

Alexander Kröller  
Henning Hasemann  
Stephan Friedrichs

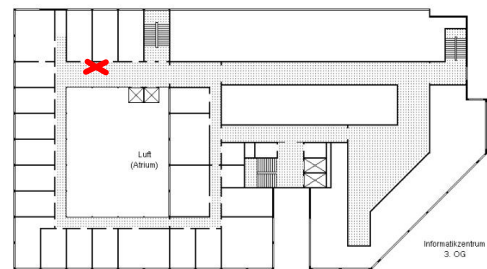
## Verteilte Algorithmen Übung 2 vom 12. 5. 2012

Abgaben zu A und T am Dienstag, dem  
22. 5. 2012, entweder

- vor der Übung im IZ358, oder
- bis 9:40 im Hausaufgabenrückgabeschrank.

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

Abgaben zu P zum 12. 6. 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:** Gib einen Algorithmus an, der Leader Election in einem ungerichteten Baum löst, für ASYNC und CONGEST. Die Knoten haben IDs und kennen ihre Nachbarschaft. Zeige für Deinen Algorithmus:

- Korrektheit
- Zeitkomplexität — und, dass diese asymptotisch optimal ist.
- Nachrichtenkomplexität — und, dass diese asymptotisch optimal ist.

(„Asymptotisch optimal“ bedeutet Optimalität in  $\mathcal{O}(\cdot)$  bzw.  $\Omega(\cdot)$ ) (4 P.)

### T — Theoretischer Track

Diese beiden Aufgaben schliessen sich mit P aus:

**Aufgabe T1:** Ein *Matching* in einem Graphen  $(V, E)$  ist eine Menge  $M \subseteq E$  von Kanten, bei der keine zwei Kanten aus  $M$  einen gemeinsamen Endknoten haben. Gib einen möglichst effizienten Algorithmus (SYNC, CONGEST, IDs) an, der ein maximales Matching bestimmt.

Maximal bedeutet: Es gibt kein Matching  $M'$  mit  $|M'| > |M|$  und  $M' \supseteq M$ . (3 P.)

**Aufgabe T2:** Zeige: Leader Election ist in Bäumen ohne IDs nicht lösbar. Genauer: Für jedes  $n \in \mathbb{N}$  gibt es ungerichtete Bäume mit mindestens  $n$  Knoten, auf denen in SYNC und LOCAL kein deterministischer Algorithmus einen Leader bestimmen kann. (3 P.)

## P — Praktischer Track

Diese Aufgabe schliesst sich mit T aus. Sie läuft über **Blatt 2** und **Blatt 3** zusammen, ihr habt Zeit bis zum 12. 6.

**Aufgabe P1:** Entwirf und implementiere (in der Wiselib, klar) einen Algorithmus für Leader Election in allgemeinen Graphen, mit IDs.

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.

Euer Algorithmus soll jeweils dafür sorgen, dass sich genau ein Knoten im Netz zum Leader erklärt. Ein geteiltes Netz mit mehreren Zusammenhangskomponenten ist dabei als mehrere Netze anzusehen. (6+6 P.)