

# Selbstorganisierende, adaptive Systeme - Blatt 04

Gruppe 04 – Alex Oks, Markus Görlich, Simon Stieber

13. November 2016

## 1 Aufgabe 3 - Entwurf Simon

Formale Definition von Spielen: (Skript Nr. 4 F. 16):

Ein strategisches Spiel  $G = \langle N, (A_i)_{i \in N}, (u_i)_{i \in N} \rangle$  besteht aus

- einer endlichen Menge  $N = 1, \dots, n$  an Spielern
- einer Menge an Aktionen  $A_i$  für jeden Spieler
- einer Nutzenfunktion (utility)  $u_i : A_1 \times \dots \times A_n \rightarrow R$  für jeden Spieler

### 1.1 a

Wenden Sie die formale Definition an, um die Entscheidungssimulation zu modellieren. Hier:

- $N$ : Computer in einem Netzwerk
- $A_i$ : Datei im Cache behalten oder Datei anfordern
- $u_i$ : Die Nutzenfunktion ergibt sich aus den Kosten für Anfordern oder Behalten

### 1.2 b

Bestimmen Sie die weiteren Nash-Gleichgewichte (nur in reinen Strategien) in Abbildung 1.

- Nash Gleichgewicht: Für jeden Agenten  $i = 1, \dots, n$  gilt, dass  $u_i(a) \geq u_i((a_{-i}, a)) \forall a \in A_i$
- Reine Strategie: Immer nur eine Aktion wählen

*TODO* Kann ich mir noch keinen richtigen Reim drauf machen, was da alles mit was verglichen werden muss. Habs versucht zu skizzieren, aber bin noch nicht ganz durchgestiegen.

### 1.3 c

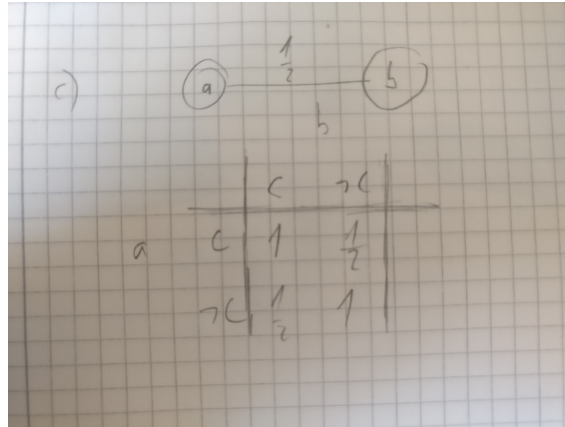
*TODO* Skizze: siehe Abb. 1

### 1.4 d

Untersuchung Algorithmus:

Durchgänge:

1.  $i = b$ , weil  $d(i) = 3$ .  $S = \{b\}$ ,  $N = \{a, c, d, e, f\} \rightarrow N = \{c, e\}$
2.  $i = c$  oder  $e$ , weil  $d(c)=2$ ,  $d(e)=2$ . Wähle  $c$ .  $S = \{b, c\}$ ,  $N = \{e\} \rightarrow N = \{e\}$



**Abb. 1:** Skizze zu 3c

3. Dasselbe passiert bei der Wahl von  $e \rightarrow$  Gleichgewicht  $S = \{b, c, e\}$

Erzeugt dieser Algorithmus stets ein Nash Gleichgewicht? Zum Schluss bleiben immer Knoten  $S$  übrig, die cachen sollen. Ist das immer ein Gleichgewicht? TODO