

Tweede Herdeeltoets Concurrency

3 januari 2014, 11.00 – 13.00, Educ- α .

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. Te behalen 13pt, cijfer is som gedeeld door 1,2.

- Integer vermenigvuldigen:** Gegeven integers A en B , als rijen van n cijfers. Je moet het product C berekenen, als rij van $2n$ cijfers. Sequentieel kan dit in $O(n^2)$ tijd door elk cijfer van A te combineren met elk cijfer van B .
 - Geef een parallel algoritme met kwadratische work en lineaire span.
 - Is dit algoritme *efficient*, en is het *optimaal*?
 - Is een beter algoritme dan onder (a) mogelijk?
- Ondergrens:** In een netwerk met n stations heeft station i een invoergetal x_i in het bereik $1 \dots 6$. We willen dat elk station een uitvoer y_i berekent die voldoet aan (1) GELDIG: de uitvoer komt voor als invoer, dus $\exists j : y_i = x_j$; (2) LAAG: de uitvoer moet door minstens de helft worden gedomineerd, dus er zijn minstens $n/2$ stations j met $x_j \geq y_i$. De berekening moet asynchroon zijn en robuust tegen het crashen van t stations, waar $t < n/2$.
 - Is dit probleem een Consensusprobleem (in de betekenis van Fischer, Lynch en Paterson)?
 - Bewijs dat het onoplosbaar is **of** geef een algoritme.
- Gerandomiseerde renaming:** In een systeem van vijf stations moet elk station een unieke vierbits identiteit krijgen. De gekozen strategie is: Elk station kiest een random getal, de getallen worden uitgewisseld en geteld of er vijf verschillende zijn. Als vijf verschillende getallen gekozen zijn, is het klaar. Bij minder dan vijf wordt de procedure herhaald (zo vaak als nodig).
 - Is dit een Monte Carlo of een Las Vegas strategie? Waarom?
 - Wat is de complexiteit (uitgedrukt in aantal ronden)?
 - Vanwege een limiet op de tijd wordt besloten, het algoritme in ieder geval na drie ronden te stoppen (en de stations het laatstgetrokken nummer te laten houden). Wat is de kans dat de uitkomst correct is?
- Synchronisatie: Barrier met Semafoor:** Een *barrier* (voor n threads) kent een methode `signalAndWait()` (in Java: `await`) die door alle threads kan worden aangeroepen. In deze opgave kijken we naar een implementatie waarin elke thread i een eigen *semafoor* $s[i]$ heeft. Aan het eind van de `await()` doet thread i een reeks van n Acquires op zijn eigen semafoor:

```
void signalAndWait()
{ ...
  for (int j=0; j<n; j++) s[i].Acquire;
}
```

 - Wat is de eigenschap waaraan de methode `signalAndWait()` moet voldoen?
 - Welke Release-operaties kun je op de stippeltjes invullen om aan de barrier-eis te voldoen?
- Welordeningen:** Welordeningen zijn van belang bij het bewijzen dat een algoritme termineert. Zeg van deze verzamelingen of ze een welordering zijn en waarom:
 - Natuurlijke getallen, met normale ordening: $(\mathbb{N}, <)$.
 - Strings over $\{a, b, c\}$, alfabetisch geordend.
 - Paren natuurlijk getallen, met lexicografische ordening, dus $(a, b) < (c, d)$ als $a < c$ of $(a = c \wedge b < d)$.
 - Rationele getallen, met normale ordening: $(\mathbb{Q}, <)$.