

Structure d'un compilateur

Fabienne Carrier

Université Grenoble Alpes

4 novembre 2016

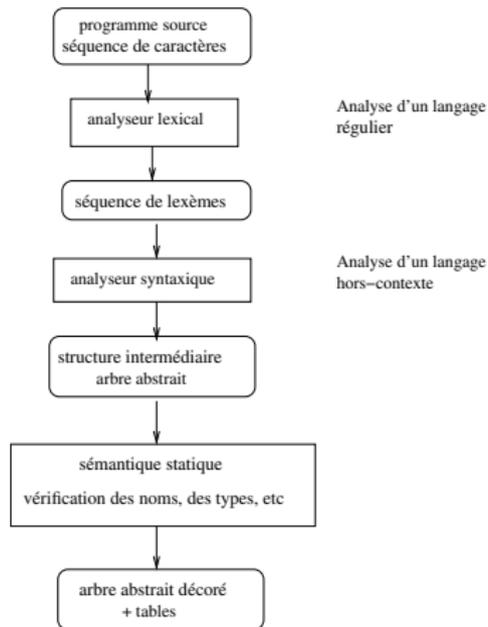
Plan

- 1 Organisation d'un compilateur
- 2 Structures intermédiaires dans un compilateur

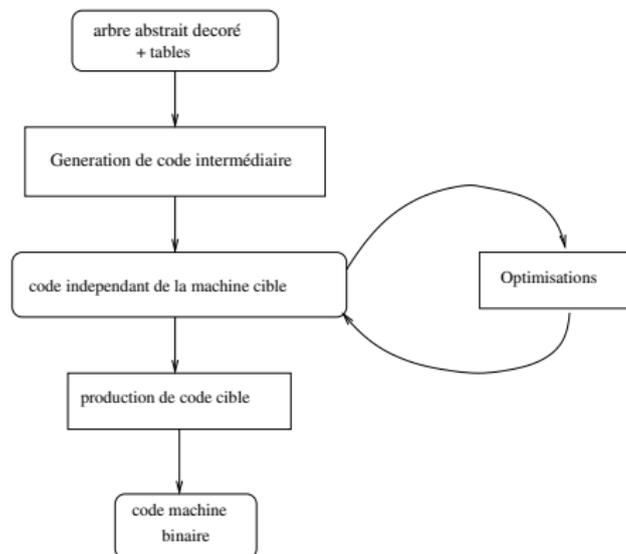
Plan

- 1 Organisation d'un compilateur
- 2 Structures intermédiaires dans un compilateur

Organisation du "Front-End"



Organisation du "Back-End"



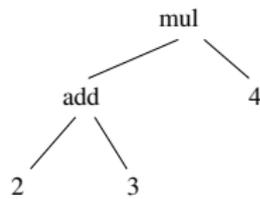
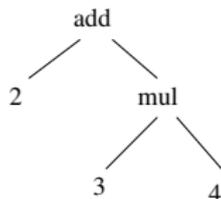
Plan

- 1 Organisation d'un compilateur
- 2 Structures intermédiaires dans un compilateur**

Notion d'arbre abstrait

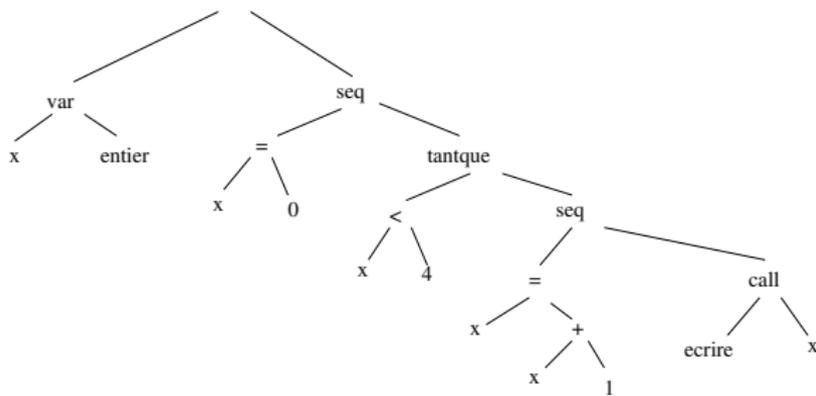
Représentation arborescente simplifiée du programme source

Exemple : $2 + 3 * 4$ et $(2 + 3) * 4$



Arbre abstrait : un exemple

```
int x;
x=0;
while (x<4){
    x=x+1;
    ecrire (x);
}
```



Grammaire abstraite

Un arbre abstrait peut être décrit par une grammaire abstraite

Exemple :

$P ::= B$

$B ::= D C$

$D ::= \text{var } x \text{ t} \mid D ; D$

$C ::= x := E \mid C ; C \text{ si } E \text{ alors } C \text{ sinon } C \mid \text{tantque } E C$

$E ::= n \mid b \mid x \mid E + E \mid E * E \mid E \text{ et } E \mid \text{non } E$

Table des symboles

Pour chaque symbole : nom, type, emplacement mémoire, etc

- Tables locales
- Table globale

Code intermédiaire : exemple

```
int x;  
x=0;  
while (x<4){  
    x=x+1;  
    ecrire (x);  
}
```

```
1: x = 1  
2: si x>=4 allera 5  
3: x = x +1  
4: ecrire x  
5:
```