Technische Universität München
Institut für Informatik
Prof. Dr. Angelika Steger
Dr. Stefanie Gerke
Alexander Offtermatt-Souza

Wintersemester 2002
Übungsblatt 09

Randomisierte Algorithmen

Abgabetermin. Montag, den 13.1.2003 vor der Vorlesung

Byzantinisches Protokoll für Prozessor P_i mit $i \in \{1, 2, \ldots, n\}$

Coin enthält in jeder Runde das Ergebnis eines globalen Münzwurfes.
Initialisierung.
$G := n - t$,$H := G - t$,$L = H - t$,$L \geq \frac{n}{2} + t$
$vote := b_i$, $(b_i \in \{0, 1\})$
Wiederhole.
(1) Sende $vote$ an alle Prozessoren
(2) Empfange $vote$-Werte aller Prozessoren, inkl. eigenem
(3) $maj :=$ Ergebnis der Mehrheit aller Prozessoren
(4) $tally :=$ Anzahl der Prozessoren von denen maj empfangen wurde
(5) **Wenn** Coin $= Kopf$ **dann** $threshhold := L$
sonst $threshhold := H$
(6) **Wenn** $tally \geq threshhold$ **dann** $vote := maj$
sonst $vote := 0$
(7) **Wenn** $tally \geq G$ **dann** sende ab jetzt immer $vote$

Aufgabe 1 Byzantinisches Problem I (15 Punkte)

Wie kann man das Protokoll verändern, damit mehr als $\frac{n}{8}$ „schlechte“ Prozessoren verkraftet werden können?

Aufgabe 2 Byzantinisches Problem II (10 Punkte)

Ist es in diesem Protokoll immer richtig, daß alle guten Prozessoren in der gleichen Runde ihre Entscheidung fällen?

Asynchrones Choice Coordination Protokoll für Prozessor P_i mit $i \in \{1,2\}$

Initialisierung.
Register C_0 und C_1 initialisiert mit $(0,0)$
P_i scannt zu Beginn ein zufällig gewähltes Register.
Die lokalen Variablen T_i und B_i sind mit 0 initialisiert.
Wiederhole.
(1) P_i erhält Zugriffsrechte für sein aktuelles Register und liest (t_i, R_i).
(2) P_i entscheidet sich für einen von 5 Fällen:
(2.1) **Wenn** $R_i = \surd$ **dann** Stop.
(2.2) **Wenn** $T_i < t_i$ **dann** $T_i := t_i$ und $B_i := R_i$.
(2.3) **Wenn** $T_i > t_i$ **dann** schreibe $\surd$ in das aktuelle Register und Stop.
(2.4) **Wenn** $T_i = t_i$ und $R_i = 0$ und $B_i = 1$
dann schreibe $\surd$ in das aktuelle Register und Stop.
(2.5) **Sonst** $T_i := T_i + 1$, $t_i := t_i + 1$;
schreibe ein, mit einer fairen Münze zufällig bestimmtes bit in B_i und
schreibe (t_i, B_i) in das aktuelle Register.
(3) P_i gibt sein aktuelles Register frei, geht zum anderen Register
und fährt mit (1) fort.

Aufgabe 3 Choice Coordination Problem (15 Punkte)

Zeigen Sie, daß obiges Protokoll zum asynchronen „Choice Coordination Problem“ aus der Vorlesung auch für mehr als 2 Prozessoren bzw. Register funktioniert.

Hinweis. Da die Aufgaben für dieses Übungsblatt erst am 13.1.2003 abgegeben werden, findet in der Übungsstunde am *Dienstag, den 7.1.2003* eine Fragestunde statt.

Übungsleitung

Alexander Offtermatt-Souza
Raum: MI 03.09.037 – Telefon: 289-17742 – eMail: offterma@in.tum.de