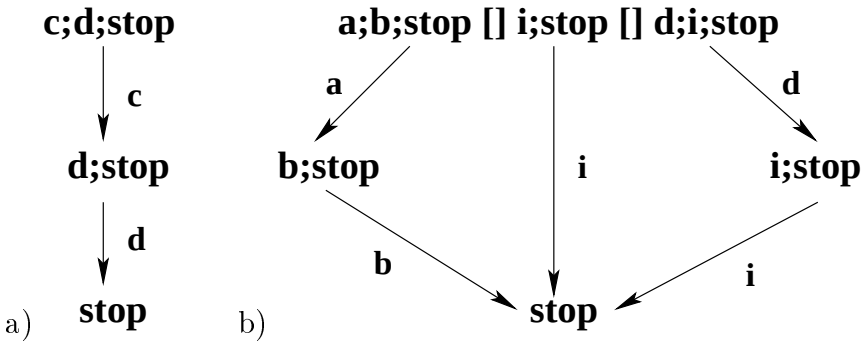
Spesifioinnin ja verifioinnin perusteet

Harjoitus 5, 15.2.2008

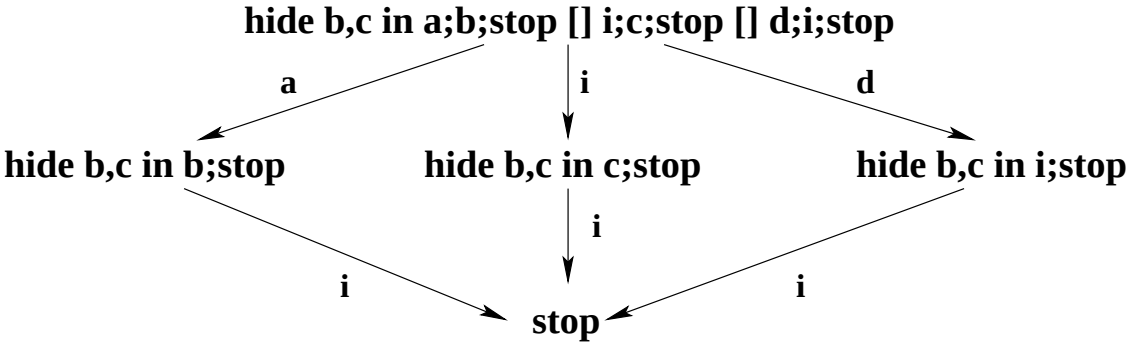
1. Piirrä seuraavia Lotos-lausekkeita vastaavat siirtymäsysteemit. Käytä tilan nimenä Lotoslauseketta.

   (a) `c;d;stop`

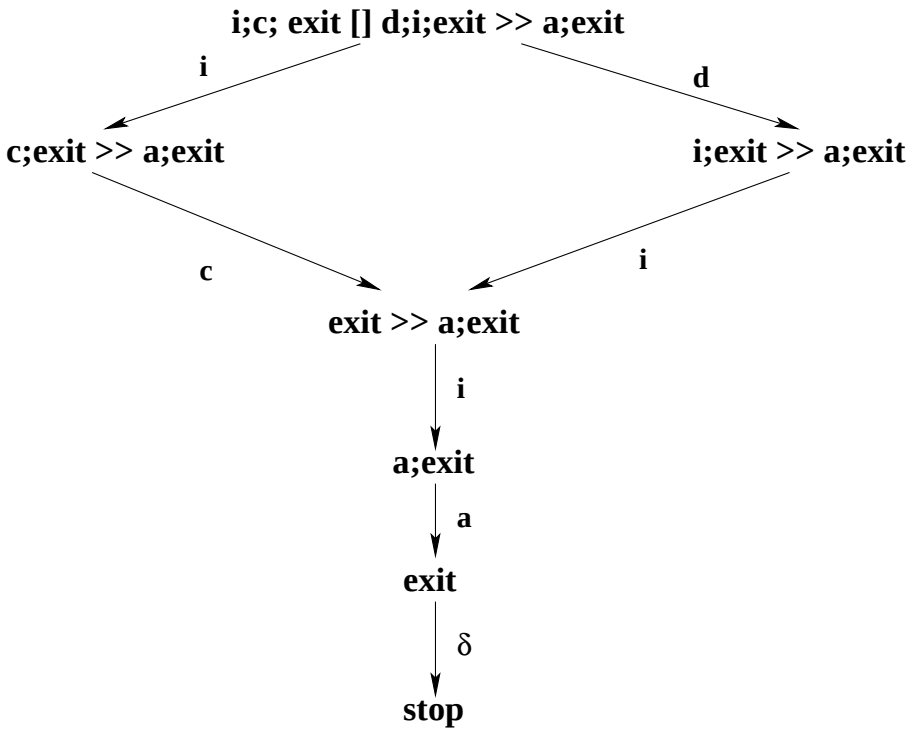
   (b) `a;b;stop [] i;stop [] d;i;stop`



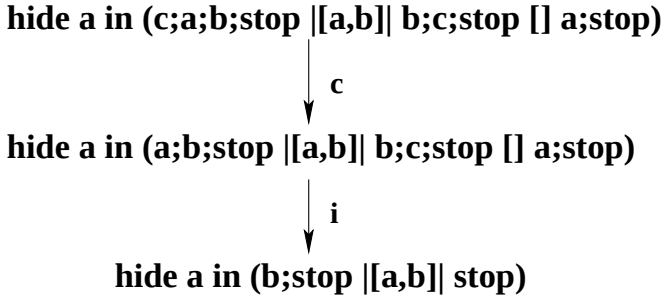
   (c) `hide b, c in a;b;stop [] i;c;stop [] d;i;stop`



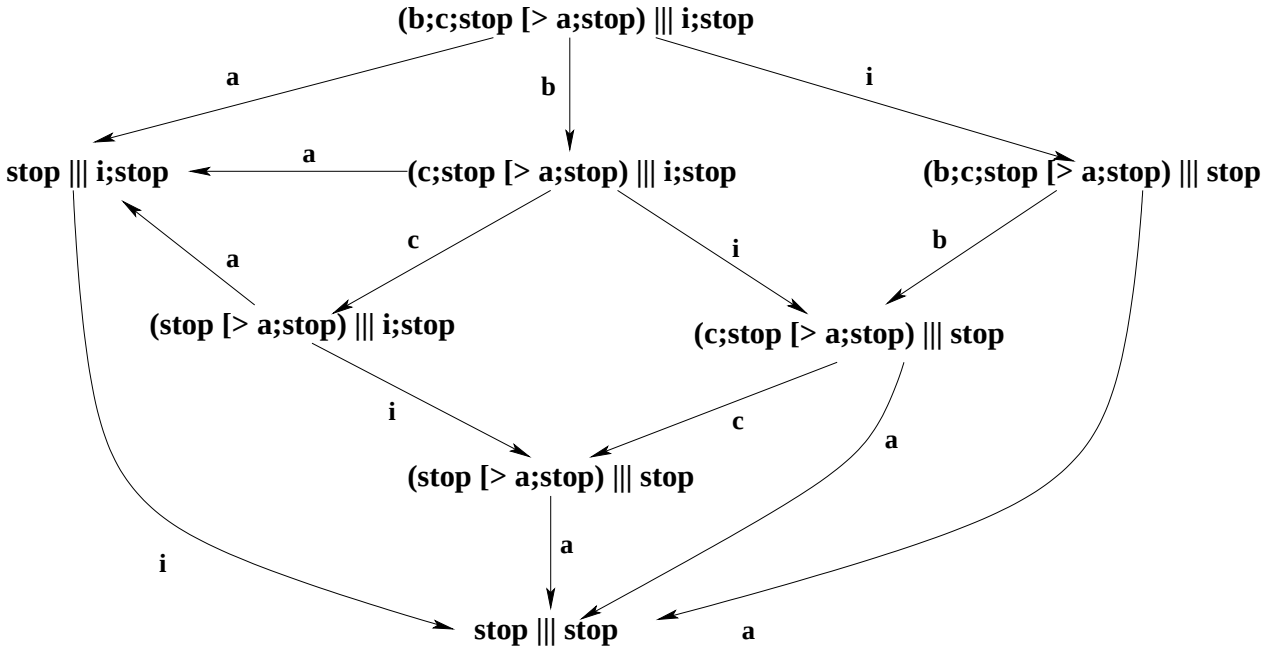
(d) `i;c; exit [] d;i;exit >> a;exit`



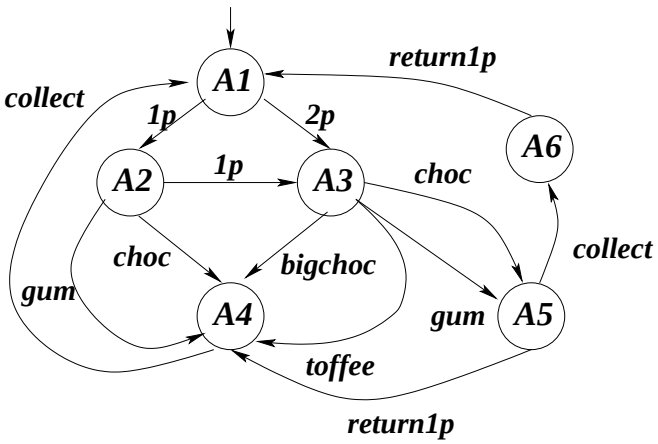
(e) `hide a in (c;a;b;stop |[a,b]| b;c;stop [] a;stop)`



(f) `(b;c;stop [> a;stop) ||| i;stop`



2. Muodosta Lotoksen avulla siirtymäsysteemi, joka kuvaa seuraavaa harjoituksissa kaksi ollutta karkkiautomaattia.



```
specification machine [p1, p2, choc, bigchoc, toffee, gum, collect, return1p] : noexi

        A1 [p1, p2, choc, bigchoc, toffee, gum, collect, return1p]
where

process A1 [p1, p2, choc, bigchoc, toffee, gum, collect, return1p] :
noexit :=
        p1; A2[p1, p2, choc, bigchoc, toffee, gum, collect, return1p]
         []
        p2; A3[p1, p2, choc, bigchoc, toffee, gum, collect, return1p]
endproc

process A2 [p1, p2, choc, bigchoc, toffee, gum, collect, return1p] :
noexit :=
        p1; A3[p1, p2, choc, bigchoc, toffee, gum, collect, return1p]
         []
        choc; A4[p1, p2, choc, bigchoc, toffee, gum, collect, return1p]
         []
        gum; A4[p1, p2, choc, bigchoc, toffee, gum, collect, return1p]
endproc

process A3 [p1, p2, choc, bigchoc, toffee, gum, collect, return1p] :
noexit :=
        bigchoc; A4[p1, p2, choc, bigchoc, toffee, gum, collect,
return1p]
         []
        toffee; A4[p1, p2, choc, bigchoc, toffee, gum, collect,
return1p]
         []
        choc; A5[p1, p2, choc, bigchoc, toffee, gum, collect, return1p]
         []
        gum; A5[p1, p2, choc, bigchoc, toffee, gum, collect, return1p]
endproc

process A4 [p1, p2, choc, bigchoc, toffee, gum, collect, return1p] :
noexit :=
        collect; A1[p1, p2, choc, bigchoc, toffee, gum, collect,
return1p]
endproc

process A5 [p1, p2, choc, bigchoc, toffee, gum, collect, return1p] :
noexit :=
```

```
        return1p; A4[p1, p2, choc, bigchoc, toffee, gum, collect,
return1p]
         []
        collect; A6[p1, p2, choc, bigchoc, toffee, gum, collect,
return1p]
endproc

process A6 [p1, p2, choc, bigchoc, toffee, gum, collect, return1p] :
noexit :=
        return1p; A1[p1, p2, choc, bigchoc, toffee, gum, collect,
return1p]
endproc
endspec
```

Kaikista tiloista ei välttämättä tarvitse tehdä omaa Lotos-prosessia vaikka edellä esitetyssä ratkaisussa on näin suoraviivaisesti tehty.

3. Spesifioi Lotoksella muuttuja, jonka arvona voi olla 1 tai 2. Spesifioi Lotoksella totuusarvomuuttuja, joka voi saada arvoja *false* ja *true*.

```
process in1f[in1rf,in1rt,in1wf,in1wt] : noexit :=
    in1rf;in1f[in1rf,in1rt,in1wf,in1wt] []
    in1wf;in1f[in1rf,in1rt,in1wf,in1wt] []
    in1wt;in1t[in1rf,in1rt,in1wf,in1wt]
endproc
process in1t[in1rf,in1rt,in1wf,in1wt] : noexit :=
    in1rt;in1t[in1rf,in1rt,in1wf,in1wt] []
    in1wt;in1t[in1rf,in1rt,in1wf,in1wt] []
    in1wf;in1f[in1rf,in1rt,in1wf,in1wt]
endproc

process K1[kr1,kr2,kw1,kw2] : noexit :=
    in1f[kr1,kr2,kw1,kw2]
endproc
```

4. Spesifioi Lotoksella Hymannin poissulkemisprotokolla kahden prosessin tapauksessa. Pseudokoodi jäljessä.

```
boolean in1,in2;
int k;
```

Alussa $in1 = in2 = false$ ja $k = 1$. Muuttujat voidaan mallintaa Lotosprosesseina. Totuusarvomuuttujat voivat siis saada arvoja $false$ ja $true$. Muuttuja k voi saada arvoja 1 ja 2.

Prosessin P1 koodi:

```
while (true)
{
    /* ei-kriittinen ohjelmanosa */
    in1 = true;
    while (k != 1)
    {
          while (in2) {skip};
          k = 1
    }
    /* kriittinen ohjelmanosa */
    in1 = false;
}
```

Komento *skip* ei tee mitään, joten se voidaan mallintaa i-tapahtumana.

Prosessin P2 koodi:

```
while (true)
{
    /* ei-kriittinen ohjelmanosa */
    in2 = true;
    while (k != 2)
    {
          while (in1) {skip};
          k = 2
    }
    /* kriittinen ohjelmanosa */
    in2 = false;
}
```

Mallinna $P1$:n kriittisen alueen laskenta kahdella toiminnolla $incs1$ ja $outcs1$, missä ensimmäinen suoritetaan kriittisen alueen alussa ja jälkimmäinen juuri ennen poistumista kriittiseltä alueelta, vastaavasti $P2$:lle toiminnot $incs2$ ja $outcs2$.

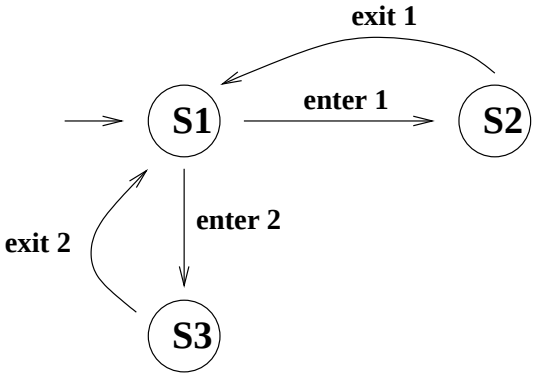
Tapahtumat on nimetty niin, että in<i>rt tarkoittaa, että luetaan arvo true muuttujasta in<i>. Vastaavasti in<i>rf tarkoittaa false arvon lukemista. Kirjoittamiselle

on vastaavasti tapahtumat in<i>wt ja in<i>wf. Vastaavalla tavalla on nimetty muuttujaa k vastaavan prosessin tapahtumat, mutta nyt arvoina mahdollisina arvoina on vain 1 ja 2. Tapahtumat ovat siis kr1,kr2,kw1,kw2.

```
specification Hyman[incs1, outcs1, incs2, outcs2] : noexit behaviour
hide in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,kr1,kr2,kw1,kw2
in
(P1[incs1,outcs1,in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,
    kr1,kw1,kr2,kw2]
|||
P2[incs2,outcs2,in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,
           kr1,kw1,kr2,kw2])
|[in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,kr1,kr2,kw1,kw2]|
(in1f[in1rt,in1rf,in1wt,in1wf]
|||
in2f[in2rt,in2rf,in2wt,in2wf]
|||
K1[kr1,kr2,kw1,kw2])

where

process P1[incs1,outcs1,in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,
           kr1,kw1,kr2,kw2] : noexit :=

in1wt;P11[incs1,outcs1,in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,
           kr1,kw1,kr2,kw2]
endproc

process P11[incs1,outcs1,in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,
           kr1,kw1,kr2,kw2] : noexit :=
  kr1;P13[incs1,outcs1,in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,
           kr1,kw1,kr2,kw2] []
  kr2;P12[incs1,outcs1,in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,
           kr1,kw1,kr2,kw2]
endproc

process P12[incs1,outcs1,in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,
           kr1,kw1,kr2,kw2] : noexit :=

in2rf;kw1;P11[incs1,outcs1,in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,
           kr1,kw1,kr2,kw2] []

in2rt;i;P12[incs1,outcs1,in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,
```

```
          kr1,kw1,kr2,kw2]
endproc

process P13[incs1,outcs1,in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,
          kr1,kw1,kr2,kw2] : noexit :=

incs1;outcs1;in1wf;P1[incs1,outcs1,in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,
          kr1,kw1,kr2,kw2]
endproc
process P2[incs2,outcs2,in1rt,in1rf,in1wt,in1wf,in2rt,in2rf,in2wt,in2wf,
          kr1,kw1,kr2,kw2] : noexit :=
  P1[incs2,outcs2,in2rt,in2rf,in2wt,in2wf,in1rt,in1rf,in1wt,in1wf,
          kr2,kw2,kr1,kw1]
endproc

process in1t[in1rt,in1rf,in1wt,in1wf] : noexit :=
               in1rt;in1t[in1rt,in1rf,in1wt,in1wf] []
               in1wt;in1t[in1rt,in1rf,in1wt,in1wf] []
               in1wf;in1f[in1rt,in1rf,in1wt,in1wf]
endproc
process in1f[in1rt,in1rf,in1wt,in1wf] : noexit :=
               in1rf;in1f[in1rt,in1rf,in1wt,in1wf] []
               in1wf;in1f[in1rt,in1rf,in1wt,in1wf] []
               in1wt;in1t[in1rt,in1rf,in1wt,in1wf]
endproc


process in2f[in2rt,in2rf,in2wt,in2wf] : noexit :=
in1f[in2rt,in2rf,in2wt,in2wf]
endproc

process K1[kr1,kr2,kw1,kw2] : noexit := in1t[kr1,kr2,kw1,kw2]
endproc

endspec (* Hyman *)
```

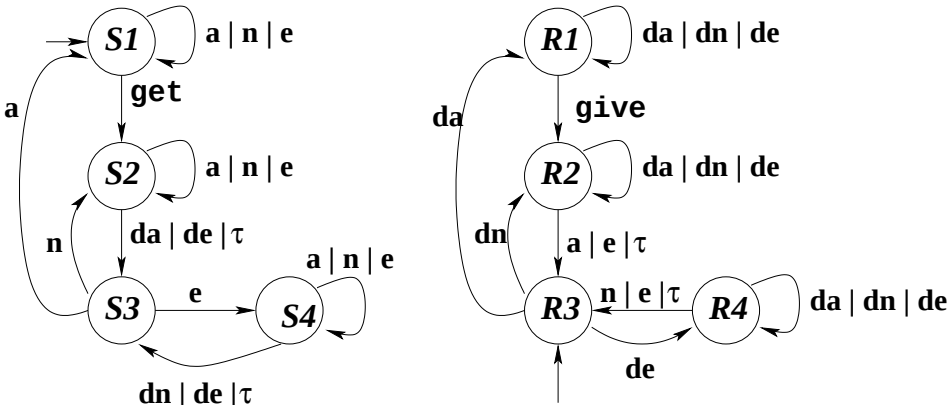
Mikä olisi nyt sopiva palvelukuvaus, joka siis kuvaisi tilanteen *prosessit eivät ole koskaan yhtäaikaa kriittisellä alueella.*



5. Muodosta edellisen tehtävän yhteistilaverkko Aldebarania käyttäen.

   Yhteistilaverkkoon tulee tiloja 81 ja siirtymiä 162, jos käytetään edellä esitettyä mallinnustapaa.

6. Spesifioi Lotoksella oheisen virheellisen protokollan FE lähettäjä- ja vastaaanottajaprosessit. Sanoman katoamista on mallinnettu τ-siirtymillä. Mallinna Lotos-lausekkeena, miten koko systeemi koostetaan. Muodosta yhteistilaverkko CADP-ohjelmistoa käyttäen.




```
specification testi [get,give,da,dn,de,a,n,e] : noexit behaviour

        S1 [get,da,dn,de,a,n,e] |[da,dn,de,a,n,e]|R3 [give,da,dn,de,a,n,e]
where

process S1 [get,da,dn,de,a,n,e] : noexit :=
        a; S1[get,da,dn,de,a,n,e]
         []
        n; S1[get,da,dn,de,a,n,e]
```

```
          []
        e; S1[get,da,dn,de,a,n,e]
          []
        get; S2[get,da,dn,de,a,n,e]
endproc

process S2 [get,da,dn,de,a,n,e] : noexit :=
        a; S2[get,da,dn,de,a,n,e]
          []
        n; S2[get,da,dn,de,a,n,e]
          []
        e; S2[get,da,dn,de,a,n,e]
          []
        da; S3[get,da,dn,de,a,n,e]
          []
        de; S3[get,da,dn,de,a,n,e]
          []
        i; S3[get,da,dn,de,a,n,e]
endproc

process S3[get,da,dn,de,a,n,e] : noexit :=
        a; S1[get,da,dn,de,a,n,e]
          []
        n; S2[get,da,dn,de,a,n,e]
          []
        e; S4[get,da,dn,de,a,n,e]
endproc

process S4[get,da,dn,de,a,n,e] : noexit :=
        a; S4[get,da,dn,de,a,n,e]
          []
        n; S4[get,da,dn,de,a,n,e]
          []
        e; S4[get,da,dn,de,a,n,e]
          []
        dn; S3[get,da,dn,de,a,n,e]
          []
        de; S3[get,da,dn,de,a,n,e]
          []
        i; S3[get,da,dn,de,a,n,e]
endproc

process R3[give,da,dn,de,a,n,e] : noexit :=
```

```
        S3[give,a,n,e,da,dn,de]
endproc
endspec
```

Yhteistilaverkkoon tulee tiloja 8 ja siirtymiä 15, jos käytetään edellä esitettyä mallinnustapaa. Alla on ohjelmiston vaihtoehdon "Properties" antamat tiedot yhteistilaverkosta.

```
bcg_io FE2.aut /tmp/xeuca_31980_32084_0.bcg
Converted FE2.aut to /tmp/xeuca_31980_32084_0.bcg
bcg_info /tmp/xeuca_31980_32084_0.bcg
/tmp/xeuca_31980_32084_0.bcg:
created by bcg_io from aldebaran format
8 states
15 transitions
9 labels
initial state: 0
list of deadlock state(s): 4
branching factor: average = 1.88, minimal = 0, maximal = 3
4 transition(s) with a hidden label ("i")
deterministic behavior for all labels
```

Yhteistilaverkko on seuraava: