

## Tentamen Concurrency, 19 april 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. Werk netjes, formuleer scherp en zorgvuldig. Schrijf duidelijk leesbaar.

**Opgave 1** (25 %). Gegeven is een systeem met twee processen

```

var active[0: 1]: bool := ([2] false) ;
var turn: int := 0 ;

process Partner (self := 0 to 1)
  do true ->
10:      NCS
11:      active[self] := true
12:      if active[1-self] ->
13:          turn := 1-self
14:          await (turn = self or not active[1-self])
          fi
15:      CS
16:      active[self] := false
  od
end Partner

```

NCS is de niet-critische sectie; dit commando hoeft niet te eindigen. CS is de kritische sectie, die altijd gegarandeerd eindigt en die alleen onder wederzijdse uitsluiting uitgevoerd mag worden.

- Geef een formele specificatie van wederzijdse uitsluiting voor dit systeem.
- Bewijs dat dit systeem aan wederzijdse uitsluiting voldoet, volgens je specificatie van onderdeel (a). Geef alle benodigde invarianten en bewijs hun invariantie.

**Opgave 2** (15 %). Gegeven zijn de commando's  $nA$ ,  $nB$  en  $nC$ ,  $cA$ ,  $cB$  en  $cC$ , en een systeem van een groot aantal processen die alle de volgende oneindige lus uitvoeren:

```

do
  true → nA ; preA ; cA ; postA
|| true → nB ; preB ; cB ; postB
|| true → nC ; preC ; cC ; postC
od .

```

We stellen de drie synchronisatieeisen:

$$\begin{aligned}
 \#\{q \mid q \text{ in } cA\} &= 0 \vee \#\{q \mid q \text{ in } cB\} = 0, \\
 \#\{q \mid q \text{ in } cB\} &= 0 \vee \#\{q \mid q \text{ in } cC\} = 0, \\
 \#\{q \mid q \text{ in } cC\} &\leq 1.
 \end{aligned}$$

Implementeer deze synchronisatieeisen zo nondeterministisch mogelijk door het invullen van de commando's  $preA$ ,  $preB$ ,  $preC$ ,  $postA$ ,  $postB$  en  $postC$  met behulp van gedeelde variabelen, atomiciteitshaakjes en simpele en compound await statements. Houd het eenvoudig. Laat zien dat je oplossing correct is.

**Z.O.Z.**

**Opgave 3** (40 %). Gegeven zijn gehele getallen  $1 \leq M \leq N$  en een systeem met  $N$  processen van de vorm:

```

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

process Member (self := 0 to N - 1)
do true →
10:    TNS0
11:    ( await ¬b then n ++ ; b := (n = M) )
12:    ( await b )
13:    TNS1
14:    ( k ++ )
15:    ( await k = n then k -- ; n -- ; b := (n ≠ 0) )
od
end Member .

```

De commando's *TNS0* en *TNS1* zijn ook gegeven. Ze wijzigen de gedeelde variabelen  $k$ ,  $n$ ,  $b$  niet en zijn verder niet van belang.

(a: 20%) Formuleer en bewijs invarianten die de waarden van  $n$ ,  $k$ ,  $b$  en  $M$  relateren aan elkaar en aan het aantal processen die op bepaalde locaties in hun programma zijn. Deze invarianten moeten sterk genoeg zijn om daarmee te bewijzen dat geldt:

(\*)  $q \text{ at } 13 \Rightarrow n = M$ .

Bewijs dat (\*) inderdaad uit je invarianten volgt.

(b: 20%) Implementeer dit systeem met één of meer gesplitste binaire semaforen.

**Opgave 4** (20 %). Gegeven is een natuurlijk getal  $k$ .

(a: 10%) Laat zien hoe een semafoor  $s$  met initialisatie  $s := k$  geïmplementeerd kan worden met de pthread primitieven mutex en conditievariabele.

(b: 10%) Laat zien hoe een semafoor  $s$  met initialisatie  $s := k$  geïmplementeerd kan worden met boodschappen (messages).