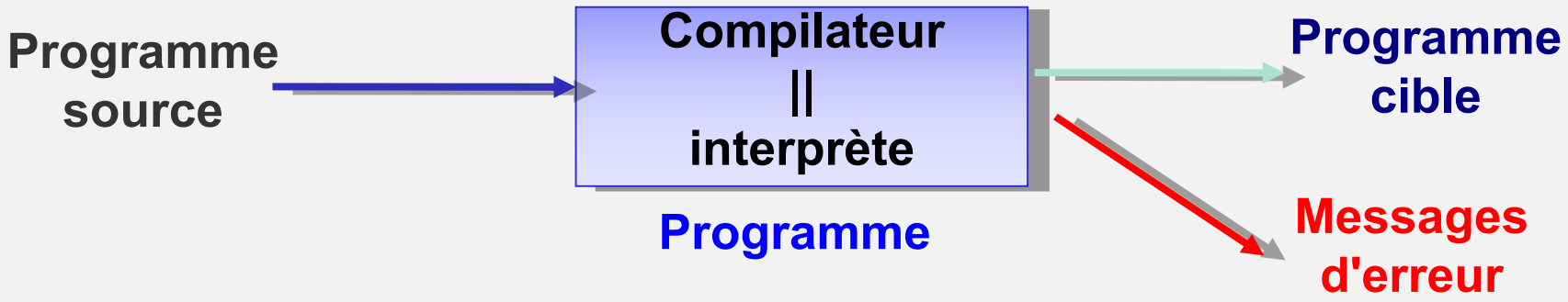
Techniques de compilation

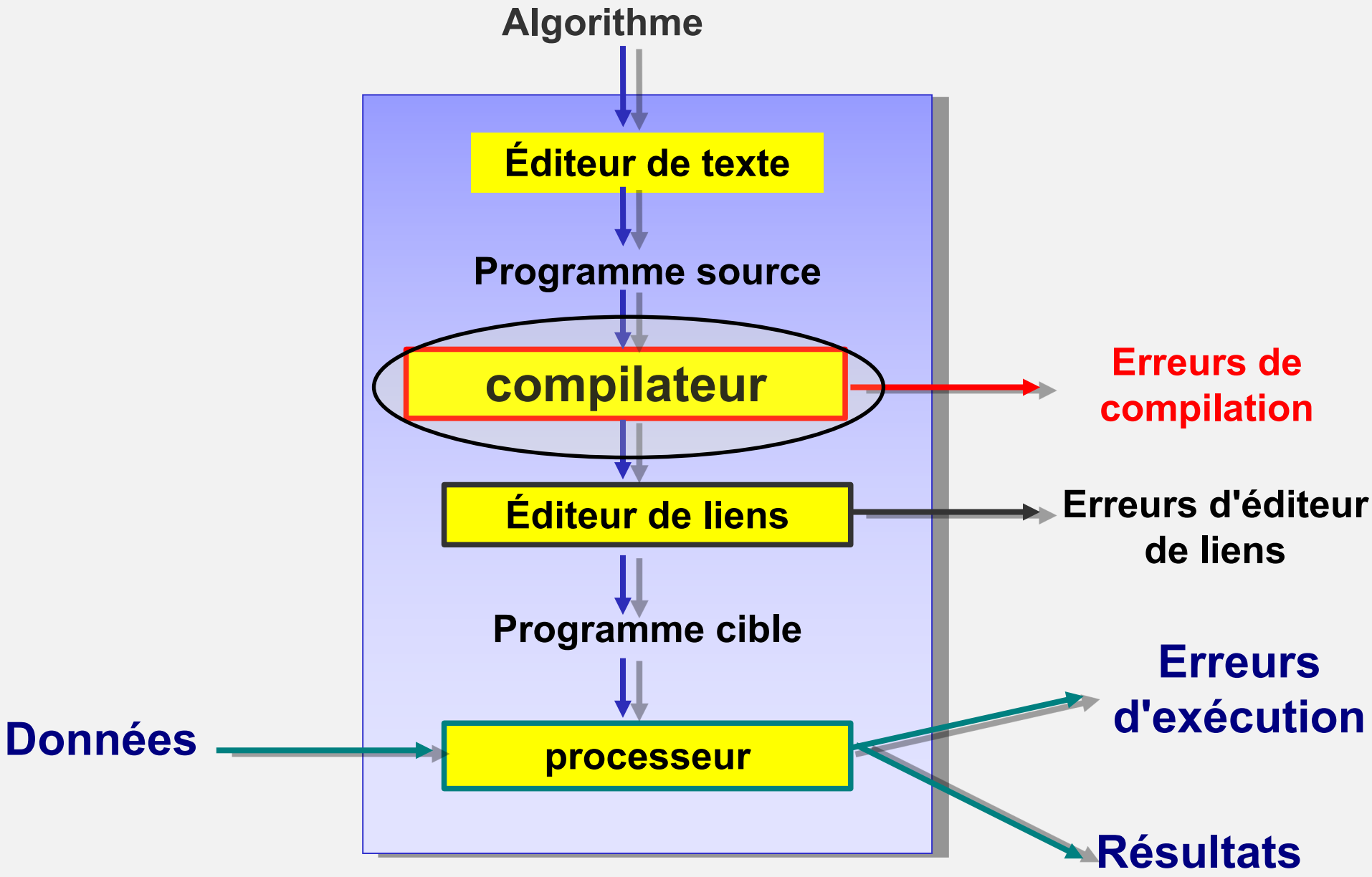


Le compilateur est un programme qui **lit un texte** (programme) écrit dans **un langage source** et le **traduit** en un texte (programme) équivalent **écrit dans un langage cible**

Introduction

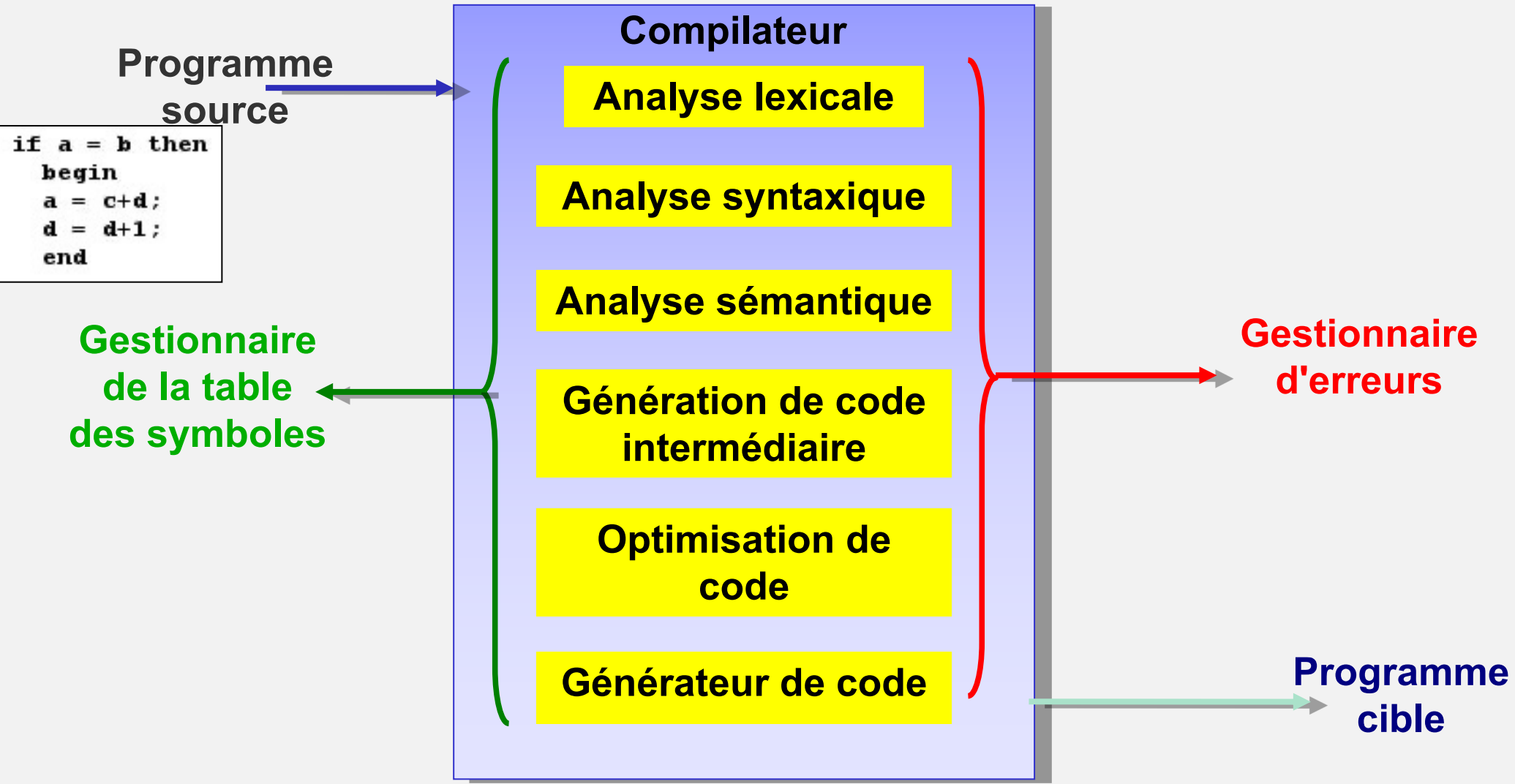
- Qu’est-ce qui est important dans un compilateur ?
  - Le compilateur est rapide (compile time)
  - Les messages d’erreurs sont précis
  - Il supporte un débogueur
  - Le code produit est correct
  - Le code produit est rapide (run time)
  - …

Chaîne de développement d'un programme



Phases de compilation

- Un compilateur est découpé en plusieurs phases
  - ... chacune d'elles transformant le programme source d'une représentation en une autre



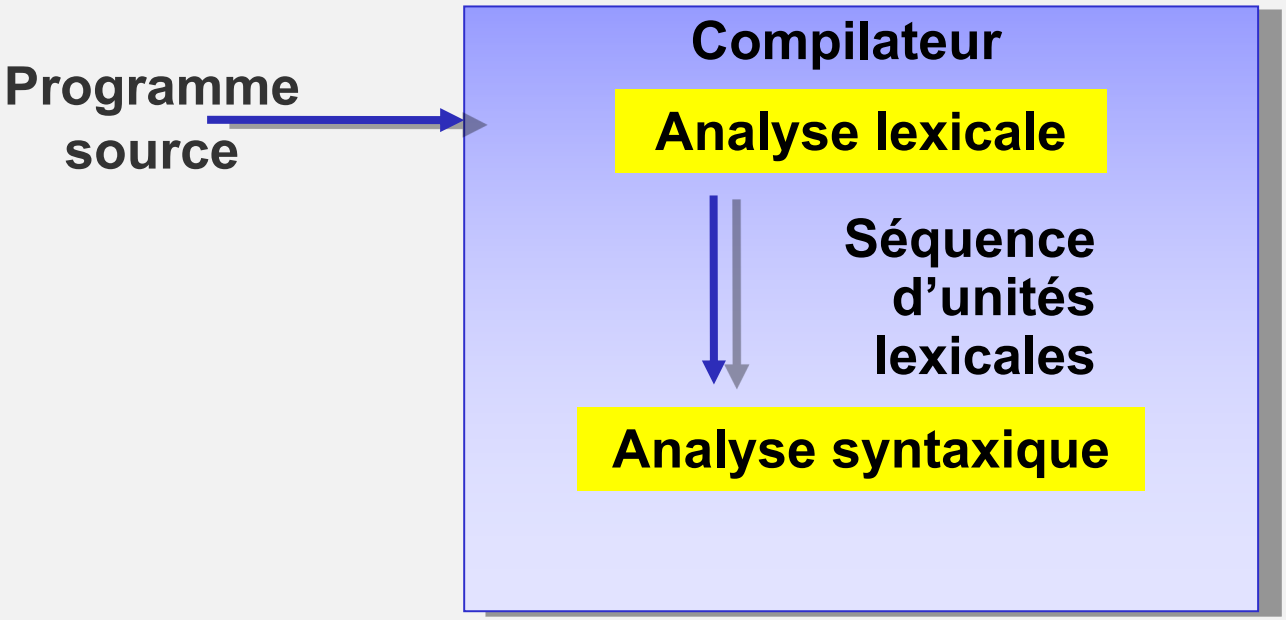
Phases de compilation

Analyse lexicale

- L'analyseur lexicale convertit le flot des caractères d'entrée en un flot d'unités lexicales (mots, tokens, lexèmes)
- Exemple : soit l'instruction d'affectation
  - position := initiale + vitesse * 60
  - Les unités lexicales sont :
    - L'identificateur position
    - Le symbole d'affectation :=
    - L'identificateur initiale
    - Le signe plus
    - L'identificateur vitesse
    - Le signe de multiplication
    - Le nombre 60
- Les commentaires et les blancs sont éliminés au cours de l'analyse lexicale

Phases de compilation

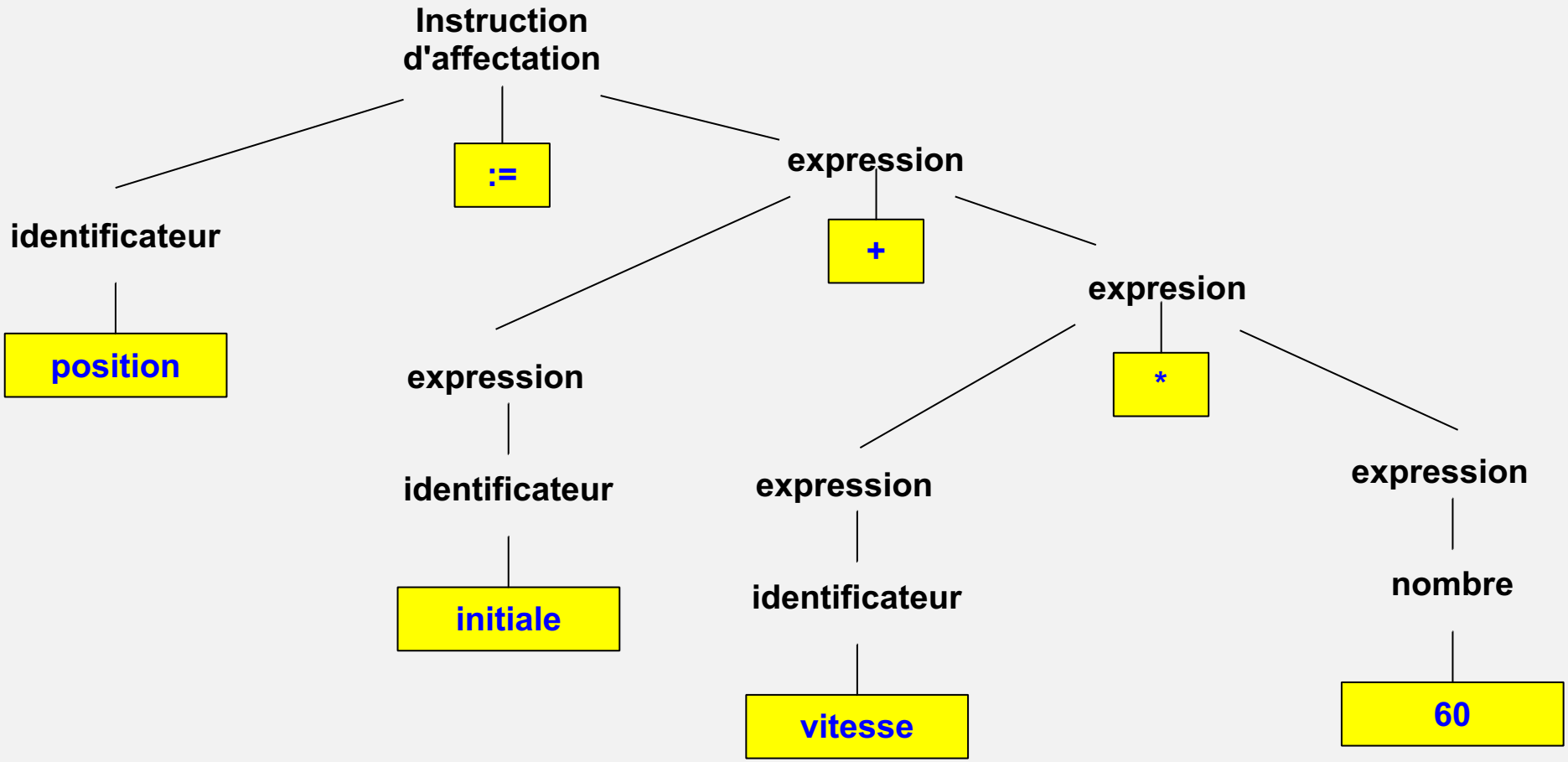
- Un compilateur est découpé en plusieurs phases



Phases de compilation

Analyse syntaxique

- L'analyse syntaxique vérifie que l'ordre des unités lexicales correspond à l'ordre défini pour le langage
  - Vérification de la syntaxe à partir de la définition de la grammaire du langage
  - Ces phases sont représentées par un arbre syntaxique
- **Exemple :** Arbre syntaxique pour position := initiale + vitesse*60

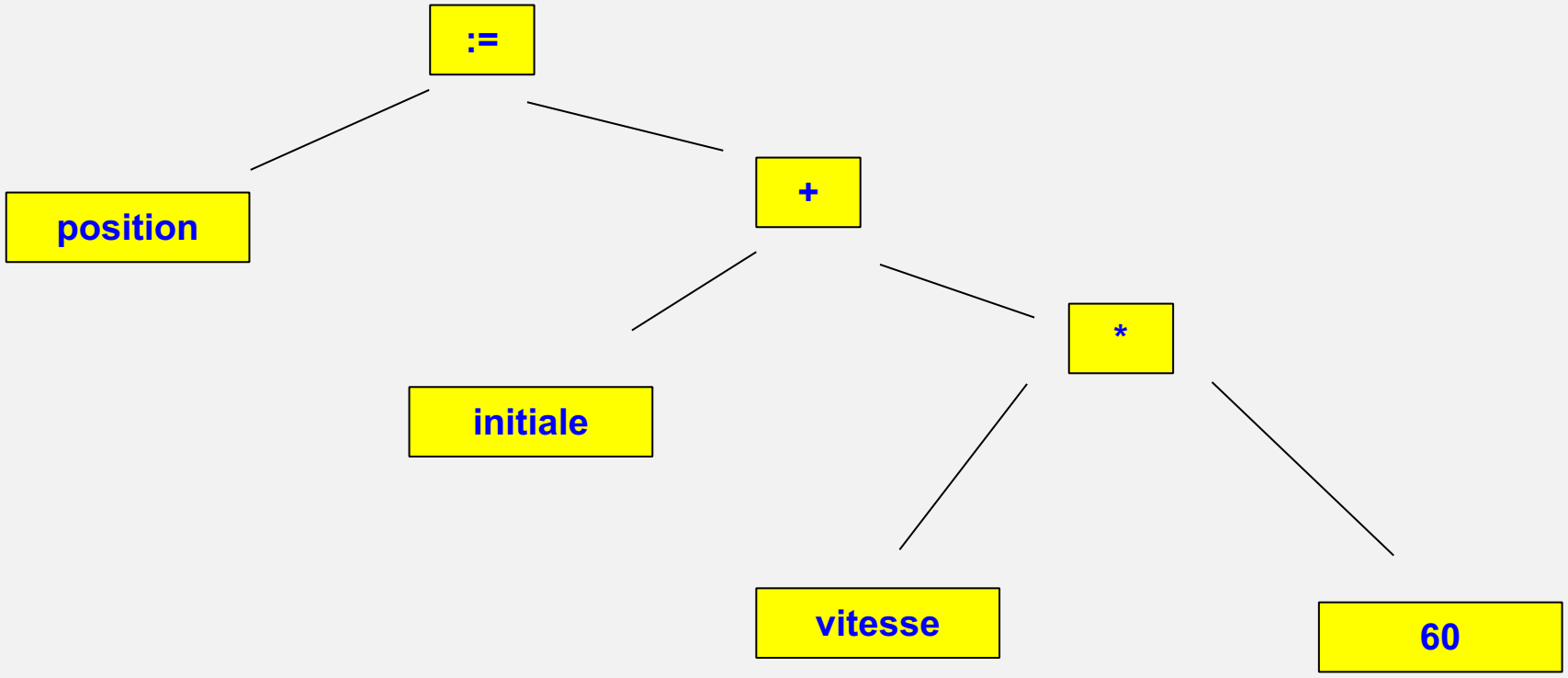


Phases de compilation

Analyse syntaxique

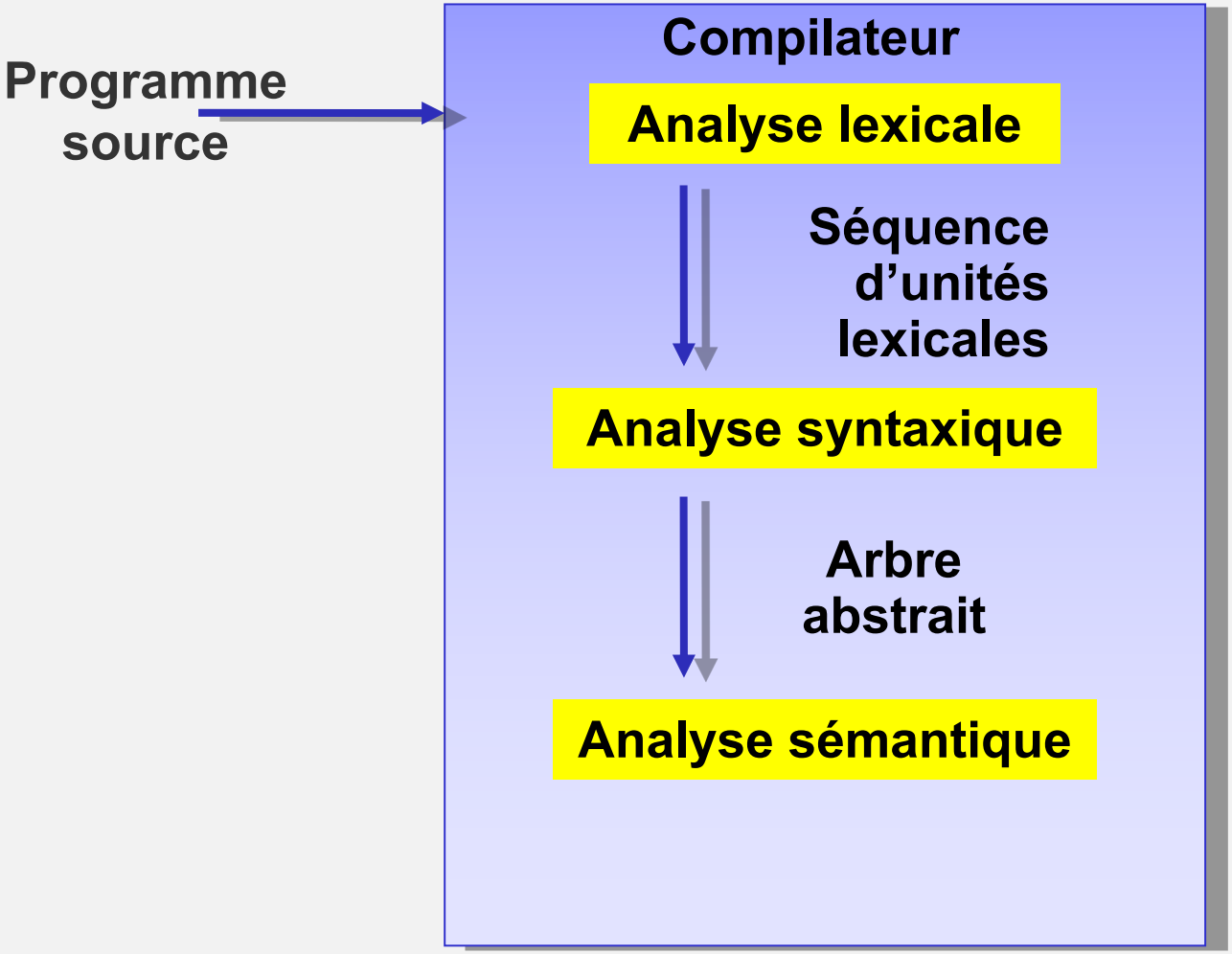
- Arbre abstrait : est une représentation compacte de l'arbre syntaxique dans laquelle :
  - Les opérateurs sont des noeuds internes
  - Les opérandes d'un opérateur sont les fils du noeud correspondant à cet opérateur

Exemple : Arbre abstrait pour position := initiale + vitesse*60



Phases de compilation

- Un compilateur est découpé en plusieurs phases



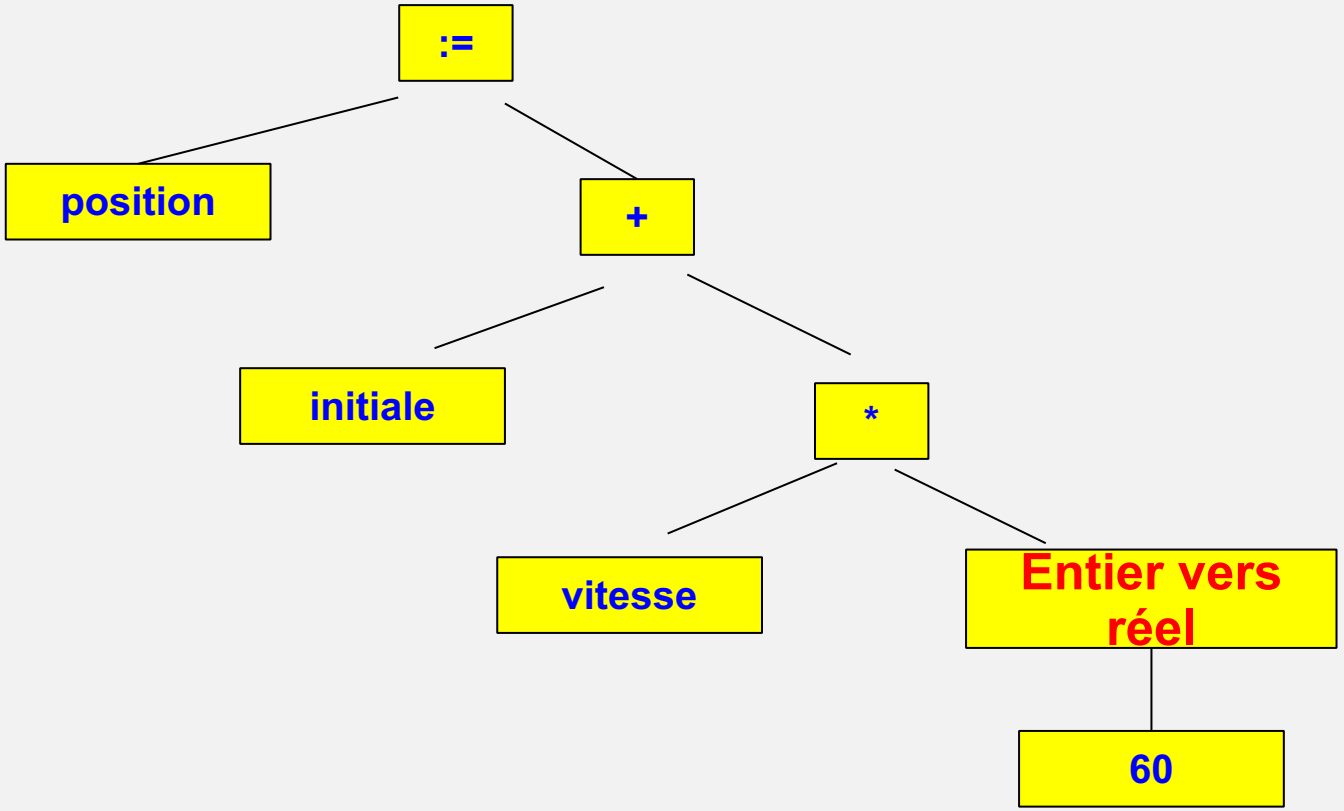
Phases de compilation
Analyse sémantique

- Dans cette phase on vérifie si le programme source contient des erreurs sémantiques
- Le constituant important de l'analyse sémantique est le contrôle de type
  - On vérifie que les variables ont un type correct
  - Cette opération s’effectue en parcourant l’arbre syntaxique et en vérifiant à chaque niveau que les opérations sont correctes

Phases de compilation
Analyse sémantique

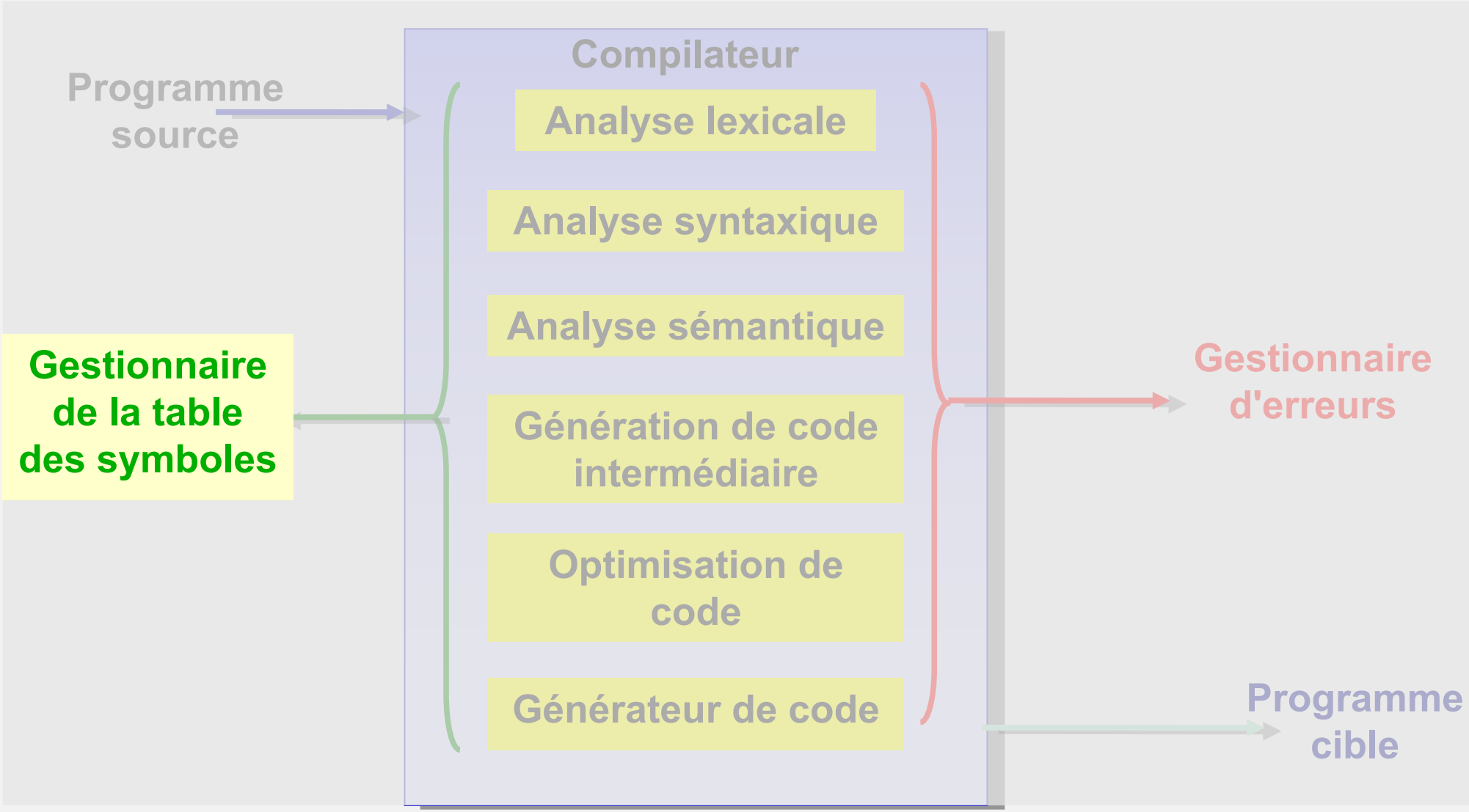
- Exemple :

L'analyse sémantique insère une conversion d'entier en réel



Phases de compilation

- Un compilateur est découpé en plusieurs phases



Phases de compilation

Table des symboles

- Une table des symboles est une structure de données contenant un enregistrement pour chaque identificateur utilisé dans le programme source
  - C'est l'analyseur lexical qui détecte et ajoute les identificateurs dans la table des symboles (s'il n'y est pas déjà) : le type, l'emplacement mémoire, la portée, ...
- Exemple : Table des symboles : position := initiale + vitesse*60

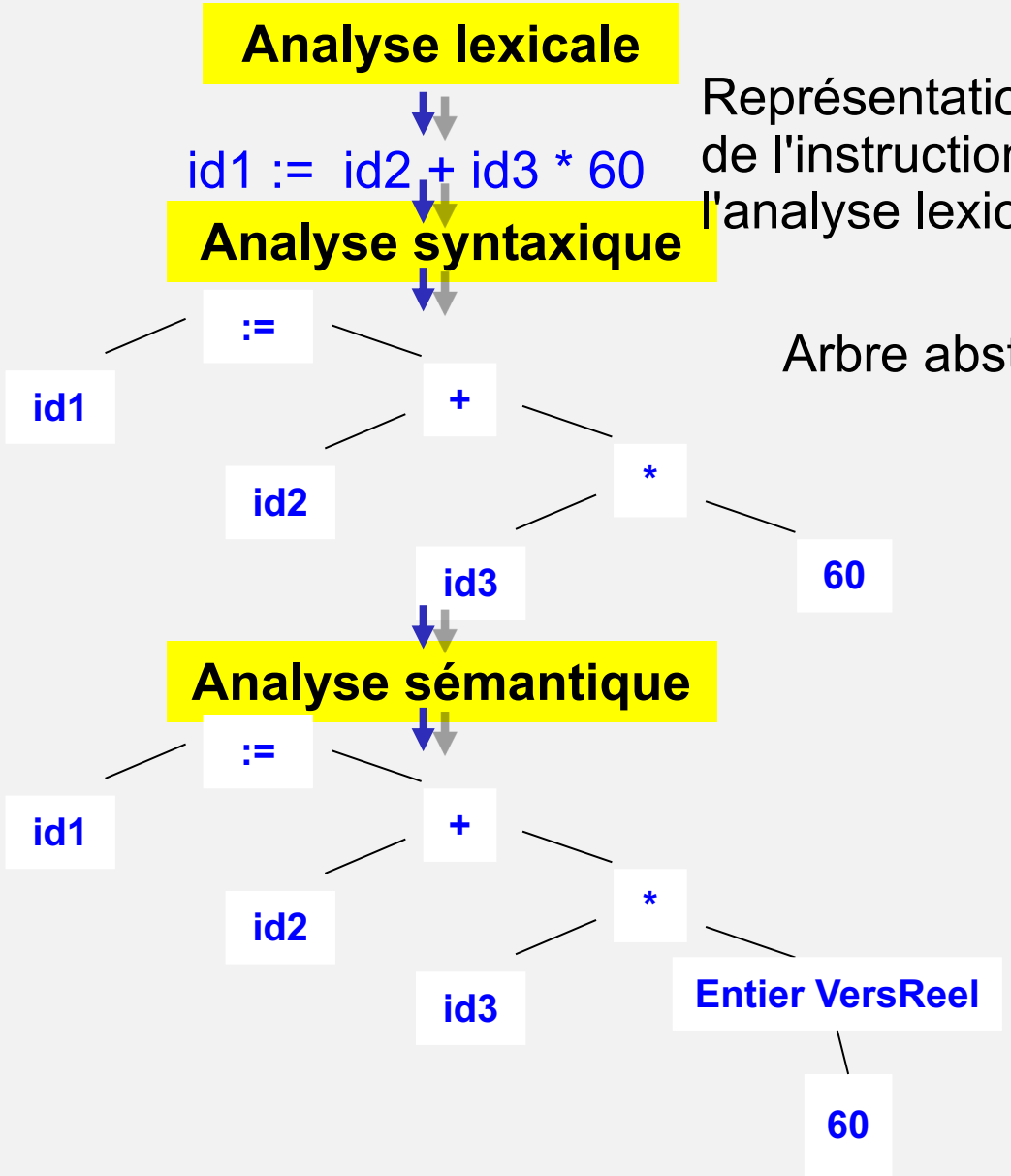
Id	**token**	**...**
1	position	...
2	initiale	...
3	vitesse	...
4	...	...

Phases de compilation

- Traduction de l'instruction : position := initiale + vitesse*60

Table des symbole

Id	token	...
1	position	...
2	initiale	...
3	vitesse	...
4	...	...

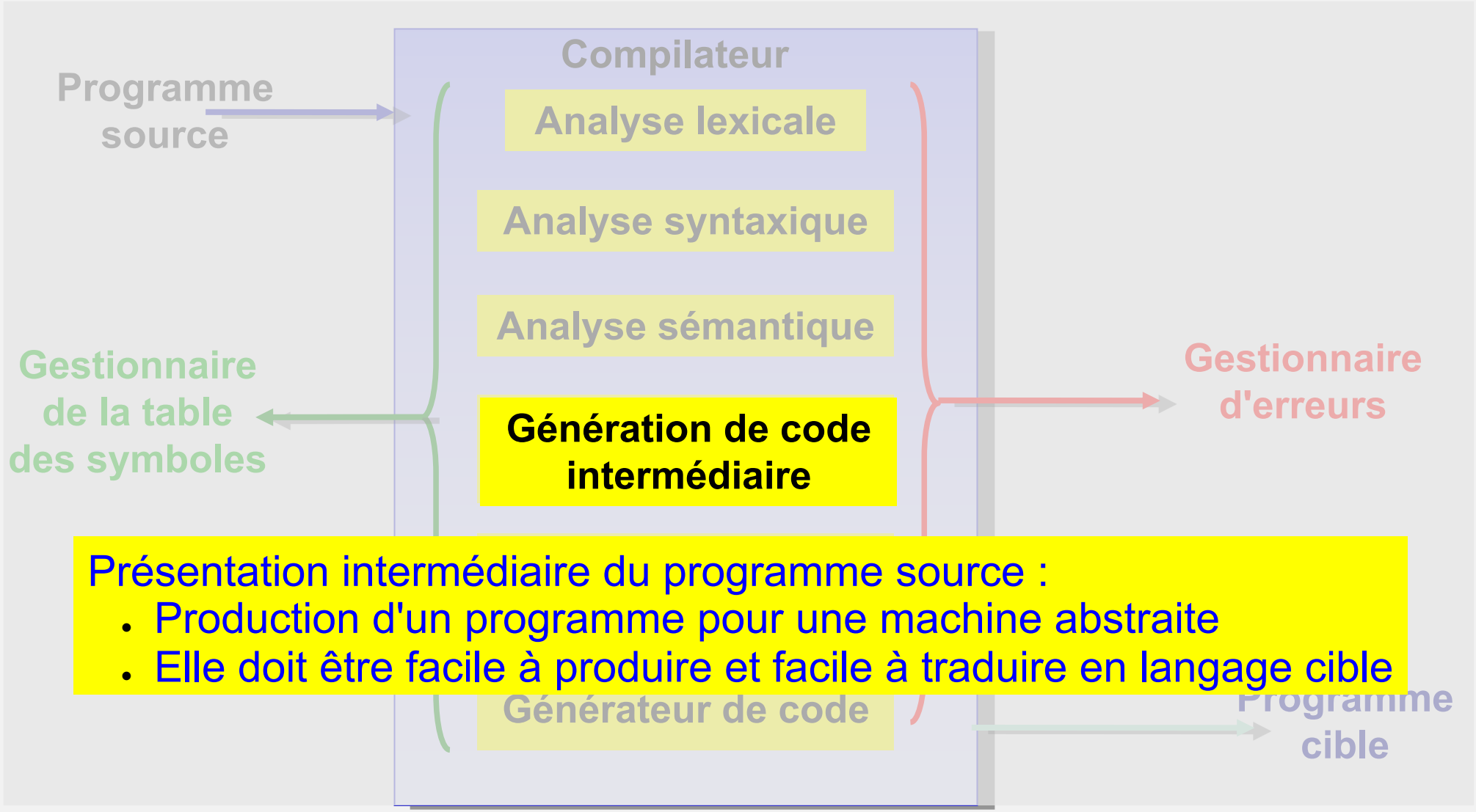


Représentation simplifiée de l'instruction à l'issue de l'analyse lexicale

Arbre abstrait

Phases de compilation

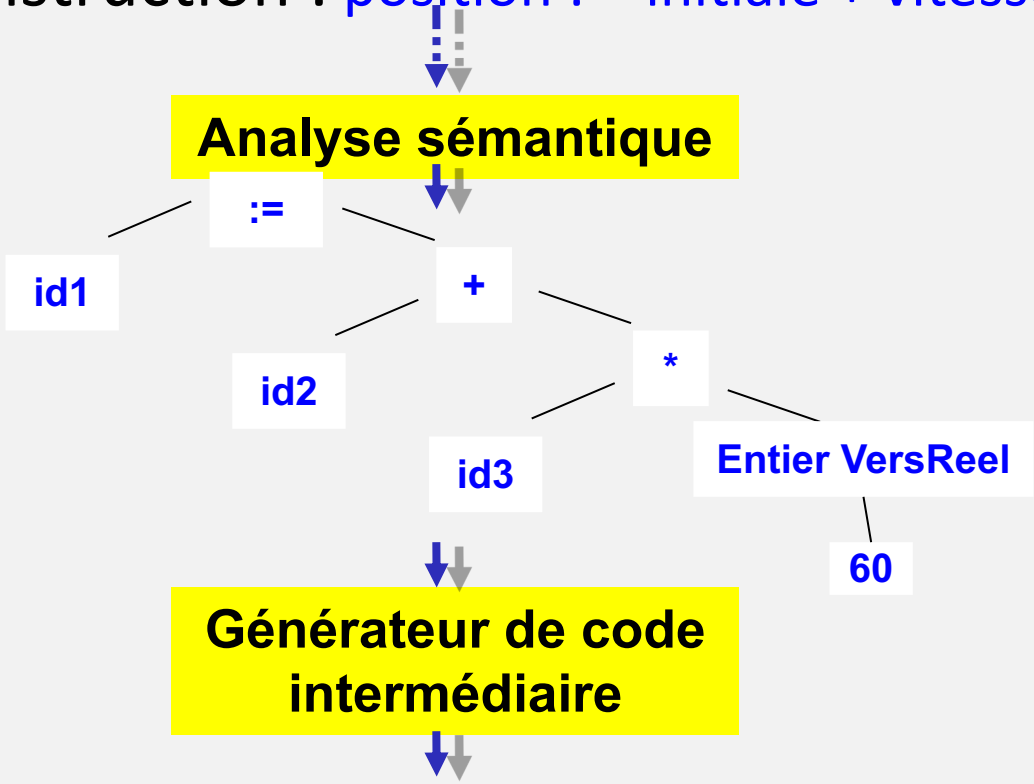
- Un compilateur est découpé en plusieurs phases



Phases de compilation

- Traduction de l'instruction : position := initiale + vitesse*60

Id	token	...
1	position	...
2	initiale	...
3	vitesse	...
4	...	...

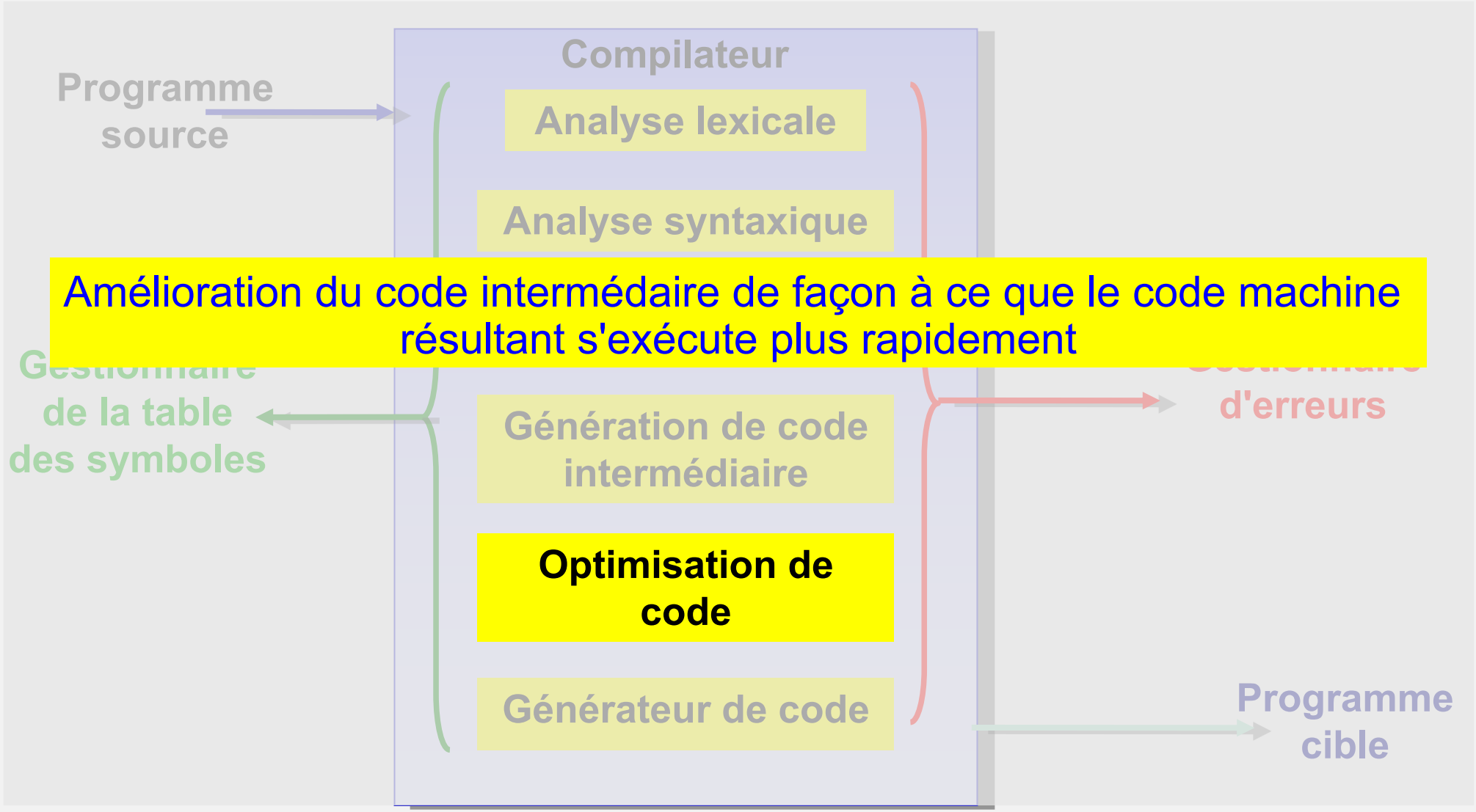


temp1:= EntierVersReel(60)
temp2:= id3 * temp1
temp3:= id2 + temp2
id1:= temp3

Code à trois adresses

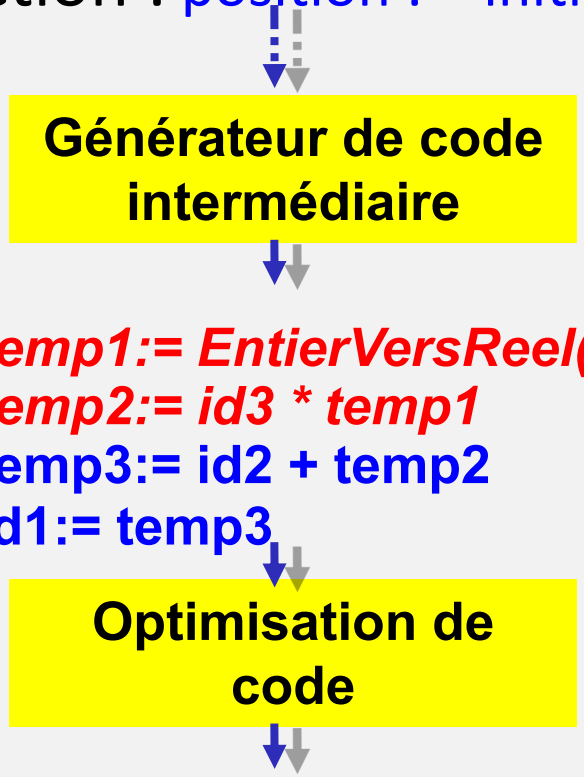
Phases de compilation

- Un compilateur est découpé en plusieurs phases



Phases de compilation

- Traduction de l'instruction : position := initiale + vitesse*60



Id	token	...
1	position	...
2	initiale	...
3	vitesse	...
4	...	...

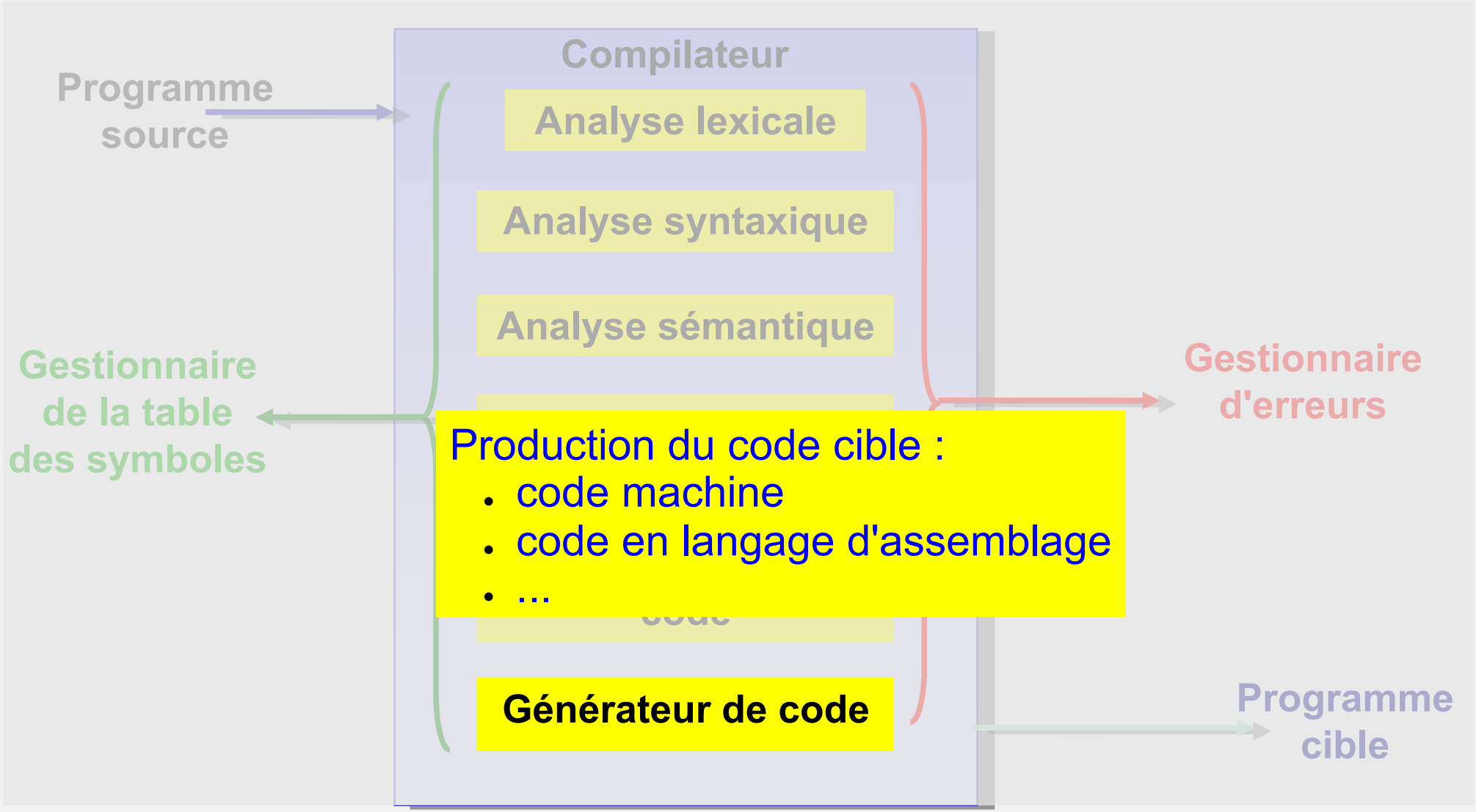
Code à trois adresses

Conversion de 60 en réel

Code à trois adresses

Phases de compilation

- Un compilateur est découpé en plusieurs phases



Phases de compilation

- Traduction de l'instruction : position := initiale + vitesse*60

Optimisation de code

Temp1:= id3 * 60.0 Conversion de 60 en réel

id1:= id2 + temp1

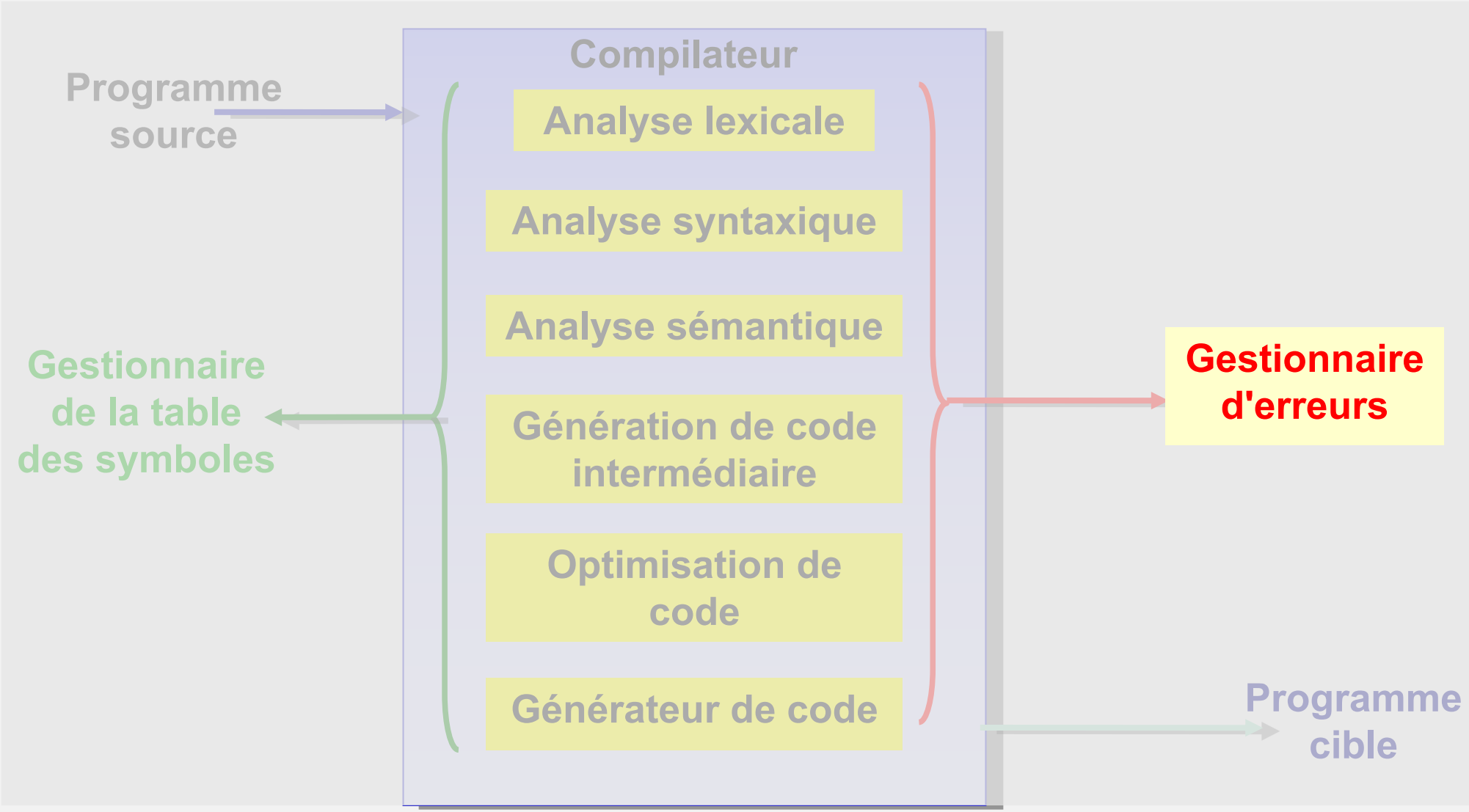
Générateur de code

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

Id	token	...
1	position	...
2	initiale	...
3	vitesse	...
4	...	...

Phases de compilation

- Un compilateur est découpé en plusieurs phases



Phases de compilation

Détection et compte rendu des erreurs

- Chaque phase peut rencontrer des erreurs
- Les phases d'analyse syntaxique et d'analyse sémantique traitent une grande part des erreurs
  - Le compilateur ne doit pas s'arrêter à la première erreur rencontrée
  - La phase lexicale : signale une erreur quand les caractères restant à lire ne peuvent former aucune unité lexicale
    - l'écriture erronée d'un identificateur, mot clé, ...
  - La phase syntaxique : détecte les erreurs dues au fait que la séquence d’unités lexicales n'est pas conforme aux règles structurelles du langage
    - expression arithmétique mal parenthésée
  - La phase sémantique :
    - un opérateur appliqué à un opérande non compatible
      - nom_procedure + id

Phases de compilation
récapitulative

- Un compilateur est découpé en plusieurs phases