

TD09 - GRAPHE D'ANALYSE DE CODE PRÉ-ASSEMBLEUR

Exercice 1. Résolution des équations de vie des temporaires

Soit le programme suivant en pré-assembleur :

```

1 main : push ebp                ;sauvegarde la valeur de ebp
2       mov ebp,                esp    ;nouvelle valeur de ebp
3       sub esp,                4      ;allocation des variables locales
4   10 : mov r0,                1      ;Affect
5       cmp dword [ebp-4*1], 10      ;JumpIfLess 1
6       jl 12                   ;JumpIfLess 2
7       mov r0,                0      ;Affect
8   12 : cmp r0,                0      ;JumpIfEqual 1
9       je 11                   ;JumpIfEqual 2
10      mov r1,                dword [ebp-4*1] ;
11      add r1,                1      ;
12      mov dword [ebp-4*1], r1    ;Affect
13      jmp 10                   ;Jump
14   11 : add esp,                4      ;désallocation des variables locales
15      pop ebp                 ;restaure la valeur de ebp
16      ret                     ;

```

- (1) Dessiner le graphe d'analyse de P .
- (2) Calculer les ensembles **use** et **def** de chaque instruction de P .
- (3) Calculer les ensembles **in** et **out** de chaque instruction de P à l'aide de l'algorithme itératif :

Algorithm 1 Calcul itératif de $in(s)$ et $out(s)$

```

1: for all  $s$  do
2:    $in(s) = \{\}$ 
3:    $out(s) = \{\}$ 
4: end for
5: repeat
6:   for all  $s$  do
7:      $in'(s) = in(s)$ 
8:      $out'(s) = out(s)$ 
9:      $in(s) = use(s) \cup (out(s) - def(s))$ 
10:     $out(n) = \cup_{s \in succ(s)} in(s)$ 
11:   end for
12: until  $in'(s) = in(s)$  et  $out'(s) = out(s), \forall s$ 

```

- (4) Calculer les ensembles **in** et **out** de chaque instruction de P en parcourant les instructions dans l'ordre inverse et en permutant les lignes 9 et 10 de l'algorithme. En combien d'itérations l'algorithme trouve-t-il la solution ? pourquoi ?