Translating PDDL Tasks into Multi-valued Conformant Planning Tasks

Jingjing Zhao
College of Computer Science and Technology,
Jilin University, Changchun, China, 130012
Key Laboratory of Symbolic Computation
and Knowledge Engineering of Ministry
of Education, Jilin University,
Changchun, China, 130012
College of Business Administration,
Changchun University of Technology
jing.jing.zhao@163.com

Jigui Sun and Minghao Yin
College of Computer Science and Technology,
Jilin University, Changchun, China, 130012
Key Laboratory of Symbolic Computation
and Knowledge Engineering of Ministry
of Education, Jilin University,
Changchun, China, 130012
jgsun@jlu.edu.cn
mhyin@nenu.edu.cn

Abstract

In this paper, we formally define the problem of conformant planning with multi-valued state variables, namely multi-valued conformant planning task (MCPT), and present an algorithm for transforming conformant planning problems specified in planning domain definition language (PDDL) into MCPTs. The algorithm can deal with the uncertainties of initial conditions and disjunctive goals according to two important relaxed conditions.

1. Introduction

Planning domain definition language (PDDL) is a standard for planning description languages in international planning competition (IPC) [1, 2, 3, 4]. Multi-valued planning task (MPT) is a concise state description for single state or symbolic exploration of classical planning task. It is well-known that MPT can minimize the state description length and compress state space during planning [5]. Translating PDDL tasks into MPTs is also a crucial preprocessing procedure for planning based on causal graphs and domain transition graphs [6]. Nevertheless, MPT transform algorithm is only fit for classical planning without uncertainties, which can not satisfy the requirements of recent researches in artificial intelligence planning [7]. For the purpose of settling uncertainties, conformant planning is proposed [8, 9]. Conformant planning requires an algorithm to transform conformant planning problem specified in PDDL into a fully instantiated multi-valued representation based on the SAS+ formalism [10, 11]. In order to introduce MPT to conformant planning, we give the definition of a fully instantiated multi-valued representation, which is called multi-valued conformant planning task (MCPT); and propose a MCPT transform algorithm, which transforms conformant planning tasks described in PDDL into MCPTs. The transform algorithm extends MPT transform method [5, 6] to permit initial conditions' uncertainties and disjunctive goals according to two important relaxed conditions. One condition is that the reachable atom set obtained from initial belief state is relaxed to be the disjunctions of reachable atom sets obtained from initial world states. The other condition is that the invariants synthesized from certain initial world state can substitute for the invariants synthesized from initial belief state.

2. Multi-valued conformant planning tasks

In this section, we formally introduce the problem of conformant planning with multi-valued state variables. The PDDL tasks we use as input include all purely logical aspects of the language, excluding numerical, temporal planning problems and strong or soft constraints. Our MCPT transform algorithm permits disjunctions, conjunctions and negative atoms in initial conditions and goals. The formal definition of MCPT is given below.

Definition 1 Multi-valued conformant planning tasks (MCPTs): A multi-valued conformant planning task (MCPT) is given by a 5-tuple $\Pi = <\mathcal{V}, \mathcal{I}, \mathcal{G}, \mathcal{A}, \mathcal{O}>$ with the following components.

$\mathcal{V}$ is a finite set of state variables, each with an associated finite domain D_v. There are two kinds of state variables: fluents (affected by operators) and derived variables (computed by evaluating axioms). The domains of derived variables must contain the undefined value $\perp$. A variable

Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)
0-7695-2874-0/07 $25.00 © 2007 IEEE

assignment is a function s on some subset of $\mathcal{V}$ such that $s(v) \in D_v$ wherever $s(v)$ is defined. In the context of variable assignments, we write $v = d$ for the variable-value pairing $<v, d>$ or $v \rightarrow d$.

$\mathcal{I}$ is a set of completely specified possible initial worlds, called initial belief state. Each world is composed of the conjunctions of variable assignments over $\mathcal{V}$.

$\mathcal{G}$ is a DNF formula substantially. We denote the disjunctive relationships by a finite set. Thus, $\mathcal{G}$ is a set of goals. Each goal is composed of the conjunctions of variable assignments over $\mathcal{V}$.

$\mathcal{A}$ is a finite set of axioms over $\mathcal{V}$. Axioms are triples of the form $<cond, v, d>$, where $cond$ is composed of the conjunctions of variable assignments called the condition or body of the axiom, v is a derived variable called the affected variable and $d \in D_v$ is called the derived value for v. The pair $<v, d>$ can be written as $v := d$, called the head of the axiom.

$\mathcal{O}$ is a finite set of operators over $\mathcal{V}$. An operator can be written as $<pre, \mathit{eff}>$ where pre is composed of the conjunctions of variable assignments over $\mathcal{V}$ called its precondition, and *eff* is a finite set of effects. Effects are triples $<cond, v, d>$, where $cond$ is composed of the (possibly empty) conjunctions of variable assignments called the effect condition, v is a fluent called the affected variable and $d \in D_v$ is called the new value for v.

Definition 2 MCPT state spaces: The state space of a MCPT $\Pi = <\mathcal{V}, \mathcal{I}, \mathcal{G}, \mathcal{A}, \mathcal{O}>$, denoted as $S(\Pi)$, is a directed graph. Its vertex set is the set of states of $\mathcal{V}$. Let s be a state of a MCPT. $S(\Pi)$ contains an arc (s, s') iff there exists some operator $<pre, \mathit{eff}> \in \mathcal{O}$ or some axiom $<cond, v, d>$ such that: $pre \subseteq s$; $s'(v) = d$ for all effects $<cond, v, d>$ or axiom $<cond, v, d>$ such that $cond \subseteq s$; and $s'(v) = s(v)$ for all other fluents.

Definition 3 MCPT planning: Given a MCPT Π with initial belief state $\mathcal{I}$ and goal set $\mathcal{G}$, MCPT planning computes the paths in directed graph of $S(\Pi)$, or proves that $\mathcal{G}$ can not be reached. The paths which all encode the same action sequence, should reach one of goals in $\mathcal{G}$ respectively, regardless of which initial world we start from and which action effects occur.

3. Transform algorithm

Translation is performed in a sequence of transformation steps. The translation process is outlined in Figure 1. Dashed line box denotes the input file or the output file while solid line box denotes the process step in Figure 1. Starting from a PDDL specification, we apply some well-known logical equivalences to compile away types, simplify conditions and effects in the normalization step [6]. Secondly, we parse the initial conditions and the goals specified in CNF into DNF. Thirdly, the invariant synthesis step computes mutual exclusion relations between atoms, which will be used for synthesizing the MCPT variables. The grounding step performs a relaxed reachability analysis to compute the set of ground atoms, axioms and operators that are considered relevant for the conformant planning task; and computes a grounded PDDL representation. Since the conformant planning task introduces the uncertainties in initial conditions, the techniques for computing reachable atom sets are different from MPT's. Moreover, invariant synthesis becomes more complex. Finally, the MCPT generation step chooses the final set of state variables using the information obtained from invariant synthesis step and grounding step; and produces the MCPT output. We will discuss the key transform steps: invariant synthesis, grounding and MCPT generation.

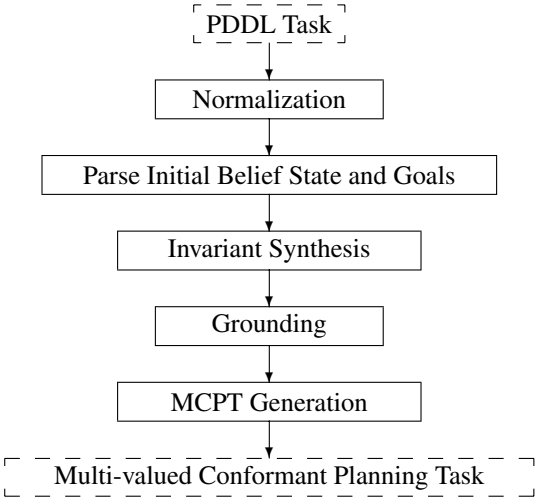

Figure 1. General architecture of the transform algorithm

3.1. Invariant synthesis

The invariant synthesis step computes mutex invariants which indicate that certain propositions can't be true at the same time. A set of propositions which are pairwise mutually exclusive can be easily encoded as a single state variable. There are a variety of machineries for computing invariants [6, 12, 13, 14, 15] which are presented in classical planning. The most direct idea is to exploit invariant information only presented in domain file if we want to introduce invariant synthesis into conformant planning. However, this eliminates mutex conditions which can not be established without checking the initial conditions. We propose a new algorithm to solve this problem.

Initial conditions defined in PDDL file are specified by

CNF with uncertainties in the form of disjunctions. We have parsed initial conditions from CNF into DNF in the previous procedure. That is to say: initial conditions with uncertainties are changed into a set of completely specified possible world states. Approximately, we take the first world state as initial conditions to compute invariants through invariant synthesis techniques adopted by MPT. If it fails to solve the problem, the second world state will be chose. We repeat this process until invariants obtained from some world state can solve the problem.

3.2. Grounding

MCPT's initial conditions, goals, axioms and operators are grounded representations, so grounding process is necessary. In this part, we want to generate a grounded representation of the normalized PDDL task. However, it would be wasteful to instantiate operators or axioms in such a way that their preconditions or bodies are inevitably false in each reachable state. Determining whether or not a given atom can ever be true is difficult. But the idea of relaxed planning in HSP and FF [16, 17], can be adopted to compute the set of reachable atoms. We compute the reachable atoms of a relaxed conformant planning task $R(\Pi)$ instead of computing the reachable atoms of a conformant planning task Π. All delete effects of operators are ignored in relaxed conformant planning. Negative literals in axiom bodies, operator preconditions, effect conditions and goal conditions are assumed to be always true. Each initial world is treated as initial condition to compute its set of reachable atoms by means of horn exploration algorithm. The set of reachable atoms of relaxed conformant planning task is composed of the disjunctions of reachable atom sets of all initial worlds.

It is easy to know that the set of reachable atoms of $R(\Pi)$ is a superset of the set of reachable atoms of Π. Computing reachable atoms is conceptually simple. Nevertheless, the set of reachable atoms can be huge in some benchmark domains such as blocksworld, coins and comm. Therefore, we choose horn exploration algorithm as instantiating algorithm for computing reachable atoms efficiently.

3.3. MCPT generation

In this process, we transform PDDL task into MCPT with the invariants synthesized earlier, the grounded PDDL task obtained from previous stage, disjunctive initial belief state and disjunctive goals. We firstly define variables and variable domains. Then we convert initial belief state, goals, axioms and operators into multi-valued representations.

Firstly, let's review MPT generation procedure. Each variable of MPT corresponds to one or more atoms of PDDL task. In order to represent as many ground atoms by a single state variable as possible, MPT generation starts with computing mutex groups by instantiating invariants in all possible ways and checking for pairwise mutual exclusions in initial state. Next, greedy algorithm [6] iteratively converts mutex groups into multi-valued variables. The suitable variables and variable domains serve as the foundation for the rest converting. For each atom p in initial state, which is the instance of modifiable fluent predicate or derived predicate, MPT variable $var(p)$ is set p. MPT variables v for which there is no initial state atom p with $var(p) = p$, are initialized to $\perp$. Positive literal p is also translated into $var(p) = p$ not only in operator effects but also in conditions. Converting delete effects or negative literals is more complicated. It is not correct to set $var(p)$ to $\perp$ when another effect of the same operator triggers simultaneously and adds another atom represented by the same variable. Or it would be the case that p was not true when the operator was applied, but some other atom represented by $var(p)$ was. Converting negative literals is complex for grounded conditions which appear in goals, operator preconditions, effect conditions and axiom bodies. Considering negative literal $\neg p$, if the condition also contains some positive literal concerning variable v which is the variable for encoding p, then there is no need to encode $\neg p$ at all, because it is implied by the other literal. However, MPT lacks simple way to represent $\neg p$ as a condition directly.

MCPT generation process is similar to MPT generation procedure. The differences exist in trivial negative literal converting. Because we can not determine whether some atoms are true or not in the states of conformant planning, known true atoms and known false atoms are important for computing conformant plan [18]. Nevertheless, classical planning is based upon closed world assumption: the atoms which do not exist in the state are false. So it is unnecessary to represent negative literals directly in classical planning. On the contrast, conformant planning requires considering negative literals. We adopt a new derived variable $not{-}p$ with domain $\{\top,\perp\}$ and generate an axiom $(v = d) \rightarrow (not{-}p := \top)$ for each value $d \in D_v \setminus \{p\}$. $Not{-}p = \top$ can serve as a translation of the literal $\neg p$.

4. Results

We performed the experiments testing the validity of MCPT transform algorithm. MCPT transform algorithm has been tested by different conformant domains from IPC-5, including adder, blocksworld, coins, comm, sortnet and uts. Most of these benchmarks can be translated into MCPTs successful. The translation result file consists of five sections: variable section, initial belief state section, goal set section, operator section and axiom section. To illustrate the validity of MCPT transform algorithm, we give the MCPT transform result of a example from uts domain, namely k01.pddl.

```
var0:
0: Atom at(n1)
1: Atom at(n2)
2: <none of those>
var1:
0: Atom visited(n2)
1: <none of those>
var2:
0: Atom visited(n1)
1: <none of those>
var3:
0: Atom started()
1: <none of those>
```

Figure 2. Variable section

The variable section encoded by the translator is illustrated in Figure 2. The translator generates a format encoding four state variables. Variable $var0$ with domain $\{0, 1, 2\}$ denotes which node the current position of universal transversal sequence is at. Value 0 means that the current position is at node $n1$. Value 1 means that the current position is at node $n2$. Value 2 means that current position is at no node. Variable $var1$ and variable $var2$ denote which nodes have been visited. Variable $var3$ denotes the start of constructing universal transversal sequence.

```
(                                   begin_initial_beliefstate
 :init                              2
 (                                  begin_world_state
  and                               0
  (edge n1 n2)                      1
  (edge n2 n1)                      1
  (                                 1
   oneof                            end_world_state
   (at n1)                          begin_world_state
   (at n2)                          1
  )                                 1
 )                                  1
)                                   1
                                    end_world_state
                                    end_initial_beliefstate
```

Figure 3. Initial belief state section

The initial belief state section of the output MCPT file for the uts example is illustrated in Figure 3 (right). The left part of Figure 3 is the initial belief state section of the input PDDL file for the uts example. There are two world states in the initial belief state as shown in Figure 3 (right). In the first world state, the initial values of $var0$, $var1$, $var2$ and $var3$ are 0, 1, 1 and 1, respectively, while their initial values of the second world state are 1, 1, 1 and 1, respectively. The meaning of MCPT initial belief state is that the current position is at node $n1$ or at node $n2$. Atom $edge(n1, n2)$ as well as $edge(n2, n1)$ is neither fluent nor derived variable. Thus there is no need to encode them. The meaning of this section keeps the same as that of the input PDDL task.

```
(                                   begin_goal_set
 :goal                              1
 (                                  begin_goal
  and                               2
  (visited n1)                      1 0
  (visited n2)                      2 0
 )                                  end_goal
)                                   end_goal_set
```

Figure 4. Goal set section

The goal set section of MCPT is similar in structure to initial belief state section of MCPT. The goal set section of the output MCPT file for the uts example is shown in Figure 4 (right). The left part of Figure 4 is the goal set section of the input PDDL file for the uts example. There is one goal in the goal set. The goal has two goal conditions: "1 0", "2 0". The condition "1 0" means that $var1$ shall assume the value 0 implying that node $n2$ has been visited. The condition "2 0" means that $var2$ shall assume the value 0 implying that node $n1$ has been visited. In other words, the goal is reached if all nodes are visited, which is the meaning of the input PDDL file.

```
(:action travel                     4
 :parameters (?x ?y - node)         begin_operator
 :precondition (started)            travel n2 n1
 :effect                            1
 (when                              3 0
  (and                              2
   (at ?x)                          1 0 1 0 -1 0
   (edge ?x ?y)                     1 0 1 2 -1 0
  )                                 end_operator
  (and                              begin_operator
   (visited ?y)                     travel n1 n2
   (at ?y)                          1
   (not (at ?x))                    3 0
  )                                 2
 )                                  1 0 0 0 -1 1
)                                   1 0 0 1 -1 0
                                    end_operator
                                    ...2 operators omitted
```

Figure 5. Operator section

For the purpose of verification, operator section of the translation result is shown in Figure 5 (right). Two operator definitions of MCPT for the uts example are omitted because of the limited space. Only the operator "$travel(?x, ?y)$" described by PDDL is shown in Figure 5 (left). We explain the operator $travel(n2, n1)$ in detail. The operator has one precondition "3 0", which means that the value of $var3$ is 0, implying the beginning of the construc-

tion of universal transversal sequence. The operator has two effects. The first effect is "1 0 1 0 -1 0". The first number of the effect "1" means that the effect has only one effect condition. The follow numbers "0 1" denote the effect condition implying the value of $var0$ is 1. The rest numbers "0 -1 0" mean that the variable affected by the effect is $var0$; there is no particular value that the variable must have for the operator to be applicable; and the new value for the affected variable is 0. The other effect is "1 0 1 2 -1 0". The first three numbers are the same as those of the first effect. The rest numbers "2 -1 0" mean that $var2$ currently has value -1 and its new is value 0. After the application of the operator $travel(n2, n1)$, node $n1$ has been visited and the current position is at node $n1$. The meaning of $travel(n2, n1)$ is unchangeable. That is to say, the validity of the translation for the operator section is verified.

The axiom section is similar in structure to the operator section, as axiom rules can be considered to be operators that are automatically executed whenever applicable. However, the section is simpler in structure because axiom rules only affect a single state variable. With the knowledge of previous introduction, the axiom rule section is easy to understand. For the uts example, the axiom section is "0". This shows that there are no axiom rules in this domain, which is the case for all pure STRIPS benchmarks. The research on conformant planning is in preliminary phase. Most benchmarks are based on STRIPS or ADL with uncertainties in initial conditions. So it is universal within conformant planning benchmarks.

5. Conclusions

In this paper, we give the definition of a fully instantiated multi-valued representation, called multi-valued conformant planning task. And then, we have presented MCPT transform algorithm which can deal with the uncertainties of initial conditions and disjunctive goals, with two useful relaxed conditions. The validity of the transform algorithm has been verified by the experiments. However, there are several aspects of MCPT transform algorithm that need to be worked on, such as pruning huge search space, dealing with uncertainties of action effects and conformant planning algorithm based on MCPT.

6. Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 60473003, Program for New Century Excellent Talents in University (NCET) and the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 20050183065.

References

[1] McDermott, D. The 1998 AI planning systems competition. *AI Magazine*, 21(2), 2000.

[2] M. Fox, D. Long. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, 20(2003):61-124, 2003.

[3] Edelkamp, S, Hoffmann, J. *PDDL2.2: The Language for the Classical Part of the 4th International Planning Competition*. Albert Ludwigs University, Freiburg, Germany, Institute for Information, Technical Report, 2003.

[4] Gerevini, A., and Long, D. *Plan constraints and preferences in PDDL3: The language of the fifth international planning competition*. University of Brescia, Italy, Technical report, 2005.

[5] Stefan Edelkamp, Malte Helmert. Exhibiting Knowledge in Planning Problems to Minimize State Encoding Length. *Recent Advances in AI Planning*, 5th European Conference on Planning (ECP'99), Vol.1809 of Lecture Notes in Artificial Intelligence, Springer-Verlag, New York, 135-147, 1999.

[6] Malte Helmert. The Fast Downward Planning System. *Journal of Artificial Intelligence Research*, 26(2006):191-246, 2006.

[7] Daniel S. Weld. Recent Advances in AI Planning. *AI Magazine*, 20(2), 1999.

[8] Daniel S. Weld, Corin R. Anderson, David E. Smith. Extending Graphplan to Handle Uncertainty & Sensing Actions. *AAAI98*, 1998.

[9] Goldman,R., Boddy,M. Expressive Planning and Explicit Knowledge. *Proceedings of the 3rd International Conference on Artificial Intelligence Planning Systems*, AIPS-96, AAAI Press,110-117,1996.

[10] Bäckström, C., Nebel, B. Complexity results for SAS+ planning. *Computational Intelligence*, 11(4):625-655, 1995.

[11] Jonsson, P., Bäckström, C. State-variable planning under structural restrictions: Algorithms and complexity. *Artificial Intelligence*, 22(3), 281-296, 1998.

[12] Fox, M., Long, D. The automatic inference of state invariants. *Journal of Artificial Intelligence Research*, 9(1998):367-421, 1998.

[13] Gerevini, A., Schubert, L. Inferring state constraints for domain-independent planning. *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, AAAI Press, 905-912, 1998.

[14] Rintanen, J. A planning algorithm not based on directional search. *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference*, KR, Morgan Kaufmann, 617-624, 1998.

[15] Rintanen, J. An iterative algorithm for synthesizing invariants. *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, AAAI Press, 806-811, 2000.

[16] Bonet, B., Geffner, H. Planning as heuristic search. *Artificial Intelligence*, 129(1):5-33, 2001.

[17] Hoffmann, J., Nebel, B. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14(2001):253-302, 2001.

[18] Hoffmann J., Ronen I. Brafman. Conformant Planning via Heuristic Forward Search: A New Approach. *Artificial Intelligence*, 170(6):507-541, 2006.