

Compilation



Prof. Santucci Jean-François

SPE – UMR 6134

UNIVERSITE DE CORSE

email : santucci@univ-corse.fr

Références

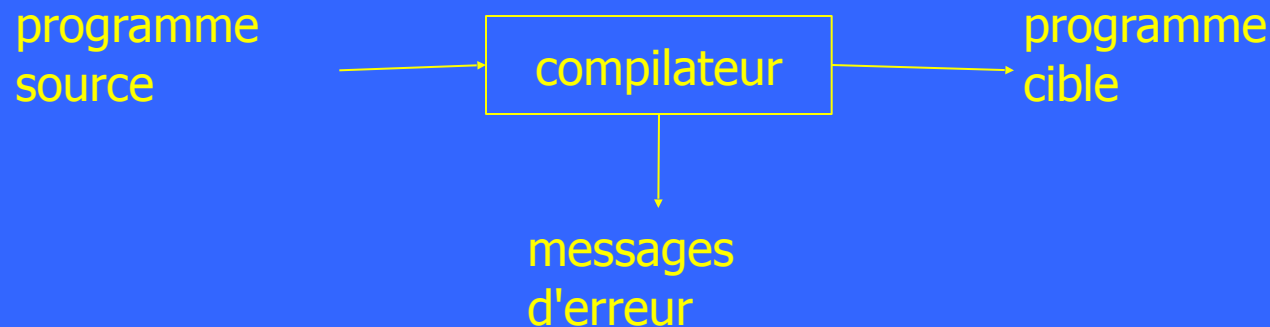
Aho, A. V. and Ullman, J. D. (1977) The Principles of Compiler Design, Addison Wesley, Reading, Mass.

Bornat, R. (1979), Understanding and Writing Compilers, Macmillan.

Gries, D. (1971), Compiler Construction for Digital Computers, Wiley, N.Y.

Introduction à la compilation

Un compilateur est un programme qui traduit un autre programme



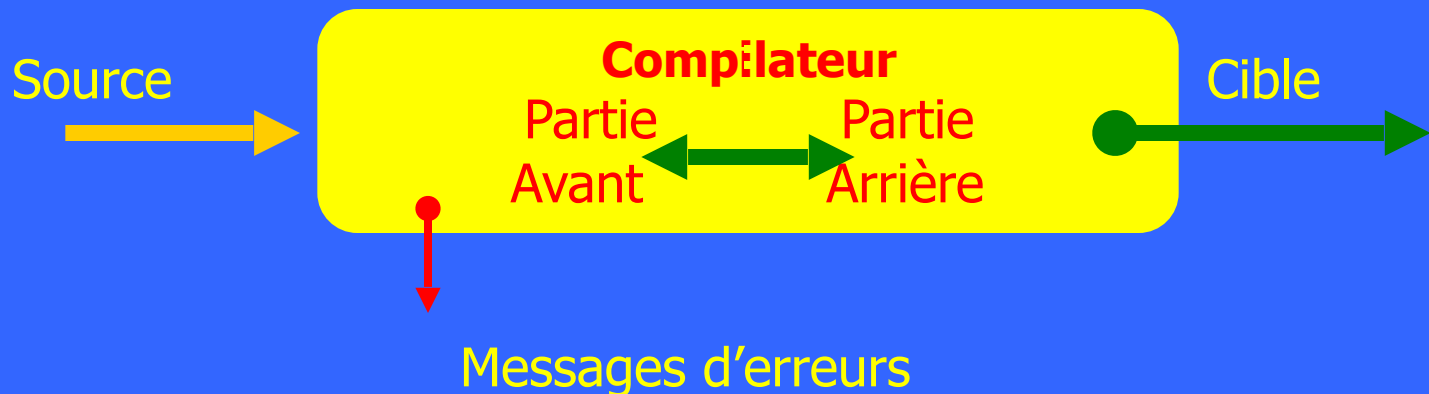
Premier compilateur : compilateur Fortran de J. Backus (1957)

Langage source : langage de haut niveau (C, C++, Java, Pascal, Fortran)

Langage cible : langage de bas niveau (assembleur, langage machine)

Compilation : Introduction

- [?] Traduire un programme source dans un langage cible
 - [?] analyse le programme conformément à la définition du langage source
 - [?] produit d'éventuels messages d'erreurs
 - [?] synthétise un code équivalent au programme source dans le langage cible



Cousins des compilateurs



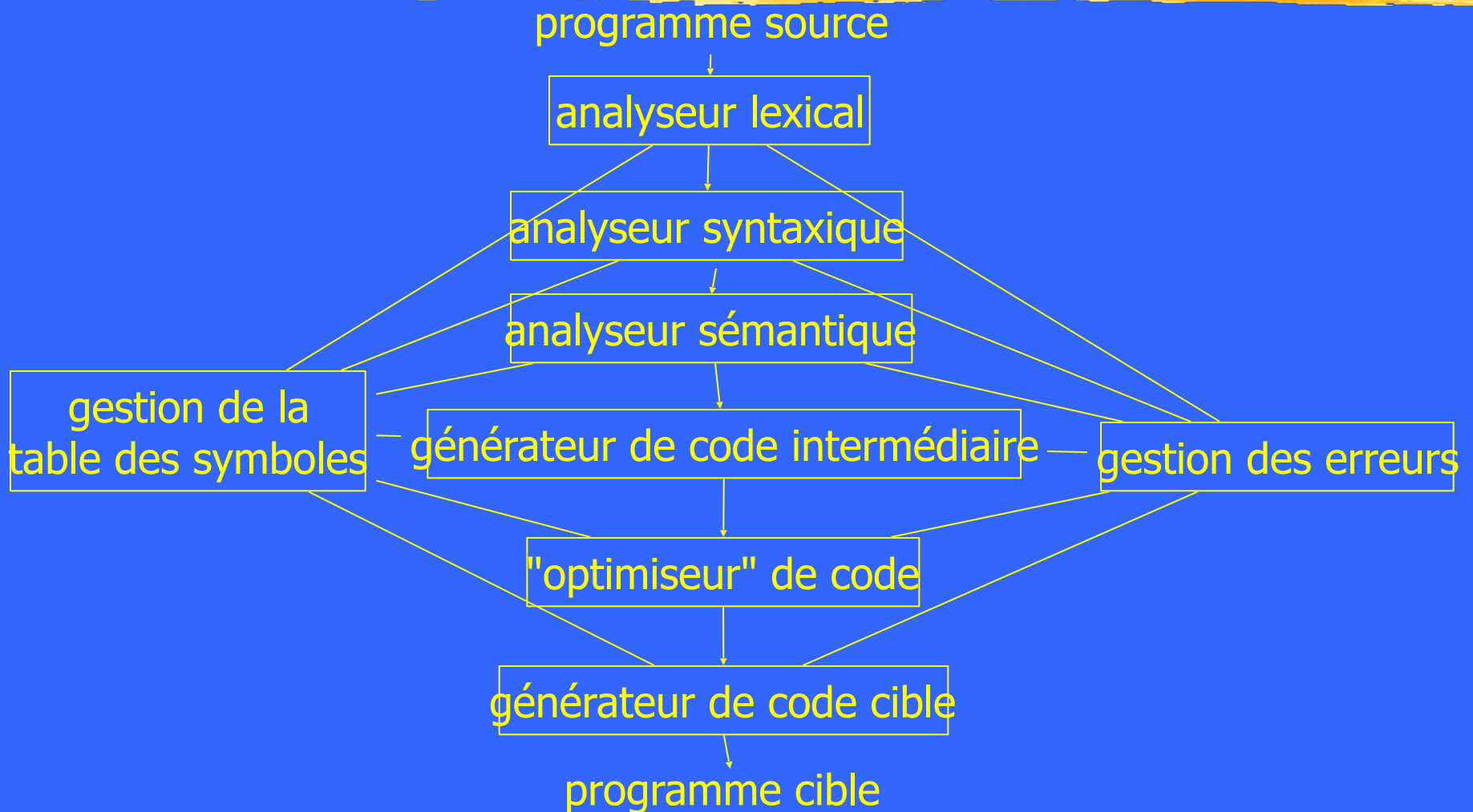
Interpréteurs

Diffèrent d'un compilateur par l'intégration de l'exécution et de la traduction. Utilisés pour les langages de commande

Formateurs de texte

Traduit le code source dans le langage de commande d'une imprimante

Les phases de la compilation

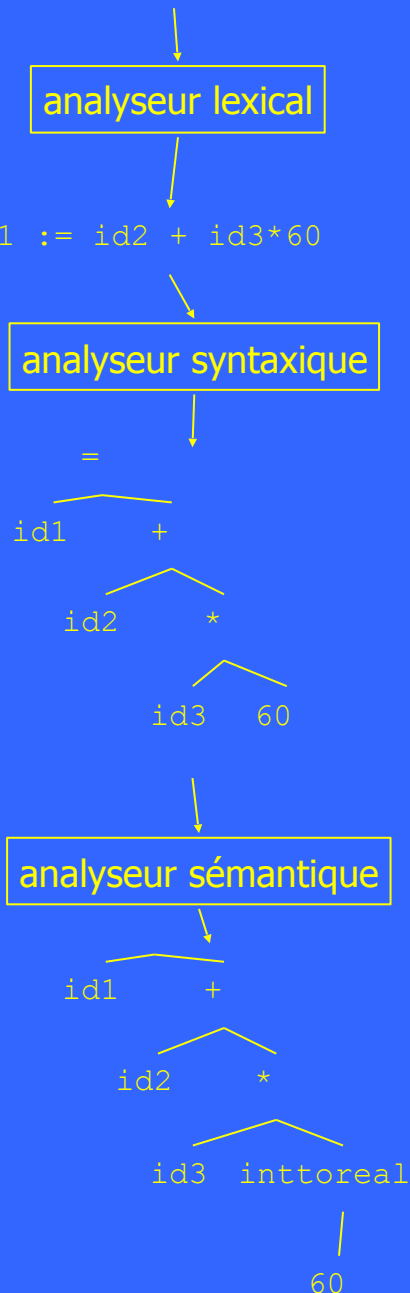


position = initial + vitesse*60

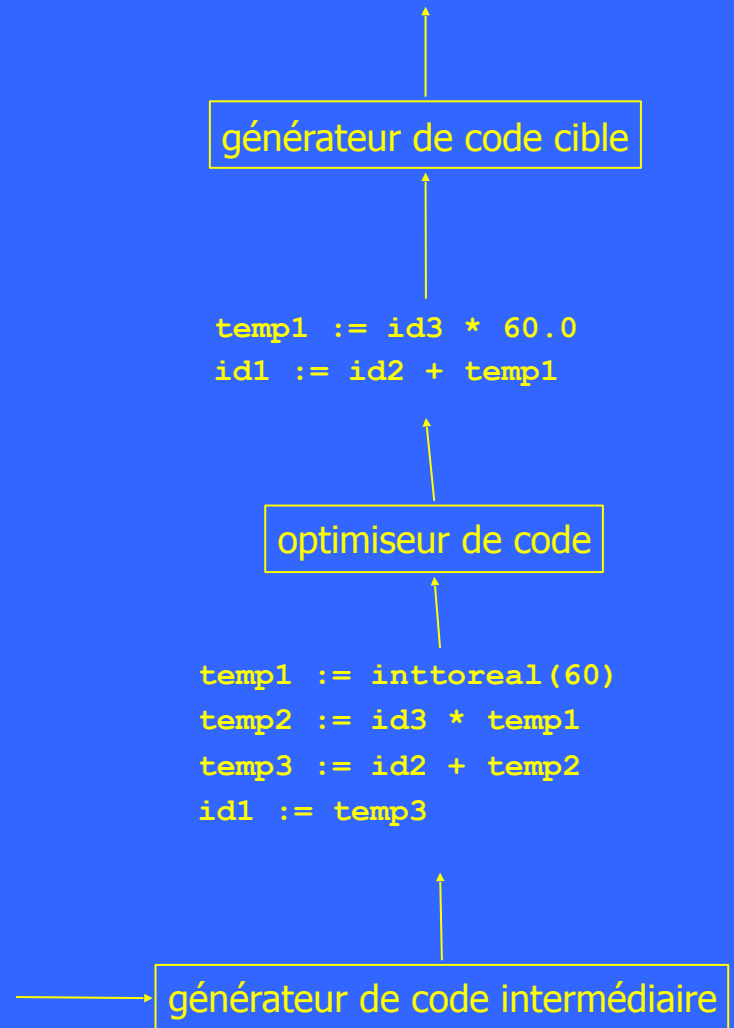
Table des symboles

1	position	...
2	initial	...
3	vitesse	...
4		
5		
...		

id1 := id2 + id3*60



```
MOVF id3, R2
MULF #60.0, R2
MOVF id2, R1
ADDF R2, R1
MOVF R1, id1
```



Langages d'assemblage

Les langages d'assemblage ou assembleurs sont des versions un peu plus lisibles du code machine avec des noms symboliques pour les opérations et pour les opérandes

```
MOV a, R1
```

```
ADD #2, R1
```

```
MOV R1, b
```

Chaque processeur a son langage d'assemblage

Groupement des phases

Partie frontale (front end)

Regroupe tout ce qui dépend du langage source plutôt que de la machine cible. On peut utiliser la même partie frontale sur une machine différente

Partie arrière (back end)

Regroupe le reste

Passes

Plusieurs phases peuvent être regroupées dans une même passe consistant

à lire un fichier et en écrire un autre

Analyse lexicale, syntaxique, sémantique et génération de code intermédiaire

peuvent être regroupées en une seule passe

Outils logiciels

Outils d'aide à la construction de compilateurs

Générateurs d'analyseurs lexicaux

Engendrent un analyseur lexical (scanner, lexer) sous forme d'automate

fini à partir d'une spécification sous forme d'expressions rationnelles
Flex, Lex

Générateurs d'analyseurs syntaxiques

Engendrent un analyseur syntaxique (parser) à partir d'une grammaire

Bison, Yacc

Générateurs de traducteurs

Engendrent un traducteur à partir d'un schéma de traduction (grammaire +

règles sémantiques) Bison, Yacc

PLAN



HISTORIQUE COMPILATEUR

ANALYSE LEXICALE

ANALYSE SYNTAXIQUE

TABLE DES SYMBOLES

GENERATION DE CODE

ECRITURE D'UN COMPILATEUR