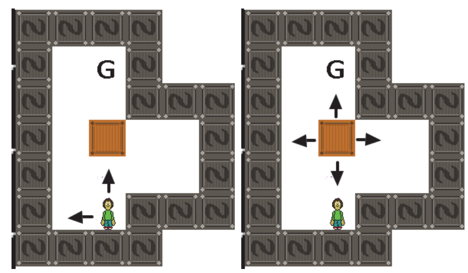
Automatically Extracting Axioms in Classical Planning (Extended Abstract)

Shuwa Miura and **Alex Fukunaga**
Department of General Systems Studies
Graduate School of Arts and Sciences
The University of Tokyo
miura-shuwa@g.ecc.u-tokyo.ac.jp, fukunaga@idea.c.u-tokyo.ac.jp

Abstract

Axioms can be used to model derived predicates in domain-independent planning models. Formulating models which use axioms can sometimes result in problems with much smaller search spaces than the original model. We propose a method for automatically extracting a particular class of axioms from standard STRIPS PDDL models. More specifically, we identify operators whose effects become irrelevant given some other operator, and generate axioms that capture this relationship. We show that this algorithm can be used to successfully extract axioms from standard IPC benchmark instances, and show that the extracted axioms can be used to significantly improve the performance of an IP-based planner.



(a) Sokoban (b) Sokoban with Axioms

Figure 1: The search space of the Sokoban domain

1 Introduction

Previous work on derived predicates and axioms for planning has focused on the advantages of expressivity (compactness) of domain modeling using axioms, (Thiébaux, Hoffmann, and Nebel 2005), as well as search algorithms which are aware of axioms (Ivankovic and Haslum 2015b). Previous work has shown that axioms are a type of structure which can be exploited by a search algorithm to solve problems more efficiently, compared to a version of the problem without explicit axioms.

Consider the single-agent puzzle Sokoban, in which the player pushes stones around in a maze. The goal is to push all the stones to their destinations. The standard PDDL formulation of Sokoban used in the International Planning Competition(IPC) consists of two kinds of operators, push and move. push lets the player push a box in one direction, while move moves the player into an unoccupied location.

Ivankovic and Haslum (2015a) proposed a new formulation of Sokoban with axioms and showed that this leads to a problem with a smaller search space (Ivankovic and Haslum 2015b). They remove the move operators entirely, and introduce axioms to check whether the player can reach a box to push it. The reformulated push operators now have a derived predicate reachable(loc) instead of at-player=loc as their precondition. The values of the derived predicates are determined by the following axioms:

1. reachable(loc) $\leftarrow$ at-player=loc
2. reachable(loc) $\leftarrow$ reachable(from), clear(loc), adj(from,loc)

Intuitively, the first axiom means that the current location of the player is reachable. The second axiom means a location next to a reachable location is also reachable. Figure1 illustrates the search space with and without axioms.

While previous work focused on axioms as expressive domain modelling constructs used by human domain modellers, this abstract shows that axioms can also be viewed as structure to be discovered and exploited. We investigate a *completely automated*, reformulation approach which extracts axioms from standard PDDL domains, solves the reformulated problem instance with an axiom-aware planner, and then converts the solution to the axiom-enhanced problem back into a valid plan for the original problem instance.

We propose methods for automatically extracting two classes of axioms from standard (axiom-free) SAS+(STRIPS) planning models. Our first method partitions the operators in a SAS+ model into *observable operators*, which are in some sense "fundamental" operators, and τ*-operators*, which are "auxiliary" operators. Given a list of observable operators to execute, a set of τ-operators to be executed is implied. For example, in Sokoban, push are observable operators while move are τ-operators, since once a push operator is applied, there must have been a way to reach the location using move operators. We reformulate the problem such that the preconditions of the observable operators are derived predicates which are established by τ-operators, and τ*-axioms* representing how these derived predicates are implied are added to the domain.

Our second method extracts axioms based on groups of fluents where exactly one of the fluents must be true at all times. We generate *ϵ-axioms* which describe the truth value of one member of such a "exactly-1" group in terms of the other members of the group. For example, in Sokoban, each location can be occupied by at most one object, so {at-player=loc,stone1=loc,...,stonen=loc,clear(loc) } is an ϵ-group. We can derive an axiom [clear(loc) $\leftarrow$ not at-player=loc, not stone1=loc,...,not stonen=loc] which determines when loc is vacant.

We show that, using our methods, axioms can be extracted from a number of standard IPC benchmark problems. We then show that the axioms extracted from IPC standard benchmarks can then be exploited to speed up an IP-based planner.

2 Evaluation of Axiom Extraction

We applied the τ-axiom and ϵ-axiom extraction algorithms to all IPC domains (IPC1998-IPC2014), *excluding* domains containing conditional effects, and domains with preexisting derived predicates. Out of duplicate sets of domains (e.g., pegsol-opt11, sat11, pegsol-opt08, pegsol-sat08), we only include one (in this case, pegsol-opt11). This results in a set of 1442 instances from 44 domains on which we executed axiom extraction (5min., 2GB /instance). Overall, τ-axioms or ϵ-axioms were found in 22/44 domains, so these axioms are relevant to a substantial fraction of IPC domains. In the grid domain, where the player walks around a maze to retrieve the key to the goal, we identified the player movements as τ-operators. miconic is a elevator domain where we must transport a set of passengers from their start floors to their destination floors using a single elevator. Operators for moving up/down the elevator were identified as τ-operators.

3 Improving Planner Performance Using Extracted Axioms

The main contributions of this research are the novel methods for extracting τ-operators and ϵ-axioms described in the previous sections. However, having extracted the axioms, a natural question is whether the axioms can be successfully exploited to improve planner performance.

We investigate our reformulation approach by integrating the axioms extracted using the techniques proposed above into an integer programming (IP) based planner. Our baseline IP-based planner, which we call IPlan, is based on the state change variable model, where the variables represent changes in state values (Vossen et al. 1999). The state change variable model was the basis of Optiplan (van den Briel and Kambhampati 2005), which incorporated variable elimination based on a relaxed planning graph. Our baseline IP-based planner improves upon Optiplan, by some additional, straightforward mutex constraints on the values of SAS+ variables.

Axioms can be integrated into an IP-based model (or, similarly, a constraint programming model), as follows: Given an axiom of the form $a \leftarrow b_1, ..., b_m, \text{not} c_1, ..., \text{not} c_m$, the corresponding normal logic program (NLP) P_t for time step t, is the rule:

$$x^{\text{sat}}_{a,t} \leftarrow x^{\text{sat}}_{b_1,t}, ..., x^{\text{sat}}_{b_n,t}, \text{not} x^{\text{sat}}_{c_1,t}, ..., \text{not} x^{\text{sat}}_{c_m,t} \quad (1)$$

where $x^{\text{sat}}_{f,t}$ is an auxiliary boolean variable which denotes whether f is true at step t. The models for P_t correspond to the truth values for the derived variables. Each NLP P_t is then translated to a integer program (IP) using the method by Liu, Janhunen, and Niemelä (2012), and these linear constraints are added to the IPlan model.

In an axiom-enhanced encoding, a single action can trigger many axioms (implied actions), so it is possible that solutions to axiom-enhanced models can require fewer steps than a model without axioms.

We compared the following: (1) IPlan, (2) IPlan$(\tau)^+$ (τ-axiom + model selection), and (3) IPlan(τ, ϵ) (τ-axiom + ϵ-axiom + model selection). The *model selection* policy used in IPlan(τ) and IPlan$(\tau, \epsilon)^+$ chooses the model with axioms, if τ-axioms are discovered; otherwise, the standard IPlan model is used.

Domain (5min, 2GB)	IPlan	IPlan$(\tau)^+$	IPlan$(\tau, \epsilon)^+$
all(732)	244	305	**313**

Table 1: Coverage (# solved) for IPlan.

The IP models generated are solved using Gurobi Optimizer 6.5.0, with 5 minute runtime (single-threaded) and 2GB of RAM per problem. The runtime includes all phases of IPlan with axioms, including the PDDL to SAS+ conversion, axiom extraction, solving the model, and axiom decoding. We evaluate all 22 IPC benchmark domains (732 instances total) for which either τ-operators, ϵ-axioms, or both were extracted in *at least one instance*.

The results are shown in Table 1. IPlan$(\tau, \epsilon)^+$ achieves significantly higher coverage than the baseline model (IPlan). Improvements due to axioms are particularly apparent on gripper, miconic, pegsol and Sokoban, where many τ-axioms are discovered.

References

Ivankovic, F., and Haslum, P. 2015a. Code supplement for "optimal planning with axioms". IJCAI2015.

Ivankovic, F., and Haslum, P. 2015b. Optimal planning with axioms. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 1580–1586.

Liu, G.; Janhunen, T.; and Niemelä, I. 2012. Answer set programming via mixed integer programming. *Proc. Int. Conf. on Principles of Knowledge Representation and Reasoning* 32–42.

Thiébaux, S.; Hoffmann, J.; and Nebel, B. 2005. In defense of PDDL axioms. *J. Artif. Intell. Res.(JAIR)* 168(1):38–69.

van den Briel, M., and Kambhampati, S. 2005. Optiplan: Unifying ip-based and graph-based planning. *J. Artif. Intell. Res.(JAIR)* 24:919–931.

Vossen, T.; Ball, M. O.; Lotem, A.; and Nau, D. 1999. On the use of integer programming models in AI planning. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 304–309.

More info: (https://github.com/dosydon/aaai17-student).