RESEARCH REVIEW

Review on Need of Representational Language for Search and Planning

Submitted by: Sumit Binnani

The planning problem in Artificial Intelligence is about the decision making performed by intelligent creatures like robots, humans, or computer programs when trying to achieve some goal. It involves choosing a sequence of actions that will (with a high likelihood) transform the state of the world, step by step so that it will satisfy the goal.

The General Problem Solver (GPS)[1] system was an early planner that reduced the difference between some state and a goal state. It used Means-Ends Analysis to compare what is given or known with what is desired and select a reasonable thing to do next. It was a state space planner that operated in the domain of state space problems specified by an initial state, some goal states, and a set of operations. While GPS solved simple problems such as the Towers of Hanoi that could be sufficiently formalized, it could not solve any real-world problems because the search was easily lost in the combinatorial explosion. Put another way, the number of "walks" through the inferential digraph became computationally untenable.

This leads to the first approach to planning known as Situation Calculus[2]. In Situation Calculus, the idea is to use first-order logic, exactly as you know about it, to do planning. We have variables and constants in logical language that ranges over the possible situations, or states of the world. This allows to state that world states have certain properties, but avoids enumerating all of them. Theorem proving is used to “prove” that a particular sequence of actions, when applied to the situation characterizing the world state, will lead to the desired result. This is fine in theory, but problem-solving (search) is exponential in the worst case. Also, resolution theorem proving only finds a proof (plan), and not necessarily a good plan.

Reducing specific planning problem to general problem of theorem proving is not efficient. To the extent that we can decompose a problem and solve the individual problems and stitch the solutions back together, we can often get a more efficient planning process. We can take this advantage in the construction of a planning algorithm. STRIPS[3] is such a representational language for constructing planning algorithms. It was made at Stanford Research Institute and stands for the Stanford Research Institute Problem Solver. It was used to power a robot named Shakey, a real, actual robot that went around from room to room, and push boxes from here to there. They built this planner so that Shakey could figure out how to achieve goals by going from room to room or pushing boxes around. It worked on a very old computer running in hardly any memory, so they had to make it as efficient as possible. The current class of planners like ADL[4], PDDL[5] have essentially grown out from STRIPS. The algorithms have changed, but the basic representational scheme has been found to be quite useful.

---

[1] Newell, A.; Shaw, J.C.; Simon, H.A. (1959). Report on a general problem-solving program. Proceedings of the International Conference on Information Processing. pp. 256–264.

[2] J. McCarthy and P. Hayes (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, Machine Intelligence, 4:463–502. Edinburgh University Press, 1969

[3] Richard E. Fikes, Nils J. Nilsson (Winter 1971). "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving" (PDF). Artificial Intelligence. 2 (3–4): 189–208. doi:10.1016/0004-3702(71)90010-5

[4] Edwin P.D. Pednault. ADL. Exploring the Middle Ground Between STRIPS and the Situation Calculus. In Proceedings of KR-89, 324-332

[5] McDermott, Drew; Ghallab, Malik; Howe, Adele; Knoblock, Craig; Ram, Ashwin; Veloso, Manuela; Weld, Daniel; Wilkins, David (1998). "PDDL---The Planning Domain Definition Language" (PDF). Technical Report CVC TR98003/DCS TR1165. New Haven, CT: Yale Center for Computational Vision and Control. CiteSeerX 10.1.1.51.9941

REFERENCES

Techniques in Artificial Intelligence (SMA 5504), MIT: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-825-techniques-in-artificial-intelligence-sma-5504-fall-2002/index.htm

Knowledge Representation and Reasoning (CS 227), Stanford University: https://web.stanford.edu/class/cs227/

Introduction to Artificial Intelligence (CMSC 671), University of Maryland, Baltimore County: https://www.csee.umbc.edu/courses/671/fall10/

STRIPS, Wikipedia: https://en.wikipedia.org/wiki/STRIPS