

CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL  
Institut für Informatik und Praktische Mathematik

Prof. Dr. W.-P. de Roever  
Harald Fecher



## Verteilte Algorithmen

Sommersemester 2006

**Serie 11**

28. Juni 2006

### Thema: Endsemestertest (Einzelabgabe)

**Ausgabetermin: 28. Juni 2006****Abgabe: 12. Juli 2006 um 11:00 (strikt)**

**Endsemestertest:** Diese Aufgabenserie ist von jedem *allein* zu bearbeiten! Die Aufgaben sind etwas breiter gestreut. Bitte beachten Sie auch den Abgabetermin. Schreinabgabe oder elektronisch (an Harald und an Meiko) möglich. Jede Aufgabe soll auf einem **neuem** Blatt bearbeitet werden (sonst gibt es Punktabzüge). Viel Erfolg!

**Aufgabe 1 (AsynchBFS (6 Punkte))** Lösen Sie Aufgabe 23 aus Kapitel 15. Informeller Beweis für die Korrektheit ist ausreichend.

**Aufgabe 2 (GHS-Ablauf (3 Punkte))** Lösen Sie Aufgabe 34 aus Kapitel 15 (Seite 528), d.h. beschreiben Sie einen Ablauf des *GHS* bei dem eine reject-Nachricht als Antwort auf Test zum einem Zeitpunkt zurückkommt, bei der "Fragesteller" diese Kante als **branch**, also zum MST gehörig, klassifiziert hat. Argumentieren Sie, dass das in Ordnung geht.

**Aufgabe 3 (GHS (3 Punkte))** Lösen Sie Aufgabe 35 aus Kapitel 15. Informelle Argumentation ist ausreichend.

**Aufgabe 4 (Deadlock-detection (6 Punkte))** [Aufgabe 13 (a),(b) aus Kapitel 19] Betrachte eine Anzahl von Prozessen, die möglicherweise auf Ihre Nachbarprozesse warten. Genauer: jeder Prozess hat eine lokale Variable *waiting-for*, in der die Menge der Prozesse enthalten ist, auf die der Prozess wartet.

1. Geben Sie einen precondition-effect code an, der eine Kreisabhängigkeit dieser verteilten Prozesse entdeckt. Genauer: Der Algorithmus soll feststellen ob es einen Kreis zwischen mindestens 2 Prozessen gibt, so dass der Nachfolger auf den Vorgänger wartet und so dass sich keine Nachrichten in den Channels befinden, die von einem Vorgänger zu einem Nachfolger (des Kreises) gesendet ist.
2. Zeigen Sie (informell), dass ihr Algorithmus korrekt ist. Geben Sie die Komplexität des Algorithmus an.

**Aufgabe 5 (Leader election (4 Punkte))** Problem: Gegeben sind eine nicht bekannte Anzahl von Prozessen, die mit einem bidirektionalen Ring verbunden sind. Weiter besitzen diese Prozesse keine eindeutige UID. Insbesondere weiss ich nicht, zu wem man seine Nachricht schickt (könnte man auch selber sein), man weiss nur, dass es ein Ring ist. Das synchrone Modell wird betrachtet.

Zeigen Sie, dass es keinen Algorithmus gibt, der sicherstellt, dass keine zwei Prozesse sich als leader bestimmen und dass die Wahrscheinlichkeit der Nichttermination gegen Null geht.

Diskutieren Sie auch, ob es einen Algorithmus geben kann, wenn man einen Spanning-tree als Netzwerk hat und wenn man einen vollständigen Graphen als Netzwerk hat.

## Literatur

- [1] Nancy Lynch. *Distributed Algorithms*. Kaufmann Publishers, 1996.