Rolling in Classical Planning with Conditional Effects and Constraints

Matteo Cardellini, **Enrico Giunchiglia**
DIBRIS, University of Genova, Genova, Italy
{matteo.cardellini, enrico.giunchiglia}@unige.it

Abstract

In classical planning, conditional effects (CEs) allow modelling non-idempotent actions, where the resulting state may depend on how many times each action is consecutively repeated. Though CEs have been widely studied in the literature, no one has ever studied how to exploit *rolling*, i.e., how to effectively model the consecutive repetition of an action. In this paper, we fill this void by (i) showing that planning with CEs remains PSPACE-complete even in the limit case of problems with a single action, (ii) presenting a correct and complete planning as satisfiability encoding exploiting rolling while effectively dealing with constraints imposed on the set of reachable states, and (iii) theoretically and empirically showing its substantial benefits.

1 Introduction

In *classical planning*, the environment in which agents operate is represented through Boolean variables, and actions are *idempotent*, i.e., applying the same action once or multiple times results in the same state. The idempotence property falls if we introduce *conditional effects* (CEs), i.e., state-dependant effects of actions. Dealing with *a single* application of an action with CEs has been extensively studied and two main approaches exist: either (i) one deals with CEs in a *native* way, i.e., by encapsulating CEs directly in the procedure searching for a plan [Rintanen, 2011; Röger *et al.*, 2014; Katz, 2019], or (ii) one *compiles-away* actions with CEs [Gazen and Knoblock, 1997; Nebel, 2000; Gerevini *et al.*, 2024]. Logic-based native approaches (see, e.g. [Rintanen, 2011]), are all based on *planning as satisfiability* (PaS) [Kautz and Selman, 1992] where a planning task Π is solved by first encoding Π into a corresponding logic formula Π_n incorporating a *bound* (or *number of steps*) n, and then searching for a model of Π_n, increasing n upon failure. In all existing PaS approaches, an action a can be applied at most once per step, and thus, a plan with r consecutive repetitions of a will be found in a number of steps $n \geq r$. The *rolling* technique, introduced for numeric planning in [Scala *et al.*, 2016], allows modelling consecutive repetitions of a in a single step, thus reducing the bound n. Rolling has proved to be very effective in numeric planning [Scala *et al.*, 2016; Cardellini *et al.*, 2024]. Similarly to the numeric case, constraints on the state-space must be carefully handled while rolling to assure the plan's validity [Scala *et al.*, 2016].

In this paper, we ask if rolling can be effectively exploited also in classical planning with CEs and constraints. Firstly, we show that planning with CEs remains PSPACE-complete [Nebel, 2000] even in the limit case of problems with a single action, and thus, finding *how many times* the action has to be consecutively applied is a difficult problem on its own. Secondly, we present a PaS encoding in which an action can be consecutively repeated in a single step. Given an action a, we (i) compute its *transition relation* $\mathcal{T}_a$ representing all the states reachable after *one* application of a, (ii) compute its *transitive closure* (TC) relation $\mathcal{T}_a^+$ (see, e.g., [Matsunaga *et al.*, 1993]) representing the states reachable after *at least one* application of a, and (iii) define a PaS encoding Π^+, exploiting a propositional variable a^+ to denote if action a is applied at least once, and $\mathcal{T}_a^+$ to compute the states reachable with a. Given a model for the encoding Π^+ at bound n, we determine the number of times each action a is applied by exploiting the intermediate formulae used to compute $\mathcal{T}_a^+$.

Since deciding the reachability of a state by repetitions of an action with CEs is PSPACE-complete, the computation of the TC $\mathcal{T}_a^+$ can become impractical. To alleviate this burden, we (i) show how constraints can be leveraged to simplify $\mathcal{T}_a^+$, and (ii) construct $\mathcal{T}_a^+$ through *Reduced Ordered Binary Decision Diagrams* (simply BDDs) [Bryant, 1985], which have been widely employed for this purpose in *Model Checking* [Burch *et al.*, 1992; Clarke *et al.*, 1996] due to its often compact representation and canonicity. If the computation of $\mathcal{T}_a^+$ becomes unpractical even then, we can limit the time on the construction of $\mathcal{T}_a^+$ and exploit the intermediate formulae $\mathcal{T}_a^m$ with $m \geq 0$ produced while constructing $\mathcal{T}_a^+$ – in the best case being $\mathcal{T}_a^m = \mathcal{T}_a^+$ and in the worst case being $\mathcal{T}_a^0 = \mathcal{T}_a$ – modelling the states reachable in *up to* 2^m repetitions of a, reducing by a factor of 2^m the bound required to find a plan.

We prove that our approach is correct and complete, and run an experimental analysis on novel domains where a plan must have repetitions of an action. The analysis confirms our theoretical results and, by comparing to the case where rolling is disabled, confirms the benefits of rolling.

2 Complexity of Rolling

In this section, we present our main complexity result for classical planning with CEs.

Theorem 1. *Deciding whether a valid plan exists for a classical planning task with* CE*s and with only one action is* PSPACE*-complete.*

Sketch Proof. Let $\Pi = \langle V, \{a\}, I, G, \top \rangle$. *(Membership)* The problem is in PSPACE because the size of a state is bounded by $|V|$ and we iteratively apply a from I until we reach $s_n \models G$. Since there are at most $2^{|V|}$ possible states, no more than $2^{|V|}$ repetitions of a are required to reach s_n. *(Hardness)* The proof relies on encoding the mechanism of a *Deterministic Turing Machine* (DTM) with bounded tape into a planning task with CEs Π, and that a valid plan for Π exists iff the DTM accepts. □

3 Rolling Actions with Conditional Effects

In this section, we show how to exploit rolling of actions with CEs. Firstly, we describe the concept of *transition relation*, i.e., a logic formula denoting the states reachable with one application of an action. Then, we show how to compute its *transitive closure* (TC), obtaining a logic formula denoting all the states reachable by at least one repetition of an action.

Let $\Pi = \langle V, A, I, G, C \rangle$ be a planning task. The *transition function* of $a \in A$ is a function $T_a : S \times S \mapsto \{\top, \bot\}$ such that, for each $s, s' \in S$, we have $T_a(s, s') = \top$ iff (i) a is applicable in s and (ii) $s' = res(s, a)$. Notice how, for now, we allow $s, s' \not\models C$, which we will disallow later on. As standard for PaS encodings (see, e.g., [Rintanen, 2011]), we model an action's transition function through a *transition relation* $\mathcal{T}_a$.

Let a be an action of Π with transition function T_a. The TC *function* of a is $T_a^+ : S \times S \mapsto \{\top, \bot\}$ such that for each state $s, s' \in S$, $T_a^+(s, s') = \top$ iff (i) a is applicable in s, and (ii) there exists a valid sequence a^k with $k \geq 1$ such that $s' = res(s, a^k)$.

The TC function T_a^+ can be computed iteratively (see, e.g. [Matsunaga *et al.*, 1993]) using the TC *relation* $\mathcal{T}_a^+$, starting from the transition relation $\mathcal{T}_a$. Let $\mathcal{T}_a^i(\mathcal{X}, \mathcal{X}')$ be the *i-th step transition relation of* a, such that $\mathcal{T}_a^0(\mathcal{X}, \mathcal{X}') = \mathcal{T}_a(\mathcal{X}, \mathcal{X}')$ and for each $i \geq 0$,

$$\begin{aligned} \mathcal{T}_a^{i+1}(\mathcal{X}, \mathcal{X}'') = {} & \mathcal{C}(\mathcal{X}) \wedge \mathcal{C}(\mathcal{X}'') \rightarrow \\ & \mathcal{T}_a^i(\mathcal{X}, \mathcal{X}'') \vee \exists \mathcal{X}' \colon \mathcal{T}_a^i(\mathcal{X}, \mathcal{X}') \wedge \mathcal{C}(\mathcal{X}') \wedge \mathcal{T}_a^i(\mathcal{X}', \mathcal{X}''), \end{aligned} \quad (1)$$

where $\mathcal{C}(\mathcal{X})$ is obtained by replacing the variables in C with the ones in $\mathcal{X}$ (and similarly for $\mathcal{X}'$ and $\mathcal{X}''$). The models of $\mathcal{T}_a^i$ represent the states $s, s'' \in S$ such that the *i-th step transition function* $T_a^i(s, s'') = \top$. Notice how $T_a^{i+1}(s, s'')$ is true if either $s \not\models C$ or $s'' \not\models C$. The requirement that $s, s'' \models C$ will be enforced in Sec. 4.

Continuing the computation, by the finiteness of the states, there exists a $p \geq 0$ – called the *fix-point index* of $\mathcal{T}_a$– such that $\mathcal{T}_a^p(\mathcal{X}, \mathcal{X}')$ is logically equivalent to $\mathcal{T}_a^{p+1}(\mathcal{X}, \mathcal{X}')$. By representing each $\mathcal{T}_a^i$ with BDDs, which are canonical representations of propositional formulae, we check logical equivalence of $\mathcal{T}^p$ and $\mathcal{T}^{p+1}$ by assessing if they have the same BDD. The TC relation $\mathcal{T}_a^+$ is thus the *fix-point relation* $\mathcal{T}_a^p$. The function $T_a^i(s, s')$ thus specifies if s' is reachable from s in *at most* 2^i repetitions of a.

As stated in the introduction, the TC computation can become intractable, due to formulae becoming exponentially long, even when employing BDDs. However, the computation of the TC can be stopped at any time, before its fix-point index p, returning the last computed relation $\mathcal{T}_a^m$ with $m \in [0, p]$, which models the states reachable with up to 2^m repetitions of a. We name m the *timeout index*.

4 The Closure-Encoding

Let $\Pi = \langle V, A, I, G, C \rangle$ be a planning task. For each action $a \in A$, let $\mathcal{T}_a(\mathcal{X}, \mathcal{X}')$ be the transition relation of a and let $\mathcal{T}_a^+(\mathcal{X}, \mathcal{X}')$ be its TC relation. We now present the closure encoding Π^+ for Π. As standard for PaS approaches, the sets $\mathcal{X}$, $\mathcal{A}$, $\mathcal{X}'$ represent the *current state*, *action*, and *next state* variables. In $\mathcal{A}$ there is a propositional variable a^+ for each action a in A, denoting if a is repeated at least one time. The current state variables $\mathcal{X}$ are equal to V and the set $\mathcal{X}'$ is a copy of $\mathcal{X}$. The closure-encoding for Π is thus $\Pi^+ = \langle \mathcal{X}, \mathcal{A}, \mathcal{I}(\mathcal{X}), \mathcal{T}^+(\mathcal{X}, \mathcal{A}, \mathcal{X}'), \mathcal{G}(\mathcal{X}) \rangle$ in which $\mathcal{I}(\mathcal{X})$ and $\mathcal{G}(\mathcal{X})$ are the formulas denoting the initial state and the goal condition, respectively, and $\mathcal{T}^+(\mathcal{X}, \mathcal{A}, \mathcal{X}')$ is a *closure symbolic transition relation*, i.e., the conjunction of the union of the following sets:

1. $\text{closure}^+(A, V)$, which contains, for each action $a \in A$,
$$a^+ \rightarrow \mathcal{T}_a^+(\mathcal{X}, \mathcal{X}'),$$
i.e., if a^+ is executed, $\mathcal{X}'$ must be reachable from $\mathcal{X}$ in at least one repetition of a,
2. $\text{frame}^+(V)$, which contains, for each $v \in V$,
$$v' \neq v \rightarrow \Big(\bigvee_{a : v \in \text{add}(a)} a^+ \vee \bigvee_{a : v \in \text{del}(a)} a^+ \Big)$$
i.e, an action must have triggered the change of v,
3. $\text{amo}^+(A)$, where for each $a_1 \neq a_2 \in A$ we have,
$$\neg(a_1^+ \wedge a_2^+)$$
i.e., at each step, at most one action is applied,
4. $\text{constraints}^+(V, C)$, where we have,
$$\mathcal{C}(\mathcal{X}) \wedge \mathcal{C}(\mathcal{X}'),$$
ensuring the respect of C, skipped in Eq. 1.

5 Conclusions and Future Work

We presented a technique to perform rolling in classical planning with CEs and constraints, which was previously unexplored. We theoretically and experimentally demonstrated that the approach is interesting and beneficial. In the future, we plan to (i) minimize the bound even further by integrating our TC approach in the latest SOTA PaS approach [Cardellini *et al.*, 2024] where a finite sequence of actions is employed to apply different actions in the same step and (ii) study how to compute the TC of multiple actions, effectively combining the possible effects of different actions in one transition relation.

References

[Bryant, 1985] Randal E. Bryant. Symbolic manipulation of Boolean functions using a graphical representation. In Hillel Ofek and Lawrence A. O'Neill, editors, *Proceedings of the 22nd ACM/IEEE conference on Design automation, DAC 1985, Las Vegas, Nevada, USA, 1985*, pages 688–694. ACM, 1985.

[Burch *et al.*, 1992] Jerry R. Burch, Edmund M. Clarke, Kenneth L. McMillan, David L. Dill, and L. J. Hwang. Symbolic Model Checking: 10^20 States and Beyond. *Inf. Comput.*, 98(2):142–170, 1992.

[Cardellini *et al.*, 2024] Matteo Cardellini, Enrico Giunchiglia, and Marco Maratea. Symbolic Numeric Planning with Patterns. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 20070–20077. AAAI Press, 2024.

[Clarke *et al.*, 1996] Edmund M. Clarke, Kenneth L. McMillan, Sérgio Vale Aguiar Campos, and Vasiliki Hartonas-Garmhausen. Symbolic Model Checking. In Rajeev Alur and Thomas A. Henzinger, editors, *Computer Aided Verification, 8th International Conference, CAV '96, New Brunswick, NJ, USA, July 31 - August 3, 1996, Proceedings*, volume 1102 of *Lecture Notes in Computer Science*, pages 419–427. Springer, 1996.

[Gazen and Knoblock, 1997] B. Cenk Gazen and Craig A. Knoblock. Combining the Expressivity of UCPOP with the Efficiency of Graphplan. In Sam Steel and Rachid Alami, editors, *Recent Advances in AI Planning, 4th European Conference on Planning, ECP'97, Toulouse, France, September 24-26, 1997, Proceedings*, volume 1348 of *Lecture Notes in Computer Science*, pages 221–233. Springer, 1997.

[Gerevini *et al.*, 2024] Alfonso Emilio Gerevini, Francesco Percassi, and Enrico Scala. An Effective Polynomial Technique for Compiling Conditional Effects Away. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 20104–20112. AAAI Press, 2024.

[Katz, 2019] Michael Katz. Red-Black Heuristics for Planning Tasks with Conditional Effects. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7619–7626. AAAI Press, 2019.

[Kautz and Selman, 1992] Henry A. Kautz and Bart Selman. Planning as Satisfiability. In Bernd Neumann, editor, *10th European Conference on Artificial Intelligence, ECAI 92, Vienna, Austria, August 3-7, 1992. Proceedings*, pages 359–363. John Wiley and Sons, 1992.

[Matsunaga *et al.*, 1993] Yusuke Matsunaga, Patrick C. McGeer, and Robert K. Brayton. On Computing the Transitive Closure of a State Transition Relation. In Alfred E. Dunlop, editor, *Proceedings of the 30th Design Automation Conference. Dallas, Texas, USA, June 14-18, 1993*, pages 260–265. ACM Press, 1993.

[Nebel, 2000] Bernhard Nebel. On the Compilability and Expressive Power of Propositional Planning Formalisms. *J. Artif. Intell. Res.*, 12:271–315, 2000.

[Rintanen, 2011] Jussi Rintanen. Heuristics for Planning with SAT and Expressive Action Definitions. In Fahiem Bacchus, Carmel Domshlak, Stefan Edelkamp, and Malte Helmert, editors, *Proceedings of the 21st International Conference on Automated Planning and Scheduling, ICAPS 2011, Freiburg, Germany June 11-16, 2011*. AAAI, 2011.

[Röger *et al.*, 2014] Gabriele Röger, Florian Pommerening, and Malte Helmert. Optimal Planning in the Presence of Conditional Effects: Extending LM-Cut with Context Splitting. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 765–770. IOS Press, 2014.

[Scala *et al.*, 2016] Enrico Scala, Miquel Ramírez, Patrik Haslum, and Sylvie Thiébaux. Numeric Planning with Disjunctive Global Constraints via SMT. In Amanda Jane Coles, Andrew Coles, Stefan Edelkamp, Daniele Magazzeni, and Scott Sanner, editors, *Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, ICAPS 2016, London, UK, June 12-17, 2016*, pages 276–284. AAAI Press, 2016.