Heuristic Analysis

AIND Project 3

PDDL

Planning problems are the problem of searching and finding the necessary actions until a state (initial state) in a state is set to the target state (end state) and the order (operation procedure) thereof. One of the most typical examples of this problem is the problem of building blocks so to be called the blocks world. A method considered to be applicable to planning problems of various areas is a general-purpose language so-called PDDL. PDDL was derived from STRIP (Fikes and Nilsson, 1971) planning language.
Thanks to PDDL, many planning problems of various kinds can now be expressed in the manner that human being can understand, which gave us a significant impact to solve the problems.

Graphplan

It is a plan generator which, STRIPS format planning problem as input, outputs a partial order plan which achieves target state as a goal. The remarkable feature of this method is to encode the plan search space into a data structure once called a planning graph. In addition, since relations between actions and fluent which are mutually incompatible with each other are checked as a mutual relation, it is possible to narrow the search space to a large extent, thereby realizing efficient searching.

SAT

In Graphplan, there was a problem that it takes time for large-scale problems, even if with utilizing mutual exclusion relation. Therefore, it was found that Graphplan can be converted to SAT(Satisfiability) and realized high-speed processing by applying SAT solver there. [Kautz 92] [Kautz 96]. This is a method combined with a search method that uses a planning graph that narrows the search space of the plan roughly for classical planning. Currently, it is said that heuristic space planners are predominant, but even with the SAT-based approach, by using a mitigated plan definition and a speculative search strategy on the plan, It has been shown to have performance comparable to that of heuristic space [Rintanen 06]. That is, there is an impact that the search efficiency in the planning problem has greatly advanced.

References

[Timeline of AI]
https://en.wikipedia.org/wiki/Timeline_of_artificial_intelligence

[SAT]
https://www.ihi.co.jp/var/ezwebin_site/storage/original/application/481ec9f103d96fc740be9a5709922926.pdf

[Graphplan & SAT solver]
https://www.google.co.jp/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwitrrPr2a3WAhXKvrwKHeTVCkMQFggnMAA&url=https%3A%2F%2Fipsj.ixsq.nii.ac.jp%2Fej%2F%3Faction%3Drepository_action_common_download%26item_id%3D50684%26item_no%3D1%26attribute_id%3D1%26file_no%3D1&usg=AFQjCNGxmDl2UO1QBvhrqvh-UQKJ61U9uw

[SAT Solver]
http://www.kochi-tech.ac.jp/library/ron/2013/g27/M/1165063.pdf

---

Related References above links

[Kautz 92] Kautz, H. and Selman, B: Planning as Satisfiability,
Proc, 10th European Con. on Artificial Intelligence, pp.359-379

[Kautz 96] Katz, H, and Seleman, B: Pushing the envelope: Planning, propositional logic and stochastic search, Proc. AAAI-9, pp. 1194-1201(1996)

[Rintanen 06] Rintanen, J., Heljianko, K. and Niemela, I,: Planning as satisfiability: Parallel plans and algorithms for plan search, Artif. Intel., Vol. 170, No. 12-13, pp-1031-1080(2006)