A* Algorithm - Report

Nicolas Boileau, Simon Stastny

September 18, 2012

1 General A* algorithm

We decided to create own own implementation of A* algorithm for better learning outome. Our general A* algorithm implementation is found in the package `edu.ntnu.simonst.tdt4136.astar` and contains following classes:

- Class `BestFirstSearch` with general A* algorithm implementation
- Class `SearchNode` for general search-node used in the algotithm
- Class `SearchState` for general search-state used in the algotithm
- Class `Fringe` used to store nodes in agenda (list of unexpanded nodes)

Apart from those general classes, this package contains class `App` which is used to run both puzzles from command-line using maven.

To run the puzzles, please use maven command `mvn exec:java` inside the project folder, or run the project within NetBeans (maven enabled).

2 Fractions puzzle

2.1 Initial state of the puzzle

The initial state of the puzzle is state identified by permutation `123456789` which represents following fraction:

$$\frac{1234}{56789}$$

2.2 Description of a goal state

This puzzle involves finding fractions equal to fractions:

$$\frac{1}{2} \quad \frac{1}{3} \quad \frac{1}{4} \quad \frac{1}{5} \quad \frac{1}{6} \quad \frac{1}{7} \quad \frac{1}{8} \quad \frac{1}{9}$$

For example for the first fraction (½) the goal state would have permutation `796215384`, because following two fractions are equal:

$$\frac{7692}{15384} = \frac{2^2 \times 3 \times 641}{2^2 \times 3 \times 641} = \frac{1}{2}$$

.

Similarily for other fractions there exist other permutations of `123456789` which represent fractions equal to them respectively.

2.3 Method of assessing arc costs

Since we are not really interested in the path to the goal node as much as we are interested in the goal-node's state itself, we are not concerned about cost of the solution (lenght of the path).

For that reason, the arc cost for transition from one state to another is fixed value 1.

2.4 Heuristic function description

A clear, concise description (using mathematical expressions and text) of the heuristic function (h) used to solve the puzzle.

2.5 Successor generation procedure

A thorough description of the procedure used to generate successor states when expanding a node.

2.6 Overview description of a solution

An overview description of a solution (i.e. path from start to goal) found by A*. The sequence of search states from the start to the goal node must be presented along with a brief summary (in Norwegian or English) of the main state-to-state transitions within sequence.

3 Checkers puzzle

1. The initial state of the puzzle. 2. A general description of a goal state. 3. The method of assessing arc costs. It may be a

xed value or a more complex procedure. 4. A clear, concise description (using mathematical expressions and text) of the heuristic function (h) used to solve the puzzle. 5. A thorough description of the procedure used to generate successor states when expanding a node. 6. An overview description of a solution (i.e. path from start to goal) found by A*. The sequence of search states from the start to the goal node must be presented along with a brief summary (in Norwegian or English) of the main state-to-state transitions within sequence.