

## Eerste Toets Concurrency

17 december 2015, 11.00–13.00, Educ-Γ.

Motiveer je antwoorden *kort!* Zet je mobiel uit. Stel geen vragen over deze toets; als je een vraag niet duidelijk vindt, schrijf dan op hoe je de vraag interpreteert en beantwoord de vraag zoals je hem begrijpt. Opgaven 1 t/m 4 zijn elk 2pt en opgaven 5 t/m 7 zijn elk 3pt. Maak opgaven 1, 2 en 3 op bladzijde 1, opgaven 4 en 5 op bladzijde 2, en opgaven 6 en 7 op bladzijde 3.

- Hardware:** Een processor met vier fysieke cores doet zich naar het operating system voor als een processor met acht cores.
  - Welke technologie wordt hier beschreven?
  - Een zekere applicatie heeft een 20% hogere throughput op deze processor dan op een verder gelijkwaardige processor zonder de bedoelde technologie (dus met “gewoon” vier cores). Hoe valt deze snelheidswinst te verklaren?
- Parallellisme:** Wat is het verschil tussen potential en mandatory concurrency?
- Vectorizatie:** In een simulatie wordt de volgende data gebruikt:  
class Particle, met de properties 'argb' van type Vector4, en 'size', van type float.  
En array particles met 16384 instances van de Particle class.
  - Hoe wordt, in de context van vectorizatie, deze data layout genoemd?
  - Herstructureer de data layout zodat deze geschikt is voor vectorizatie. (Ga uit van 4-wide SIMD.)
- Speculative Execution:** Leg (in circa 10 regels) uit hoe speculative execution kan leiden tot een super-linear speedup bij vectorizatie.
- Het TaS-lock:** Met een enkel TaS-register kun je een multi-threaded lock maken.
  - Geef de code voor dit TaS-lock (lock en unlock methode).
  - Welke van deze eigenschappen heeft het TaS-lock *niet*: **Mutal Exclusion**, **NoDeadlock**, **NoStarvation** en leg uit waarom niet.
  - Welke instructie(s) in C# kun je gebruiken om de TaS-operatie te implementeren?
- Register Implementaties:** Voor het implementeren van wacht-vrije registers bestaan o.a. deze technieken: (A) Copying, (B) Silent Write, (C) Sequence Numbers, (D) Unary Representation.
  - Welke hiervan gebruik je om (i) van een Safe Bit een Regular Bit te maken; (ii) van een Regular Register een Atomic Register te maken; (iii) van een Regular Bit een Regular *m*-Valued Register te maken?
  - Leg uit hoe de Silent Write werkt.
  - Kun je deze technieken combineren om met registers een TaS-instructie te implementeren?
- Consensus Number:** (a) Wat wordt bedoeld met het *Consensus Number* van een object?
  - Wat is het Consensus Number van (i) Het Snapshot Object; (ii) Een Register; (iii) Een Compare-and-Swap; (iv) de Test-And-Set?
  - Is het mogelijk om, uitgaande van alleen registers, een wachtvrije Stack te implementeren? Bewijs je antwoord!