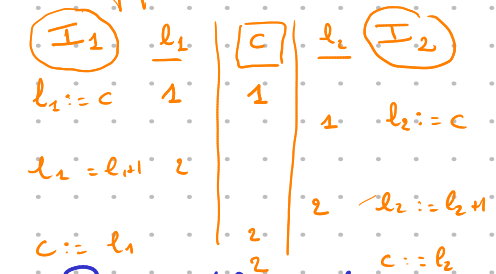


TD5 Modélisation et Vérification

Ex 1: Compteur 10x

Pb: entre laient & on ne protège pas les affectations.



PROBLEME: comment modéliser ce système

- Déclarer le compteur `int c = 0;`
- Décrire la routine des threads INC

```

prototype inc (chan Fin){
    int l, i
    for (i: 1..10){
        l := c
        l := l + 1
        c := l
    }
}
  
```

nombre de message stocké
sur le canal
0 = synchrone

```

• Pmain:
  init {
    chan Fin = [0] of {int}
    for (j: 1..10){
      run inc (Fin)
    }
    for (j: 1..10){
      Fin? m -> printf("Fini")
    }
  }
  
```

c = compteur

Pmain

```

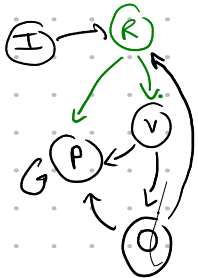
crée canal Fin
vise 10 inc
attend 10 fin Fin? m
vérifie le prog fait ce qu'il doit:
c == 100
  
```

Argument

envoie 0 sur le canal Fin

Vérifie condition
assert (c == 100)

Exo 2 : Feu tricolore



Feu

bool cli
mtype col

initial :

- cli = true
- col = 0

rouge

- goto rouge
- cli = false
- col = R

orange

vert

panme

- cli = true
- col = 0
- goto panme

mtype = {R, V, O, I}

chan obs = [0] of {mtype, bool}

OBS

comportement non déterministe "select"

étiquette permet de créer un état

Syntaxe de promela

select = choix non déterministe

if :: actions

:: actions

fi

condition : "if A then B else C"

A → B : C

étiquettes : "états de l'automate"

label : goto label

atomic = protège des sections de code des autres agents.

atomic { }

boucle : for (i: 1..10) { }

select répétitif

do :: action 1

od :: action

scripture du flot de contrôle

ou une transition dans l'automate

entité isolée minée pour entité alive d'observateur.

Observateur / Vérification

mtype actuel = I

mtype suivant = I

boucle

do :: obs ? m →

actuel = suivant ;

suivant = m

gauche

suivant == V → assert (actuel == R)

si suivant == V alors

assert (actuel == R)

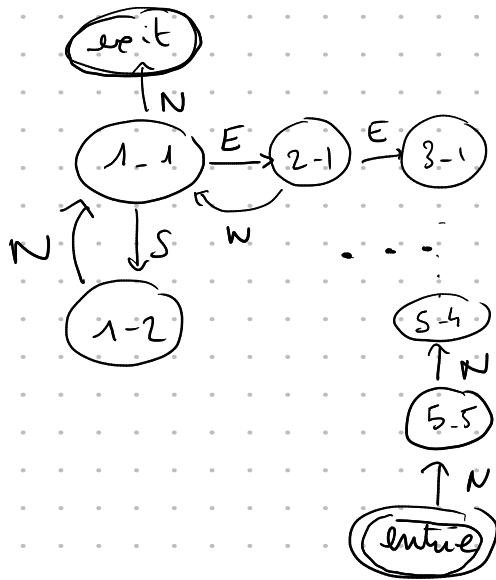
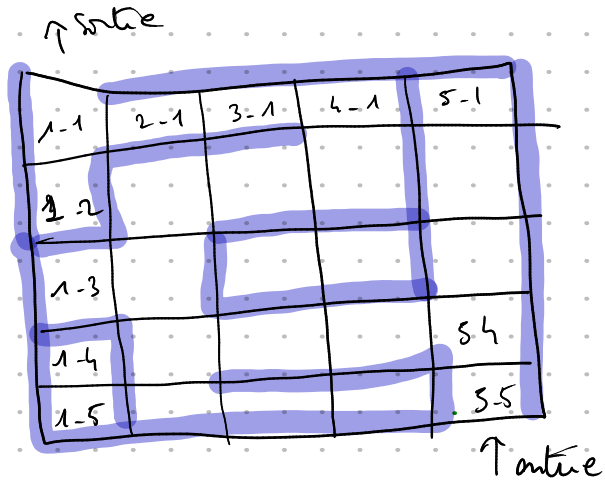
suivant == 0 → assert (actuel == V || cli == true)

si suivant == 0 alors

vérifie que (actuel == V ou cli == true)

od

TRES Wesynthe



But trouver 1 mot reconnu par l'autorité qui va décrire un chemin allant de l'entrée à la sortie

lab4 obs! entire; goto cell 5.5

cell_1-1:

if :: { obs! Exit
goto exit

:: goto cell 1-2

fi :: goto cell 1-1

cell 5-5

the 1st. fun

$$mtype = \{N, S, E, W, \text{true}, \text{Error}\}$$

→ der →

chan OBS = [o] of {ntyrc}

une ét. forte par état

attend de recevoir le
message soit

observation

obs? m

Boole if :: m = Exit
→ goto exit;

$\therefore \text{abe} \rightarrow$

f_e

lip it: