



# APPLICATION PROGRAMMER PERSPECTIVE

shared memory

```
{  
t → f(y);  
}
```

com/nom  
seq server

```
{  
call f(y);  
}
```

create req struct  
connect to server socket  
write request  
read answer  
return answer

com server

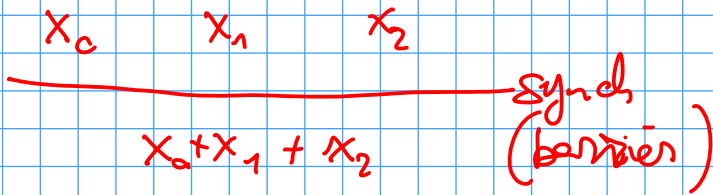
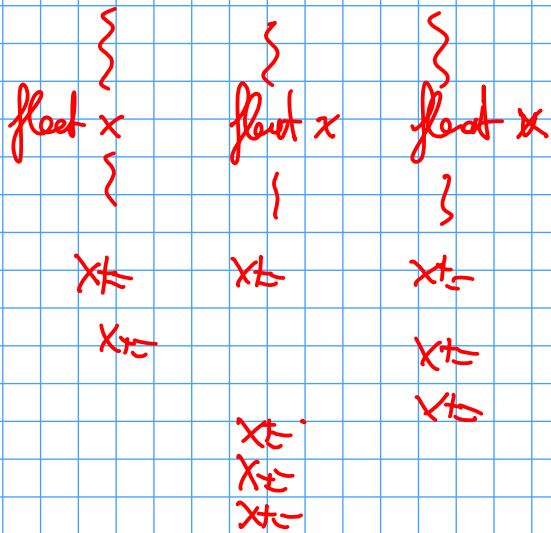
```
{  
call f(y)  
}
```



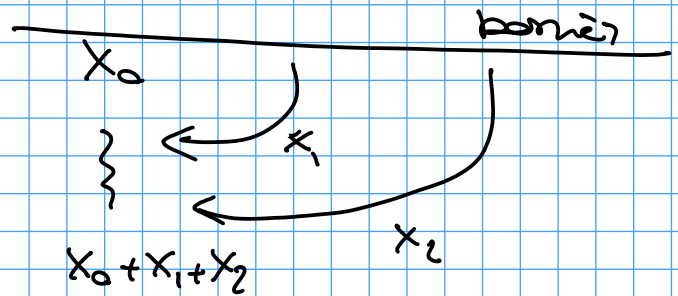
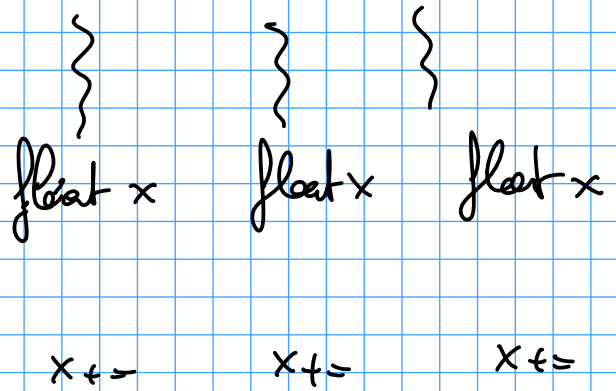
||

Manager can "accumulate" kind of state

she mem



can/won



als snel  
 $s_1 s_2 s_3 \dots$

$\int_a$  implementatie

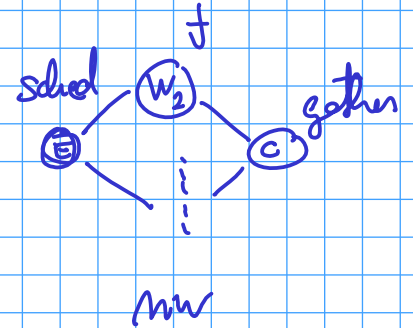
TEMPLATE BASED IMPLEMENTATION

Template : parametric process network implementing a skeleton

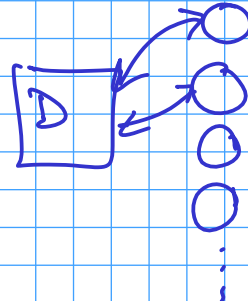
CA

form is a skeleton

① form template



② form template

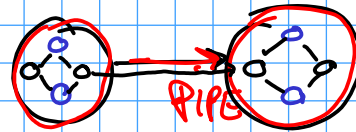


TEMPLATE LIBRARY

Ski	hw	Templij	Cost model
	hw	Templij+1	"
	hw	Templij+2	"

Compiler

labelled SKELETON TREE

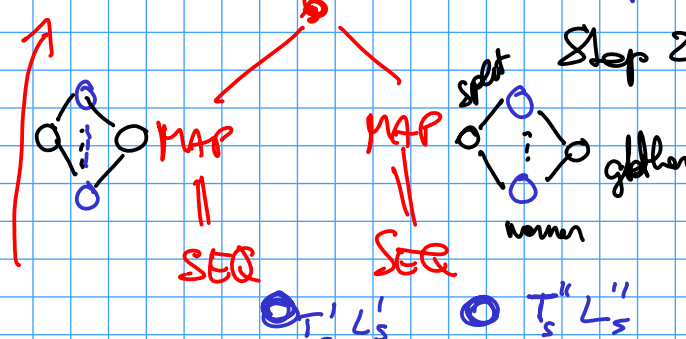


Step 1

Step 2

Step 3

Other templates

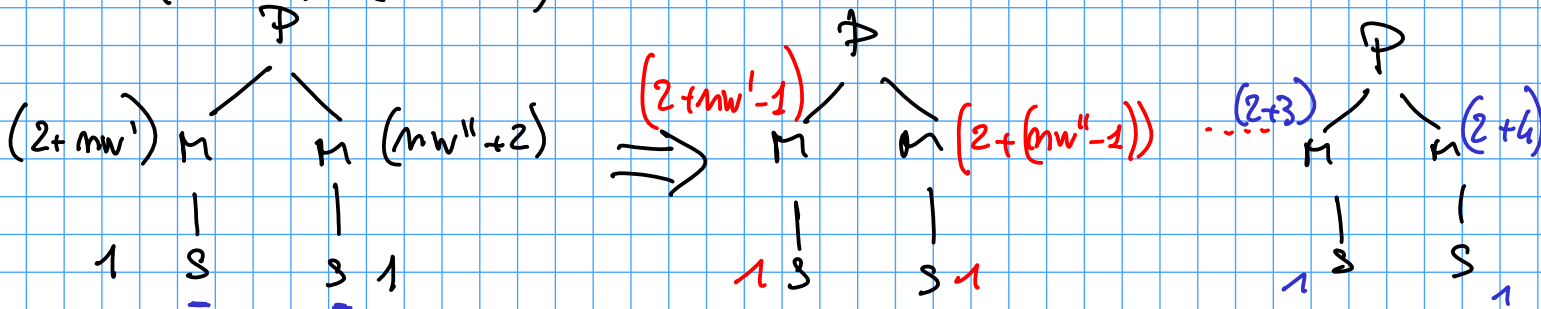


$$nw \approx \left\lceil \frac{tw}{te} \right\rceil$$

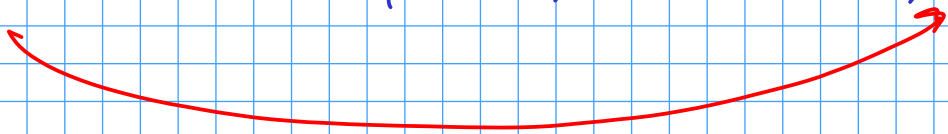
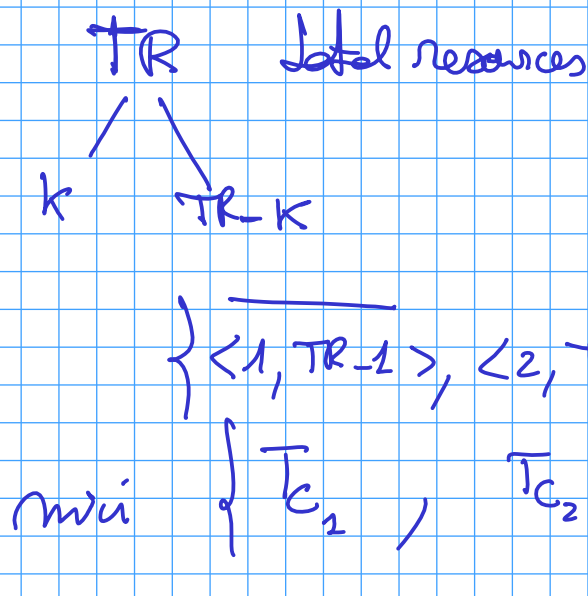
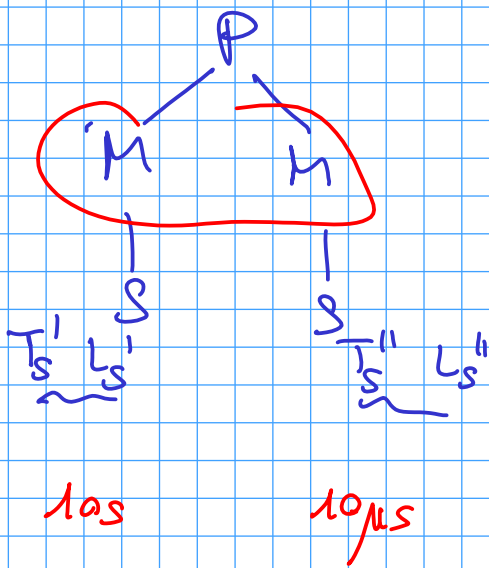
$$TR = \sum \text{resources}$$

$ln \ll \# \text{ resources} \ll TR$

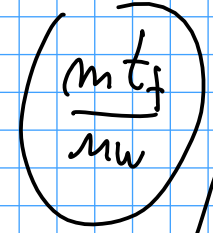
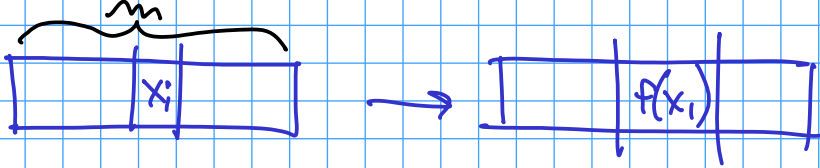
$(2+mw') + (2+mw'') \Rightarrow \# \text{ resources}$



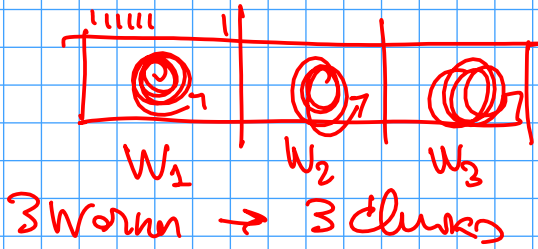
respect the constraints by the perf models of the skeleton/templates



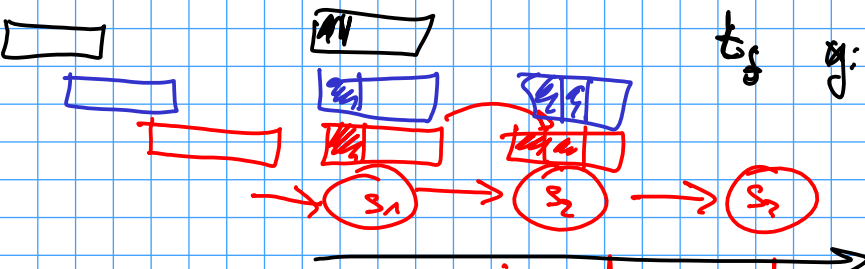
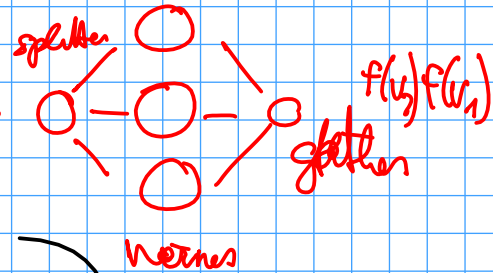
MAP Skeleton or vectors



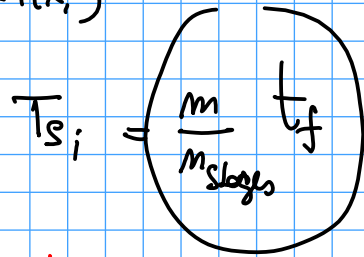
template 1



stream of vectors



$t_f$   $y_i := f(x_i)$



$i$ : gets a vector

visits in its  $i$ th position

$x_i = f(x_i)$

outputs the vector

