

Verteilte Algorithmen und Datenstrukturen

SS 2016

Übungsblatt 7

Aufgabe 1:

- a) Implementieren Sie das Build-Cycle Protokoll in der Subject-Erweiterung von Java und testen Sie dieses für eine Menge an Prozessen, die initial in einer schwach zusammenhängenden, ungeordneten Liste verbunden sind.
- b) Begründen Sie, die für Phase 1 und 3 formulierten Ziele im Beweis von Satz 5.13 irgendwann erreicht werden.

Aufgabe 2:

- a) Überlegen Sie sich ein geeignetes Build-d-List Protokoll, das die Knoten aus jedem schwachen Zusammenhang heraus zu einer d -fachen Liste organisieren soll und begründen Sie, warum es korrekt ist.
Hinweis: Angenommen, u habe zurzeit als linke Nachbarn v_1, \dots, v_d mit $v_1 < v_2 < \dots < v_d$ und als rechte Nachbarn w_1, \dots, w_d mit $w_1 > w_2 > \dots > w_d$ (d.h. u hat bereits d linke und rechte Nachbarn). Zur Vereinfachung der Regel nehmen wir weiterhin an, dass $u = v_0 = w_0$ ist. Dann reicht es in timeout, wenn u dem Knoten v_d v_{d-1} vorstellt, dem Knoten w_d w_{d-1} vorstellt, und alle v_i mit $i \in \{0, \dots, d-1\}$ dem Knoten w_1 sowie alle w_i mit $i \in \{0, \dots, d-1\}$ dem Knoten v_1 vorstellt.
- b) Implementieren Sie Ihr Build-d-List Protokoll in der Subject-Erweiterung von Java und testen Sie dieses für eine Menge an Prozessen, die initial in einer schwach zusammenhängenden, ungeordneten Liste verbunden sind.