

Eerste Herdeeltoets Concurrency

24 dec 2014, 13.30 – 15.30, Educ-Γ.

Maak vragen 1 en 2 op pagina 1, 3 en 4 op pagina 2, en 5 en 6 op pagina 3. Vragen 2 en 6 zijn 2pt, de andere elk 3pt. Te halen 16pt, cijfer is totaal gedeeld door 1,5.

1. **Fairness:** In dit lusje zitten twee concurrente statements, die i op 0 en 1 zetten:

```
i = 0
while (i == 0)
  { cobegin { i=0 }   { i=1 } coend
    print i }
```

- (a) Laat zien dat er voor elke $k \in \mathbb{N}$ een executie is die eerst k nullen en dan een 1 print.
- (b) Geef de definitie van *fair* scheduling
- (c) Is bovenstaand programma terminerend onder de aanname van fair scheduling? Leg uit!
2. **Safety of Liveness:** Zeg van elk van deze uitspraken of ze een *Safety* of een *Liveness* eigenschap beschrijven en waarom (in 1 zin).
- (a) Berichten worden afgeleverd in dezelfde volgorde als waarin ze zijn verstuurd.
- (b) Als een bericht aan een niet-bestaand adres wordt verzonden, wordt binnen 1500 ms een foutmelding getoond.
- (c) Na klikken op *LOAD* wordt het bestand in de edit-buffer geladen.
- (d) Een krant die in de zon ligt, wordt geel.
- (e) De waarde van *counter A* is strikt stijgend.
3. **Celebrities in Bier:** Twaalf studenten drinken elk drie flesjes bier. Elk biertje bevat een plaatje van een random artiest, waarbij de kans op Jan Smit 0,10 is en de kans op Patty Brard 0,20.
- (a) Hoe groot is de kans dat er op het feest minstens 1 Jan Smit is?
- (b) Hoe groot is de kans, voor *een willekeurige student*, om zowel Patty als Jan te krijgen?
- (c) Wat is de verwachting van het aantal Patty Brards dat op deze avond gevonden wordt?
4. **Fairness en TaS-lock:** Het TaS-lock heeft als lock-methode het herhalen van de Test-and-Set operatie, tot het resultaat 0 wordt gekregen: `while (L.TaS ==1) {}`.
- (a) Voldoet het TaS-lock aan *No Starvation* als fairness niet wordt aangenomen?
- (b) Voldoet het TaS-lock aan *No Starvation* als fairness wel wordt aangenomen?
- (c) Voldoet het TaS-lock aan *No Deadlock* als fairness wel wordt aangenomen?
5. **Wisselgeheugen en Consensus:** Valerie heeft een geheugenchip gemaakt (het wisselgeheugen) die je niet alleen per geheugenlocatie i atomair kunt lezen en schrijven (`read(i)` en `write(i,x)`), maar ook een instructie `wissel(i,j)` heeft die de inhoud van locaties i en j atomair verwisselt. Zij claimt dat haar product ConsensusGetal oneindig heeft. Voor consensus tussen threads 0 t/m $n - 1$ stoppen we eerst een 0 in locaties 0 t/m $n - 1$ en een 1 in plek n .
- (a) Bewijs: als elke thread i eenmaal `wissel(i,n)` doet, verandert de toestand van het geheugen alleen in de eerste aanroep.
- (b) Geef een wacht-vrije Consensus-implementatie voor n threads met wisselgeheugen.
- (c) Klopt de claim van Valerie?
6. **Technieken voor Register-Implementatie:** Om sterke registers te implementeren kunnen deze technieken worden gebruikt: *Silent Write*, *Unaire notatie*, *Serienummers*.
- (a) Hoe werkt de Silent Write?
- (b) Welke techniek gebruikt men om van een regular *bit* naar een *m-valued register* te gaan? Welke techniek gebruikt men om van een *Safe bit* naar een *Regular bit* te gaan?