The Planning Net: A Structure to Improve Planning Solvers with Petri Nets

Marcos A. Schreiner, Department of Informatics, Federal University of Paraná, Curitiba, Brazil
Marcos A. Castilho, Department of Informatics, Federal University of Paraná, Curitiba, Brazil
Fabiano Silva, Department of Informatics, Federal University of Paraná, Curitiba, Brazil
Luis A. Kunzle, Department of Informatics, Federal University of Paraná, Curitiba, Brazil
Razer A. N. R. Montaño, Department of Informatics, Federal University of Paraná, Curitiba, Brazil

ABSTRACT

In this paper the classical planning problem is formalized as a Petri Net. The authors review the Graphplan notions of mutex relation and maintenance actions based on the Petri Net flow. They also classify pairs of conflicting actions in terms of four different control structures, which are used to build the Plan Net. In addition the authors present the order relation of propositions, i.e., pairs of conflicting propositions that allow inclusion of more information in the Planning Net. The planning problem represented on Planning Net is translated into a SAT instance and solved by a modern SAT solver. The authors show the advantages provided by the inclusion of the propositions ordering relation and compare their experimental results with Satplan.

Keyword: Conflict, Order Relations, Planning Net, Planning Problem, Propositions

INTRODUCTION

The planning problem in AI aims to find a set of actions (a plan) in order to obtain a goal state from a given initial state. The classical planning problem is PSPACE-Complete (Bylander, 1994), nevertheless industry, transport and robotics are examples of areas that are directly benefited from the recent advances on automated planning.

One of the breakthroughs in methods for solving planning problems was Graphplan (Blum & Farst, 1995; Weld, 1999). It is based on a data structure that represents the search space of a planning problem in a smart and economic way. It allows parallelism among
actions through the identification of mutually exclusive actions and propositions. The so called Planning Graph is the basis for many algorithms which were proposed in the past few years.

The strategy of encoding planning problems as satisfiability ones was shown to be competitive. The transformation of a Planning Graph into as satisfiability one presented good results on intrinsically hard problems. A good example is Satplan (Kautz & Selman, 2006).

The idea of representing the planning problem in a Petri Net also seems promising since Petri Nets is a model to represent a system structure and behavior based on preconditions and effects.

In this paper we present a data structure based on Petri Nets in order to solve a planning problem. Our idea is to use the dynamics of Petri Nets in order to improve the classical Planning Graph, which is a static structure, replacing it by a Petri Net, which is a dynamic one, in order to obtain a faster planner (Silva et. al. 2000; Schreiner et. al, 2012).

A planning method based on Petri Nets, called Petriplan, was proposed by Silva (2000). This method converts the planning problem into the reachability problem in Petri Nets. The solver was initially based on integer programming techniques and was not as efficient as it could be.

In this paper we present the Planning Net, which is a data structure based on Petri Nets. It includes a more precise way to classify relations between actions in terms of order relations.

We explore Petri Nets in a more effective way. We propose (i) a smarter way to use the Petri Net flow; (ii) a more precise way to classify the relations between actions; and (iii) a more precise way to classify the relations between propositions.

In Graphplan, pairs of conflicting actions are classified as mutually exclusive based on two criteria: interference and competing needs. The conflicts between pairs of action on the Planning Net are an improvement of the interference conflict. The Petri Net flow analysis showed the uselessness of the competing needs mutexes in the Planning Net. The reason is that the net flow ensures that pairs of actions in this mutex type do not occur in parallel. The net flow also allows to reduce the number of maintenance actions on the net when compared to the Planning Graph.

We specify a partial ordering on actions, which theoretically captures information in a richer way. Pairs of actions are classified in four different ways. Given two actions a and b, we define that they may occur in parallel, non-parallel, a precede b or they are mutually exclusive (Schreiner, 2012).

We also present a new form to capture order relations on propositions. Partial orderings between two propositions represents the information from one or more order relation of actions. In this way, the order relation of propositions encapsulates restrictions of the order relations of actions (Schreiner, 2013).

As a result, we define ordering relation of propositions and control structures between propositions for the Planning Net. We got a data structure with more information, allowing a more detailed SAT encoding to solve planning problems. In what follows we present the new structure and compare it experimentally with Satplan.

ADVANCES IN AI PLANNING

The planning problem in AI was defined in the late ‘60s, when logic was the underlying basis. However, due to the frame problem, this approach was abandoned (McCarty & Hayes, 1969). The frame problem is the need to represent the propositions that do not change after the execution of an action. When an action is applied, we need to add a formula stating that some proposition remains after the execution of this action. These formulas are called frame axioms.

Unfortunately, the number of frame axioms can be intractable. According to McCarthy and Hayes (1969), given a set of n actions and m propositions, the number of frame axioms is approximately n X m (Shanahan, 1997).
Recognition of Human Silhouette Based on Global Features
www.igi-global.com/chapter/recognition-human-silhouette-based-global/66788?camid=4v1a