

## Deel A

### Opgave 1

a. Massa-balans vergelijkingen:

$$\begin{aligned}\lambda_{W_1} &= \lambda + \delta_{W_4W_1} = \lambda + 0.6\delta_{W_4} \\ &= \delta_{W_1} + \text{'scrap'} = \delta_{W_1} + 0.2\lambda_{W_1} \\ \lambda_{W_2} &= \delta_{W_1W_2} = 0.5\delta_{W_1} \\ &= \delta_{W_2} + \text{'scrap'} = \delta_{W_2} + 0.1\lambda_{W_2} \\ \lambda_{W_3} &= \delta_{W_1W_3} = 0.5\delta_{W_1} \\ &= \delta_{W_3} \\ \lambda_{W_4} &= \delta_{W_2W_4} + \delta_{W_3W_4} = \delta_{W_2} + \delta_{W_3} \\ &= \delta_{W_4}\end{aligned}$$

b. Resultaat:

$$\begin{aligned}\lambda_{W_1} &= 1.838\lambda \\ \lambda_{W_2} &= 0.735\lambda \\ \lambda_{W_3} &= 0.735\lambda \\ \lambda_{W_4} &= 1.397\lambda \\ \delta &= 0.559\lambda\end{aligned}$$

c. De maximale doorzet  $\delta_{\max} = 0.2$  [lots/uur] wordt bereikt bij:

- Werkstation 1: 3 werknemers
- Werkstation 2: 1 werknemer
- Werkstation 3: 1 werknemer
- Werkstation 4: 2 werknemers

### Opgave 2

Volgorde  $M_2 \rightarrow M_1$  resulteert in de kortste doorlooptijd.

### Opgave 3

- a. WAAR
- b. NIET WAAR
- c. NIET WAAR
- d. NIET WAAR
- e. NIET WAAR

## Deel B

### Opgave 4

Omdat  $1 + 1/2 + 1/4 + 1/8 + \dots = 2 < 3$  zal het programma altijd blijven doorlopen.  
(2.0 wordt ook geaccepteerd als antwoord)

### Opgave 5

- `2*(3*nat)`
- `[]`

### Opgave 6

```
proc AP(chan a?,b?,c?,e! : lot, d! : asy)=
|[ var p1, p2, p3 : lot, g : bool
:: *( ( g:= true; g *> ( a?p1
                    ; delay 5.0
                    ; g:= not check(p1)
                    ; not g -> e!p1
                    )
      || b?p2
      || c?p3
      )
; delay 10.0
; d!<p1,p2,p3>
)
]|
```

### Opgave 7

```
proc Bufreg(chan a?, b! : lot, val t0 : real)=
|[ var x : lot, xs : [lot]
:: *( a?x; xs:= xs ++ [x] )
|| *( len(xs)>0 -> b!hd(xs)
    ; xs:= tl(xs)
    ; delay t0
    )
]|
```

Opmerking: tijdens het "delay" moet de buffer ook lots kunnen ontvangen. In bovenstaande oplossing is daarom gebruik gemaakt van een parallelle compositie. Het is echter ook mogelijk dit probleem op te lossen met behulp van alternatieve compositie:

```
proc Bufreg(chan a?, b! : lot, val t0 : real)=
|[ var x : lot, xs : [lot], go : bool = true, t : real = t0
:: *( a?x; xs:= xs ++ [x]
    | not go -> delay (t-time); go:= true
    | (go and len(xs) > 0) -> b!hd(xs); xs:= tl(xs); go:= false; t:= time + t0
    )
]|
```