Tentamen Concurrency, 7 februari 2007

Tijdsduur 3 uur. Gesloten boek tentamen.

Voorzie alle in te leveren bladen van je naam, en nummer ze. Schrijf op het eerste blad het aantal ingeleverde bladen.

Geef steekhoudende argumenten voor de correctheid van je oplossingen en je andere beweringen.

Werk netjes. Formuleer scherp en zorgvuldig. Schrijf duidelijk leesbaar.

Opgave 1 (40 %). Gegeven zijn gehele getallen M en N met $1 \leq M \leq N$. We beschouwen een systeem met N processen en twee gedeelde variabelen volgens

```
var n : int := 0 , b : bool := false ;

process ThreadC (self := 0 to N − 1)
   do true  →
10:      TNS1
11:      ⟨ await ¬b then n ++ ; b := (n = M) ⟩
12:      ⟨ await b ⟩
13:      TNS2
14:      ⟨ n −− ; b := (n > 0) ⟩
   od
end ThreadC .
```

De commando's *TNS1* en *TNS2* zijn gegeven, eindigen gegarandeerd en modificeren de gedeelde variabelen b en n niet.

(a: 10 %) Formuleer en bewijs zo sterk mogelijke zinvolle invarianten van de vorm:

(J0) $\quad n = \#\{q \mid \ldots\}$,
(J1) $\quad \neg b \;\Rightarrow\;$ "*iets over* n",
(J2) $\quad b \;\Rightarrow\;$ "*iets over* n",
(J3) $\quad q \text{ in } \{\ldots\} \;\Rightarrow\; b$.

(b: 10 %) Er gelden de voorgangscondities:

(V0) $\quad \neg b \;\; o\!\!\rightarrow \;\; b$,
(V1) $\quad b \;\; o\!\!\rightarrow \;\; \neg b$.

Wat betekent dit (geef de definities)? Bewijs deze voortgangscondities.

(c: 20 %) Implementeer dit system (in het bijzonder de atomaire commando's 11, 12 en 14) met één of meer gesplitste binaire semaforen. Het moet hierbij mogelijk zijn dat meerdere processen tegelijk *TNS1* of *TNS2* (of beide) uitvoeren.

Opgave 2 (30 %). We beschouwen een systeem waarin $N > 1$ processen alle van tijd tot tijd een gedeelde variabele x moeten wijzigen met behulp van privé gegevens en een gegeven functie f volgens de specificatie:

```
var x : Item := x0 ;

process Thread(self := 0 to N − 1)
   var priv : Data
   do true  →
      NCS   {kan priv wijzigen}
      ⟨ x := f(priv, x) ⟩
   od end .
```

Voorgesteld wordt dit te implementeren volgens:

Allard


```
        var list : set of Process := ∅ ;

        process Thread(self := 0 to N − 1)
           var priv : Data ; it : Item ; bb : bool
           do true  →
10            NCS   {kan priv wijzigen}
11            bb := false
12            do ¬ bb  →
13               ⟨ it := x ; list := list ∪ {self} ⟩ ;
14               it := f(priv, it)
15               ⟨ if  self ∈ list  →
                     x := it ; list := ∅ ; bb := true  fi ⟩
              od
           od end Thread .
```

Commando 13 heet LL (*load-linked*); commando 15 heet SC (*store-conditional*).
(a) Bewijs, dat een process p de binnenlus van 12 tot 15 dan en alleen dan verlaat als hij een toekenning aan x doet. Welke invariant heb je hier nodig?
(b) Bewijs, dat als process p een toekenning aan x doet, deze variabele de waarde $f(priv.p, \mathrm{x})$ krijgt. Formuleer en bewijs hiertoe geschikte invarianten.

Opgave 3 (30 %). Beschouw een broadcast-systeem met één zendend proces en N ontvangende processen, die communiceren middels gedeelde variabelen volgens

```
var boodschap: Boodschap := null

process Zender
   var b: Boodschap
   do true ->
      b := verzin()
      boodschap := b
   od end Zender

process Ontvanger (self := 0 to N-1)
   var b: Boodschap
   do true ->
      b := boodschap
      verwerk(b)
   od end Ontvanger
```

Dit systeem dient met gedeelde variabelen zo gesynchroniseerd te worden, dat elke boodschap van de zender door *elke* ontvanger *precies één* keer ontvangen wordt.

Gebruik hiertoe simpele **await** statements en gedeelde en/of privé variabelen, zodanig dat geen van de processen onnodig hoeft te wachten. Je mag desgewenst onbegrensde integers gebruiken. Een gedeelde integer variabele mag atomair met 1 verhoogd of verlaagd worden. Maak de correctheid aannemelijk met behulp van invarianten.