Présentation du Cours

Didier Rémy
Octobre 2000

`http://cristal.inria.fr/~remy/poly/compil/0/`
`http://w3.edu.polytechnique.fr/profs/informatique/Didier.Remy/compil/0/`

Avertissement

La version distribuée des notes de cours est la compilation des transparents du cours de l'année 2000. Elles seront mises à jour et parfois détaillées au fur et à mesure du cours. Ces notes de cours sont également disponibles sur le WEB à l'adresse

http://cristal.inria.fr/r̃emy/poly/compil/

dans leur version la plus récente. Une version HTML portable (pour être installé sur une machine personnelle et lue sans connection réseau) peut aussi être trouvée à cette adresse et installée sur votre machine.

Emploi du temps
Introduction
La chaîne de compilation

Informations utiles

Votre Chef de Travaux pratiques Luc Maranget

Le calendrier

Les lundi

12/01, 16/01, 30/01, 06/02, 13/02, 27/03, 06/03, 13/03, 20/03

- Cours en Amphi Fresnel de 8h30 à 10h
- Travaux dirigés en salle 33 de 10h15 à 12h15

Examens

- écrit le lundi 23/03 de 10h à 12h.
- oraux, entre le 26/03 et le 29/03.

Quelques buts du cours de compilation

- Démystifier et comprendre comment les programmes sont compilés puis exécutés par un ordinateur.
- Apprendre des techniques, des méthodes et l'utilisation d'outils classiques utiles : expressions régulières, analyseurs lexicaux et grammaticaux, interprètes, transformations par découpage en passes successives.
- Réaliser un mini-compilateur :
  - La compilation est un domaine difficile ; L'erreur ne pardonne pas car elle peut se manifester bien plus tard...
  - L'ensemble est compliqué : il faut bien le découper pour le simplifier.
  - \+ Des méthodes et des techniques bien comprises.

Qu'est-ce que la compilation ?

C'est une traduction, en général d'un langage évolué vers un langage moins expressif.

Quelques exemples :

- Compilation des expressions régulières (utilisées dans les outils `sed`, `awk`, l'éditeur `Emacs` et les bibliothèques des langages `C`, `Perl` et `Ocaml` ;
- Minimisation des automates (compilation d'un automate vers un automate plus efficace) ;
- De Tex vers Postscript ou HTML (Hevea) ; *etc.*

La *décompilation* : on va dans le sens inverse, d'un langage moins structuré vers un langage évolué. Ex : retrouver le source C à partir d'un code compilé (d'un programme C). Applications : piratage, récupération de vieux logiciels.

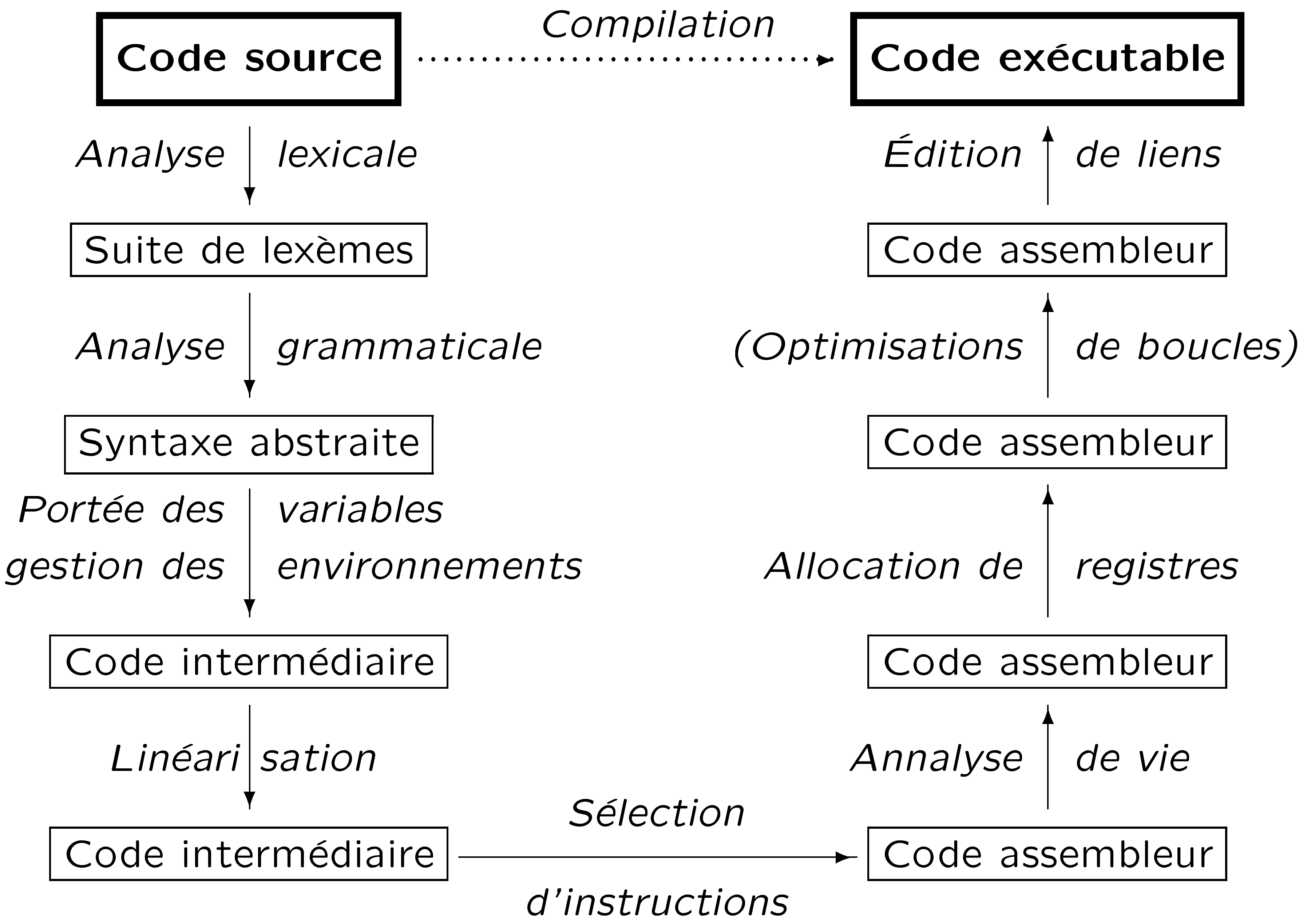
Compilation des langages de programmation

Il s'agit de transformer un fichier (suite de caractères) décrivant un programme en un fichier (suite d'instructions) directement exécutable par une machine.

Principe :

- Découper la compilation en une succession de transformations simples.
- Introduire des représentations intermédiaires à des points importants.

Différence avec la compilation des expressions régulières où l'on peut passer (presque) directement de la chaîne de caractères décrivant l'expression régulière à l'automate.


Code source
Compilation
Code exécutable
Analyse lexicale
Suite de lexèmes
Analyse grammaticale
Syntaxe abstraite
Portée des variables
gestion des environnements
Code intermédiaire
Linéarisation
Code intermédiaire
Sélection
d'instructions
Code assembleur
Annalyse de vie
Code assembleur
Allocation de registres
Code assembleur
(Optimisations de boucles)
Code assembleur
Édition de liens