreover, our encoding uses less additional actions with respect to the previous compilations in general. In fact, to force the simultaneous execution of two starting or two ending snap-actions we do not use any additional durative action while both previous approaches require one additional action regardless of the type of snap actions involved.

Considering the example shown in Figure 1, the make-treatment action needs to be transformed into two actions aimed at capturing the effect at time 50. With the clip-action construction, we can create two actions each with duration 50, and can enforce the end of the first action to happen simultaneously (by means of an additional action) to the starting of the second. The first action sets $done$ true at its ending. With the container construction, instead, we can create two additional actions that are forced to happen in sequence and both during the make-treatment. The first lasts $50 - \epsilon$ and sets $done$ at its end, while the second has duration $50 - 2 \times \epsilon$. In this way, these two actions are forced to happen at specific times within the duration of make-treatment and the ending of the first action lays exactly at time 50 after the start of make-treatment. Our compilation, instead, only uses one (not two!) additional action that lasts 50 units of time with an ending effect that sets $done$, and forces it to start together with make-treatment.

Conditional effects are an interesting modeling feature of temporal planning that has been largely ignored in the literature. We observe that neither the naïve exponential compilation of conditional effects nor the polynomial one presented in the seminal paper by Nebel (2000) for classical planning can be easily adapted to the context of temporal planning. In fact, neither compilation supports conditions specified at times different than the timing of the effect.

The naïve exponential compilation for classical planning works by creating one action for every possible combination of the conditional effects. For example, two conditional

effects having preconditions ϕ and ψ respectively are compiled away by creating four copies of the action they belong to. Such actions have additional preconditions $\phi \wedge \psi$, $\phi \wedge \neg\psi$, $\neg\phi \wedge \psi$, and $\neg\phi \wedge \neg\psi$ respectively and each action has the appropriate effect given the precondition. This compilation is simple but is in general exponential and is not applicable in temporal planning for effects conditioned on the over-all constraint. For example, consider the situation depicted in Figure 2: the over-all constraint $\neg blowing$ cannot be easily negated to construct the copy of the action without the very-hot effect. In fact, the over-all constraint is falsified by the existence of a point where $blowing$ is true, and this is not easily expressible in a condition[1].

The polynomial compilation for classical planning works by encoding a set of conditional effects for a certain action a into a sequence of pairs of artificial actions that must be executed after a. Each pair encodes a single conditional effect and the two alternative actions are used to make the choice between applying the effect or skipping it.

For example, the pair of actions associated with a conditional effect having precondition ϕ is composed of one action having precondition ϕ and the (unconditional) effect, and another action with precondition $\neg\phi$ and no effect. Additional propositions are used to enforce the execution of exactly one action from each pair corresponding to each conditional effect after the execution of a.

In the context of temporal planning, this compilation suffers from the same limitation of the naïve exponential one. Moreover, in the case of ϵ-separation semantics, the sequence of actions introduced by the compilation would require a non-zero amount of time (in particular $\epsilon \times N$ where N is the number of contemporary conditional effects in case of ϵ-separation semantics). This timing is in general non-negligible and breaks the semantics of the compilation.

Our compilation of conditional effects takes inspiration from the polynomial compilation for classical planning: we also create a number of actions for each conditional effect. However, we overcome the over-all complication mentioned above by fully actively monitor each over-all condition appearing in conditional effects. Moreover, the simultaneity construction we use works regardless of the ϵ-separation semantics, thus maintaining the compilation polynomial without introducing artificial delays in the plan.

Both our techniques build on and generalize a construction to force the simultaneity of the start of two durative actions that to the best of our knowledge was first identified in the (unpublished) PhD thesis of William Cushing (Cushing 2012). The simultaneity construction for two starting points was used as an argument to reject the semantics of PDDL 2.1 and develop the message of the thesis. In this paper, we instead use and generalize this construction to force the simultaneity of any pair of clip actions and we show how it can be used to encode in the semantics of PDDL 2.1 (with or without ϵ-separation) both conditional effects and ICE.

Our construction is also related to a compilation of quantitative Allen relations presented in (Coles et al. 2019); in fact, our simultaneity operators can be seen as *starts-with*, *ends-with* or *meets* Allen constraints depending on the kind of clip actions involved. We remark that all previous compilations are limited to the ϵ-separation semantics, while our approach works with or without this assumption.

ICE and conditional effects, together with *over-all* conditions, also allows one to state that if during the interval execution of an action some condition ϕ holds, then there are two sub-intervals where two conditions ϕ' and ϕ'' hold. In the field of *interval temporal logics* (Della Monica et al. 2011), this effect is obtained by the *chop operator*. The interval temporal logic CDT, that results from considering this temporal operator, is *undecidable*, but here we can express the operator thanks to an implicit restriction to *existential* modal operators: we cannot state that *for all* the ways of splitting the interval in two subintervals, the two satisfies ϕ' and ϕ'', but only that one such splitting exists.

Finally, our computational complexity results build on the work by Gigante et al. (2022): we adopt their abstract language for temporal planning and we consider conditional effects and ICE, showing that the addition of such features does not impact the computational complexity of the planning problems regardless of the ϵ-separation semantics.

8 Conclusions

In this paper, we analyzed the expressive power of Intermediate Conditions and Effects (ICE) and conditional effects in action-based temporal planning languages. We showed that both these constructs do not add expressive power and that both features can be compiled away in an equivalent formulation irrespective of the assumptions on the semantics of time and the presence of ϵ-separation. Our encodings exploit a construction that forces the simultaneous execution of either two action endpoints or two entire durative actions. A secondary yet important consequence of our compilations is that they generalize recent results by Gigante et al. (2022). That is, the addition of ICE and conditional effects do not have any impact on the computational complexity of temporal planning, which remains PSPACE-complete when self-overlapping is prohibited. Last, but not least, we highlight that our results are constructive: our compilations can be used to approach problems featuring ICE or conditional effects even with planners that do not natively support them.

Following on from this last observation, in the future, we plan to empirically experiment with our compilations to evaluate the practical applicability on existing planners: this could open new venues for further research as the simultaneity constraints we used as a basis for our compilation could be in principle natively supported, hence providing a more efficient implementation of the ideas behind our encodings. Other directions of research include the analysis of other planning features such as the quantitative Allen-algebra constraints identified in (Coles et al. 2019). Speaking of Allen relations, it comes natural to see whether our constructions can be used to support a kind of *temporally extended goals* where, instead of LTL conditions, more expressive *interval* conditions can be expressed, for example in terms of (some fragment of) the Halpern and Shoham interval temporal logic (Della Monica et al. 2011).

[1]Note that this is *not* equivalent to an over-all $\neg blowing$ constraint, that would require $blowing$ to be false during the action.

Acknowledgements

Andrea Micheli and Enrico Scala have been partially supported by AIPlan4EU, a project funded by EU Horizon 2020 research and innovation programme under GA n. 101016442. Nicola Gigante has been partially supported by the TOTA project (*"Temporal Ontologies and Tableaux Algorithms"*) and by the STAGE project (*"Synthesis for Timeline-based Planning Games"*) by the Faculty of Computer Science, Free University of Bozen-Bolzano.

References

Bit-Monnot, A.; Ghallab, M.; Ingrand, F.; and Smith, D. E. 2020. FAPE: a constraint-based planner for generative and hierarchical temporal planning. *CoRR* abs/2010.13121.

Bonassi, L.; Gerevini, A. E.; Percassi, F.; and Scala, E. 2021. On planning with qualitative state-trajectory constraints in PDDL3 by compiling them away. In *Proceedings of the 31st International Conference on Automated Planning and Scheduling*, 46–50. AAAI Press.

Cesta, A.; Cortellessa, G.; Fratini, S.; Oddi, A.; and Rasconi, R. 2009. The APSI Framework: a Planning and Scheduling Software Development Environment. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling*.

Cimatti, A.; Do, M.; Micheli, A.; Roveri, M.; and Smith, D. E. 2018. Strong temporal planning with uncontrollable durations. *Artif. Intell.* 256:1–34.

Coles, A. J.; Coles, A.; Martínez, M.; Savas, E.; Delfa, J. M.; de la Rosa, T.; E-Martín, Y.; and Olaya, A. G. 2019. Efficiently reasoning with interval constraints in forward search planning. In *Proceedings of the The Thirty-Third AAAI Conference on Artificial Intelligence*, 7562–7569. AAAI Press.

Cushing, W. A. 2012. *When is Temporal Planning Really Temporal?* Ph.D. Dissertation, Arizona State University. https://rakaposhi.eas.asu.edu/cushing-dissertation.pdf.

Della Monica, D.; Goranko, V.; Montanari, A.; and Sciavicco, G. 2011. Interval temporal logics: a journey. *Bull. EATCS* 105:73–99.

Fox, M., and Long, D. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research* 20:61–124.

Fox, M.; Long, D.; and Halsey, K. 2004. An investigation into the expressive power of PDDL2.1. In de Mántaras, R. L., and Saitta, L., eds., *Proceedings of the 16th Eureopean Conference on Artificial Intelligence*, 328–342. IOS Press.

Frank, J., and Jónsson, A. 2003. Constraint-based Attribute and Interval Planning. *Constraints* 8(4):339–364.

Gigante, N.; Montanari, A.; Cialdea Mayer, M.; and Orlandini, A. 2017. Complexity of timeline-based planning. In *Proc. of the 27th International Conference on Automated Planning and Scheduling*, 116–124. AAAI Press.

Gigante, N.; Micheli, A.; Montanari, A.; and Scala, E. 2020. Decidability and complexity of action-based temporal planning over dense time. In *Proc. of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 9859–9866. AAAI Press.

Gigante, N.; Micheli, A.; Montanari, A.; and Scala, E. 2022. Decidability and complexity of action-based temporal planning over dense time. *Artif. Intell.* 307:103686.

Grastien, A., and Scala, E. 2020. CPCES: A planning framework to solve conformant planning problems through a counterexample guided refinement. *Artif. Intell.* 284:103271.

Micheli, A., and Scala, E. 2019. Temporal planning with temporal metric trajectory constraints. In *Proceedings of The Thirty-Third AAAI Conference on Artificial Intelligence*, 7675–7682. AAAI Press.

Nebel, B. 2000. On the compilability and expressive power of propositional planning formalisms. *J. Artif. Intell. Res.* 12:271–315.

Palacios, H., and Geffner, H. 2007. From conformant into classical planning: Efficient translations that may be complete too. In *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling*, 264–271. AAAI.

Rintanen, J. 2007. Complexity of concurrent temporal planning. In *Proc. of the 17th International Conference on Automated Planning and Scheduling*, 280–287.

Rintanen, J. 2015. Models of action concurrency in temporal planning. In Yang, Q., and Wooldridge, M. J., eds., *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 1659–1665. AAAI Press.

Scala, E., and Grastien, A. 2021. Non-deterministic conformant planning using a counterexample-guided incremental compilation to classical planning. In *Proceedings of the Thirty-First International Conference on Automated Planning and Scheduling*, 299–307. AAAI Press.

Smith, D.; Frank, J.; and Cushing, W. 2008. The anml language. In *KEPS 2008*.

Smith, D. E. 2003. The case for durative actions: A commentary on PDDL2.1. *J. Artif. Intell. Res.* 20:149–154.

Torres, J., and Baier, J. A. 2015. Polynomial-time reformulations of LTL temporally extended goals into final-state goals. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 1696–1703. AAAI Press.

Valentini, A.; Micheli, A.; and Cimatti, A. 2020. Temporal planning with intermediate conditions and effects. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 9975–9982. AAAI Press.