

## RESUMEN

La temática que se investiga en este trabajo es la planificación continua mediante especificaciones en un lenguaje de definición de dominios de planificación denominado PDDL. El objetivo es implementar un traductor de un subconjunto de este lenguaje a fin de que el Framework de Planificación Continua, también presentado aquí, pueda aprovechar las características de PDDL. Este enfoque es relevante ya que plantea la posibilidad de combinar la expresividad de un lenguaje estándar, como PDDL, con un planificador continuo capaz de resolver problemas en ambientes reales. Es esperable alcanzar un mayor nivel de abstracción para tratar dominios más complejos y, además, realizar comparaciones empíricas de performance con otras soluciones para un mismo problema.

El lenguaje del Planificador Continuo, y uno de los lenguajes fundacionales de representación de problemas de planificación, es STRIPS. Este formalismo permite modelar acciones simples como listas de precondiciones y efectos. Además, cada lista es una conjunción de proposiciones. Por su parte, PDDL es un lenguaje sensiblemente más expresivo que STRIPS. Este lenguaje provee características adicionales que definen varios niveles de expresividad permitiendo enriquecer la definición de dominios y problemas de planificación.

En base a lo expuesto, se plantea la necesidad de tratar con lenguajes más complejos con el objetivo de modelar acciones aplicables en ambientes reales. No obstante, esto resulta en un mayor costo computacional de los algoritmos al momento de resolver problemas de planificación y, por lo tanto, es importante encontrar un balance aceptable entre expresividad y complejidad.

Este trabajo presenta un traductor, desarrollado en Ciao Prolog, de un subconjunto PDDL que soporta algunas de sus características. El subconjunto conforma el lenguaje fuente del traductor y su salida consiste en una representación similar a STRIPS que permite adaptar este traductor a diferentes planificadores, entre ellos, al Planificador Continuo.

La investigación también presenta otros dos resultados teóricos. La redefinición de la arquitectura modular del Framework de Planificación Continua para soportar PDDL y una nueva variante del lenguaje STRIPS para modelar acciones con cuantificación universal en sus precondiciones. Sobre este último resultado, también introducimos un esquema de compilación que permite traducir las acciones, definidas en esta variante, a STRIPS estándar. Además, esta traducción conserva los resultados de planificación que se obtienen con ambas especificaciones. Este esquema, junto con otros esquemas de compilación enunciados, son usados para el desarrollo del traductor propuesto.

La experimentación sobre la implementación presentada se realiza sobre distintas instancias del dominio del “Mundo de Bloques” (un problema clásico de la literatura de Inteligencia Artificial). Cada instancia de este problema es modelada en PDDL empleando las distintas características del subconjunto considerado.

## ABSTRACT

The researched topic in this work is related to the continuous planning using specifications written in the Planning Domain Definition Language named PDDL. The objective is to implement a translator of a PDDL language subset in order to allow that the Continuous Planning Framework (also, it is introduced here) supports PDDL features. This approach is relevant because it supposes combining the expressiveness of PDDL standard language with a continuous planner that tries to solve problems in real environments. We hope to reach a high-level abstraction in order to define more complex domains and to compare the performance with other solutions for the same problem.

The continuous planner language is STRIPS. This formalism allows modeling simple actions using precondition and effect lists. Also, each list is a conjunction of propositions. Particularly, PDDL is more expressive than STRIPS because it provides other features defining expressiveness levels and allowing to enrich the definition of domains and problems of planning.

Based on the above, it is necessary to try with more expressive languages in order to model actions in real environments. Nevertheless, this implies to work with more complex algorithms too. Then, a trade-off between complexity and expressiveness is important.

This thesis presents a translator implemented in Ciao Prolog of a PDDL subset with additional features. This subset defines the source language of translator and its output is a STRIPS-like notation. This output allows adapting the translator to different planning algorithms as the Continuous Planner.

The research also presents other two theoretical results. First, the architecture of Planning Continuous Framework is redefined. After, a new STRIPS variant with universal preconditions is presented. In the last result, actions defined in this variant could be translated to STRIPS preserving results for both specifications. This result together with other compilation schemes are used in the translator implementation.

The experimentation on the implementation has been done on different instances of the “Blocks World” problem (a classic problem of Artificial Intelligence bibliography). Each instance of this problem is modeled in PDDL by using different features.