Finding Parallel Regions with Temporal Planning

Claudio Scheer
Master's Degree in Computer Science
Pontifical Catholic University of Rio Grande do Sul - PUCRS
Porto Alegre - RS, Brazil
claudio.scheer@edu.pucrs.br

Abstract

Abstract.

Introduction.

Bibliography

Temporal Plannign

According to (Haslum et al. 2019), actions in temporal planning have a duration. Therefore, the planner will try to find a schedule in which some actions can be executed in parallel.

There are different approaches that can be used to formalize temporal actions with PDDL. In this paper, I used `:durative-actions`. This action is represented in four sections, as listed below.

- `:parameters`: parameters needed to execute the action;
- `:duration`: time the action takes to run;
- `:condition`: conditions that need to be respected to apply the effects;
- `:effect`: effects that will be applied to the state;

The sections `:condition` and `:effect` are separated in three categories: `at start`, `over all` and `at end`. As described by (Haslum et al. 2019), these categories represent the conditions and effects used at each stage of the action. The `at start` statements are used when starting the action. The `over all` statements are used during the time the action is being executed. The `at end` statements are used at the end of the action.

The following section may not be necessary.

I must mention that `:durative-actions` can be translated into instantaneous actions. Here is the paper: (Scala et al. 2016).

STP

I used the STP (Simultaneous Temporal Planner) planner, introduced by (Blanco et al. 2018), to find the best temporal plan. STP uses a modified version of the Fast Downward (Helmert 2006) planner that can generate a temporal plan. The version used in this paper is provided by the Artificial Intelligence and Machine Learning Group - Universitat Pompeu Fabra[1].

The STP planner needs to receive as a parameter the maximum number of actions that can be executed at the same time. When finding parallel regions, this parameter is a problem, because we do not know how many instructions can be executed in parallel. Therefore, in some cases, it is necessary to test different values for this paramenter.

Formalization

Describe here how I did the formalization.

Results

Describe here the results.

References

[Blanco et al. 2018] Blanco, D. F.; Jonsson, A.; Verdes, H. L. P.; and Jimenez, S. 2018. Forward-search temporal planning with simultaneous events.

[Haslum et al. 2019] Haslum, P.; Lipovetzky, N.; Magazzeni, D.; and Muise, C. 2019. *An Introduction to the Planning Domain Definition Language*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

[Helmert 2006] Helmert, M. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research* 26:191–246.

[Scala et al. 2016] Scala, E.; Haslum, P.; Thiébaux, S.; and Ramírez, M. 2016. Interval-based relaxation for general numeric planning. In Kaminka, G. A.; Fox, M.; Bouquet, P.; Hüllermeier, E.; Dignum, V.; Dignum, F.; and van Harmelen, F., eds., *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, 655–663. IOS Press.

[1] https://github.com/aig-upf