

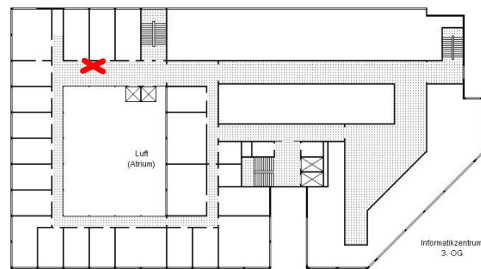
Alexander Kröller  
Henning Hasemann  
Stephan Friedrichs

## Verteilte Algorithmen Übung 5 vom 1. 7. 2012

Abgaben zu allen Teilen bis zum 15. 7. im  
Hausaufgabenrückgabeschrank.<sup>1</sup>

**Bitte die Blätter vorne deutlich mit eigenem Namen versehen!**

Abgaben zu P per Mail an  
hasemann@ibr.cs.tu-bs.de.



### A — Allgemeiner Teil

Diese Aufgaben können von jedem bearbeitet werden, egal ob sich ansonsten für T oder P entschieden wird.

**Aufgabe A1:** Betrachte Algorithmus 3.3 aus der VL (Dijkstras Token Ring). Zeige, dass der Algorithmus nicht stabilisiert, wenn  $v_0$  nicht alle  $n$  Zustände  $0, \dots, n - 1$  verwendet. **(t P.)**

### T — Theoretischer Track

Diese Aufgabe schliesst sich mit P aus:

**Aufgabe T1:** Angenommen, wir haben zwei Algorithmen  $\mathcal{A}_1$  und  $\mathcal{A}_2$  gegeben, die jeweils in Zeit  $\theta_1$  bzw.  $\theta_2$  stabilisieren. Allerdings benötigt  $\mathcal{A}_2$  ein korrektes Ergebnis von  $\mathcal{A}_1$  als Input (zum Beispiel könnte  $\mathcal{A}_1$  einen Baum konstruieren und  $\mathcal{A}_2$  darauf einen ConvergeCast durchführen).

Zeige, wie die beiden so ausgeführt werden können, dass ein kombinierter Algorithmus  $\mathcal{A}$  entsteht, der in  $\theta_1 + \theta_2$  stabilisiert. **(2 P.)**

<sup>1</sup>Analog zur vorigen Programmieraufgabe heisst „bis zum 15.7.“ genauer: „Bis zum 15. 7. 2012, 24:00, Hawaiianische Zeit“. Zu diesem Zeitpunkt ist in Braunschweig der 16. 7., 12:00 mittags.<sup>2</sup>

<sup>2</sup>Eine derartige Festlegung ist völlig normal z.B. bei Konferenzanmeldungen und Forschungsanträgen, wo sich immer kurz vor Deadline jammernde Autoren bei den Organisatoren melden.<sup>3</sup>

<sup>3</sup>Natürlich funktioniert so etwas nicht bei Anträgen bei der EU. Dort ist die Deadline immer Punkt fünf nachmittags, Brüsseler Zeit. Was niemanden überraschen sollte.

**Aufgabe T2:** Eine Menge  $I \subseteq V$  heisst unabhängig („independent set“), wenn es keine Kante im Graphen gibt, die zwei Knoten aus  $I$  verbindet. Wenn  $I$  unabhängig ist, aber es kein  $v \in V \setminus I$  gibt, mit dem  $I \cup \{v\}$  unabhängig ist, ist  $I$  ein MIS. Betrachte folgenden Algorithmus mit gegebenen (vom Gegenspieler unveränderlichen) IDs:

**Führe jede Runde aus:**

```

┌ if  $v$  hat einen Nachbarn mit höherer ID in MIS then
│    $v$  ist nicht im MIS
└ else
     $v$  ist im MIS

```

**Algorithm 1:** Selbststabilisierendes MIS (code für  $v \in V$ )

- Zeige, dass der Algorithmus selbst-stabilisierend und korrekt ist, und gib die zugehörige Stabilisierungszeit an ( $\mathcal{O}(\dots)$ -Notation reicht).
- Gib für beliebige  $n > 2$  Netzwerke und vom Gegenspieler erzwungene Startzustände an, bei denen die Zeit aus a) tatsächlich erreicht wird (mit Beweis, klar).

(4 P.)

## P — Praktischer Track

Diese Aufgabe schliesst sich mit T aus. Sie läuft über **Blatt 4** und **Blatt 5** zusammen, ihr habt Zeit bis zum 15. 7.

**Aufgabe P1:** Entwirf und implementiere (in der Wiselib, klar) einen Algorithmus für kürzeste Wege in allgemeinen gewichteten Graphen, mit IDs. Ein Knoten dient als Quelle (typischerweise ist dies ein Knoten mit Aussen-, d.h. Internetanbindung). Die Gewichte werden von einer anderen Komponente gestellt, sie stellen bspw. Linkqualitäten dar. Der Algorithmus soll mit dynamischen Netzen umgehen können, d.h.:

- Einige Knoten starten später („kommen dazu“),
- Knoten können sich beenden,
- Knoten können sich bewegen und darüber ihre Nachbarschaft ändern,
- Durch die Bewegungen kann das Netzwerk geteilt bzw. wieder vereint werden.
- Die Gewichte der Kanten können sich ändern.

Euer Algorithmus soll diesmal auch auf eingebetteten Systemen laufen können. Daher gibt es zwei Teilaufgaben:

- Entwirf und implementiere einen Algorithmus in der Wiselib; zur Abgabe gehört auch eine Beschreibung des Algorithmus, eine Begründung seiner Korrektheit. Dafür gibt es bis zu **6 P.**
- Implementiere den Algorithmus so, dass er auch auf den iSense-Sensorknoten des Instituts läuft. Dafür gibt es bis zu weitere **6 P.**

(6+6 P.)