

Übungen *Handlungsplanung und Allgemeines Spiel* - Blatt 2

Stefan Edelkamp

Peter Kissmann

Abgabe 22.11.2010, Besprechung 25.11.2010

1. (GDL) Ladet euch die Spielbeschreibung `ex_2_1.kif` von der Übungs-Webseite herunter. In diesem Format kommen alle Spiele bei euren Spielern an, wenn (wie bei einer Meisterschaft) die Eingabe verschlüsselt wird. Analysiert die Regeln und ersetzt die unsinnigen Relationssymbole, Funktionssymbole, Atome und Variablennamen durch aussagekräftige, um so letztlich die Spielbeschreibung zu entschlüsseln.

(a) Um welches Spiel handelt es sich? Falls ihr es nicht kennt, beschreibt, wie der Spielablauf aussieht.

(b) Wie habt ihr die Bedeutung der Regeln erkannt?

2. (Implementierung Einpersonenspiellöser) Baut auf dem bestehenden Code des Referenzspielers eurer Wahl auf und implementiert einen Löser für allgemeine Einpersonenspiele. Überlegt euch dazu eine geeignete Datenstruktur und führt eine vollständige Suche durch. Für einfache Spiele wie Maze, Blocks oder Buttons sollte nach wenigen Sekunden eine Lösung gefunden werden. Bei komplexeren Spielen mag dies länger als die zur Verfügung stehende Zeit dauern, sorgt aber auch bei diesen dafür, dass stets rechtzeitig ein gültiger Zug an den GameController geschickt wird (nutzt beispielsweise direkt den Code des Referenzspielers – der wählt immer den letzten gültigen Zug – oder wählt einen zufälligen gültigen Zug oder überlegt euch eine andere Herangehensweise).

(a) Testet eure Implementierung mit folgenden Spielen und Zeiten: Die Spiele könnt ihr alle auf dem GGP-Server

Spiel	Startclock	Playclock
Blocks	5	5
Buttons	5	5
Maze	5	5
Pancakes	10	5
Asteroids	60	10
Hanoi	60	10

<http://ggpserver.general-game-playing.de> finden. Schickt mir anschließend die Logs des GameController (auch jeweils für die restlichen Aufgaben).

(b) Testet euer Programm auch an einigen komplexeren Spielen (etwa `8puzzle` oder `peg`) und nutzt längere Start- (beispielsweise 15 bis 30 Minuten) und Playclocks (etwa bis 5 Minuten), um zu überprüfen, ob euer Spieler mit der längeren Laufzeit und der damit verbundenen höheren Speicherauslastung zurechtkommt. Auch hier sollte euer Spieler stets einen gültigen Zug zurückliefern. Experimentiert insbesondere mit folgenden Spielen und Zeiten:

Spiel	Startclock	Playclock
<code>8puzzle</code>	180	30
<code>max_knights</code>	1200	300
<code>peg_bugfixed</code>	300	120
<code>queens</code>	180	60
<code>twisty_passages</code>	180	60

3. (Implementierung Einpersonenspieler) Implementiert zumindest die Monte-Carlo Suche, besser sogar UCT, für Einpersonenspiele. Überlegt euch auch hierzu passende Datenstrukturen und sorgt dafür, dass stets ein gültiger Zug an den GameController übermittelt wird.

Führt die Aufgabenteile 2a und 2b entsprechend auch mit diesem Spieler durch. Kann euer Löser irgendwelche Spiele lösen, in denen der Spieler versagt? Oder umgekehrt? Habt ihr Erklärungen dafür?