General Problem Solving

Lorenz M"osenlechner, Thomas R"uhr

November 15, 2011
Problem Definition

- Initial state
  - example: (at-home have-money)

- Goal state
  - example: (at-uni)

- Set of actions
  - preconditions
  - postconditions
  - example:
    
    name: buy-ticket
    preconditions: have-money
    postconditions: have-ticket

- The problem solver finds a sequence of actions to get from the initial state to the goal state.
Algorithms

- **Breadth-first search:**
  - Calculates all states for a specific search depth and increases the search depth by one in each iteration.
    1. Generate all possible successor states by applying all valid actions to the current state.
    2. Expand each successor state by applying valid actions again and generate new set of states.
  - pro: will always find the shortest solution
  - con: might need a lot of memory

- **Depth-first search:**
  - Search along one branch until no actions are applicable anymore.
    1. Generate set of successor states by applying all valid actions
    2. For each successor state, make recursive call
  - pro: good memory footprint
  - cons: can get stuck in loops, not necessarily finds the shortest solution.
Possible Optimizations

▶ Iterative deepening:
  ▶ Combination of breadth-first and depth-first search
  ▶ Perform a depth-first search up to a maximal depth of $n$
  ▶ If no solution was found, iterate and search up to a depth of $n + 1$.

▶ Hashing to avoid loops
  ▶ Generate a hash for each state
  ▶ Store the hash in a table
  ▶ Before expanding a state, check if it has been traversed already before.