Research Review

Fluent Merging and Fast Downward

Originally, planning problems in PDDL notation only accounted for boolean variables. The introduction of **fluent merging** allowed to combine several boolean variables into finite-domain variables. This process enables mere efficient plan search algorithms.

Fast Downward is a classical planning system. It is based on a search heuristic and can be described in the following three stages:

1. Tanslation stage: The PDDL problem is parsed into a normalized planning task (including fluents, operators, initial state and goal).
2. Knowledge compilation stage: A relevance analysis is performed and selected data structures are prepared for the following stage.
3. Search stage: Search is executed, using several heuristics and search methods.

Fluent merging is performed between the first and second stage of this algorithm and transforms the first planning task into a new planning task, used as input for the knowledge compilation stage, where variables are now merged.
Fluent merging algorithm can be divided into two stages:

1. Selection of groups of variables than can be merged
2. Merge of the selected variables

The results of using Fluent Merging in some domains can lead to an improvement in the problem solving performance.

Pattern Database Heuristics and efficient implementations

While an heuristic provides a method for finding a solution to a problem (normally a function computed by an algorithm), a **Pattern Database Heuristics** or PDB tries to achieve the same result by using a lookup table. Pattern Database store a collection of solutions to sub-problems of the main problem and hence demands a high use of memory. On the other hand, they can be more efficient than simple heuristics as heuristics functions are pre-computed and stored in the lookup tables.

An more efficient implementation of PDB using the Fast Downward can be achieved by[1]:

- Avoiding Constructing the Transition Graph
- Avoiding Checking All Operators Individually
- Avoiding State Generation and Ranking

References

Jendrik Seipp and Malte Helmert, Fluent Merging for Classical Planning Problems
Joseph C. Culberson and Jonathan Schaeffer, Pattern Databases
Silvan Sievers and Manuela Ortlieb, Efficient Implementation of Pattern Database Heuristics for Classical Planning

[1] Due to space limitations details are not provided in this summary but can be found in references.