Plan Merging & Plan Reuse as Satisﬁability *

Amol Dattatraya Mali
Dept. of Elect. Engg. & Computer Science, P.O.Box 784
University of Wisconsin, Milwaukee, WI 53201, USA
mali@miller.cs.uwm.edu

Abstract. Planning as satisﬁability has hitherto focused only on purely generative planning. There is an evidence in traditional reﬁnement planning that planning incrementally by reusing or merging plans can be more efﬁcient than planning from scratch (sometimes reuse is not more efﬁcient, but becomes necessary if the cost of abandoning the reusable plan is too high, when users are charged for the planning solution provided). We adapt the satisﬁability paradigm to these scenarios by providing a framework where reusable or mergeable plans can be either contiguous or partially ordered and their actions can be removed and new actions can be added. We report the asymptotic sizes of the propositional encodings for several cases of plan reuse and plan merging. Our empirical evaluation shows that the satisﬁability paradigm can scale up to handle plan reuse and plan merging.

1 Introduction

Impressive results have been obtained by casting planning problems as propositional satisﬁability in [Kautz & Selman 96]. In planning as satisﬁability, a propositional encoding is generated by ﬁxing the number of plan steps (say K). If the number of steps in the solution is more than K, the value of K is increased and an encoding is regenerated. An encoding contains all action sequences of length K and solving it can be viewed as extracting a plan from it. To ensure that each model of an encoding is a valid plan, several constraints about the satisfaction of the pre-conditions of actions and resolution of conﬂicts between the actions are included in the encoding.

This paradigm has become highly popular, as can be seen from the work that followed [Kautz & Selman 96] such as [Ernst et al 97] [Giunchiglia et al 98] [Kautz et al 96]. However this work is limited to purely generative style of planning where each problem is solved from scratch without reusing or merging plans. On the other hand, there is a considerable evidence and argument in the previous literature that using plans can provide improvements in the plan synthesis time [Hanks & Weld 95] [Kambhampati & Hendler 92] [Veloso 94]. Many times, plans have to reused even if there is no efﬁciency gain, because

* I thank Subbarao Kambhampati and the anonymous referees of the ECP-99 for useful comments on the previous draft of this paper. This work was performed when the author was a graduate student at Arizona State University.
the cost of abandoning them is too high (because the users are charged for the planning solution provided). [Britanik & Marefat 95] and [Fouler et al 92] show how plans can be merged by resolving interactions between them and how this kind of plan synthesis can be sometimes faster than planning from scratch.

To examine how these arguments apply to planning as satisfiability, we set up several propositional encodings that contain the constraints from the reusable or mergeable plans. Since there are different forms in which plans can be stored (for further use, to solve a new problem) and there are different ways of encoding them in propositional logic (discussed in [Kautz et al 96]), a variety of ways of casting plan reuse and plan merging as satisfiability exist. These ways are of interest to us because of the different sizes of the encodings that they yield. Our encodings for plan merging and plan reuse are synthesized by adapting the encodings of [Kautz et al 96] to handle these scenarios and we assume some familiarity of the readers with their encodings. We identify the constraints that dominate the asymptotic sizes of the encodings for reuse and merging. Treating the number of actions and fluents as domain constants, we use the power of the variable (number of steps in an encoding) to compare the encoding sizes.

Our work makes the following contributions.

- An automated synthesis of propositional encodings for several cases of plan merging and plan reuse and a report of their asymptotic sizes.
- We show that the size of the state-based encodings which are the shown to be the smallest (have the fewest number of clauses, sum of the clause lengths and number of variables) in the generative planning scenario [Mali & Kambhampati 99], approaches the size of the causal encodings, in the plan merging scenario, when the order preservation restriction (defined in section 3) is enforced.
- We show that the causal encodings for solving planning problems by merging or reusing causal plans are smaller and also faster to solve, than the causal encodings for solving the same problems in a generative style.
- We show that the state-based encodings for merging or reusing contiguous plans are generally neither smaller nor faster to solve, than the state-based encodings that do not reuse or merge plans.
- We show that the causal encodings for merging causal plans can be smaller (and also faster to solve) than the state-based encodings for merging contiguous plans.

In section 2, we explain the notation used for representing the constraints in the plans. In section 3, we describe the semantics of plan merging and show how different cases of merging can be encoded as satisfiability. In section 4, we revisit these cases, but for reusing plans. In section 5, we report empirical results on problems in several benchmark domains and discuss the insights obtained from them. We present conclusions in section 6.

\[1\] Available at http://www.cs.yale.edu/HTML/YALE/CS/HyPlans/ mcdermott.html