



# **LOG3210: Éléments de langages et compilateurs**

## **Laboratoire 5**

**Présenté à : Alexandre Pereira**

Soumis par

Raphael Christian-Roy(1743344)

Louis-Charles Hamelin(1742949)

Section 01 (B1)

Dimanche 2 avril 2017

1. Donnez le résultat du script sur exemple3.ci avec la même variable vive de fin.  
Présentez les données sous forme de tableau où une colonne indique l'instruction et une autre indique le résultat du calcul « OUT » de cette instruction.  
(1 point)

```
LIVE --->
NODE 1 IN:
    b
    c
    d
NODE 1 OUT:
    b
    c
    d
NODE 2 IN:
    b
    c
    d
NODE 2 OUT:
    b
    c
    d
NODE 3 IN:
    b
    c
    d
NODE 3 OUT:
    b
    c
NODE 4 IN:
    b
    c
NODE 4 OUT:
    e
<--- LIVE
```

Instruction	Résultat du calcul "OUT"
$a = b + c$	NODE 1 OUT: b c d
$b = b - d$	NODE 2 OUT: b c d
$c = c + d$	NODE 3 OUT: b c
$e = b + c$	NODE 4 OUT: e

2. Que signifie la présence ou l'absence de la variable « a » dans les résultats de l'algorithme?

La variable a est absente dans les résultats de l'algorithme, car elle est définie au départ, mais n'est jamais utilisée par la suite par les autres instructions.

3. Utilisez les résultats calculés en 1, afin d'identifier la première fois que tous les registres sont pleins et qu'un registre doit être écrasé. Quel registre choisir? Décrivez (2 points).

Le registre à choisir pour l'écrasement est celui qui a été rempli il y a le plus longtemps ou celui qui a été le moins utilisé lors de l'exécution du programme. Considérant qu'il n'y a pas eu d'optimisation, le a est celui qui a été remplacé par le registre dans ce cas.

4. Si on enlève la ligne `init_stop_live`, quelle instruction arbitraire ajouteriez-vous pour avoir le même résultat de variables vives à l'instruction 4? (1 point)

```
cfg_attr stop_node 5
cfg_succ 4 5
ref 5 e
```

Si l'on veut que ce soit en encore une variable vive, il faut qu'elle soit utilisée à nouveau dans une autre instruction.