

COMPILATION

**© MOURCHID Mohammed
Département d'Informatique
Faculté des Sciences Ibn Tofail**

Plan

- 1. Introduction de la compilation**
- 2. Analyse lexicale**
- 3. Analyse syntaxique**
- 4. Analyse sémantique**

Chapitre 1

Introduction

à la compilation

© Compilation / M.MOURCHID, SMI, 2021-2022

3

Sommaire

Langages de programmation

Qu'est-ce qu'un compilateur ?

Phases d'un compilateur

L'analyseur lexical

L'analyseur syntaxique

L'analyseur sémantique

© Compilation / M.MOURCHID, SMI, 2021-2022

4

Langages de programmation

Langages de programmation :

- **Langage machine** : code binaire directement exécutable par un ordinateur.
- **Langage assembleur** : utilise un mnémonique pour coder les instructions (add, mul, mov, etc). Il est très proche du langage machine.
- **Langage de haut niveau** : utilise des structures de contrôle (tests, boucles, appels de fonctions, etc). Le programme est proche du langage naturel.

Remarques

- Seul le langage machine (code binaire) est directement exécutable.
- Un outil est nécessaire pour convertir n'importe quel langage en code binaire : cet outil s'appelle un **compilateur**

© Compilation / M.MOURCHID, SMI, 2021-2022

5

Qu'est-ce qu'un compilateur ?

Définition : un **compilateur** est un programme qui lit en entrée un programme écrit dans un premier langage (appelé **langage source**) et le traduit en un programme équivalent écrit dans un autre langage (appelé **langage cible**). Au cours de ce processus de traduction, le compilateur tentera également de repérer et de signaler les erreurs évidentes commises par le programmeur.

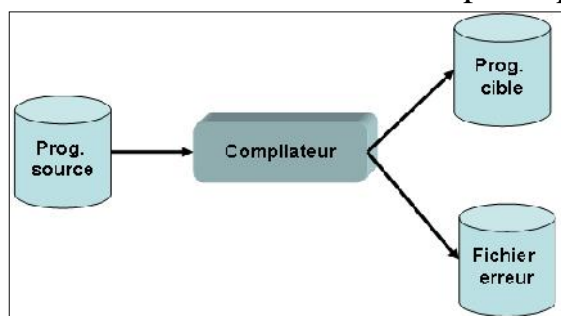


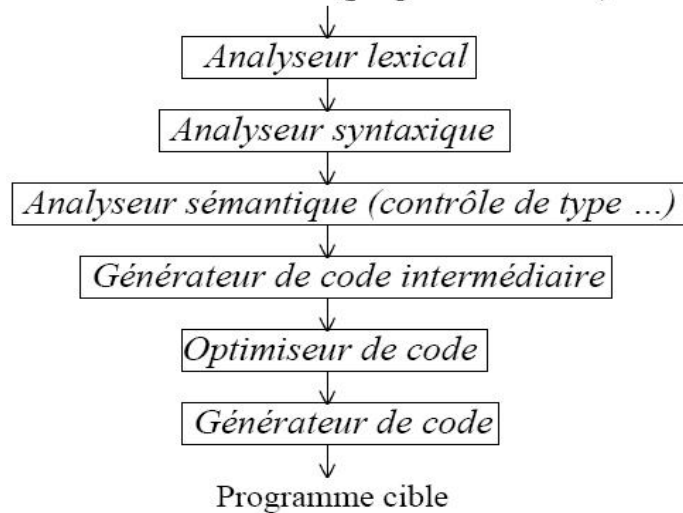
Fig.1.
Schéma
simplifié d'un
compilateur.

© Compilation / M.MOURCHID, SMI, 2021-2022

6

Phases d'un compilateur

Chaîne de caractères (programme source)



© Compilation / M.MOURCHID, SMI, 2021-2022

7

Phases d'un compilateur

Modules de la partie analyse d'un compilateur

- L'analyseur lexical (*scanner*).
- L'analyseur syntaxique (*parser*).
- L'analyseur sémantique (*Type checker*).

Modules de la partie synthèse d'un compilateur

- La génération de code intermédiaire.
- L'optimisation de code intermédiaire.
- La génération de code cible.

© Compilation / M.MOURCHID, SMI, 2021-2022

8

L'analyseur lexical

Rôle : l'analyseur lexical prend en entrée le flot de caractères constituant le programme source et délivre les unités lexicales.

- Le flot de caractères du programme source est lu de gauche à droite.
- Les caractères formant un sens sont groupés en unités lexicales (*tokens*).
- Les caractères superflus sont supprimés par l'analyseur lexical.
- Une des tâches de l'analyseur lexical est d'ignorer les commentaires.

Unité lexicale (token) : c'est une suite de caractères ayant une signification collective.

L'analyseur lexical doit aussi déterminer si chaque lexème est un mot correct du vocabulaire du langage.

© Compilation / M.MOURCHID, SMI, 2021-2022

9

L'analyseur lexical

- Exemple

- Soit l'instruction Pascal :
`aire:=base*hauteur/2`
- L'analyse lexicale reconnaît les lexèmes suivants :

aire	:=	base	*	hauteur	/	2
------	----	------	---	---------	---	---

- Les unités lexicales reconnues sont :

<identificateur> aire
<affectation> :=
<identificateur> base
<opérateur> *
<identificateur> hauteur
<opérateur> /
<nombre> 2

© Compilation / M.MOURCHID, SMI, 2021-2022

10

L'analyseur syntaxique

Rôle : l'analyseur syntaxique prend en entrée un flot d'unités lexicales fournit par l'analyseur lexical et délivre les unités syntaxiques.

- Une unité syntaxique est une structure grammaticale représentée généralement par des arbres dites « *arbres syntaxiques* ».

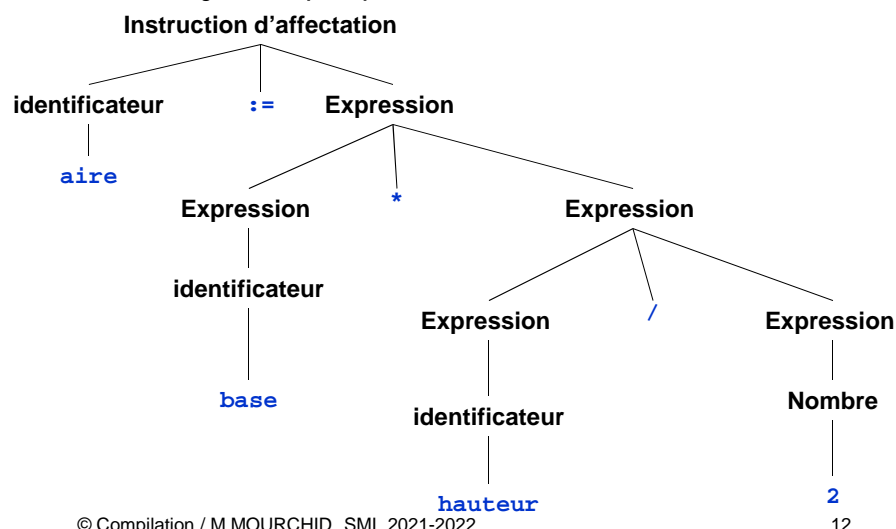
L'analyseur syntaxique doit également déterminer si chaque unité syntaxique est une phrase correcte du langage, c'est-à-dire qu'elle vérifie sa grammaire. Cette dernière est définie par un ensemble de règles de production.

© Compilation / M.MOURCHID, SMI, 2021-2022

11

L'analyseur syntaxique

- Exemple
 - L'arbre syntaxique produit est :



© Compilation / M.MOURCHID, SMI, 2021-2022

12

L'analyseur sémantique

L'analyse sémantique contrôle si le programme source contient des erreurs sémantiques et collecte des informations de type destinées à la phase de production de code qui la suit.

- L'analyseur sémantique prend en entrée l'arbre syntaxique délivré par l'analyseur syntaxique pour identifier les opérateurs et les opérandes des expressions, ainsi que les instructions.
- La tâche importante de l'analyseur sémantique est la **vérification de type**. Par exemple, dans la plupart des langages de programmation, l'indice d'un tableau doit être de type entier et non pas de type réel.

© Compilation / M.MOURCHID, SMI, 2021-2022

13

L'analyseur sémantique

- Une autre activité de l'analyseur sémantique est la **conversion implicite de type**.
- Par exemple, additionner un « float » et un « int » ne se fait pas directement, il faut transformer la valeur int en une valeur float (elles n'ont pas la même représentation binaire).

Autres activités de l'analyseur sémantique :

- Détecter les **erreurs d'incompatibilité de types** : opération sur deux types incompatibles (« fichier + ensemble » ou « fonction * tableau »).
- Détecter certaines **erreurs de calcul** : division par 0, racine carrée ou logarithme d'un nombre négatif.

© Compilation / M.MOURCHID, SMI, 2021-2022

14

L'analyseur sémantique

- Exemple
 - Ici l'analyseur sémantique va effectuer la conversion implicite de type

