

The Variety of Partial Plans

Jane Yung-jen Hsu*

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan 106, R.O.C.
yjhsu@csie.ntu.edu.tw

Abstract

The notion of *partial plans* has existed ever since the early days of planning research. However, plans can be partial in many different ways, and for many different reasons. This paper presents a characterization of partial plans and discusses the use of partial plans in incremental planning under computational and informational constraints. A heuristic for guiding the generation of plausible partial plans is proposed.

1 Introduction

Traditional planning systems assume the availability of perfect information and unlimited computational resources, so that once a plan is generated by the planner, it can be carried out successfully by the plan executor. Over the years, the problems arising from such an idealistic view of the world have been widely recognized among researchers in the planning community [22]. Besides the various formalizations of classical planning systems [16; 3; 4], many alternative approaches to dealing with complex planning domains have been proposed [12]. In particular, *incremental planning* provides a promising solution when resources are inadequate, i.e. generating a preliminary partial plan quickly, and then refining the partial plan when new information or time becomes available.

A wide variety of partial plans have been employed by different planning systems. Plans can be partial in that they can be *incomplete* [19; 13], *abstract* [20; 23], or *approximate* [8; 11; 15]. Furthermore, plans containing conditional actions, suboptimal steps, or information-gathering actions can be considered to be partial as well [18]. This paper attempts to establish the common criteria for comparing different partial plan formalisms. In Section 2, we first present the formal definitions of partial plans in different plan representations (e.g. totally ordered, partially ordered, and situation-action plans). Section 3 outlines the different dimensions that can be used to characterize partial plans. A heuristic for guiding the search in the space of partial plans during incremental planning is proposed in Section 4.

*This research was supported in part by a grant from the National Science Council of ROC under grant number NSC 82-0422-E-002-046

2 Definitions

Plan representation is a fundamental element of any planning system. It determines the space of possible (partial) plans for the planner. Although planning can generally be carried out in the state space, for a more coherent treatment of the various planning strategies, we consider planning as search in the *plan space*, where nodes denote possible plans and arcs denote plan modification steps. In what follows, we define the notions of plans, solutions to a planning problem, and plausible partial plans for three different plan representations.

Planning Problems and Plans Planning is the process of finding sequences of operators for achieving certain goal conditions from the initial state(s). Given a first-order language L , a state description is a collection of formulas that are partial descriptions of the state of the world. For simplicity, we restrict our attention to state descriptions that are finite, consistent sets of ground literals. STRIPS-style operators are used for describing actions, which are partial functions mapping states into states. A planning problem is defined by a pair of state descriptions $\langle I, G \rangle$, where I is the initial state and G is the goal state. Following the convention in planning literature, a planning problem can be represented by a two-step initial plan, in which the first step adds the propositions true in the initial world, while the final step has the goal conditions as its preconditions.

A typical plan consists of an ordered set of operators that transforms the initial state into a final state satisfying the goal conditions. A *totally ordered plan* (or simply a *TOP*) is a linear sequence of operators, in which every operator is ordered with respect to every other operator.

Definition 1 A TOP $\Pi = \langle \alpha_1, \dots, \alpha_n \rangle$ is a solution to the planning problem $\langle I, G \rangle$, if the sequence of state descriptions S_0, \dots, S_n generated by applying Π to I satisfies two conditions¹:

- (1) $\text{Pre}(\alpha_i) \subseteq S_{i-1}$, and
- (2) $G \subseteq S_n$.

In other words, the plan is correct if every operator in the sequence is applicable in the state resulting from applying the preceding operators to the initial state, and the goal conditions are satisfied in the final state.

¹This definition is from [1].

A *partially ordered plan* (or a *POP*), on the other hand, may contain operators that are unordered with respect to each other. The extreme case is called an *unordered plan*, where all operators (other than the first and final operators in the initial two-step plan) are unordered with respect to one another. A *linearization* of a partially ordered plan is a total order over the plan's set of operators consistent with the existing partial order.

Definition 2 A *POP* Π is a solution to a planning problem $\langle I, G \rangle$, if every linearization of Π is a solution to $\langle I, G \rangle$.

To cope with incomplete information concerning the effects of actions, a reactive planning system typically determines its appropriate next action based on the currently perceived situation [21; 10; 5]. A *situation-action plan* (or a *SAP*) is an unordered plan consisting of a collection of situation-action pairs. Each situation dictates the action that should be applied. Unlike classical planners, reactive planners do not assume the intended effects of operators to always hold. It follows that an operator can no longer be considered as a partial function from states into states. Instead, an operator corresponds to a partial *nondeterministic* function from states into states. In other words, the world can be in any one of a set of states after an operator is applied in the current state. We can extend the definition of linearization and define the *nondeterministic linearization* of a SAP Π to be a total order of a finite *multi-set* chosen from the operators in Π so that their preconditions match the sequence of state descriptions generated by nondeterministically applying Π to the initial state.

Definition 3 A *SAP* Π is a solution to a problem $\langle I, G \rangle$ if for any given sequence of state transitions, there exists some *nondeterministic linearization* of Π that is a solution to $\langle I, G \rangle$.

This definition is bit strong in that it requires a correct SAP solution to successfully achieve the goal conditions regardless of the dynamic changes in the world. In general, such a solution exists if and only if the goal state is reachable from every state of the world.

Partial Plans Intuitively, a plan is partial if it does not specify a complete solution for the given planning problem, and it has the potential of becoming a solution. Any plan can be complete for one problem but partial relative to another. Each plan can be transformed by plan modification steps as defined in its corresponding plan space. TOP transformations typically add operators into various points of the partial plan. At any point, the operators in a partial TOP are totally ordered with respect to each other. POP transformations are used to add additional operators or to order operators in the existing partial plan. A partial POP may be missing both operators and orderings on existing operators. In both cases, the planner may also backtrack and remove operators or orderings

from the partial plans². Similarly, SAP transformations add situation-dependent operators into the partial plan. The situation associated with each operator is assumed to be disjoint from each other. A complete SAP contains situations covering every possible state of the world.

A *completion* of a partial plan Π can be defined to be the plan resulting from applying a finite sequence of plan transformation steps to Π .

Definition 4 A partial plan Π is plausible with respect to the given planning problem $\langle I, G \rangle$ if and only if there exists a completion of Π such that it is a solution to $\langle I, G \rangle$.

In other words, there exists a finite sequence of plan transformation steps that produces a solution to the problem. A plausible partial plan possibly but not necessarily solves the given planning problem since some of its completions are not solutions. If a partial plan has no potential of becoming a solution, it should be discarded as early in the planning stage as possible. Unfortunately, it is computationally infeasible to check for every completion of a partial plan. However, one can compare partial plans based on the number of plan transformation steps required for a partial plan to become a solution, and the number of potential completions that are solutions to the given problem. In Section 4, we will define a heuristic for generating plausible partial plans.

3 Characterizations of Partial Plans

In addition to the differences due to plan representation as defined in the previous section, this paper outlines several distinguishing features characterizing partial planning systems.

Sources of Incompleteness Partial plans are useful for a variety of reasons. For example, a planning system may have imperfect knowledge about the initial state or the effects of actions. Given such incomplete information, it may be impossible to construct a complete solution to a problem. In addition, even with complete information, searching for a solution to a planning problem may take an enormous amount of time. When planning has to be carried out on-line, the search for a solution may have to be terminated before it is completed.

- *Computational considerations* — Least commitment planning using partially ordered plans [19] is an early example of partial planning to maximize execution flexibility. As was pointed out in [17], it turns out that partial-order planning also has a significant advantage in terms of its planning efficiency. On the other hand, abstraction planning

²When working with abstract or approximate partial plans, plan expansion and revision are considered as valid transformations as well.

was originally motivated by the desire to improve planning efficiency. Using a hierarchy of abstraction spaces, a planner can explore the abstract space quickly to find a skeleton for the solution, and gradually fills in the details of the final plan. To save time, it is also possible to generate an imprecise or approximate partial plan without going through the entire search space, and then to refine and modify the partial plan at a later time.

- *Informational considerations* — Due to uncertainties about its environment and actions, a planner may have to avoid committing prematurely to certain operator selections or orderings. The least-commitment strategy can be easily extended to deal with informational constraints so that the planner never commits to any operator or its orderings unless such a decision is dictated by the information at hand [13]. Similarly, when the goals for a planner are not completely known, it is impossible to generate a complete plan ahead of time. Furthermore, a planner's goals can be dynamically specified, e.g. diagnostic vs. therapeutic goals [18], or information goals and sensory actions [9]. Combining information gathering actions with task achieving actions is necessary for planning in many real world domains.

Types of Incompleteness Partial plans can be classified into three major categories:

- *Incomplete* — i.e. plans that carry out a subset of the operations necessary for achieving the goal. Such plans usually only solve a subset of the goal conditions. For example, the intermediate solutions produced during means-ends analysis or nonlinear planning [19]. Plans that prescribe a set of possible solutions based on current information are also incomplete [13].
- *Abstract* — i.e. plans that solve the problem in a high, non-operational level of abstraction. Virtually all abstraction planners [20; 23; 14] fall into this category. Bacchus and Yang [1] further showed that for abstraction hierarchies satisfying the *downward refinement property* (DRP), if a concrete level solution exists, then any abstract solution can be refined to a concrete solution without backtracking across abstraction levels.
- *Approximate* — i.e. plans that approximate a solution to the problem based on default assumptions or incomplete inference mechanisms. For example, Elkan's planner [8] generates approximate plans by allowing negated subgoals to be assumed, and employs an iterative deepening search for the negation-as-failure proofs. Ginsberg [11] uses functional truth values based on a *bilattice*, and nonmonotonically updates the truth values of the fluents and thus modifying the current plan. Korf's real-time A* algorithm performs a partial search for a solution plan [15].

Plan Generation Finally, we consider the various approaches in generating and modifying partial plans. We will focus on partial plans that are readily executable or close to a complete solution in the following discussions.

- *Two-stage planning* — planning is divided into a preliminary stage that creates a skeletal or an approximate plan, as well as a refinement stage in which the preliminary plan is gradually improved. For example, the progressive horizon planning proposed in [18] uses a quick-and-dirty process to construct a plan sketch, and applies optimization transformations to operators within the planner's horizon. Elkan's system is another such example.
- *Anytime planning* — planning is viewed as a continuous process in modifying the existing plan, and the quality of results improves with time. For example, Boddy [2] uses dynamic programming to construct the plan steps iteratively to solve a progressive subset of the goals. At any time during the planning process, the plan consists of an optimal plan for solving an intermediate goal from the initial state, and a randomly generated plan for getting from the intermediate goal to the final state. Zilberstein and Russell [24] combine anytime algorithms and an off-line compilation process. Drummond and Bresina [5; 6] uses anytime synthetic projection to generate situated control rules incrementally. The partial planning system in [13] maintains a partial plan for quick reactive behavior, and incrementally incorporates action prescriptions into the plan.

4 Incremental Planning Strategies

Incremental planning is a viable approach to planning in complex domains. The idea is to refine or revise the existing partial plan so that it becomes increasingly complete, concrete, and accurate. Such a transformation is called a *monotonic refinement*. However, search in the space of possible partial plans is notoriously intractable. It is therefore necessary to adopt some heuristics in searching for a solution to an arbitrary planning problem. Search control heuristics such as temporal coherence for goal ordering defined in [7] can significantly constrain the search. In addition, we extend the DRP defined in [1] as follows:

Definition 5 (Monotonic Completion Property)

An incremental planning strategy is said to have the monotonic completion property if for every partial plan Π , any transformation of Π based on the strategy produces a monotonic refinement of Π .

The UA planning algorithm described in [17] and the partial planning algorithm presented in [13] as well as several other anytime planning systems satisfy this property.

5 Conclusions

We have presented the formal definitions as well as some informal characterizations of partial plans. The next step is to perform more rigorous analyses of the computational advantages of partial plans for incremental planning. In particular, we hope to identify sufficient and checkable syntactic conditions that can be used for checking if an incremental planning strategy satisfies the monotonic completion property.

References

- [1] F. Bacchus and Q. Yang. The downward refinement property. In *Proceedings of the 12th IJCAI*, pages 286–292, 1991.
- [2] M. Boddy. Anytime problem solving using dynamic programming. In *Proceedings of AAAI-91*, pages 738–743, 1991.
- [3] D. Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32(3):333–377, 1987.
- [4] J. Christensen and A. Grove. A formal model for classical planning. In *Proceedings of the 12th IJCAI*, pages 246–251, 1991.
- [5] M. Drummond. Situated control rules. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, May 1989.
- [6] M. Drummond and J. Bresina. Anytime synthetic projection: maximizing the probability of goal satisfaction. In *Proceedings of the AAAI-90*, pages 138–144, 1990.
- [7] M. Drummond and K. Currie. Goal ordering in partially ordered plans. In *Proceedings of the Eleventh IJCAI*, pages 960–965, August 1989.
- [8] C. Elkan. Incremental, approximate planning. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 145–150, 1990.
- [9] O. Etzioni, S. Hanks, D. Weld, D. Draper, N. Lesh, and M. Williamson. An approach to planning with incomplete information. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, October 1992.
- [10] R. Firby. An investigation into reactive planning in complex domains. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 202–206, July 1987.
- [11] M. L. Ginsberg. Computational considerations in reasoning about action. In J. A. Allen, R. Fikes, and E. Sandewell, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pages 250–261, Morgan Kaufmann, San Mateo, CA, April 1991.
- [12] *Planning in Uncertain, Unpredictable, or Changing Environments (Working Notes of the 1990 AAAI Spring Symposium)*. SRC TR 90-45, Systems Research Center, University of Maryland, 1990.
- [13] J. Y.-j. Hsu. Partial planning with incomplete information. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence '90*, pages 474–479, Nagoya, Japan, November 1990.
- [14] C. A. Knoblock, J. D. Tenenber, and Q. Yang. Characterizing abstraction hierarchies for planning. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 692–697, AAAI, 1991.
- [15] R. E. Korf. Real-time search for dynamic planning. In *Proceedings of the AAAI Spring Symposium Series on Planning in Uncertain, Unpredictable, or Changing Environments*, American Association for Artificial Intelligence, March 1990.
- [16] V. Lifschitz. On the semantics of strips. In M. P. Georgeff and A. L. Lansky, editors, *Reasoning about Actions and Plans: Proceedings of the 1986 Workshop*, pages 1–9, Morgan Kaufmann Publishers, Los Altos, CA, 1987.
- [17] S. Minton, J. Bresina, and M. Drummond. Commitment strategies in planning: a comparative analysis. In *Proceedings of the 12th IJCAI*, pages 259–265, IJCAI, 1991.
- [18] R. Rymon, B. L. Webber, and J. R. Clarke. Progressive horizon planning – planning exploratory-corrective behavior. *IEEE Transactions on Systems, Man, and Cybernetics*, Special issue on Planning, Scheduling, and Control, 1992.
- [19] E. D. Sacerdoti. The non-linear nature of plans. In *Proceedings of the Fourth IJCAI*, pages 206–214, IJCAI, 1975.
- [20] E. D. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5(2):115–135, 1974.
- [21] M. J. Schoppers. Universal plans for reactive robots in unpredictable domains. In *Proceedings of the Tenth IJCAI*, pages 852–859, 1987.
- [22] A. Tate, J. Hendler, and M. Drummond. A review of ai planning techniques. In J. Allen, J. Hendler, and A. Tate, editors, *Readings in Planning*, pages 26–49, Morgan Kaufmann, 1990.
- [23] Q. Yang and J. D. Tenenber. Abtweak: abstracting a nonlinear, least commitment planner. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 204–209, American Association for Artificial Intelligence, 1990.
- [24] S. Zilberstein and S. J. Russell. Efficient resource-bounded reasoning in AT-RALPH. In *Proceedings of the First International Conference on Artificial Intelligence Planning Systems*, pages 260–266, 1992.