

# Analyse de vie des temporaires

Alexis Nasr  
Carlos Ramisch  
Manon Scholivet  
Franck Dary

Compilation – L3 Informatique  
Département Informatique et Interactions  
Aix Marseille Université

# Vie des temporaires

- Le nombre de temporaires d'un programme est illimité.
- Le nombre de registres d'un processeur est limité.
- Deux temporaires  $t_1$  et  $t_2$  peuvent être mis dans le même registre s'ils ne sont jamais utilisés en même temps.
- On analyse le pré-assembleur pour savoir si deux temporaires sont utilisés au même moment.

# Vie des temporaires

- Un temporaire est dit **vivant** s'il contient une valeur qui sera utilisée dans le futur.
- L'**analyse de vie** des temporaires consiste à identifier, pour chaque instruction  $i$  du pré-assembleur, quelles temporaires sont en vie au moment de l'exécution de  $i$ .
- Cette analyse est illustrée ici sur du code trois adresse, mais elle s'applique aussi au pré-assembleur.

# Graphe d'analyse

- On représente le programme sous la forme d'un graphe orienté.
- Chaque instruction du programme correspond à un sommet.
- Si l'instruction  $I_0$  peut être suivie par l'instruction  $I_1$ , on crée un arc de  $I_0$  vers  $I_1$ .

# Exemple

## Programme en L

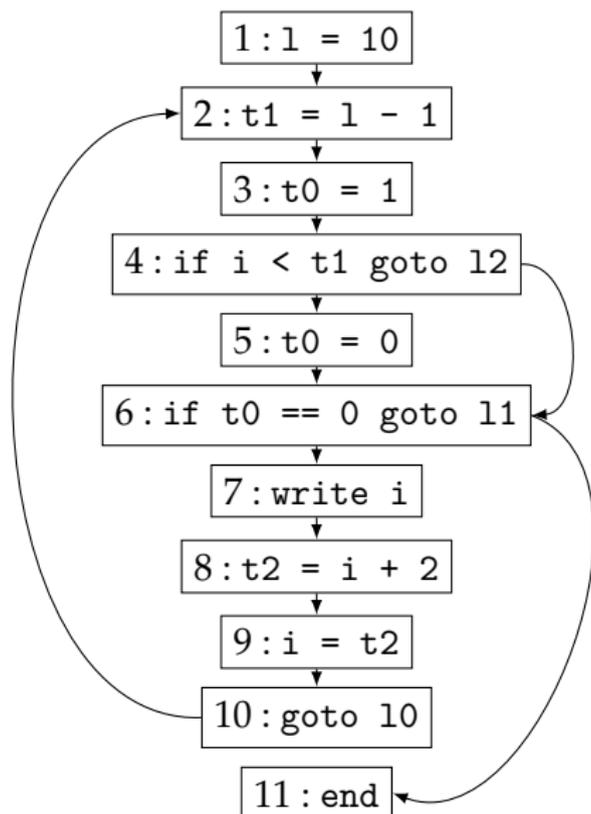
```
entier l, entier i;
main()
{
  l = 10;
  tantque i < (l - 1) faire
  {
    ecrire(i);
    i = i + 1;
  }
}
```

## Programme en C3a

```
main fbegin
  l = 10
10  t1 = l - 1
    t0 = 1
    if i < t1 goto 12
    t0 = 0
12  if t0 = 0 goto 11
    write i
    t2 = i + 1
    i = t2
    goto 10
11  fend
```

# Graphe d'analyse

```
main fbegin
  l = 10
10  t1 = l - 1
    t0 = 1
    if i < t1 goto l2
    t0 = 0
12  if t0 = 0 goto l1
    write i
    t2 = i + 1
    i = t2
    goto l0
11  fend
```

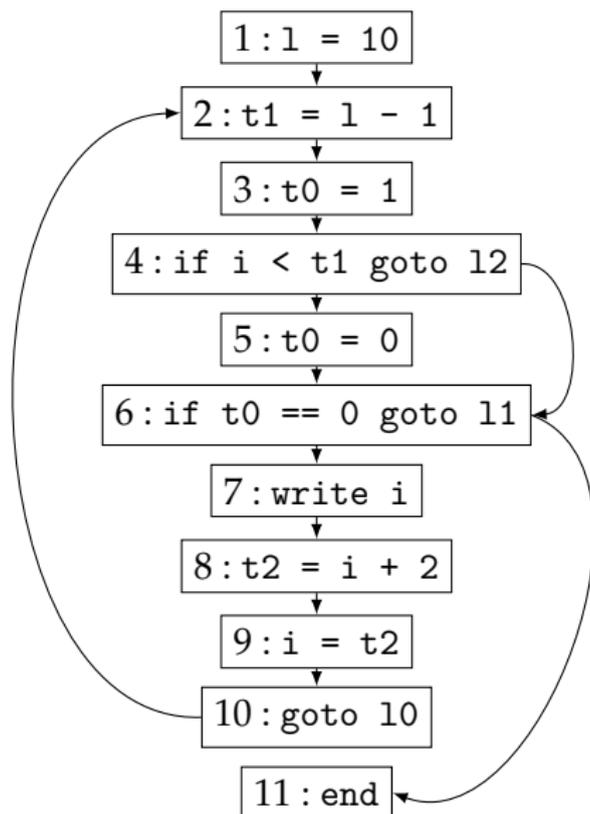


# Parcours du graphe d'analyse

- On part de l'utilisation d'une variable  $t$ .
- On "remonte" à la recherche de l'instruction qui spécifie la valeur de  $t$  (en général une affectation).
- $t$  est vivante sur tout le chemin menant de sa spécification à son utilisation.

## Exemple

- $t_2$  est utilisée en 9 et elle est définie en 8, elle est donc vivante sur l'arc  $8 \rightarrow 9$
- $t_0$  est utilisée en 6, et elle est définie en 5, elle est donc vivante sur l'arc  $5 \rightarrow 6$
- mais  $t_0$  est aussi définie en 3 et il existe un arc entre 3 et 4 puis entre 4 et 6, elle est donc vivante sur les arcs  $3 \rightarrow 4$  et  $4 \rightarrow 6$ .
- $t_1$  est utilisée en 4 et définie en 2. Elle est donc vivante sur les arcs  $2 \rightarrow 3$  et  $3 \rightarrow 4$ .



# Résultat de l'analyse

arc	t0	t1	t2
1 → 2			
2 → 3		✓	
3 → 4	✓	✓	
4 → 5			
4 → 6	✓		
5 → 6	✓		
6 → 7			
6 → 11			
7 → 8			
8 → 9			
9 → 10			✓
10 → 2			

## Conclusions

- t0 et t2 ne sont jamais en vie sur un même arc.
- t1 et t2 ne sont jamais en vie sur un même arc.
- t0 et t1 sont en vie sur l'arc 3 → 4, ils ne peuvent être mis dans un même registre.
- Deux registres sont suffisants pour cet exemple.

# Définition et utilisation des variables

- Une affectation à une variable  $x$ , **définit** cette variable.
- $def(s)$  est l'ensemble des variables que le sommet  $s$  définit.
- Une variable dans une partie droite d'une affectation (ou dans une autre instruction) **utilise** cette variable.
- $use(s)$  est l'ensemble des variables que le sommet  $s$  utilise.

		def	use
1			
2	10	t1	
3		t0	
4			t1
5		t0	
6	12		t0
7			
8		t2	
9			t2
10			
11	11		

# Calcul de la vie des variables

- Une variable est **vivante** sur un arc s'il existe un chemin depuis cet arc vers une utilisation de cette variable qui ne passe pas par une définition.
- Une variable est vivante en entrée d'un sommet  $s$  si elle est vivante sur un arc entrant de  $s$ .
- Elle est vivante en sortie d'un sommet  $s$  si elle est vivante sur un arc sortant de  $s$ .
- On associe à tout sommet  $s$  les ensembles  $in(s)$  et  $out(s)$  des variables vivantes en entrée et en sortie de  $s$ .
- Objectif : calculer  $in(s)$  et  $out(s)$  pour tous les sommets  $s$  du graphe.

# Equations pour le calcul de la vie des variables

- 1 Si une variable est dans  $use(s)$  alors elle est dans  $in(s)$ . En d'autres termes, si une instruction utilise une variable, alors elle est vivante sur les arcs entrants.
- 2 Si une variable est dans  $in(s)$  alors elle est dans  $out(s')$  pour tous les sommets  $s' \in pred(s)$
- 3 Si une variable est dans  $out(s)$  et n'est pas dans  $def(s)$  alors elle appartient aussi à  $in(s)$ . En d'autres termes, si une variable  $v$  est vivante en sortie de l'instruction  $s$  et que  $s$  ne définit pas  $v$ , alors  $v$  doit être vivante en entrée de  $s$ .

Ces trois propositions peuvent s'écrire sous la forme des équations suivantes :

$$in(s) = use(s) \cup (out(s) - def(s)) \quad (1)$$

$$out(s) = \cup_{s' \in succ(n)} in(s') \quad (2)$$

Objectif : déterminer les ensembles  $in$  et  $out$  qui vérifient ces équations.

# Algorithme de calcul de *in* et *out*

---

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

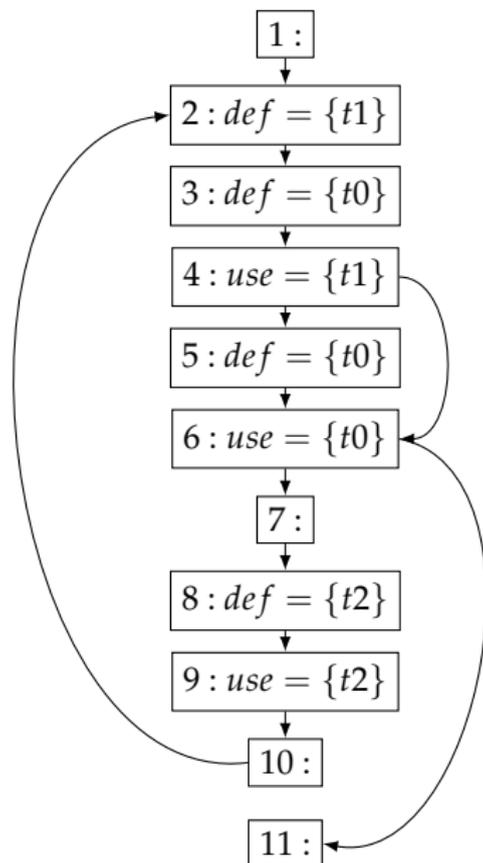
---

```
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  $out'(s) = out(s), \forall s$ 
```

---

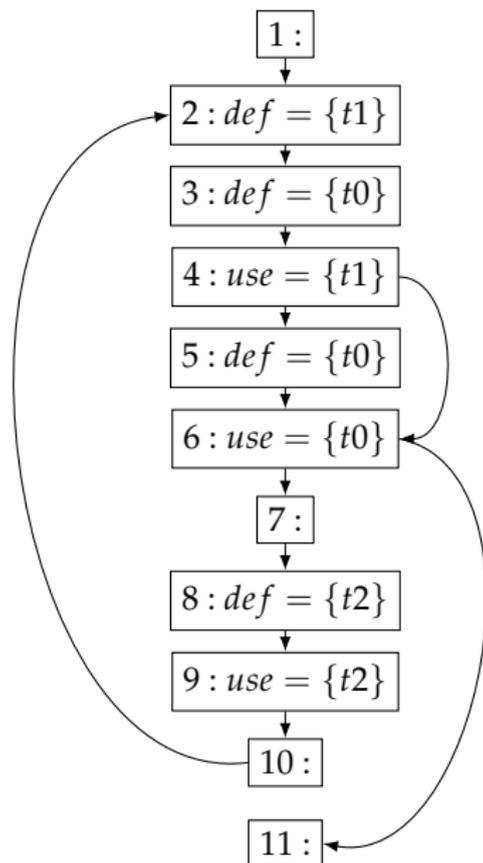
# Itération 1

	0		1	
	in	out	in	out
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				



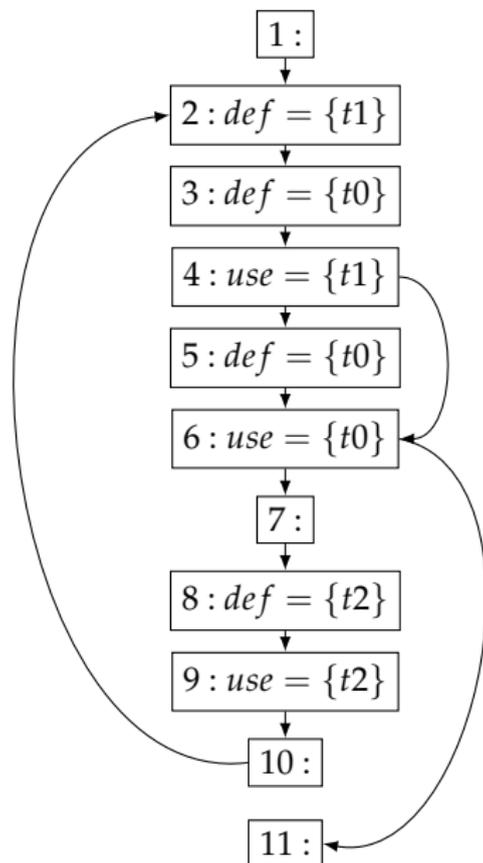
# Itération 1

	0		1	
	in	out	in	out
1				
2				
3				
4			t1	
5				
6				
7				
8				
9				
10				
11				



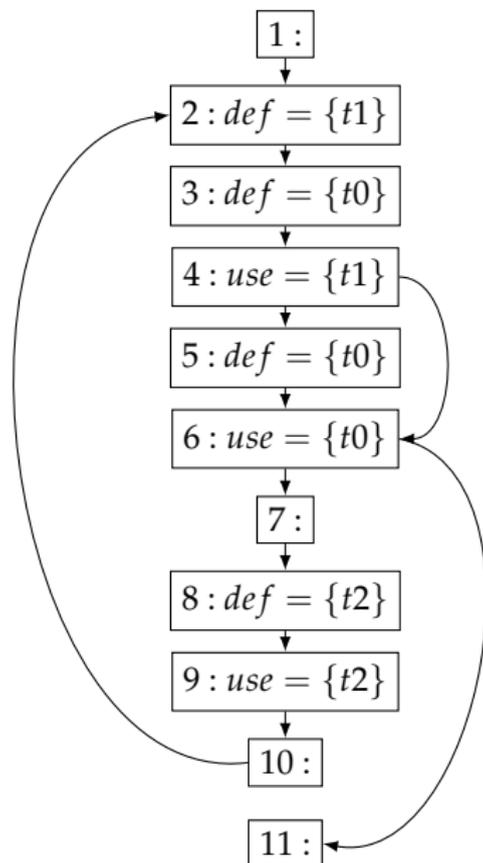
# Itération 1

	0		1	
	in	out	in	out
1				
2				
3				
4			t1	
5				
6			t0	
7				
8				
9				
10				
11				



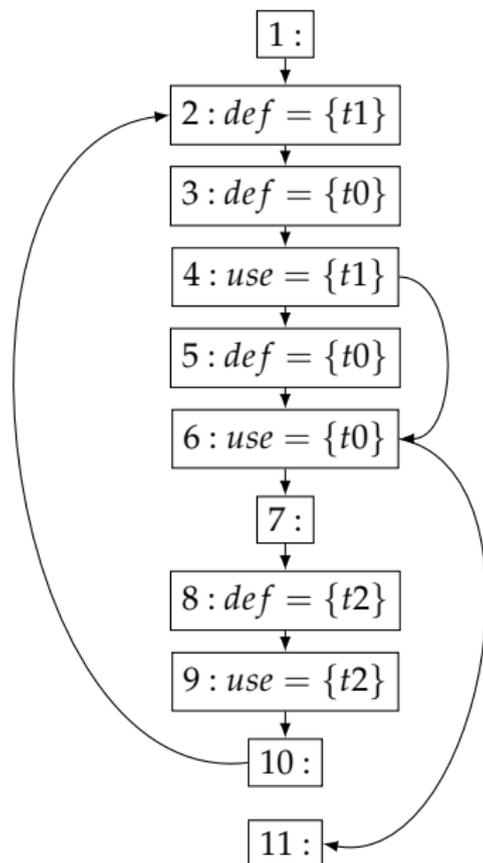
# Itération 1

	0		1	
	in	out	in	out
1				
2				
3				
4			t1	
5				
6			t0	
7				
8				
9			t2	
10				
11				



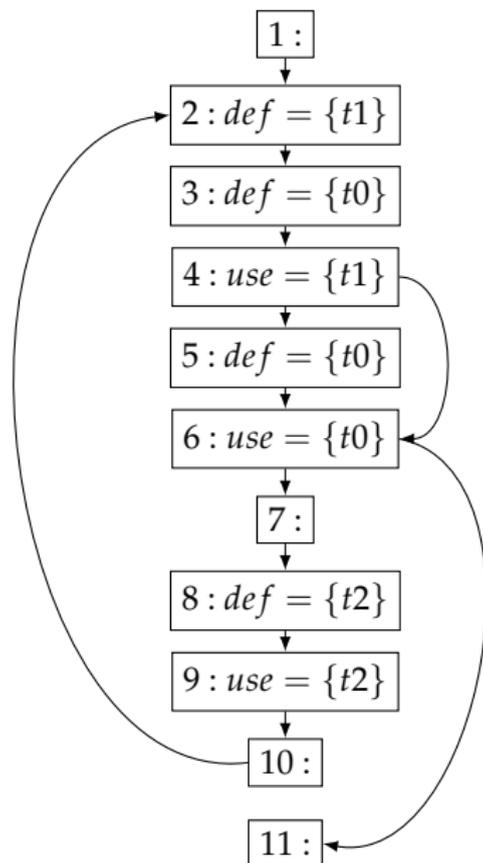
## Itération 2

	1		2	
	in	out	in	out
1				
2				
3				
4	t1			
5				
6	t0			
7				
8				
9	t2			
10				
11				



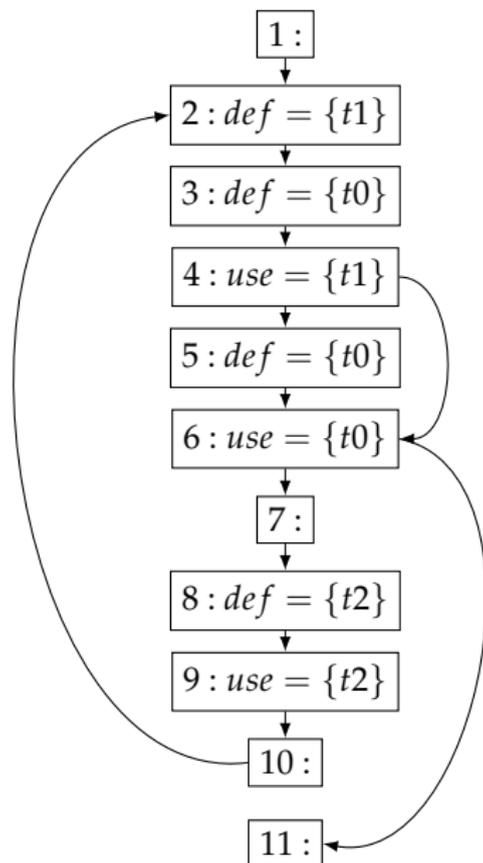
## Itération 2

	1		2	
	in	out	in	out
1				
2				
3				t1
4	t1			
5				
6	t0			
7				
8				
9	t2			
10				
11				



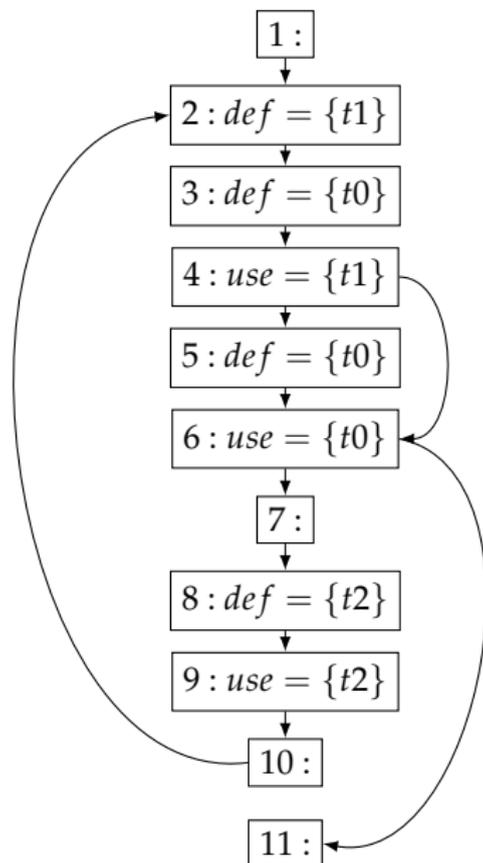
## Itération 2

	1		2	
	in	out	in	out
1				
2				
3				
4	t1		t1	
5				t1
6	t0			
7				
8				
9	t2			
10				
11				



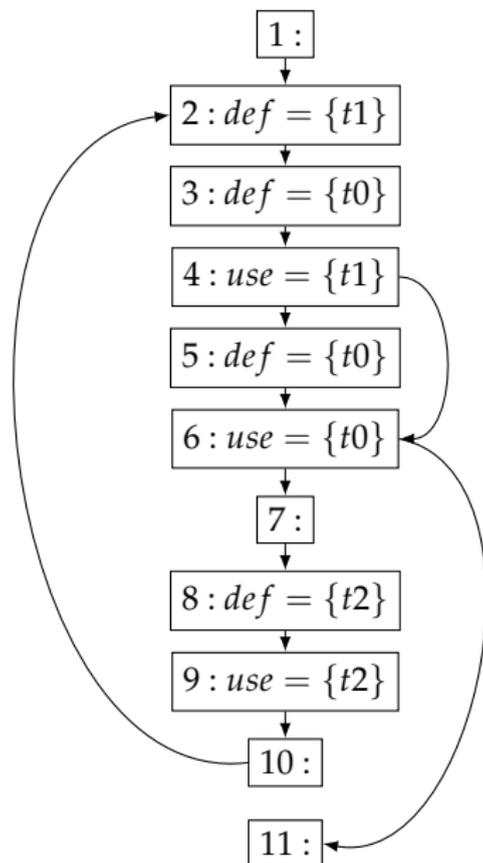
## Itération 2

	1		2	
	in	out	in	out
1				
2				
3				t1
4	t1		t1	t0
5				
6	t0			
7				
8				
9	t2			
10				
11				



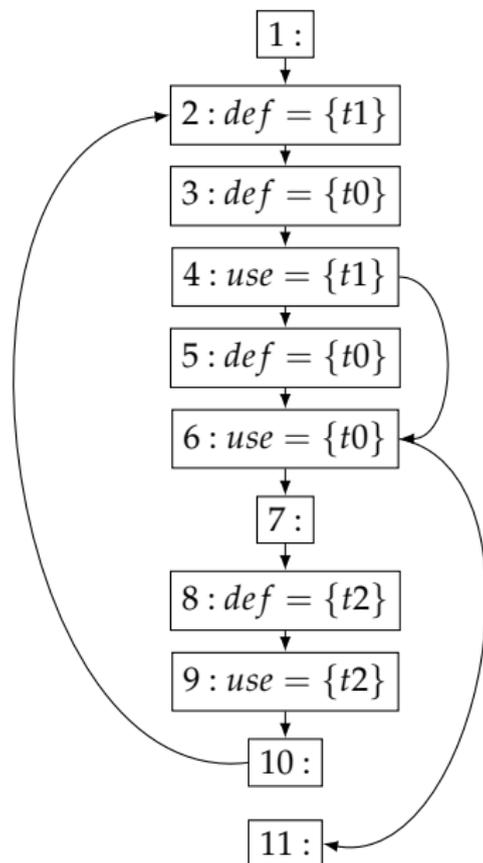
## Itération 2

	1		2	
	in	out	in	out
1				
2				
3				t1
4	t1		t1	t0
5				t0
6	t0			
7				
8				
9	t2			
10				
11				



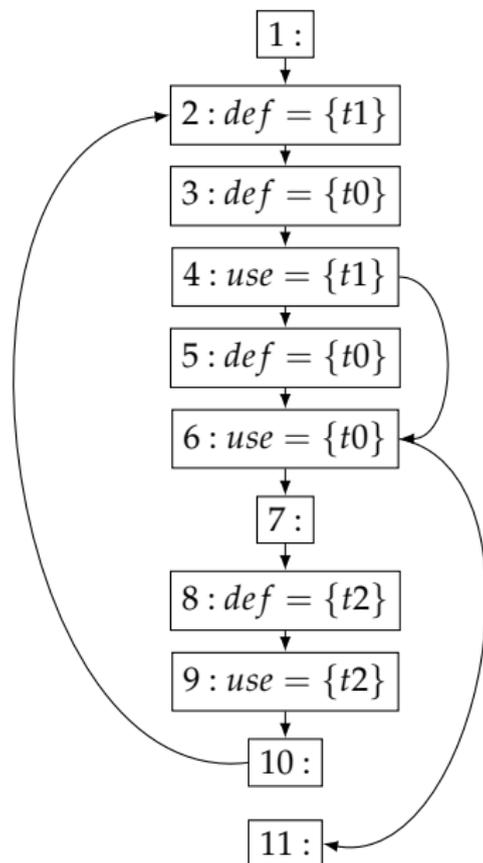
## Itération 2

	1		2	
	in	out	in	out
1				
2				
3				t1
4	t1		t1	t0
5				t0
6	t0		t0	
7				
8				
9	t2			
10				
11				



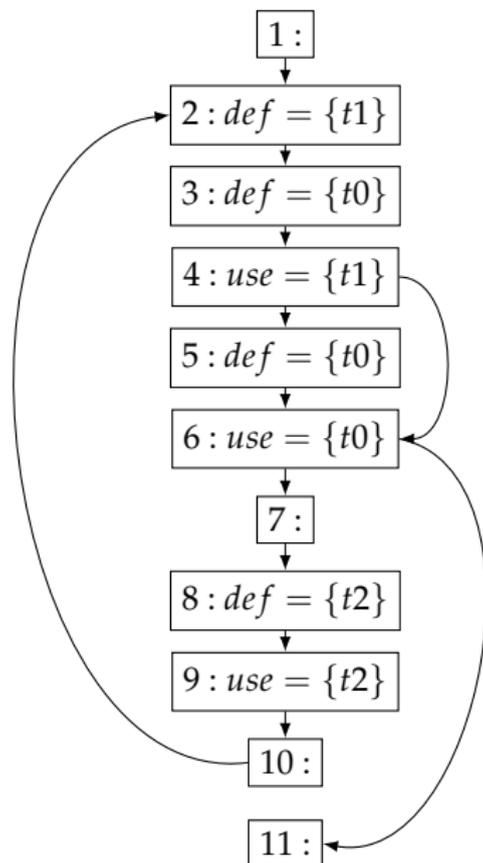
## Itération 2

	1		2	
	in	out	in	out
1				
2				
3				t1
4	t1		t1	t0
5				t0
6	t0		t0	
7				
8				t2
9	t2			
10				
11				



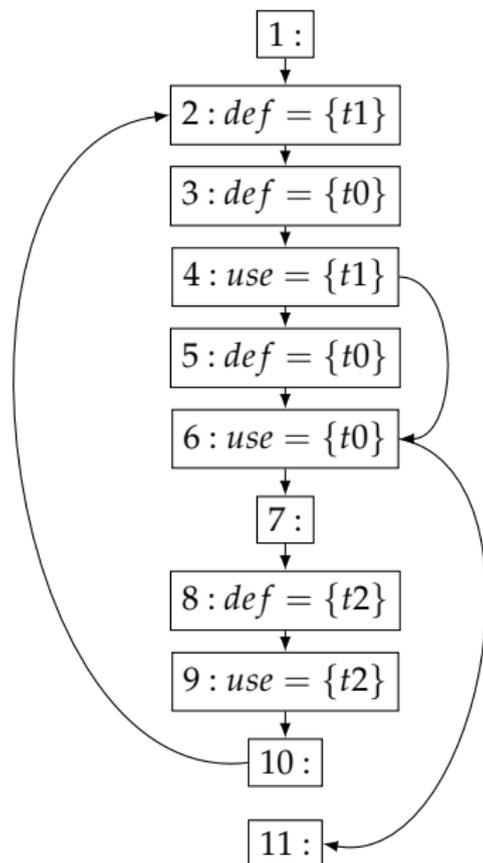
## Itération 2

	1		2	
	in	out	in	out
1				
2				
3				t1
4	t1		t1	t0
5				t0
6	t0		t0	
7				
8				t2
9	t2		t2	
10				
11				



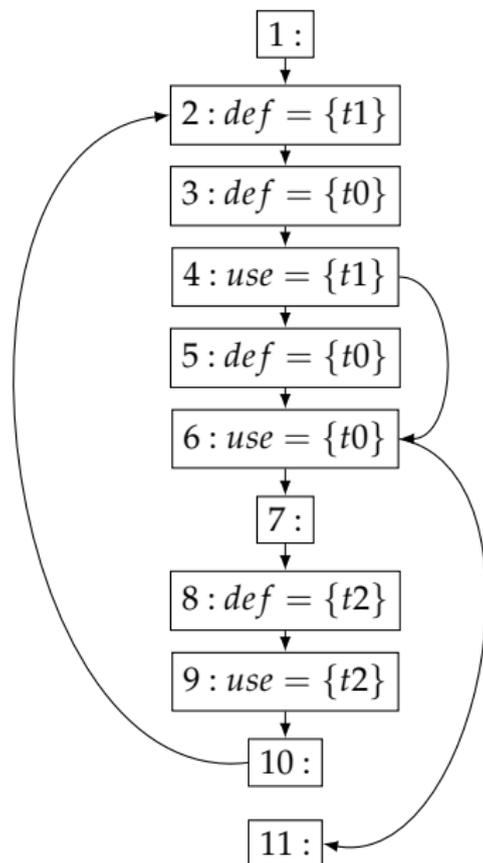
## Itération 3

	2		3	
	in	out	in	out
1				
2				
3		t1		
4	t1	t0		
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



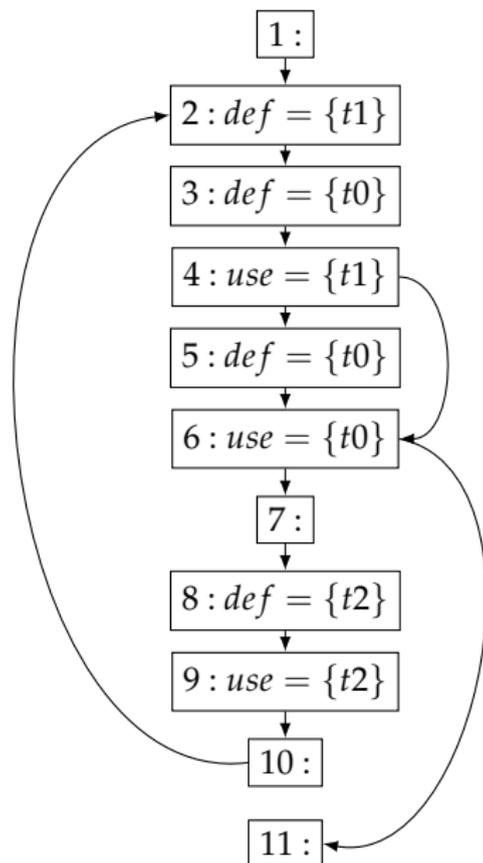
## Itération 3

	2		3	
	in	out	in	out
1				
2				
3		t1	t1	
4	t1	t0		
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



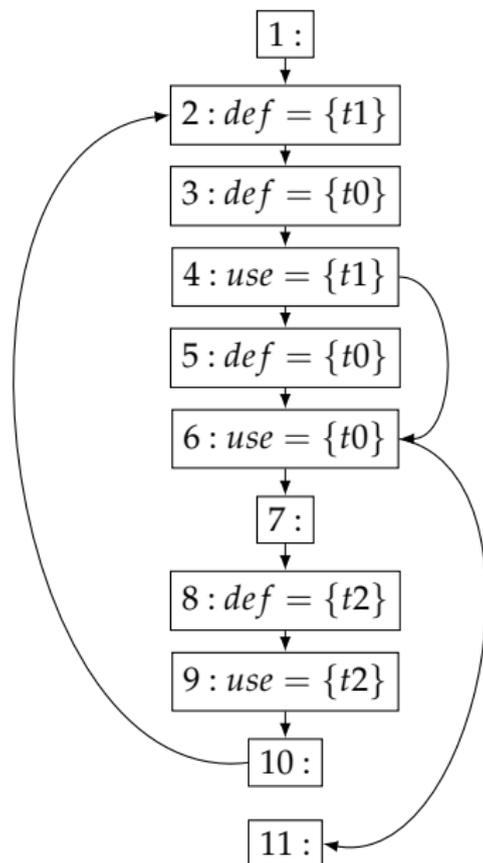
## Itération 3

	2		3	
	in	out	in	out
1				
2				
3		t1	t1	t1
4	t1	t0		
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



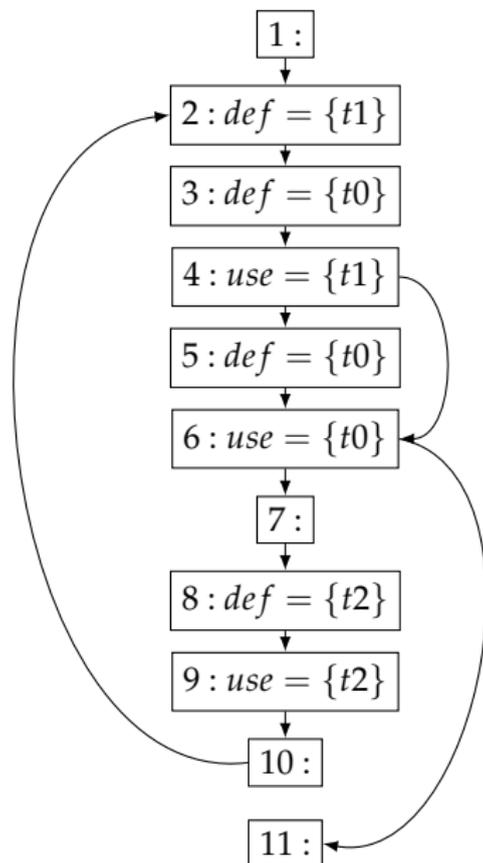
## Itération 3

	2		3	
	in	out	in	out
1				
2				
3		t1	t1	t1
4	t1	t0	t1,t0	
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



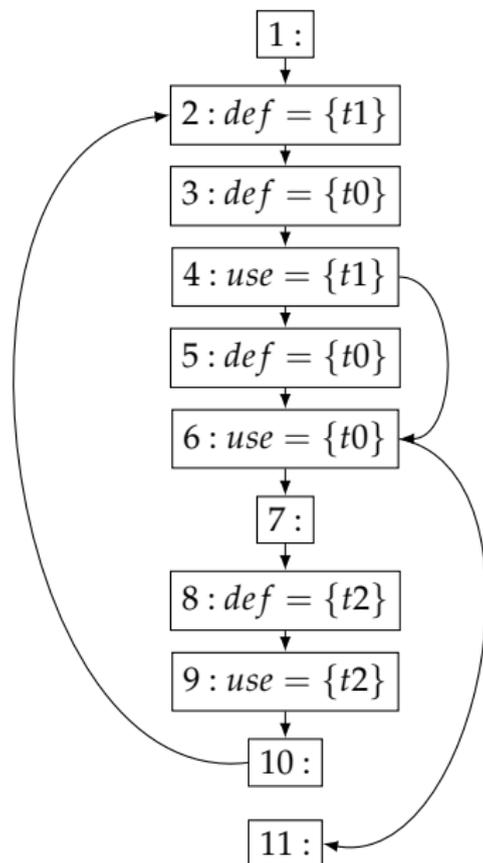
## Itération 3

	2		3	
	in	out	in	out
1				
2				
3		t1	t1	t1
4	t1	t0	t1,t0	t0
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



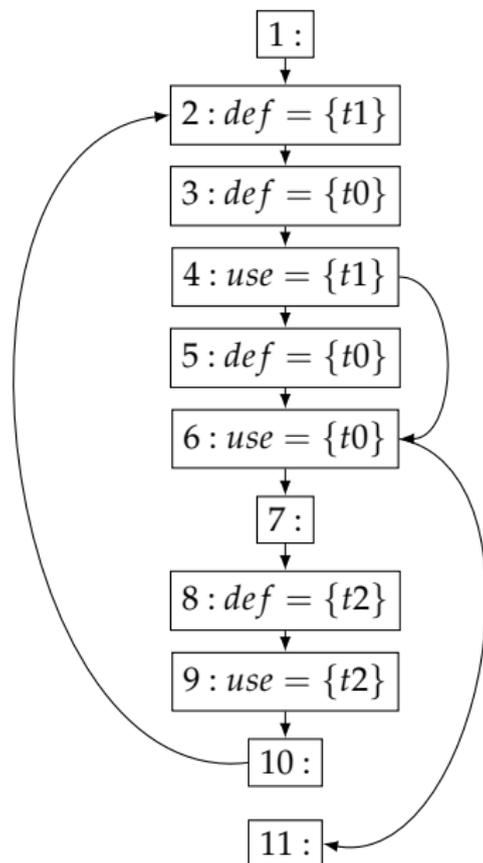
## Itération 3

	2		3	
	in	out	in	out
1				
2				
3		t1	t1	t1
4	t1	t0	t1,t0	t0
5		t0		t0
6	t0			
7				
8		t2		
9	t2			
10				
11				



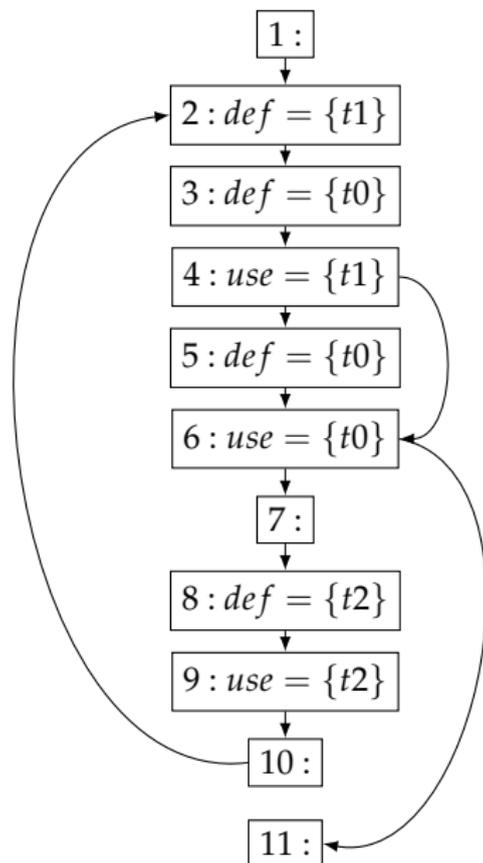
## Itération 3

	2		3	
	in	out	in	out
1				
2				
3		t1	t1	t1
4	t1	t0	t1,t0	t0
5		t0		t0
6	t0		t0	
7				
8		t2		
9	t2			
10				
11				



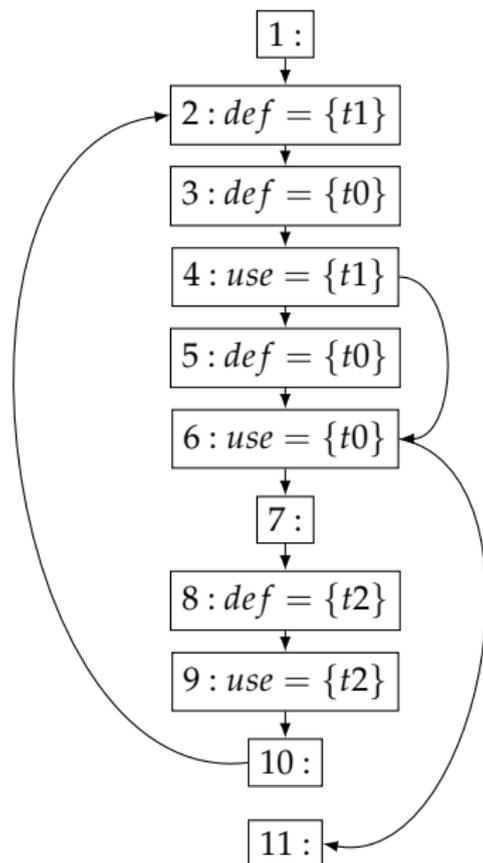
## Itération 3

	2		3	
	in	out	in	out
1				
2				
3		t1	t1	t1
4	t1	t0	t1,t0	t0
5		t0		t0
6	t0		t0	
7				
8		t2		t2
9	t2			
10				
11				



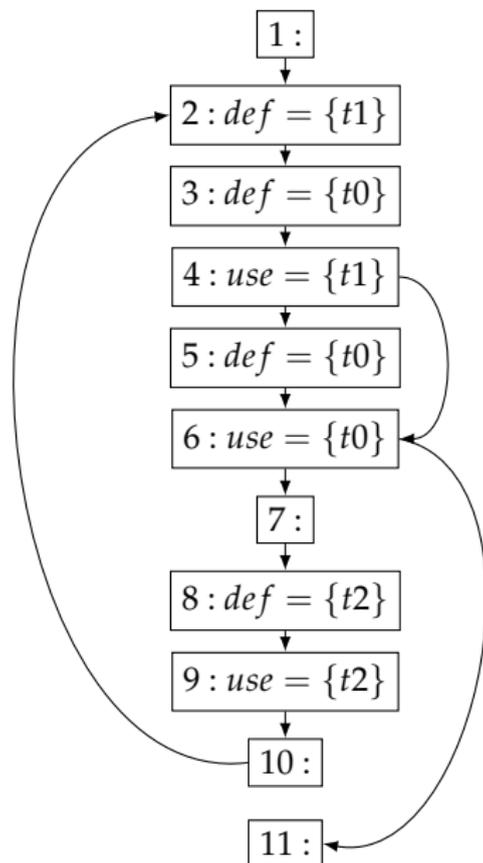
## Itération 3

	2		3	
	in	out	in	out
1				
2				
3		t1	t1	t1
4	t1	t0	t1,t0	t0
5		t0		t0
6	t0		t0	
7				
8		t2		t2
9	t2		t2	
10				
11				



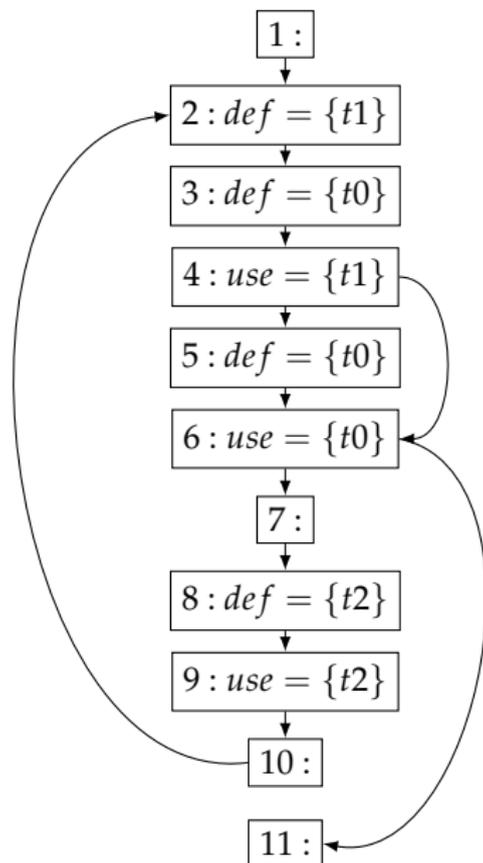
## Itération 4

	3		4	
	in	out	in	out
1				
2				
3	t1	t1		
4	t1, t0	t0		
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



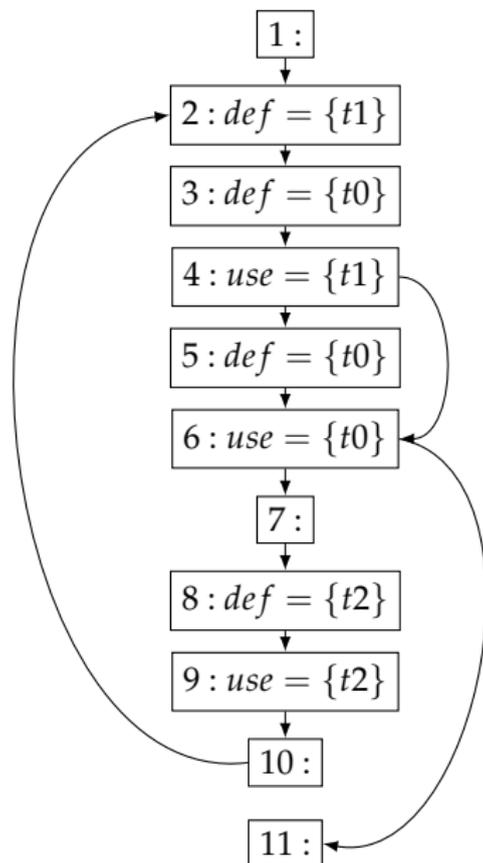
## Itération 4

	3		4	
	in	out	in	out
1				
2				t1
3	t1	t1		
4	t1, t0	t0		
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



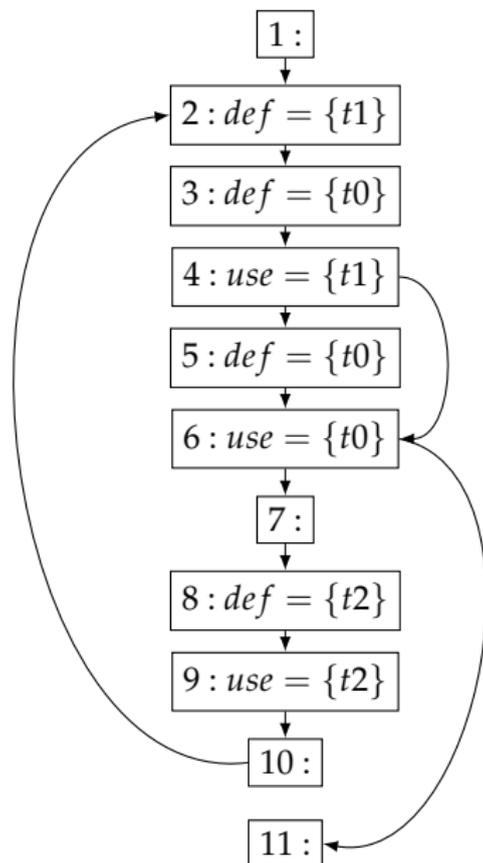
## Itération 4

	3		4	
	in	out	in	out
1				
2				t1
3	t1	t1	t1	
4	t1, t0	t0		
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



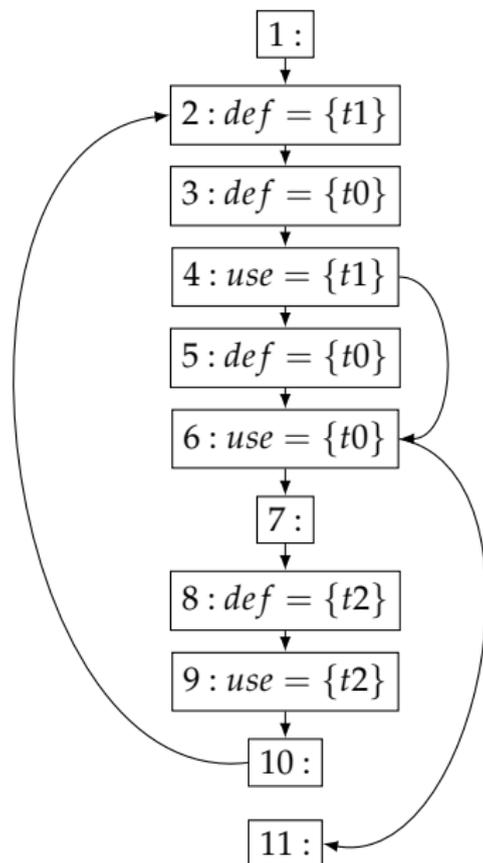
## Itération 4

	3		4	
	in	out	in	out
1				
2				t1
3	t1	t1	t1	t1,t0
4	t1,t0	t0		
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



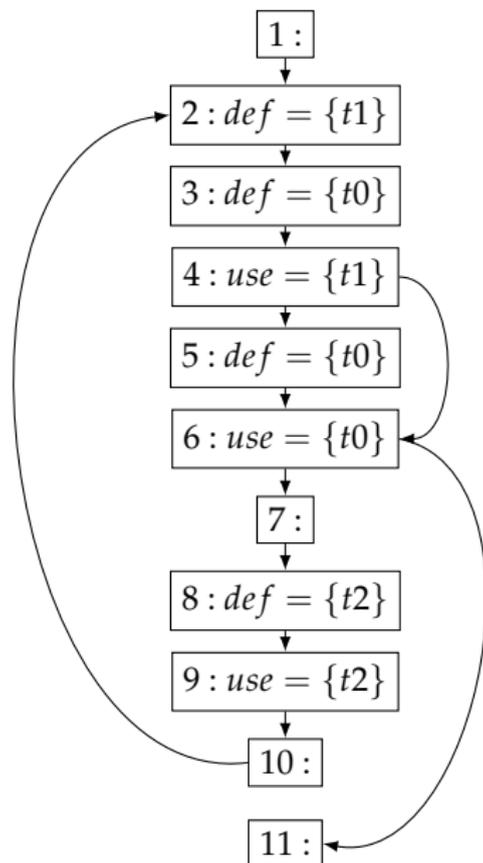
## Itération 4

	3		4	
	in	out	in	out
1				
2				t1
3	t1	t1	t1	t1,t0
4	t1,t0	t0	t1,t0	
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



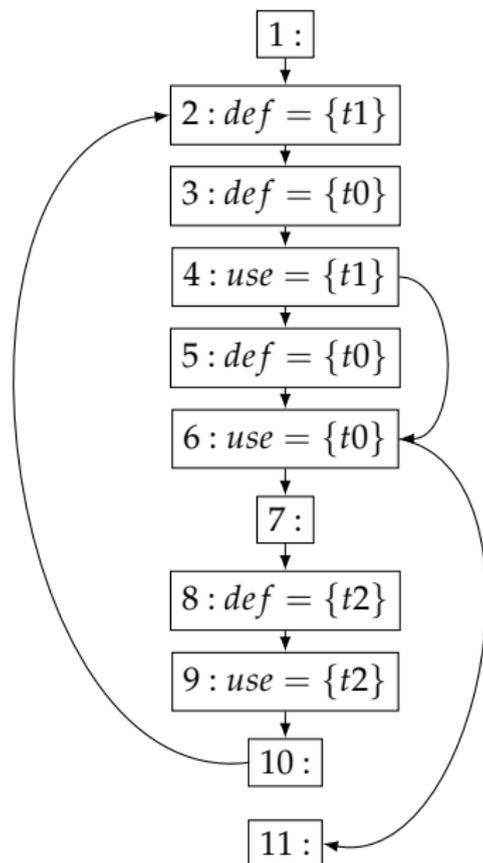
## Itération 4

	3		4	
	in	out	in	out
1				
2				t1
3	t1	t1	t1	t1,t0
4	t1,t0	t0	t1,t0	t0
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



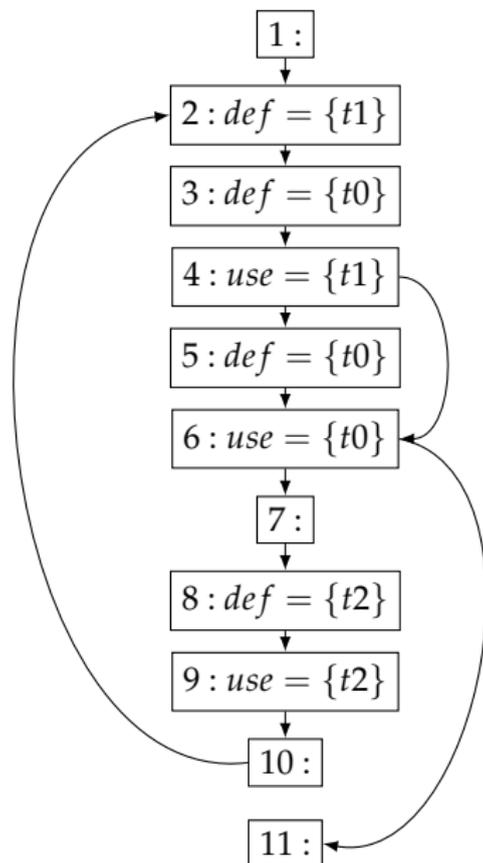
## Itération 4

	3		4	
	in	out	in	out
1				
2				t1
3	t1	t1	t1	t1,t0
4	t1,t0	t0	t1,t0	t0
5		t0		t0
6	t0			
7				
8		t2		
9	t2			
10				
11				



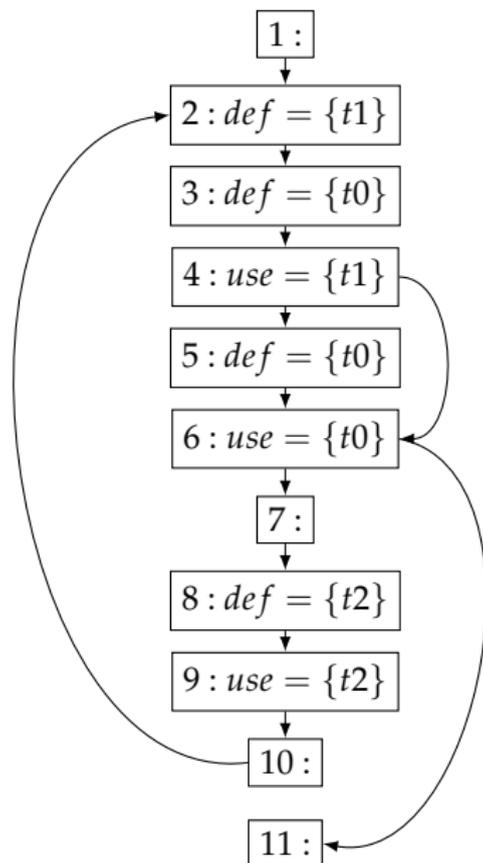
## Itération 4

	3		4	
	in	out	in	out
1				
2				t1
3	t1	t1	t1	t1,t0
4	t1,t0	t0	t1,t0	t0
5		t0		t0
6	t0		t0	
7				
8		t2		
9	t2			
10				
11				



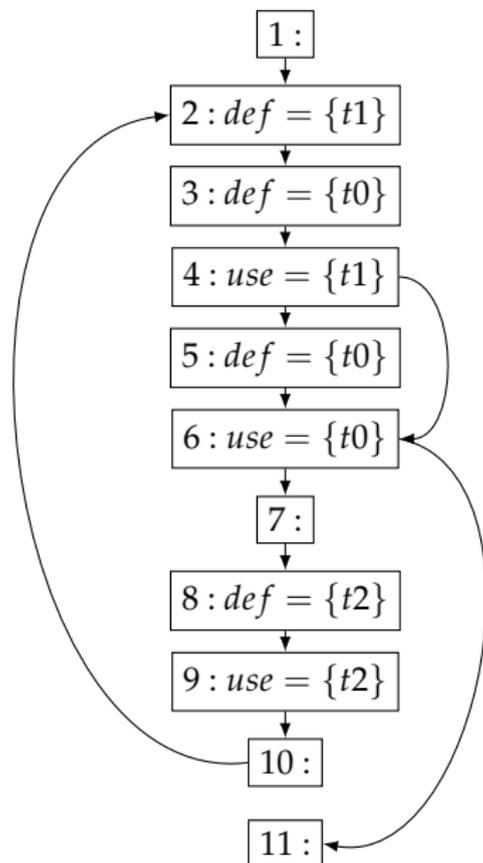
## Itération 4

	3		4	
	in	out	in	out
1				
2				t1
3	t1	t1	t1	t1,t0
4	t1,t0	t0	t1,t0	t0
5		t0		t0
6	t0		t0	
7				
8		t2		t2
9	t2			
10				
11				



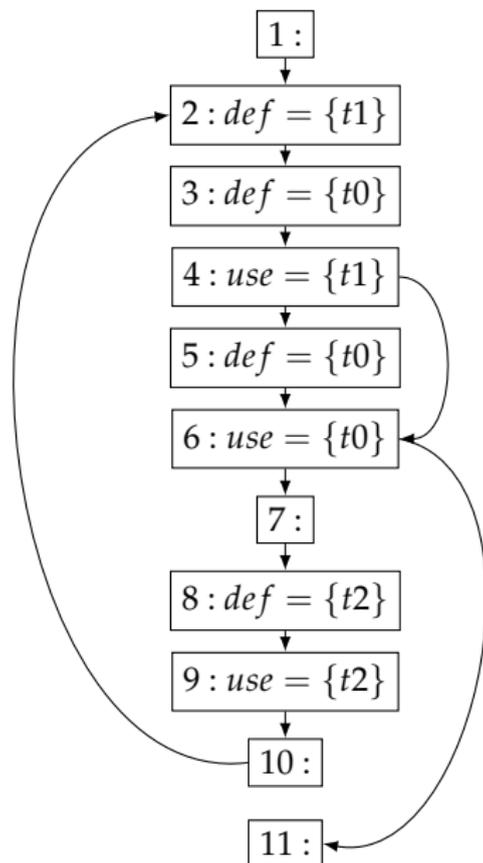
## Itération 4

	3		4	
	in	out	in	out
1				
2				t1
3	t1	t1	t1	t1,t0
4	t1,t0	t0	t1,t0	t0
5		t0		t0
6	t0		t0	
7				
8		t2		t2
9	t2		t2	
10				
11				



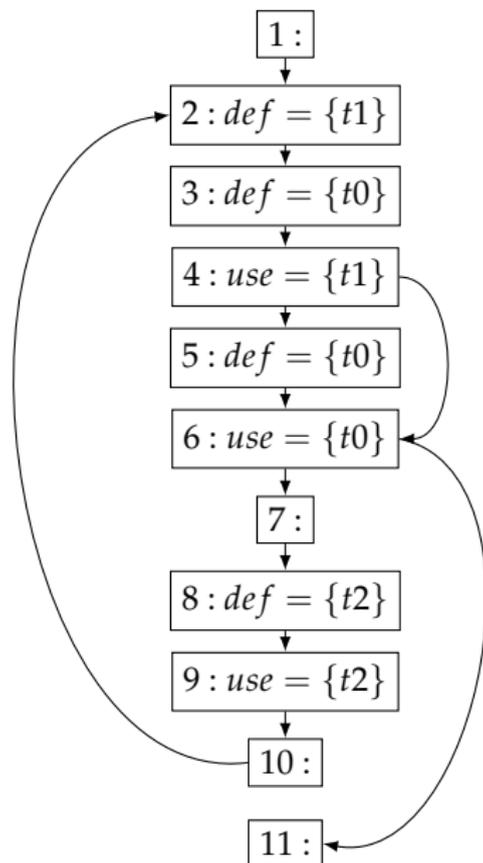
# Itération 5

	4		5	
	in	out	in	out
1				
2		t1		
3	t1	t1,t0		
4	t1,t0	t0		
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



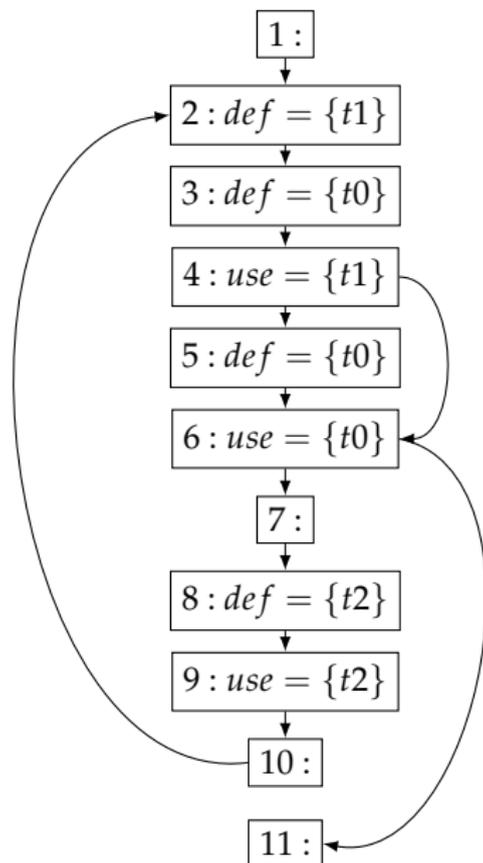
## Itération 5

	4		5	
	in	out	in	out
1				
2		t1		t1
3	t1	t1,t0		
4	t1,t0	t0		
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



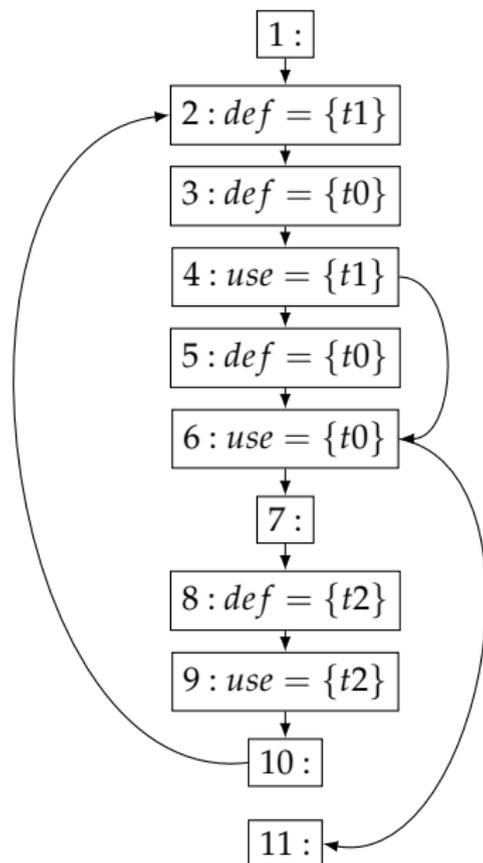
## Itération 5

	4		5	
	in	out	in	out
1				
2		t1		t1
3	t1	t1,t0	t1	
4	t1,t0	t0		
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



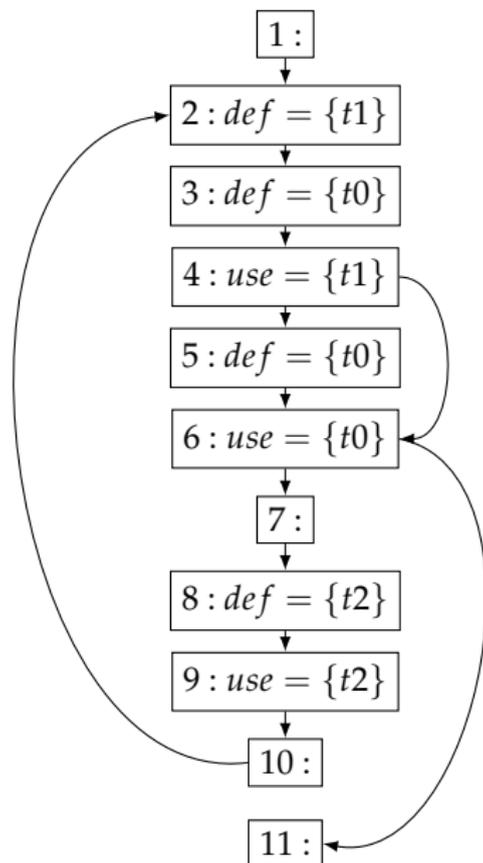
# Itération 5

	4		5	
	in	out	in	out
1				
2		t1		t1
3	t1	t1,t0	t1	t1,t0
4	t1,t0	t0		
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



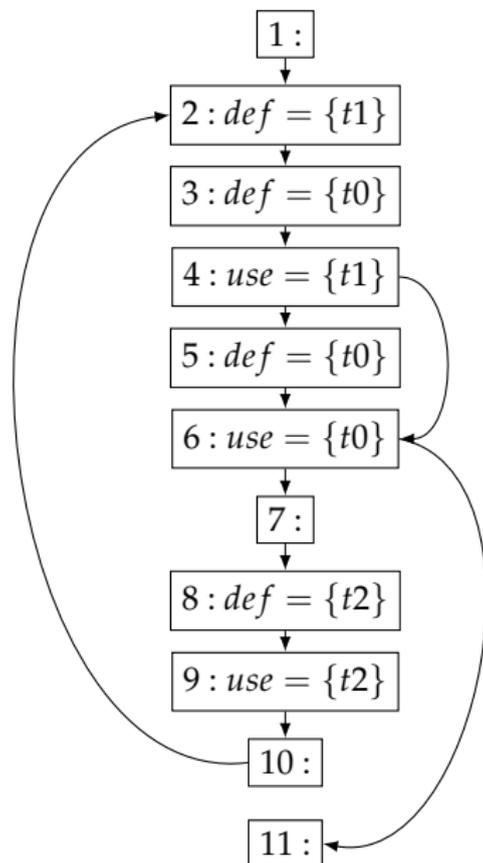
## Itération 5

	4		5	
	in	out	in	out
1				
2		t1		t1
3	t1	t1,t0	t1	t1,t0
4	t1,t0	t0	t1,t0	
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



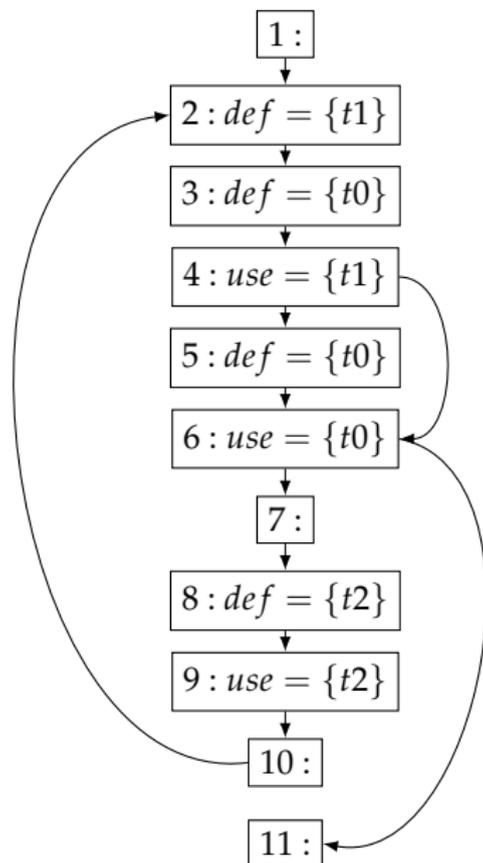
## Itération 5

	4		5	
	in	out	in	out
1				
2		t1		t1
3	t1	t1,t0	t1	t1,t0
4	t1,t0	t0	t1,t0	t0
5		t0		
6	t0			
7				
8		t2		
9	t2			
10				
11				



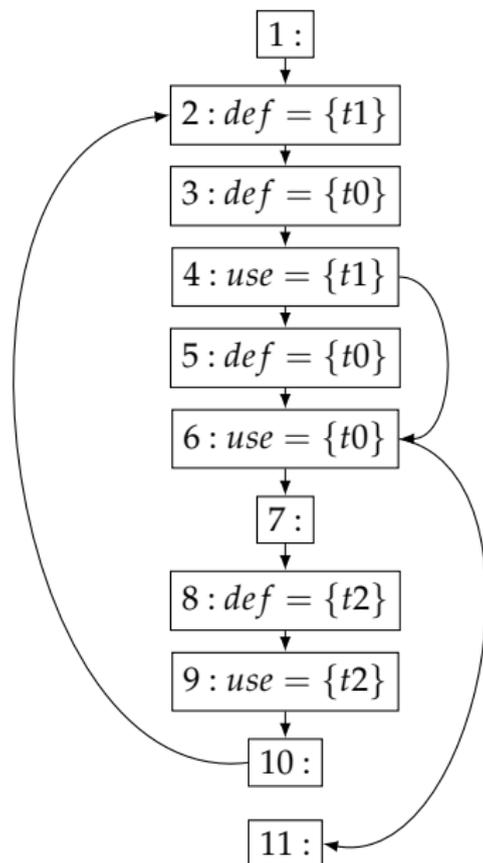
## Itération 5

	4		5	
	in	out	in	out
1				
2		t1		t1
3	t1	t1,t0	t1	t1,t0
4	t1,t0	t0	t1,t0	t0
5		t0		t0
6	t0			
7				
8		t2		
9	t2			
10				
11				



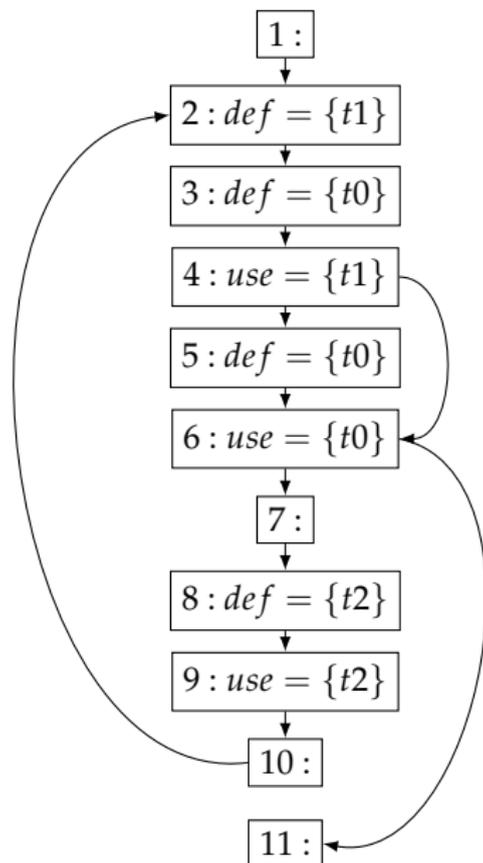
# Itération 5

	4		5	
	in	out	in	out
1				
2		t1		t1
3	t1	t1,t0	t1	t1,t0
4	t1,t0	t0	t1,t0	t0
5		t0		t0
6	t0		t0	
7				
8		t2		
9	t2			
10				
11				



# Itération 5

	4		5	
	in	out	in	out
1				
2		t1		t1
3	t1	t1,t0	t1	t1,t0
4	t1,t0	t0	t1,t0	t0
5		t0		t0
6	t0		t0	
7				
8		t2		t2
9	t2			
10				
11				



# Itération 5

	4		5	
	in	out	in	out
1				
2		t1		t1
3	t1	t1,t0	t1	t1,t0
4	t1,t0	t0	t1,t0	t0
5		t0		t0
6	t0		t0	
7				
8		t2		t2
9	t2		t2	
10				
11				

