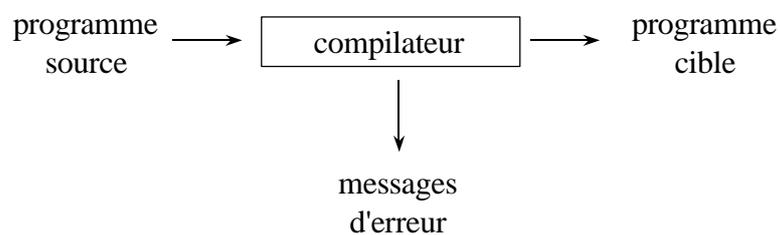


CH.1 INTRODUCTION

- 1.1 Les compilateurs
- 1.2 L'analyse de la source
- 1.3 Les phases d'un compilateur
- 1.4 Les cousins du compilateur
- 1.5 Le regroupement des phases
- 1.6 Les outils logiciels de compilation

1.1 Les compilateurs



• **Langages source :**

Fortran, Pascal, C

Cobol

Langages spécialisés :

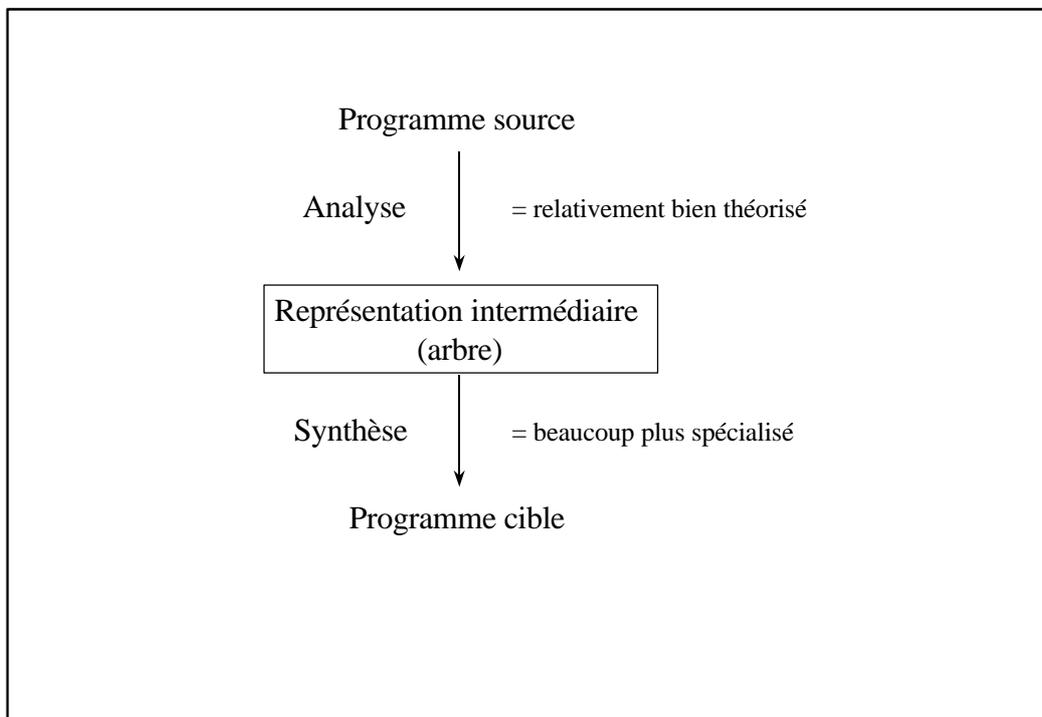
S.Q.L., TeX, Maple

• **Langages cible :**

Langage machine

Code intermédiaire

Autre langage de programmation



• **Autres logiciels d'analyse :**

Éditeurs structurels

Aide à l'écriture d'un programme

Paragrapheurs

Mise en forme du programme source

Contrôleurs statiques

Aide au débogage

Interpréteurs

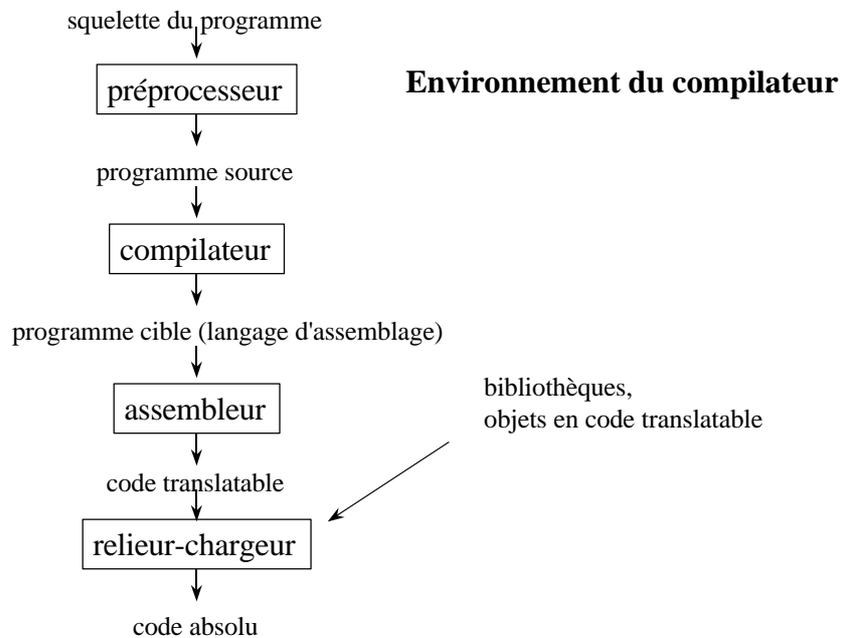
Pour langages spécifiques,

Basic, A.P.L., langages de commandes,

TeX, Maple, interprètes de requêtes

Comme aide à la programmation,

Pascal interprété



1.2 L'analyse du programme source

Découpage en trois phases :

i) Analyse lexicale

flot de caractères regroupés en **unités lexicales**

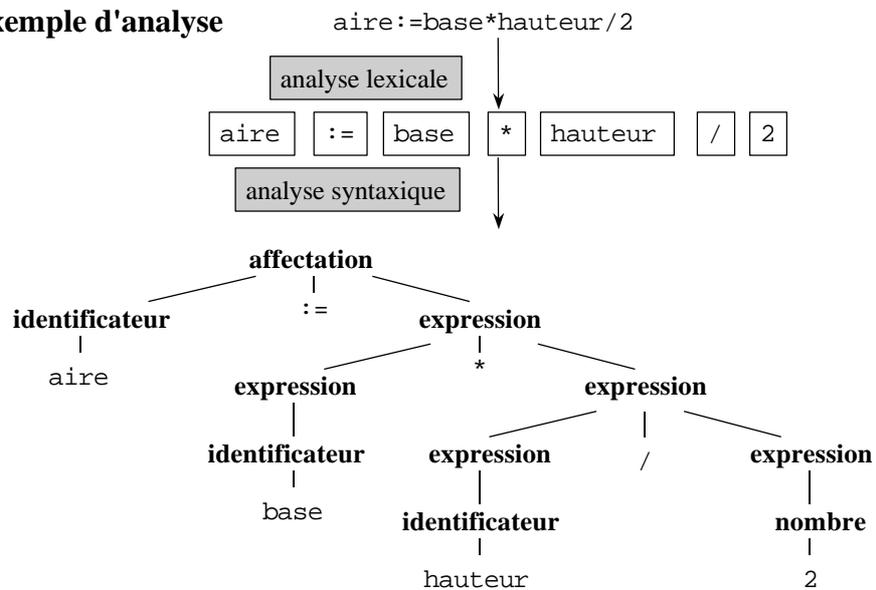
ii) Analyse syntaxique

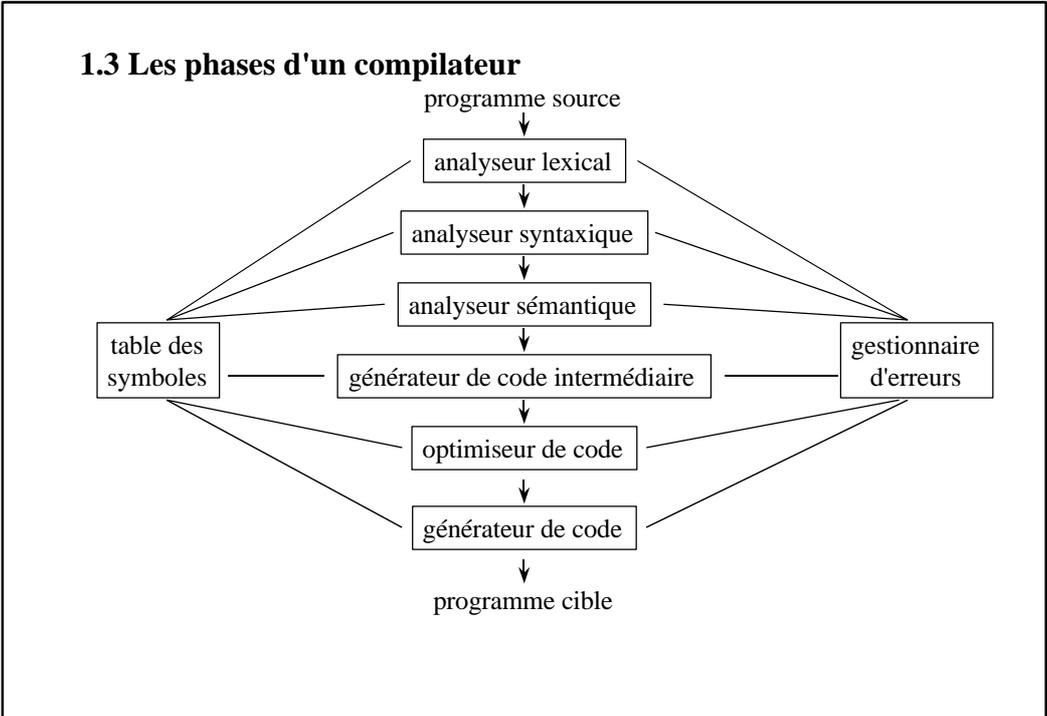
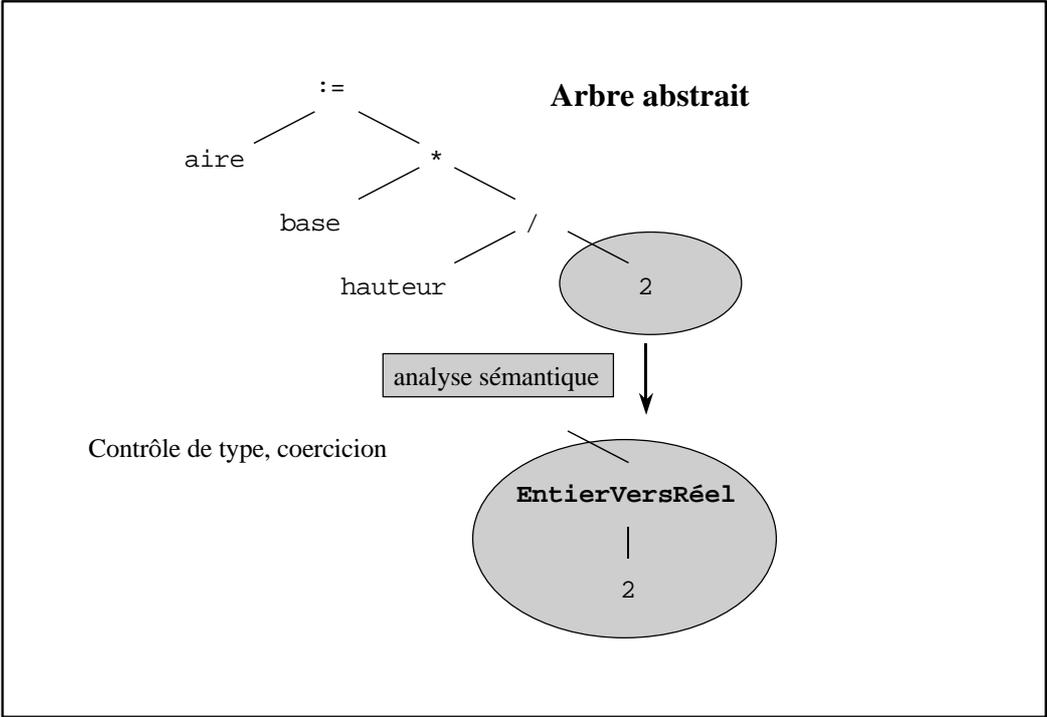
regroupement des unités lexicales en **unités grammaticales**

iii) Analyse sémantique

contrôle ou établissement de la **cohérence sémantique**

Exemple d'analyse





- **Table des symboles :**

Information concernant le type, la portée, l'emplacement-mémoire des identificateurs.

Initialisée par l'analyseur lexical, complétée et utilisée par les autres phases.

- **Gestion des erreurs :**

Erreurs lexicales (caractères interdits).

Erreurs syntaxiques (règles structurelles non respectées).

Erreurs sémantiques (incohérence des opérations : contrôle statique).

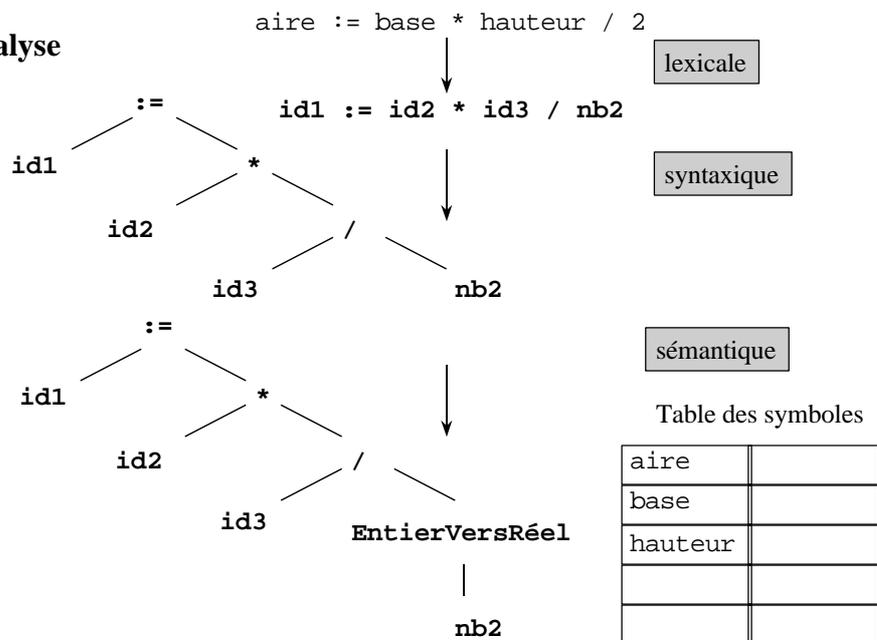
Diverses récupérations possibles, selon la finalité du compilateur :

arrêt à la première erreur ;

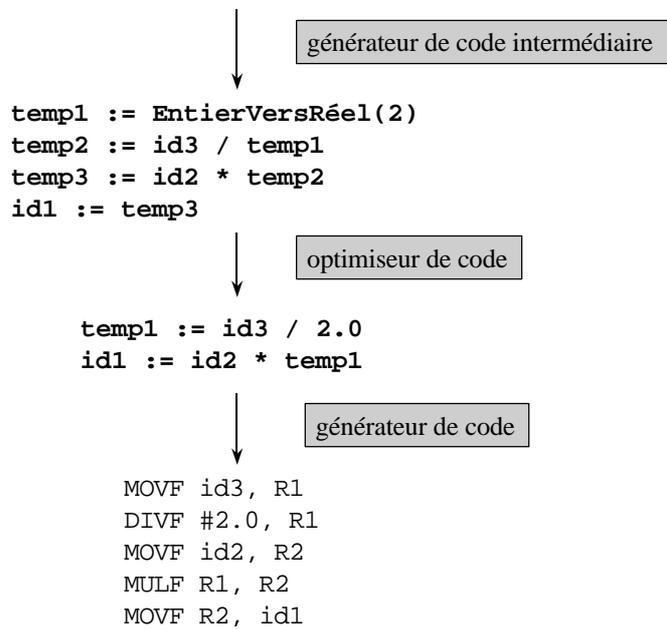
resynchronisation sur la prochaine construction correcte ;

tentatives de correction.

- **Analyse**



• Synthèse



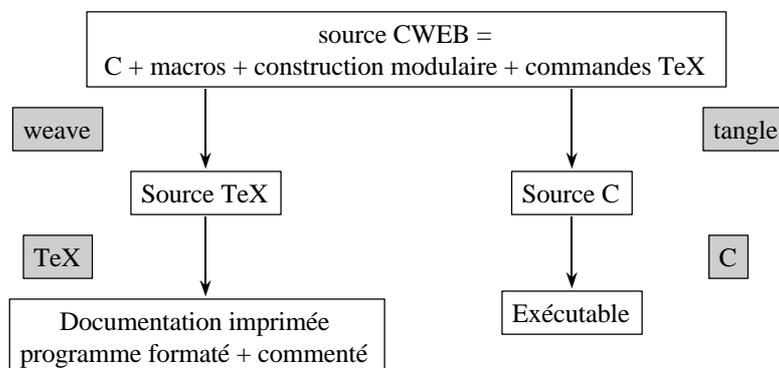
1.4 Les cousins du compilateur

- Les préprocesseurs (extension de langages)

Fournissent du langage source :

macros, inclusion de fichiers, constructions non prévues

Exemple : CWEB de D. Knuth



- Les assembleurs :

Traduire le code d'assemblage produit en
code machine translatable(adresses relatives)

MOVf id3, R1		0001 01 00 00001000 *
DIVf #2.0, R1		0101 01 10 00000010
MOVf id2, R2	→	0001 10 00 00000100 *
MULf R1, R2		0100 10 01 00000001
MOVf R2, id1	→	0010 10 00 00000000 *

0001 charger
0010 ranger
0100 multiplier
0101 diviser

00 mode absolu
01 mode registre
10 mode littéral

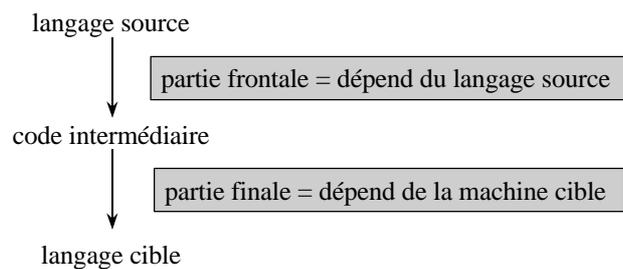
adresse
n° de registre
valeur numérique

- Les chargeurs et relieurs :

gèrent les **références externes** ;
rassemblent les codes translatables en **code absolu** ;
placent ce code à l'emplacement approprié.

1.5 Le regroupement des phases

- Partie frontale et partie finale :



- Passes :

Avantages de la réduction de leur nombre (rapidité, place), plus facile en partie frontale (**traduction dirigée par la syntaxe**).
Impossibilité de déterminer certaines caractéristiques finales :
branchements à des adresses situées plus avant (**reprise arrière**).

1.6 Les outils logiciels de compilation

Outils efficaces d'aide à la réalisation de compilateurs :

- Constructeurs d'analyseurs lexicaux (LEX) ;
- Constructeurs d'analyseurs syntaxiques ;
- Traducteurs dirigés par la syntaxe (YACC) ;
- Générateurs de code automatiques ;
- Moteurs d'analyse de flots de données.