

Eerste Deeltoets Concurrency

4 oktober 2011, 8.30 – 10.30, Educ- β .

Zet je mobiel uit.

Geef bij al je antwoorden een korte, maar duidelijke motivatie. Stel geen vragen over dit tentamen; als je de formulering onduidelijk vindt, vertel dan hoe je de vraag interpreteert en geef je antwoord.

Gebruik niet meer dan één dubbel vel. Succes!

- 1. Locking.** Voor locken kennen we een *spin-lock* en een *blocking lock*.
 - (a) Leg uit, wat het verschil is tussen deze twee type locks.
 - (b) In welke situaties is een spin-lock beter, en wanneer is een blocking lock beter?
 - (c) Wanneer is er een voordeel van een TTaS-lock boven een TaS-lock?
- 2. Register uit Bit.** Om een m -waardig register te implementeren uit bits gebruiken we een unaire representatie. De `Write(x)` set bit $r[x]$, en *reset* de lagere bits. De `Read` bekijkt bits van laag naar hoog, en returnt de eerste x waarvoor $r[x]$ true is. Deze constructie is behandeld op college.
 - (a) Een karakteristieke eigenschap van registers is: dat een `Write` alle voorgaande toestanden van het register uitwist. In deze register-implementatie zijn “oude” toestanden echter nog zichtbaar als 1-en in de array. Heeft deze implementatie dan wel de karakteristieke register-eigenschap?
 - (b) Een programmeur vindt het mooier om, in een `Write(x)`, ook de *hogere* bits te resetten (zodat na de operatie, precies één bit true is). Laat zien, dat dit tot een foute uitkomst bij `Read` leidt.
- 3. Combinaties en kansen.** Uit een vaas met 12 rode en 8 blauwe ballen worden 4 ballen willekeurig getrokken (*zonder* terugleggen).
 - (a) Wat is de kans dat alle vier deze ballen rood zijn?
 - (b) Wat is de kans dat de trekking 3 rode ballen en 1 blauwe bal oplevert?
 - (c) Wat is $\sum_{k=0}^n 3^{k+1} \binom{n}{k}$?
 - (d) Wat is $\sum_{k=0}^n (2k+1) \binom{n}{k}$?
- 4. Terminatie van threads.** Twee threads delen variabele s , die door thread 1 steeds wordt opgehoogd. Thread 0 leest s en termineert beide threads als s *even* is:

Thread0	Thread1
<code>s = 0; t=0;</code>	<code>b=True;</code>
<code>while (s%2 == 1)</code>	<code>while (b)</code>
<code>{ t++ }</code>	<code>{ s++ }</code>
<code>b=False;</code>	<code>print(t);</code>

- (a) Wanneer is een methode wait-free?
- (b) Is het programma van Thread0 waitfree? Is dat van Thread1 wait-free?
- (c) Is het programma terminerend onder de aannamen van *read-write atomicity* en *fairness*?