

Tentamen "Co-operating Sequential Processes".

Aan een grote ronde tafel zijn de plaatsen cyclisch genummerd van 0 t/m 39; de leden van een gezelschap van 40 personen zijn eveneens genummerd van 0 t/m 39, ieder mag alleen op zijn eigen plaats -dwz. de plaats, die zijn eigen nummer heeft- aan tafel aanzitten.

Hun gedragpatroon bestaat uit een niet-eindigende cyclische opeenvolging ~~van~~ van:

- leven (buitenskamers);
- borrelen (binnenskamers, maar niet aan tafel);
- eten (binnenskamers, aan tafel);

en alleen voor de heren

- sigaarroken (binnenskamers, niet aan tafel).

Hun gedragingen moeten zo gesynchroniseerd zijn dat

- 1) nimmer drie cyclisch opeenvolgende plaatsen aan tafel tegelijkertijd bezet zijn, en
- 2) geen dame van tafel opstaat, tenzij er tenminste 1 andere dame in de kamer is.

Voor de dames (die qualitate qua na tafel geen sigaar roken) is het opstaan van tafel en het verlaten van de kamer eenzelfde handeling. Eventuele personen, die met aan tafel gaan moeten wachten (wegens regel 1) op het van tafel opstaan van dame A, die ~~wag~~ (wegens regel 2) met opstaan moet wachten totdat, zeg, dame B binnenkomt, hebben ten ~~XXXX~~ aanzien van het aan tafel gaan voorrang boven dame B.

Er zijn tenminste twee dames in het gezelschap, dat met iedereen buitenskamers levend geïntialiseerd wordt.

De oplossing acht in het omvattend universum gedeclareerd:

- 1) de boolean procedure dame(u); value u; integer u;
deze heeft de waarde "true" als persoon nr. u een dame is, de waarde "false" als persoon nr. u een heer is.
- 2) de integer procedure mod40(u); value u; integer u;
deze heeft de waarde van u, gereduceerd modulo 40.

Gegeven is een oplossing van de volgende structuur.

```

begin semaphore mutex, damutex; semaphore array persem [0:39];
  boolean array honger [0:39]; integer array eettal [0:39]; integer damnr, k;
  procedure test(u); value u; integer u;
  begin if honger[u] then
    begin if eettal[mod40(u-1)] < 2 and
      eettal[u] < 2 and
      eettal[mod40(u+1)] < 2 then
        begin eettal[mod40(u-1)] := eettal[mod40(u-1)] + 1;
          eettal[u] := eettal[u] + 1;
          eettal[mod40(u+1)] := eettal[mod40(u+1)] + 1;
          honger[u] := false; V(persem[u])
        end
      end
    end
  end;
  -----
  procedure staop(u); value u; integer u;
  begin eettal[mod40(u-1)] := eettal[mod40(u-1)] - 1;
    eettal[u] := eettal[u] - 1;
    eettal[mod40(u+1)] := eettal[mod40(u+1)] - 1;
    test(mod40(u-2)); test(mod40(u-1)); test(mod40(u+1)); test(mod40(u+2))
  end;
  for k := 0 step 1 until 39 do
    begin honger[k] := false; eettal[k] := 0; persem[k] := 0 end;
  mutex := 1; damutex := 1; damnr := -1; k := 0;

  parbegin
  persoon 0: begin..... end;
    :
  persoon39: begin..... end
  parend
end

```

waarbij de structuur van persoon h is als volgt:

```

persoon h:
  begin
    Lh: leven;
      if dame(h) then
        begin P(damutex); k := k + 1;
          if damnr  $\geq$  0 then
            begin P(mutex); staop(damnr); V(mutex); V(persem[damnr]); damnr := -1 end;
            V(damutex)
          end;
        borrelen;
        P(mutex); honger[h] := true; test(h); V(mutex); P(persem[h]);
        eten;
        if dame(h) then
          begin P(damutex); k := k - 1;
            if k = 0 then
              begin damnr := h; V(damutex); P(persem[h]) end
            else
              begin P(mutex); staop(h); V(mutex); V(damutex) end
            end
          end
        else
          begin P(mutex); staop(h); V(mutex);
            sigaarroken
          end;
        goto Lh
      end
    end

```

Opgave. Bewijs, dat de gegeven oplossing aan alle specificaties voldoet.

NB. De tentaminandi wordt verzocht hun papier slechts eenzijdig te beschrijven.