

Compilation sujet 2018 session 2

(Mettez votre nom avant votre proposition comme ça on s'y retrouve, pour les commentaires il y a une fonctionnalité pour)

Exercice 1 :

PP

Proposition Clément:

```
function f( n : integer ) : boolean
if n=0 then
    f:=true ;;on retourne 0
else
    if n=1
        f:=false
    else
        f:= f(n-2)
```

UPP

Proposition Clément:

```
function f( n )
f:=0 ;
if n=0 then
    f:=true ;;on retourne 0
else
    if n=1
        f:=false
    else
        f:= f(n-2)
```

RTL

Proposition Clément:

```
function(%0):%1
var %0, %1, %2, %3, %4
entry %f8
exit f0
```

```

f8 : li %1 0 ->f7
f7 : blez %0 -> f2,f6
f6 : addiu %2 %0 -1
f5 : blez %2 ->f2,f4
f4 : addiu %3 %0 -2 ->f3
f3 : call %1 f(%3) ->f0
f2 : li %1 true -> f0
f1 : li %1 false ->f0

```

Proposition Elisabeth :

```

function(%0):%1
var %0, %1, %2, %3
entry f8
exit f0

```

```

f8 : li %1, 0 -> f7
f7 : blez %0 -> f2,f6
f6 : li %3, 1 -> f5
f5 : beq %0, %3 -> f1, f4
f4 : addiu %2, %0, -2 -> f3
f3 : call %1, f(%2) -> f0
f2 : li %1, true -> fa
f1 : li %1, false -> f0

```

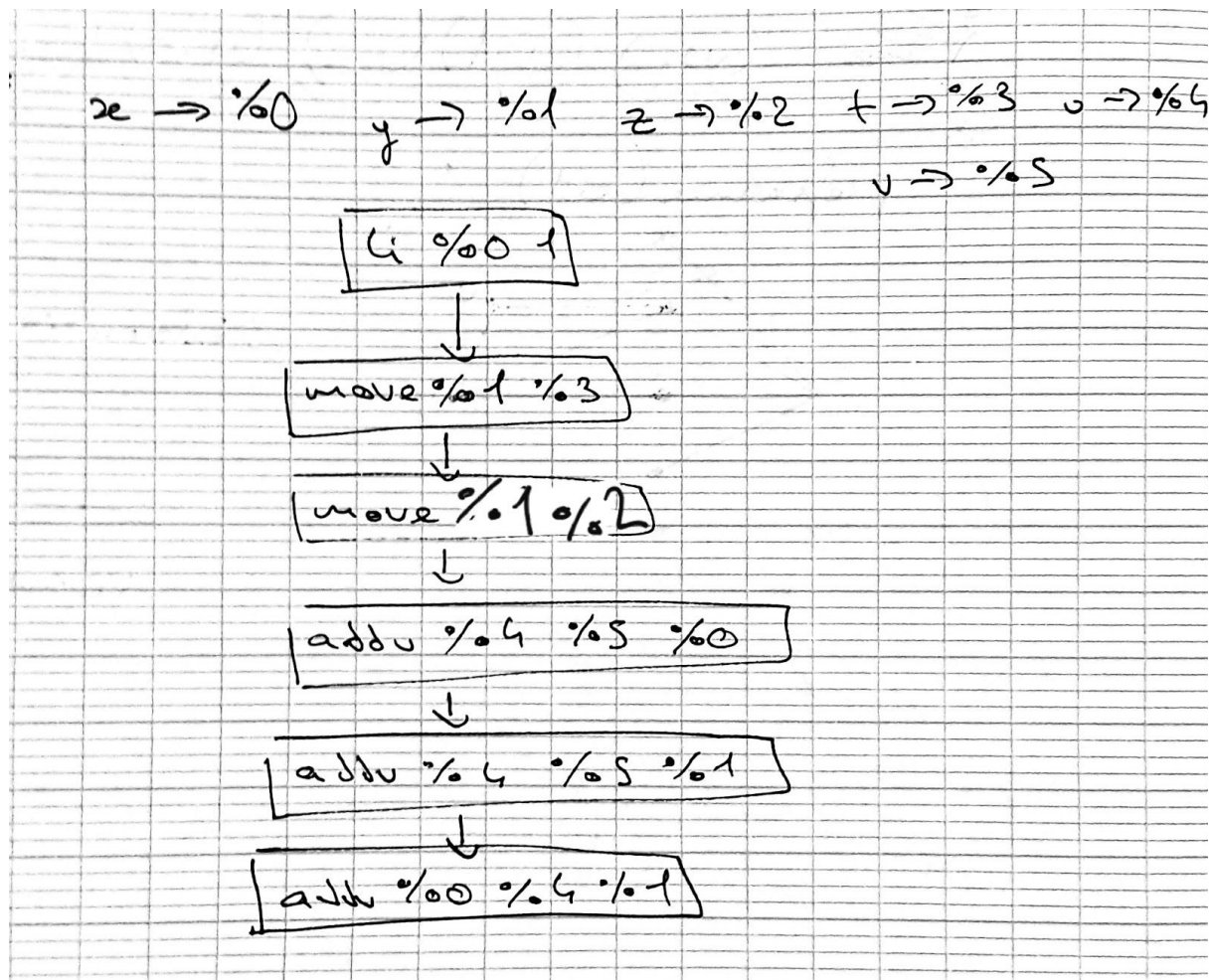
ETRL

Exercice 2

1) Proposition Clément :

(c'est un peu brouillon mais je pense que ça doit être ça, l'intérêt c'est de voir les move à mon avis, vu que là où il y a un move il y aura une arête de préférence

J'ai remplacé les noms des variables par des %, je ne sais pas si c'était nécessaire mais bon ça correspond plus aux graphes de flot de contrôle de RTL)



2) Proposition Elisabeth :

{v,z,t}

x := 1

{v,x,z,t}

y := t

{v,x,z}

y := z

{y,v,x}

u := v + x

{y,v}

u := v + y

{u,y}

$x := u + y$
 $\{x, y\}$

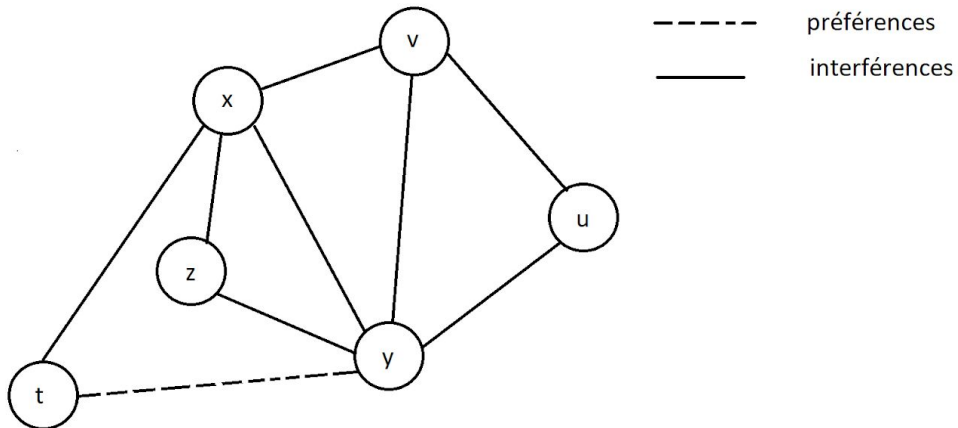
3) Proposition Elisabeth

interférences :

$(x, v), (x, z), (x, t), (y, v), (y, x), (y, z), (u, y), (u, v)$

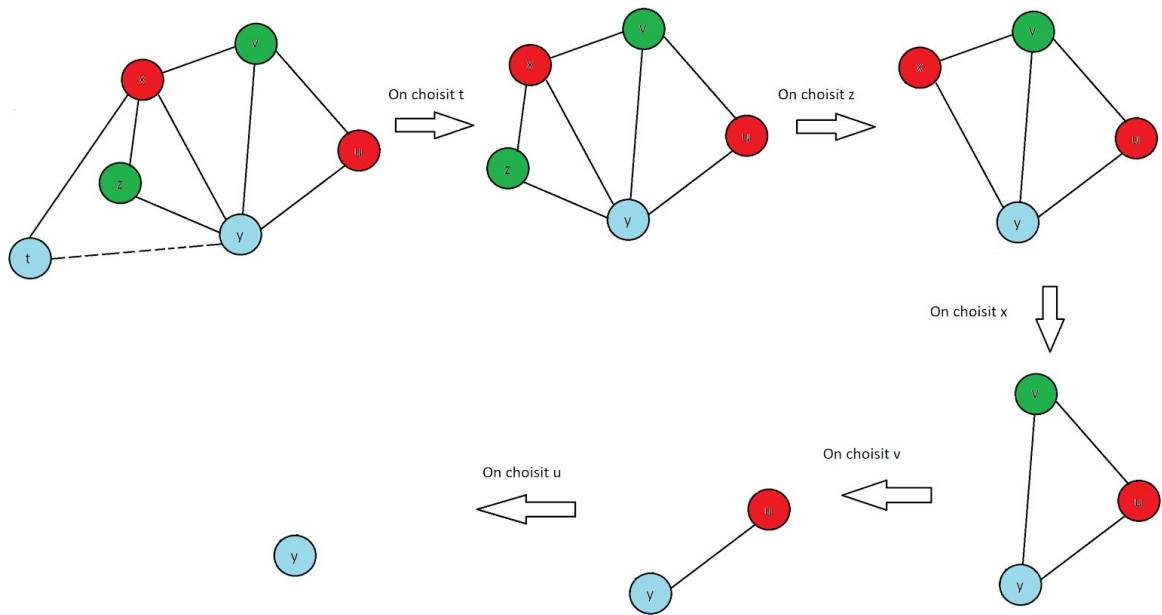
préférences :

(y, t)



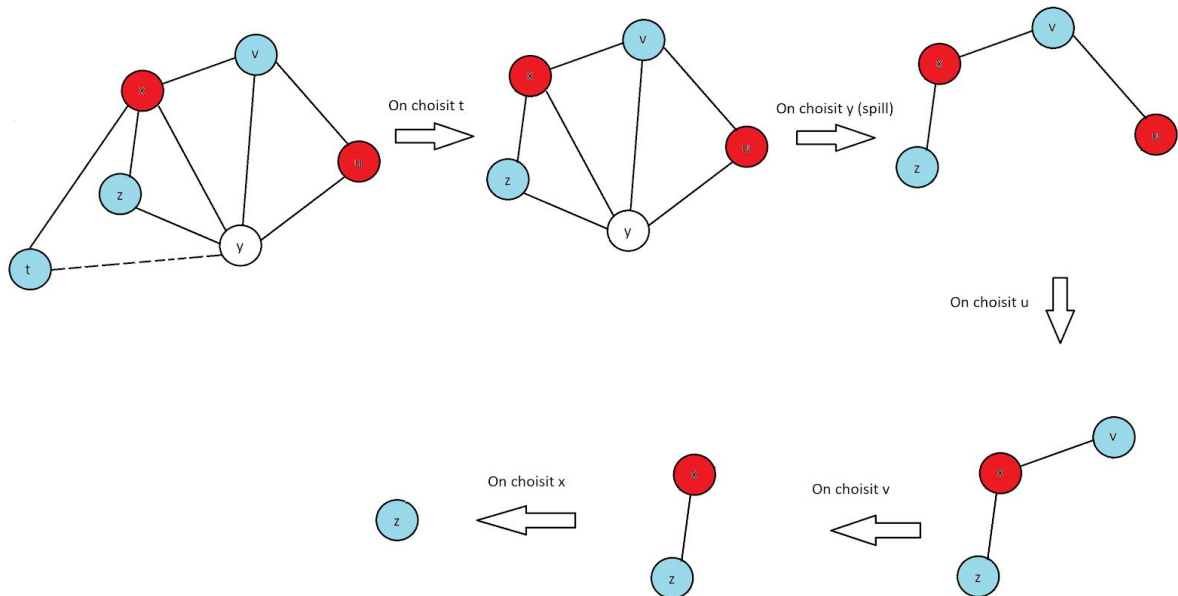
4) Proposition Elisabeth :

Avec 3 couleurs



Proposition Elisabeth :

Avec 2 couleurs : (y n'a pas de couleur)



Exercice 3

QUESTION 1

1)

Proposition Christophe:

Si on associe un label (une étiquette) à chaque état alors ces derniers seront liés à une adresse. Ainsi en utilisant l'instruction JMP on pourrait facilement passer d'un état à un autre. Ainsi une pile n'est pas nécessaire.

2)

Proposition Christophe:

(LABEL E0) ; état 0

(CAR R0 R1) ; on met la première lettre du mot courant dans R1

(CDR R0 R0) ; on retire cette première lettre du reste du mot

(CMP R1 a)

(JEQ E0) ; si la lettre dans R1 est a, on revient au début du code de l'état 0

(CMP R1 b)
(JEQ E1) ; si la lettre est b, on va à l'état 1
(MOVE NIL R0) ; on a pas de correspondance, on met NIL dans R0
(HALT) ; et on arrête l'exécution

(LABEL E1)
(CAR R0 R1)
(CDR R0 R0)
(CMP R1 a)
(JEQ E2)
(CMP R1 b)
(JEQ E0)
(MOVE NIL R0)
(HALT)

(LABEL E2)
(MOVE 'E2 R2)
(cmp R0)
(bnull H) ; si le mot est vide en arrivant ici alors le mot initial est reconnu par l'automate
(CAR R0 R1)
(CDR R0 R0)
(CMP R1 a)
(JEQ E2)
(MOVE NIL R0)
(HALT)

(LABEL H)
(MOVE R2 R0) ; on met l'état final atteint quand le mot a été reconnu dans R0
(HALT)

Proposition Clément :

#init
move nil R2
J #Etat0

#Etat0
CAR R0 R1
CMP R1
BNULL #fin
CDR R0 R0
CMP R1 #a
JEQ #Etat0
CMP R1 #b
JEQ #Etat1

J #Fin

#Etat1

CAR R0 R1

CMP R1

BNULL #fin

CDR R0 R0

CMP R1 #a

JEQ #Etat2

CMP R1 #b

JEQ #Etat0

J #Fin

#Etat2

CAR R0 R1

CMP R1

move #2 R2

BNULL #fin

CDR R0 R0

CMP R1 #a

JEQ #Etat2

move nil R2

J #Fin

#Fin

move R2 R0

Remix (Version de Clément en plus formel) By Béré :

(LABEL init)

(MOVE NIL R2)

(JMP Etat0)

(LABEL Etat0)

(CAR R0 R1)

(CMP R1)

(BNULL fin)

(CDR R0 R0)

(CMP R1 'a')

(JEQ Etat0)

(CMP R1 'b')

(JEQ Etat1)

(JMP Fin)

(LABEL Etat1)

(CAR R0 R1)

(CMP R1)

(BNULL fin)
(CDR R0 R0)
(CMP R1 'a')
(JEQ Etat2)
(CMP R1 'b')
(JEQ Etat0)
(JMP Fin)

(LABEL Etat2)
(CAR R0 R1)
(CMP R1)
(move 2 R2)
(BNULL fin)
(CDR R0 R0)
(CMP R1 'a')
(JEQ Etat2)
(MOVE NIL R2)
(JMP Fin)

(LABEL Fin)
(MOVE R2 R0)