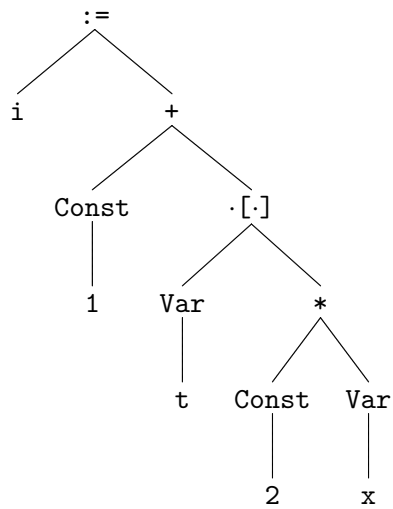


Corrigé de l'examen final de compilation

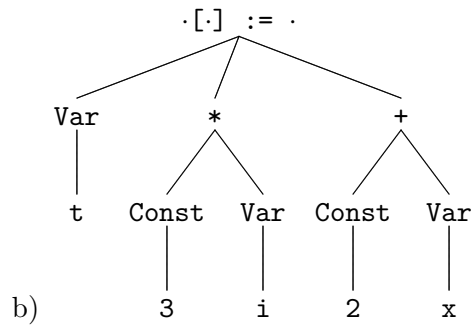
Énsiie, semestre 3

10 janvier 2011

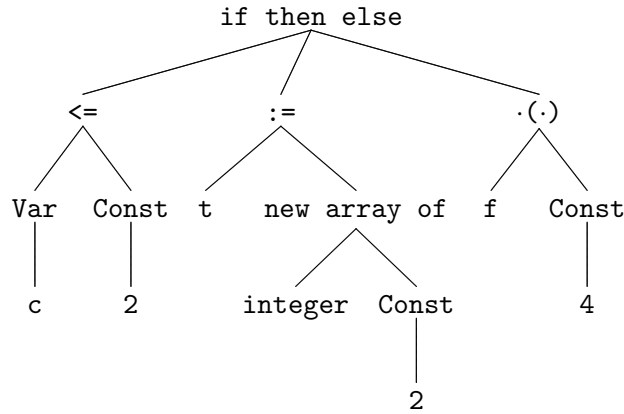
Exercice 1 : Syntaxe et sémantique (5 points)



1. a)



b)



- c)
2. On part de l'état $G = \{x \mapsto 4; t \mapsto \ell\}$, $H = \{\ell \mapsto \langle 3, 3 \rangle\}$ et $E = \{i \mapsto 0\}$. On a l'arbre de dérivation suivant :

$G, H, E/t[3*i] := 2 + x \rightarrow G, \{\ell \mapsto \langle 6, 3 \rangle\}, E$

Écriture dans un tableau

$G, H, E/t \rightarrow G, H, E/\ell$

Variable globale

$G, H, E/3 * i \rightarrow G, H, E/0$

Opérateur binaire

$G, H, E/3 \rightarrow G, H, E/3$

Constante

$G, H, E/i \rightarrow G, H, E/0$

Variable locale

$G, H, E/2 * x \rightarrow G, H, E/6$

Opérateur binaire

$G, H, E/2 \rightarrow G, H, E/2$

Constante

$G, H, E/x \rightarrow G, H, E/4$

Variable globale

$H(\ell) = \langle 3, 3 \rangle$

$0 \leq 0 < 2$

3. On évalue la sémantique du programme, il faut donc évaluer le corps principal dans l'état $G = \{x \mapsto 0\}$, $H = \emptyset$, $E = \emptyset$. Pour cela on applique la règle séquence. On évalue $x := 1$ ce qui nous mène à l'état $G' = \{x \mapsto 1\}$, H , E .

On évalue $x := f(x)$ dans ce nouvel état, ce qui nous amène à évaluer $f(x)$. Il faut donc évaluer le corps de f dans l'état G' , H , $E_f^1 = \{y \mapsto 1; z \mapsto 0; f \mapsto 0\}$.

On évalue le if (Conditionnelle (sinon)), on doit évaluer $f(y - 1)$. On doit donc évaluer le corps de f dans le contexte G' , H , $E_f^2 = \{y \mapsto 0; z \mapsto 0; f \mapsto 0\}$, ce

qui mène à l'état G' , H , $\{y \mapsto 0; z \mapsto 0; f \mapsto 2\}$. Par conséquent, l'évaluation de $f(y - 1)$ ne change pas l'état G' , H , E_f^1 et produit la valeur 2. $x := f(y - 1)$

mène donc à l'état $G'' = \{x \mapsto 2\}$, H , E_f^1 . Puis, $z := x * y$ mène à G'' , H , $\{y \mapsto 1; z \mapsto 2; f \mapsto 0\}$. Enfin, $f := z - 3$ mène à G'' , H , $\{y \mapsto 1; z \mapsto 2; f \mapsto -1\}$.

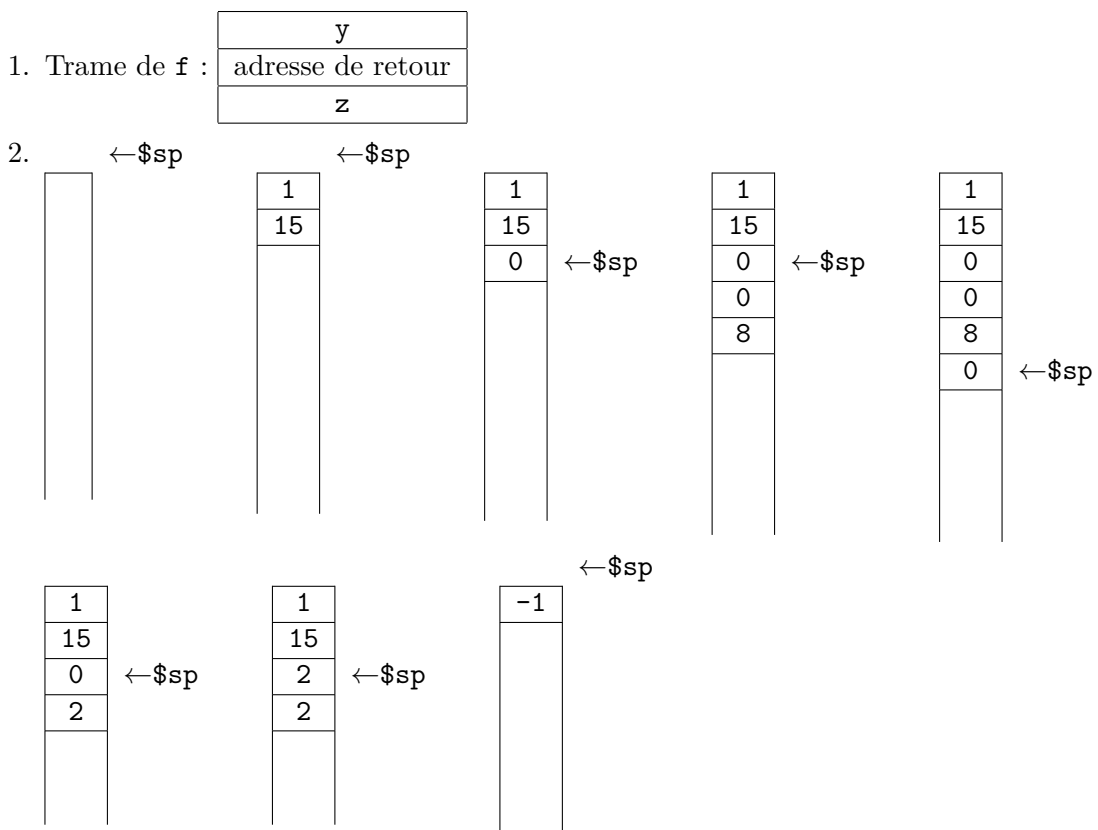
Par conséquent, $x := f(x)$ mène à l'état final $G''' = \{x \mapsto -1\}$, H , E .

Exercice 2 : Réécriture (3 points)

1. Pas confluente ($b \leftarrow a \rightarrow c$), terminant (pas de réécriture possible après b et c)
2. Confluente : seul a et b peuvent se réécrire, et l'un se réécrit en l'autre et vice versa.
Pas terminant : suite infinie de $a b$.
3. Pas confluente ($c \leftarrow a \rightarrow *d$), pas terminant : suite infinie de $a b a b a \dots$
4. Terminant : (deux fois le nombre de g) plus le nombre de d diminue strictement.
Confluente : seule paire critique $d, f(e)$ joignable.

Exercice 3 : Convention d'appel (6 points)

Convention par pile

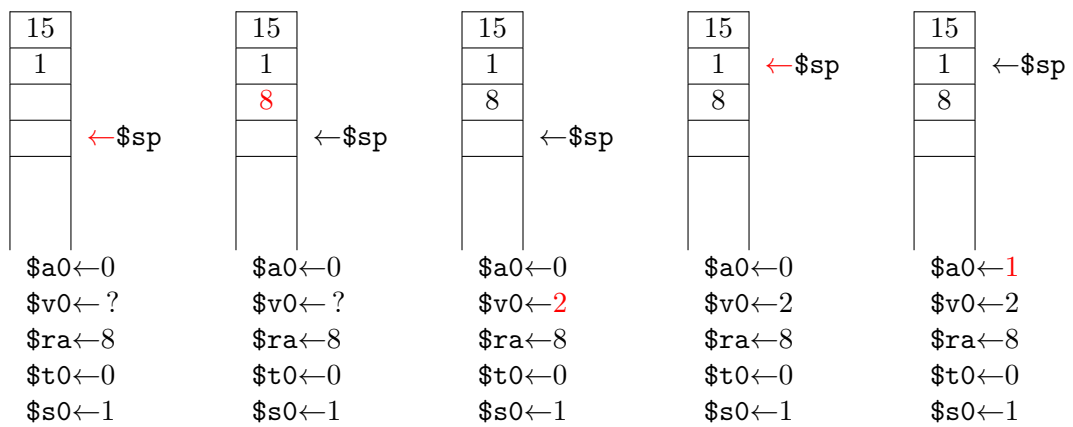
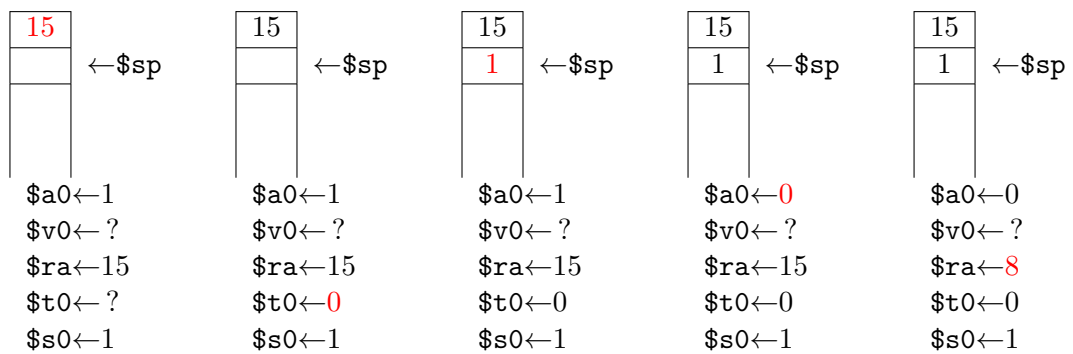
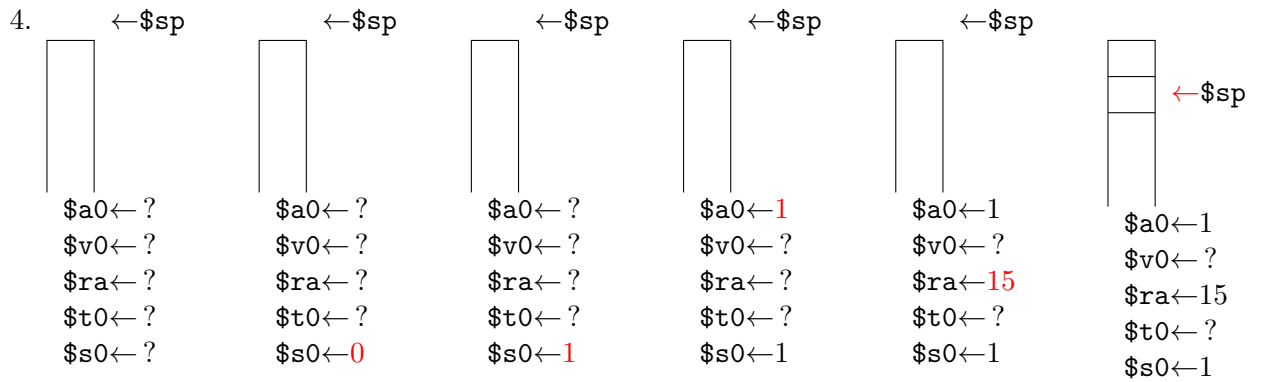


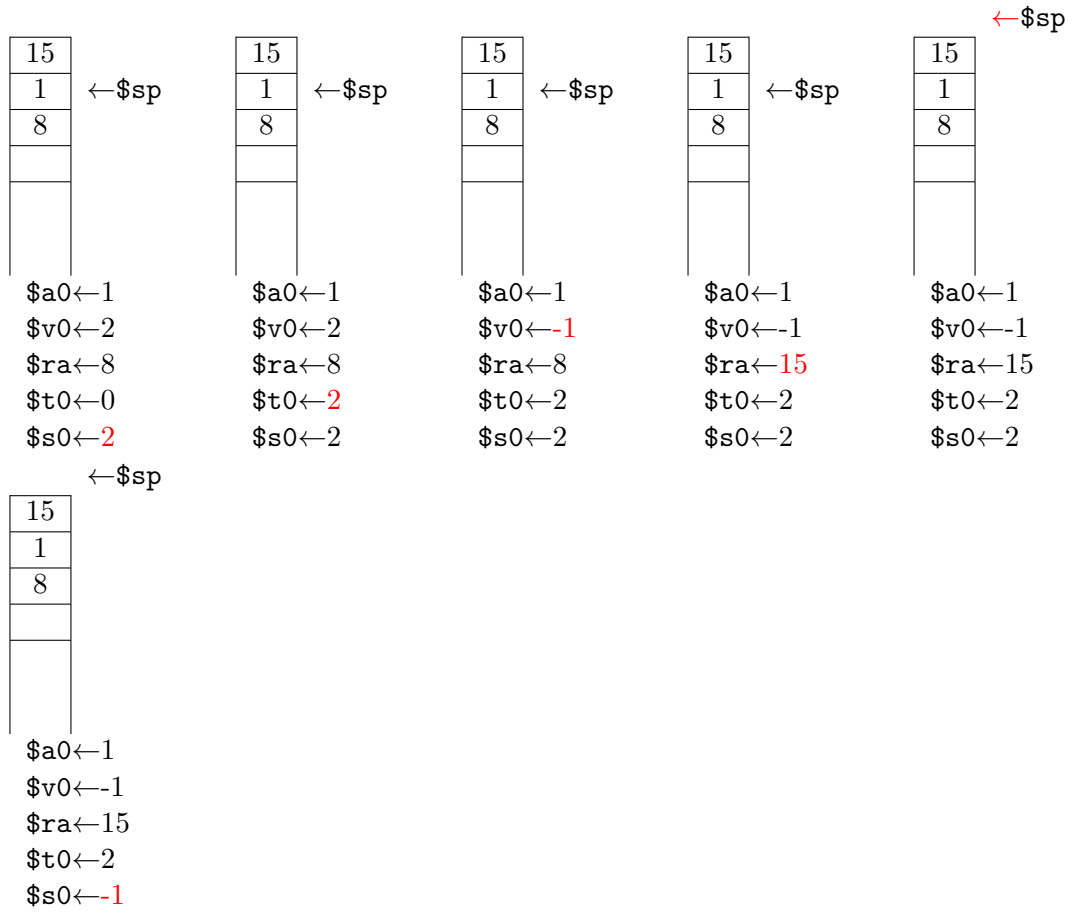
Convention d'appel MIPS

3. f modifie les registres *callee-saved* $\$ra$ (appel de fonction dans f) et $\$s0$. Comme indiqué dans l'énoncé, seul $\$ra$ est à sauvegarder.
Pour les registres *caller-saved*, seul $\$a0$, qui contient la valeur de y réutilisée ensuite, a besoin d'être sauvegardé. $\$t0$ n'en a pas besoin, car z est redéfini avant d'être utilisé, sauvegarder sa valeur est donc inutile.

Trame de f :

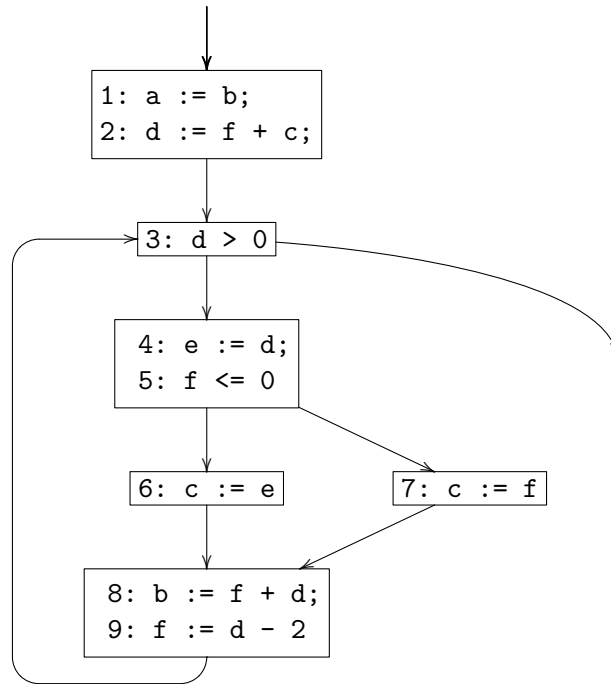
sauvegarde de \$ra
sauvegarde de \$a0





Exercice 4 : Allocation de registres (6 points)

1.

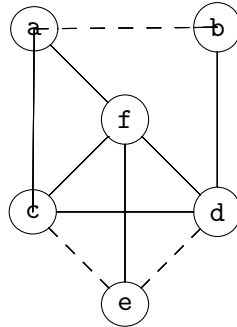


2. Numéro des instructions sur le graphe de flot.

point	vivantes après	vivantes avant
1	c f	b c f
2	d f	c f
3	d f	d f
4	d e f	d f
5	d e f	d e f
6	d f	d e f
7	d f	d f
8	d	d f
9	d f	d

3. Comme indiqué dans le cours, on peut supprimer toutes les instructions dont la variable affecté n'est pas vivante après. On peut donc dans un premier temps supprimer 1 6 7 8. Si on recalcule la durée de vie, on peut alors aussi supprimer 4 avec le même argument.

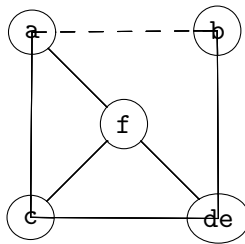
4.



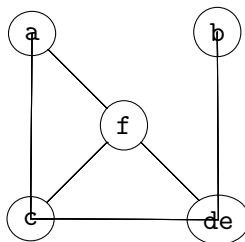
Pas d'arête de préférence entre f et c car l'instruction 6 crée une interférence entre ces deux variables.

5. On ne peut pas simplifier : les noeuds trivialement colorables b et e ont des arêtes de préférence.

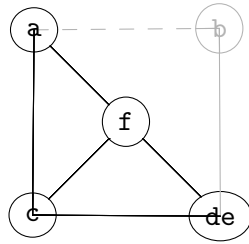
On peut fusionner d et e car le critère de George s'applique : les voisins de e , i.e. f , sont inclus dans ceux de d .



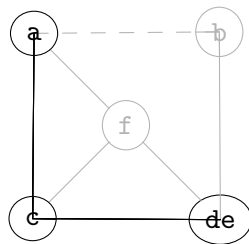
On ne peut toujours pas simplifier, on ne peut pas fusionner b et d car aucun critère ne s'applique. On gèle b .



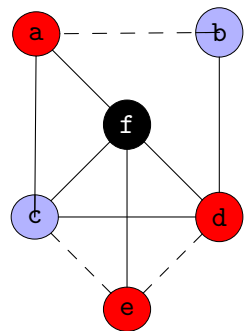
On simplifie b .



On spille par exemple f.



On simplifie a c de. Quand on dépile, on voit que f doit effectivement être spillée.
On obtient :



Remarque : il était possible de choisir de fusionner c et e au lieu de d et e, et de spiller un autre registre que f. Ce sont les seuls endroits où l'algorithme n'est pas déterministe ici.

```

6. entry 11
   exit fin
11: move $s1, $s0 -> 12
12: lw $t0, 0($sp) -> 12'
12': add $s0, $t0, $s1 -> 13
13: bgtz $s0 -> fin, 15
15: lw $t0, 0($sp) -> 15'

```



```
15': bgtz $t0 -> 16, 17
16:  move $s1, $s0 -> 18
17:  lw $t0, 0($sp) -> 17'
17': move $s1, $t0 -> 18
18:  lw $t0, 0($sp) -> 18'
18': add $s1, $t0, $s0 -> 19
19:  addi $t0, $s0, -2 -> 19'
19': sw $t0, 0($sp) -> 13
```

On peut supprimer l'instruction correspondant à 4 car l'arête de préférence est respectée. On pourrait réduire le nombre de chargements de \$t0.