

## Regelbasierte Programmierung mit XL, SoSe 2007 Übungsblatt 5

Präsentation der Lösungen am 18. 06. 2007 in der Übung

### Aufgabe 10

Schreiben Sie ein XL-Programm, welches ein zufälliges chemisches Netzwerk erzeugt.

- a) Erzeugen Sie eine Menge von Knoten für die vorhandenen Substrate (Molekülarten, durch Kugeln repräsentiert).
- b) Erzeugen Sie chemische Reaktionen, indem Sie für jede Reaktion einen Transitionsknoten erzeugen (dargestellt durch eine Box) und diesen mit zwei Substratknoten durch gerichtete Kanten (vom Edukt zum Transitionsknoten und vom Transitionsknoten zum Produkt) verbinden. Die Kanten zwischen den Substratknoten und den Transitionsknoten sollen visualisiert werden.
- c) Stellen Sie in einem Diagramm die Abhängigkeit der Zusammenhangskomponenten (z.B. durch Graphfärbung) des chemischen Netzwerks als Funktion der Anzahl der Transitionsknoten dar.
- d) Erzeugen Sie katalytische Reaktionen, indem Sie ein zufällig gewähltes Substrat als Katalysator für eine zufällig gewählte Reaktion auswählen und diesen über einen zweiten Kantenyp mit dem Transitionsknoten verbinden. Die Funktion des Substrats als Katalysator ist geeignet zu visualisieren.
- e) Stellen Sie die Anzahl der Komponenten, die katalytisch abgeschlossen sind, in einem Diagramm als Funktion der erzeugten Anzahl von katalytischen Reaktionen dar. Eine Komponente ist katalytisch abgeschlossen, wenn Sie jeden den in ihr enthaltenen Katalysator selbst erzeugen kann.

### Aufgabe 11

Entwickeln Sie ein Modell für ein Ökosystem von Tieren und Pflanzen. Pflanzen sollen dabei wachsen und als Nahrung für die Pflanzenfresser dienen. Fleischfresser ernähren sich dann von den Pflanzenfressern. Die Tiere bewegen sich durch die virtuelle Landschaft auf der Suche nach Nahrung (visualisieren). Pflanzen und Tiere sollen sich virtuell vermehren.

- a) Implementieren Sie dieses Ökosystem für Pflanzen und Pflanzenfresser
- b) Implementieren Sie zusätzlich die Fleischfresser
- c) Stellen Sie die Anzahl der Individuen der Pflanzen, Pflanzenfresser und Fleischfresser in einem Diagramm als Funktion des Simulationsschritts dar.

## Aufgabe 12

Programmieren Sie einen binären evolvierenden zellulären Automaten (ZA) mit zwei Zuständen (0 und 1). Im Gegensatz zu einem normalen zellulären Automaten kann dieser seine Ausführungsregeln verändern.

a) Dazu sind die Ausführungsregeln geeignet in einer Zahl zu kodieren (siehe Skript „Artificial Life“). Implementieren Sie also einen 1-dimensionalen ZA, der diese Ausführungsregeln entsprechend einer angegebenen Zahl simuliert.

b) Für einen gegebenen ZA erzeugen Sie 5 verschiedene zufällige Mutationen, indem Sie in der den ZA kodierenden Zahl jeweils ein zufällig gewähltes Bit invertieren. Jeder dieser 5 ZA wird mit einem (für jeden ZA gleichen) zufälligen Muster von Nullen und Einsen initialisiert. Die Anzahl an Nullen und Einsen ist gleich groß. Anschließend wird jeder ZA eine vorgegebene Anzahl an Schritten simuliert. Der Benutzer kann nun einen der 5 ZA auswählen, welcher dann für weitere Mutationen verwendet wird.

c) Die Auswahl einer Mutation ist zu automatisieren. Es werden sechs verschiedene, zufällige Testmuster mit verschiedenen Verhältnissen der Anzahl der Nullen und Einsen erstellt (0%+100%, 20%+80%, 40%+60%, 60%+40%, 80%+20%, 100%+0%). Jede der Mutationen muß alle diese Testmuster bearbeiten. Für die Bewertung, wie gut eine Mutation bei einem Testmuster abgeschnitten hat, wird ein Qualitätskriterium eingeführt. Dieses prüft nach einer vorgegebenen Anzahl an Iterationsschritten das Verhältnis der Anzahl von Nullen und Einsen. Gab es am Anfang mehr Nullen als Einsen, so ist der ZA besser, welcher hinterher die meisten Nullen besitzt. Gab es am Anfang mehr Einsen als Nullen, so ist es der ZA mit den meisten Einsen. Die Ergebnisse der mutierten ZA aus den sechs Tests werden miteinander verrechnet, um den besten ZA auszuwählen. Anschließend wiederholt sich das Verfahren.