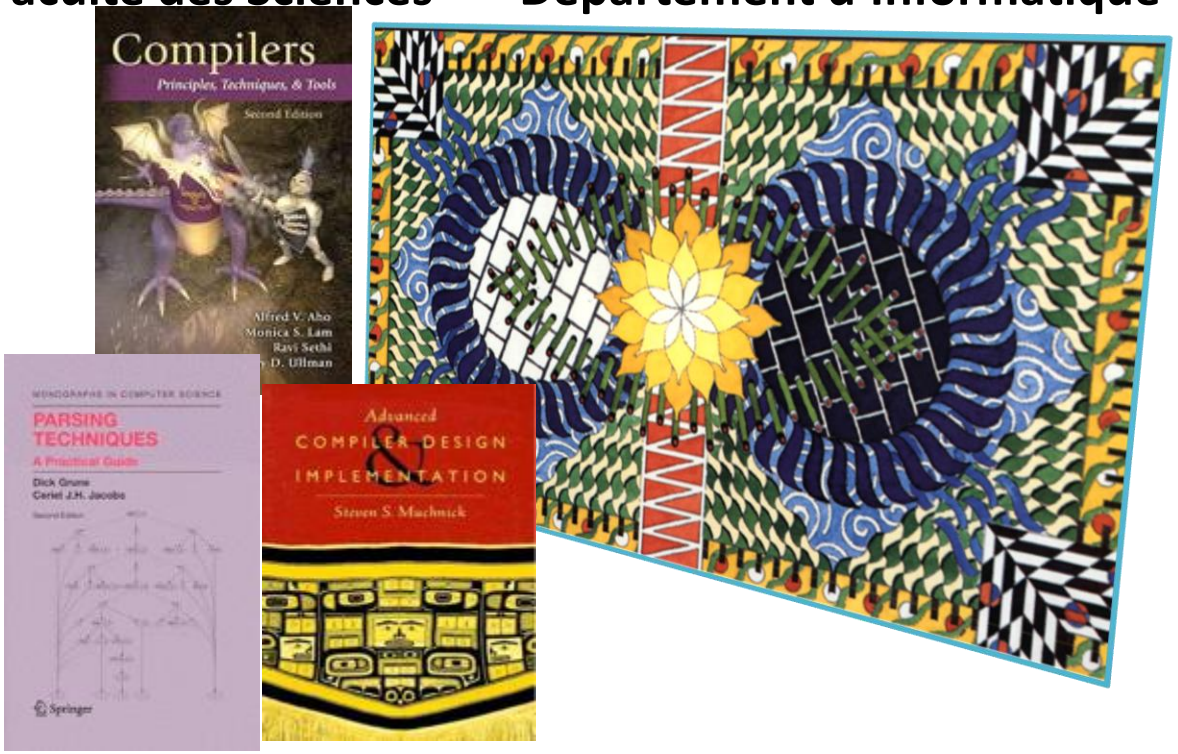
FERHAT ABBES- SETIF-1 UNIVERSITY
UNIVERSITE FERHAT ABBES – SETIF 1

جامعة فرحات عباس- سطيف

Université Ferhat Abbas Sétif 1

Faculté des Sciences - Département d'Informatique



Compilation: Introduction

3ème Année License

Dr .Abdelaziz LAKHFIF

Bibliographie

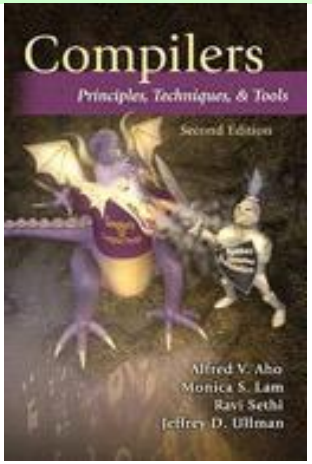
Compilers: Principles, Techniques, and Tools, Aho, Sethi and Ullman

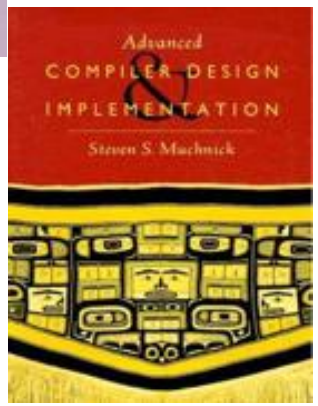
—http://dragonbook.stanford.edu/

> **Parsing Techniques, Grune and Jacobs**

—http://www.cs.vu.nl/~dick/PT2Ed.html

> **Advanced Compiler Design and Implementation, Muchnik**





Introduction aux Compilateurs

Programme :

✓Introduction, position du problème, plan du cours.
✓Syntaxe abstraite et interprétation de l'Assembleur du cours d'architecture.
✓Génération de code pour les expressions arithmétiques et pour les structures de contrôle (Assembleur cours d'Architecture).
✓Analyse lexicale – Lex.
✓Analyse syntaxique – Yacc.
✓Application : Un lexeur-parseur pour Assembleur (cours d'Architecture).
✓Blocs : principe (Assembleur cours d'Architecture).
✓Blocs : réalisation (Assembleur cours d'Architecture).
✓Fonctions : principe (Assembleur cours d'Architecture).
✓Fonctions : réalisation (Assembleur cours d'Architecture).
✓Procédures (Assembleur cours d'Architecture).

Introduction aux Compilateurs

Motivations et Histoire.

- Architecture d'un Compilateur.
- la Phase d'Analyse.
- la Phase de Génération.
- Vers un Code Exécutable: Assembleur, Chargeur (Loader) et l'Editeur de lien (Linker).

Introduction aux Compilateurs

Langages de programmation : c'est l'explosion

Langages Industriels

Apple	Swift, OpenCL
Facebook	Hack, React
Google	Go, Dart, Angular, GWT, MapReduce
Intel	Ct, Cilk
Microsoft	C#, F#, Rx
Mozilla	JavaScript2, Rust, asm.js
nVidia	CUDA
Wolfram	Wolfram
Yahoo	Hadoop

Langages académique

EPFL	Scala
Northeastern, Utah, Brown, …	Racket
Edinburgh	Haskell
CMU, ML,	O'Caml
MIT	Julia, Scratch
Stanford	D3
Berkeley	Spark

Langages créatifs

Yukihiro Matsumoto	Ruby
John Resig	jQuery
Walter Bright, Andrei Alexandrescu	D
Rich Hickey	Clojure
.....	the next big thing

Pourquoi cette évolution rapide?

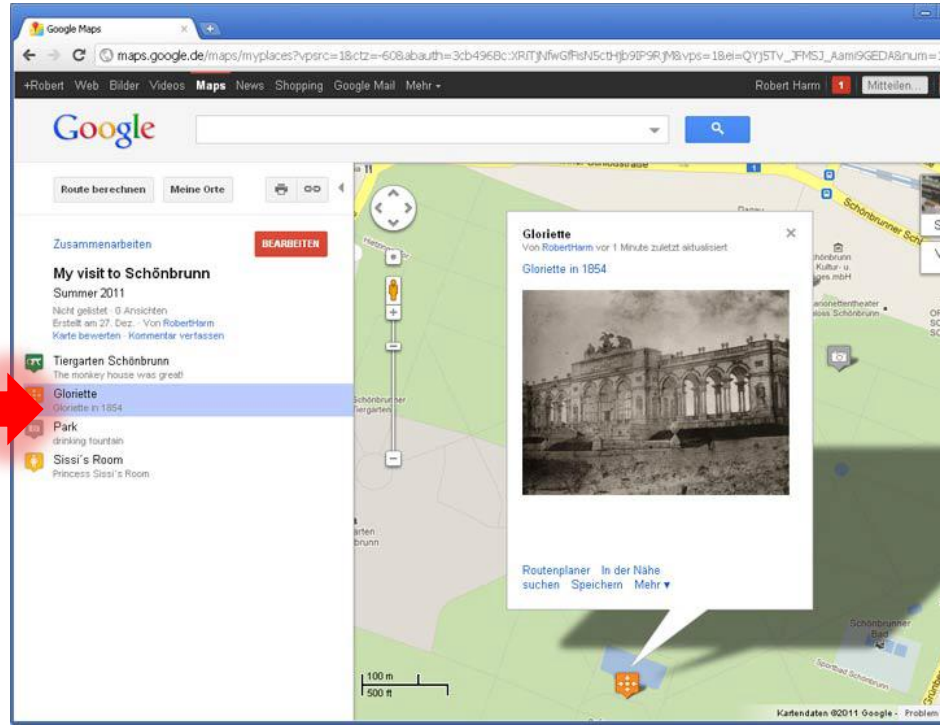
Une activité sans précédent depuis les langages:

FORTRAN, ALGOL, Smalltalk, APL, Lisp, Simula (60/70s)

Quels sont les problèmes adressés par les langages récents ?

Les Langages Recents sont révolutionnaires

Sans JavaScript, internet serais simplement un langage hypertexte



- MapReduce permet une programmation parallèle dans les (**Cloud**)
- CUDA programmes GPUs

Introduction aux Compilateurs

Comment les Langages sont implémentés?

- Deux stratégies majeures:
  1. **Compilateurs.** Traduit des programmes vers un code machine exécutable**. (un prétraitement extensive)**.

  2. **Interpréteurs.** Exécute des programmes sans traduction préliminaire : phases de traduction successives (vers la machine/code intermédiaire) et exécution.

Introduction aux Compilateurs

Histoire des Langages de Haut-Niveau

- 1953 : IBM développe le **701**: la programmation se fait en assembleur.
  - **– Problème: coût logiciel dépasse le coût du matériel!**
- John Backus: *Speedcoding: Un langage Interprété qui s'exécute 10-20 fois lent qu'un programme écrit en* assembleur!

  -Traduit un code haut niveau vers un code assembleur.

  - **– 1954-1957** le projet **FORTRAN I :**
  - **– 1958**.. , > 50% de tous les logiciels sont écrits en **FORTRAN**.
  - **– le premier compilateur** -grand impact sur l'informatique

réduction dramatique du temps de développent (semaines → heurs).

Introduction aux Compilateurs

Motivations et Histoire.

- Architecture d'un Compilateur.
- Phase d'Analyse.
- Phase de Génération.
- Vers un Code Exécutable: Assembleur, Chargeur (Loader) et l'Editeur de lien (Linker).

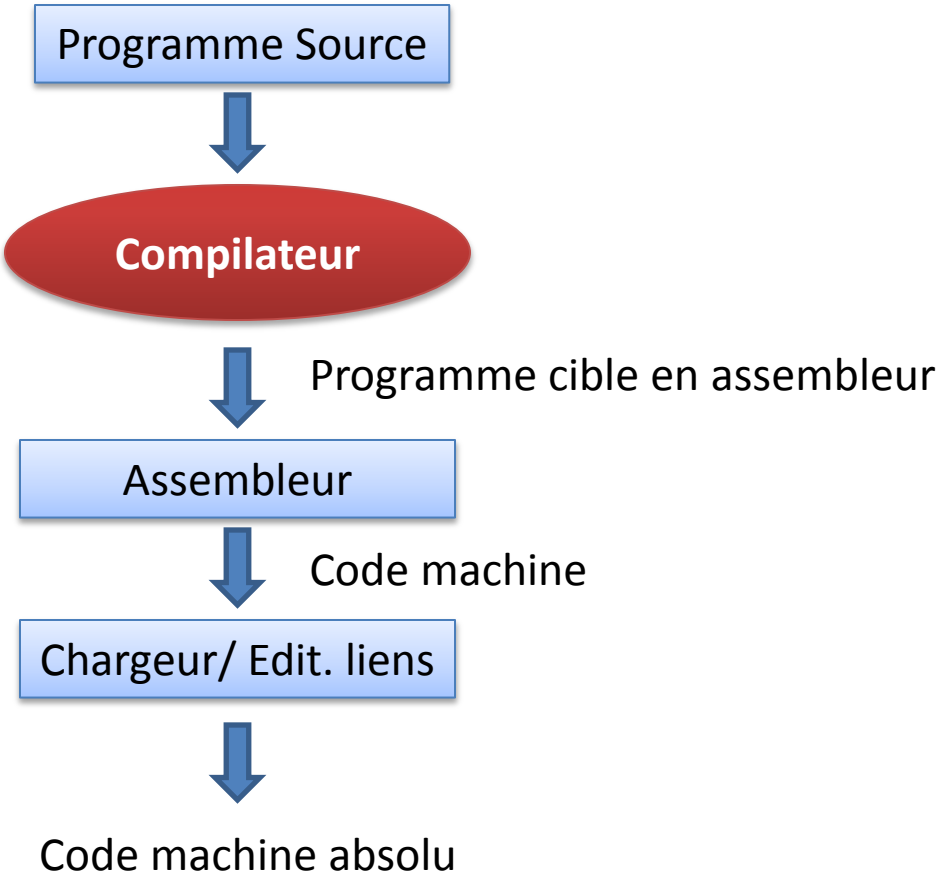
Contexte d'un Compilateur

Un Compilateur est un programme qui lit un *programme écrit dans un langage – Langage source – et le traduit vers un programme équivalent dans un autre* Langage – *langage cible*.

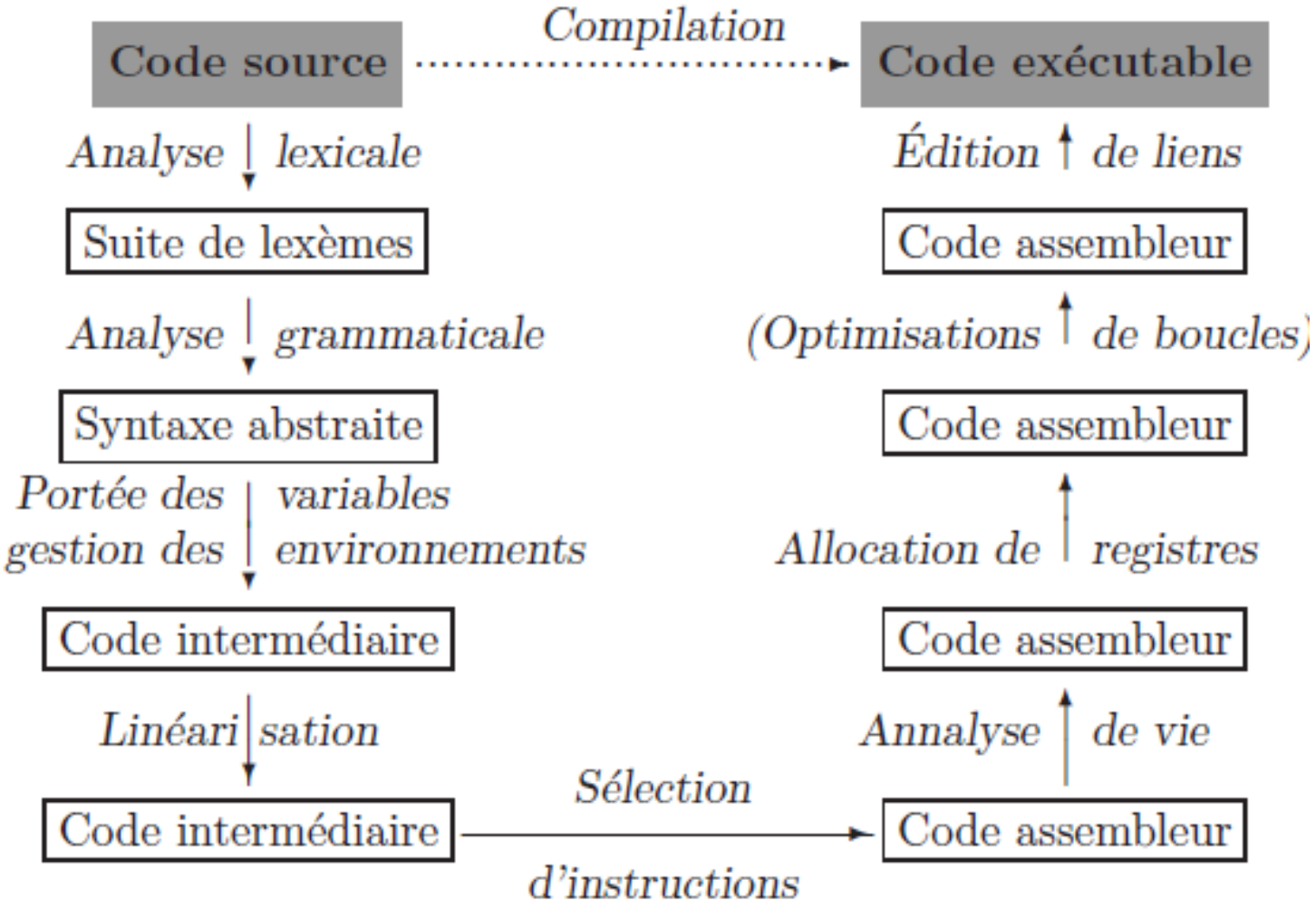
En plus du compilateur, d'autres programmes sont nécessaire pour la génération d'un code *exécutable.*

- gcc : programmes C/C++ vers assembleur/code machine
- f2c : programmes Fortran vers programmes C
- latex2html: documents Latex vers documents HTML
- javac : programmes Java vers byte code JVM
- ps2pdf: fichiers PostScript vers fichiers PDF

Introduction aux Compilateurs



Introduction aux Compilateurs



Architecture d'un Compilateur

La Compilation peut être diviser en deux parties: Analyse et Génération.

1. **Analyse. Décomposer le programme source en pièces et créer une** représentation intermédiaire.
2. **Génération génère un programme cible à partir de la** représentation intermédiaire**.**

La partie analyse comporte les phases suivantes:

1. **Analyse Lexicale;**
2. **Analyse Syntaxique;**
3. **Analyse Sémantique.**

La partie Génération comporte les phases suivantes :

1. **Générateur du Code Intermédiaire;**
2. **Optimiseur de Code;**
3. **Générateur de Code.**