Exercise 3

General Game Playing

Prof. Michael Thielscher and Stephan Schiffel
International Masters Programme in Computational Logic — winter term 2008/09
17.11.2008

Exercise 3.1

Implement exhaustive search for single player games. Your program should be able to solve small single player games like 'Maze', 'Blocks', 'Buttons' or 'Pancakes' within seconds or a few minutes (see table below). Solving a game means to find a sequenze of moves to win within the startclock. To actually play the game perfectly after that you have to remember the best move for each state you encountered during search such that you can reuse that information. For bigger games or smaller clock values your program won't be able to exhaustively search the game tree in the given timeframe but it should still play better than a random player. Make sure your program always returns a legal move in time, even if the search hasn't finished.

Game	Startclock	Playclock
Maze	10	10
Blocks	10	10
Buttons	60	10
Pancakes	180	10

Exercise 3.2

- Use a hash table to recognize repeated states, if you have not already done so. This can lead to substantial reduction of the size of the search space. It is also a good way for reusing information from a previous step search in the next step.
- Improve your algorithms, avoid unneccessary computations, object creations, memory allocations, etc.

Ideally, your program will be able to solve the following games in the given times.

Game	Startclock	Playclock
Buttons	10	10
Pancakes	30	10
Asteroids	60	10
Hanoi	60	10
Aipsrovers01	120	10

Exercise 3.3

Test your program with more difficult single player games (e.g. 8puzzle or peg) and longer start clock ($\approx 15 - 30min$) and play clock ($\approx 3 - 5min$) to check that it is able to handle longer runtime and bigger memory requirements. Although your program is probably not able to solve these games it should not crash and return legal moves.

Exercise 3.4 optional

Compare runtime (and possibly memory usage) of breadth-first search, depth-first search and iterative deepening depth-first search for solving pancakes. asteroids, hanoi and coins2. Which algorithm is better on which game and why?