



ISEN CIR ³

Théorie des langages

1. Introduction

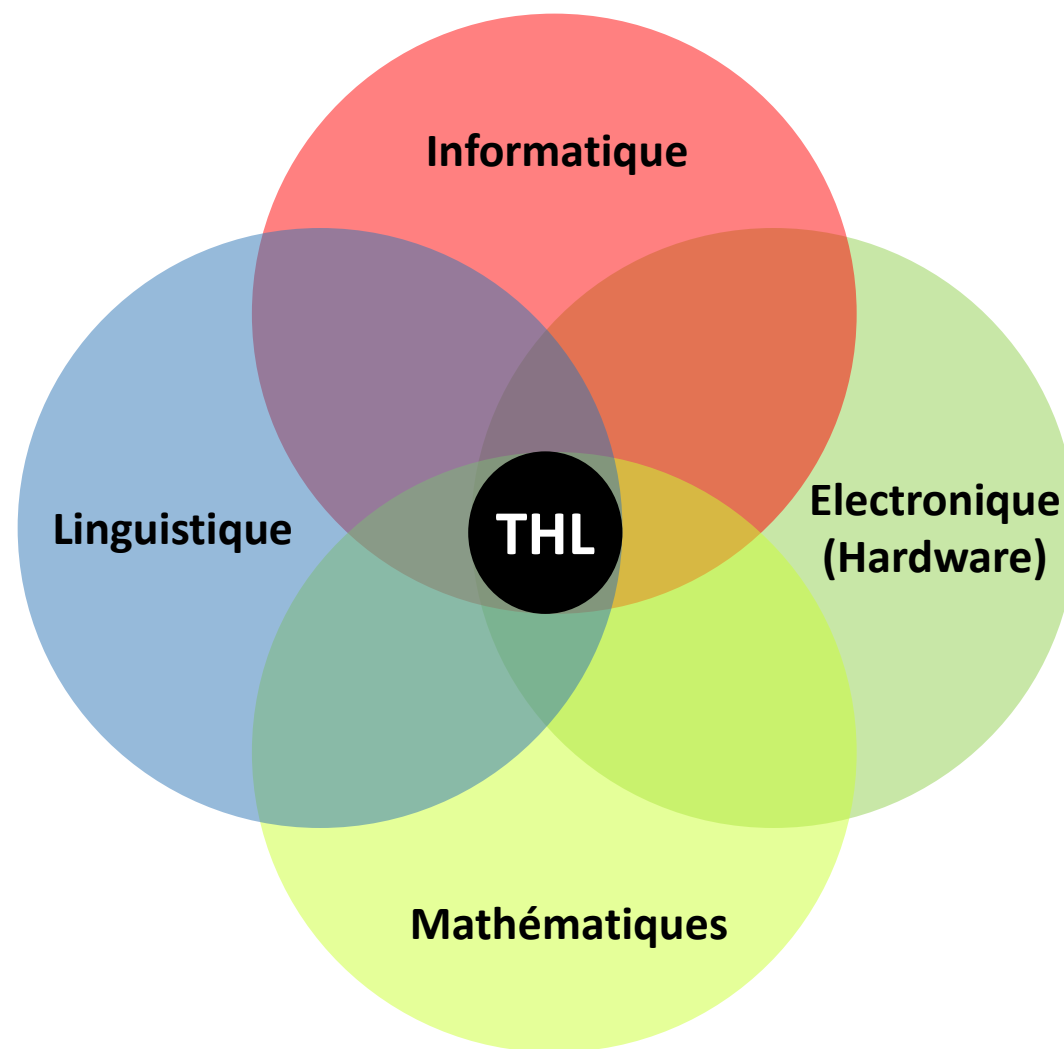
- [illegible]

- 9 séances de cours
- 5 séances de TD/TP
- 5 séances de projet en binôme (Pair Programming!)
- Evaluation:
 - Un partiel (DS)
 - Un examen
 - Un projet (par binômes)

- Algorithmique
- Théorie des graphes
- Logique formelle
- Sciences de l'information et de la communication
- Complexité algorithmique
- Calculabilité et décidabilité
- Théorie des langages
- ...

- Il n'est pas nécessaire de comprendre comment est écrit un compilateur pour savoir comment l'utiliser.
- De même, un ingénieur/informaticien a peu de chances d'être impliqué dans la réalisation ou même la maintenance d'un compilateur pour un langage de programmation majeur.
- Alors pourquoi ce cours ?

- Présenter les principes de base inhérents à la réalisation de compilateurs.
- Comprendre les "contraintes" imposées par les différents langages lorsque l'on écrit un programme.
- Prendre en main de nouveaux langages devient un tâche plus aisée.
- Les idées et techniques développées dans ce domaine sont si générales et fondamentales qu'un informaticien (ou un scientifique) les utilisera très souvent au cours de sa carrière : traitement de données, moteurs de recherche, IA, ...
- Beaucoup de problèmes font référence à la théorie des langages (En science des données, en électronique, en robotique, ...). L'usage des techniques de théories des langages augmente l'efficacité et réduit le temps des solutions.



Applications



Langages de programmation



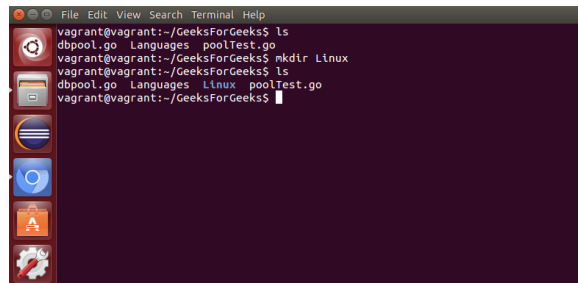
Moteurs de recherche



Protocoles de communication et Web



Reconnaissance et Synthèse vocale



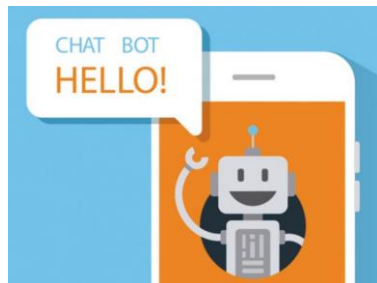
Langages de commandes et interpréteurs



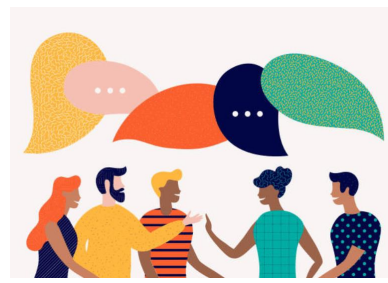
Data Science



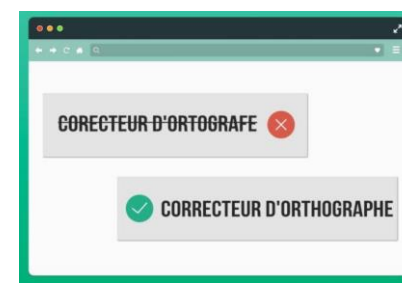
Traduction automatique



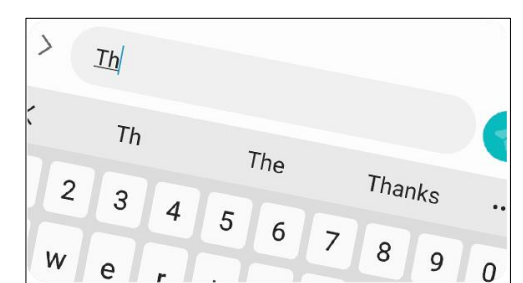
Chatbots



Natural Language Processing (NLP)



Correcteurs de d'orthographe et de grammaire



Aide à la saisie

QU'EST-CE QUE C'EST QUE CETTE CHOSE-LÀ ?

WHAT IS THIS THAT THIS IS THAT THIS THING THERE

Langage de programmation

- Déterministe : Un mot à souvent un seul sens.
- Interprétation unique de la phrase.
- Nombre de mots fini.
- Il est « facile » d'écrire un compilateur de langage.

Langage naturel

- Ambigu : Un mot peut avoir plusieurs sens.
- Figures de style, proverbes, ...
- Nombre de mots variable et évolutif.
- Il est très compliqué d'écrire un interpréteur de langage naturel.

Historique et évolution des langages de programmation





- Les langages machine
 - ↳ Assembleur
 - ↳ Les langages de haut niveau
 - ↳ Dédiés, centrés donnée, naturels, ... (Buzzwords)
 - ↳ Programmation par contraintes, IA, ... (Buzzwords)
 - ↳ Codeless / No code, Low-code

https://en.wikipedia.org/wiki/Timeline_of_programming_languages

HISTORY OF PROGRAMMING LANGUAGES

1949	ASSEMBLY LANGUAGE			OBJECTIVE C	1983
1957	FORTRAN			PYTHON	1991
1959	COBOL			RUBY	1993
1964	BASIC			JAVA	1995
1970	PASCAL			PHP	1995
1972	C			JAVASCRIPT	1995
1972	SQL			C#	2000
1983	C++			SWIFT	2014

Et aujourd'hui ??? (2021)

Rank	Language	Type	Score
1	Python	  	100.0
2	Java	  	95.3
3	C	  	94.6
4	C++	  	87.0
5	JavaScript		79.5
6	R		78.6
7	Arduino		73.2
8	Go	 	73.1
9	Swift	 	70.5
10	Matlab		68.4
11	Ruby	 	66.8
12	Dart	 	65.6
13	SQL		64.6
14	PHP		63.8
15	Assembly		63.7
16	Scala	  	63.5
17	HTML		61.4
18	Kotlin	 	57.8
19	Julia		56.0
20	Rust	  	55.6



Make it be the size of the `rocketship` times 0.75



```
text.style.left =
rocketship.offsetLeft + 'px';
text.style.top =
rocketship.offsetTop + 'px';

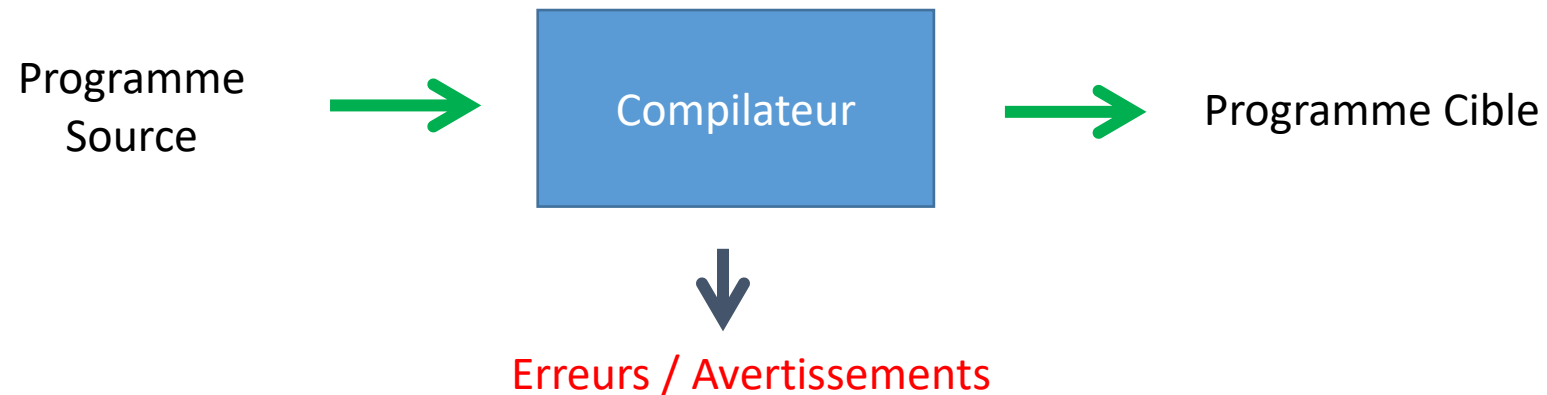
document.body.appendChild(text
);
xSpeed = 20;
setTimeout(function() {
  xSpeed = 5;

document.body.removeChild(text
);
}, 250);
};
```

```
/* Now add an image of an
asteroid:
https://d.newsweek.com/en/full
/1721338/asteroid.jpg?
w=1600&h=1600&q=88&f=9d82d35c9
de96a82b3fcdf7705eb325b */
var asteroid =
document.createElement('img');
asteroid.src =
'https://d.newsweek.com/en/ful
l/1721338/asteroid.jpg?
w=1600&h=1600&q=88&f=9d82d35c9
de96a82b3fcdf7705eb325b';
document.body.appendChild(aste
roid);
```

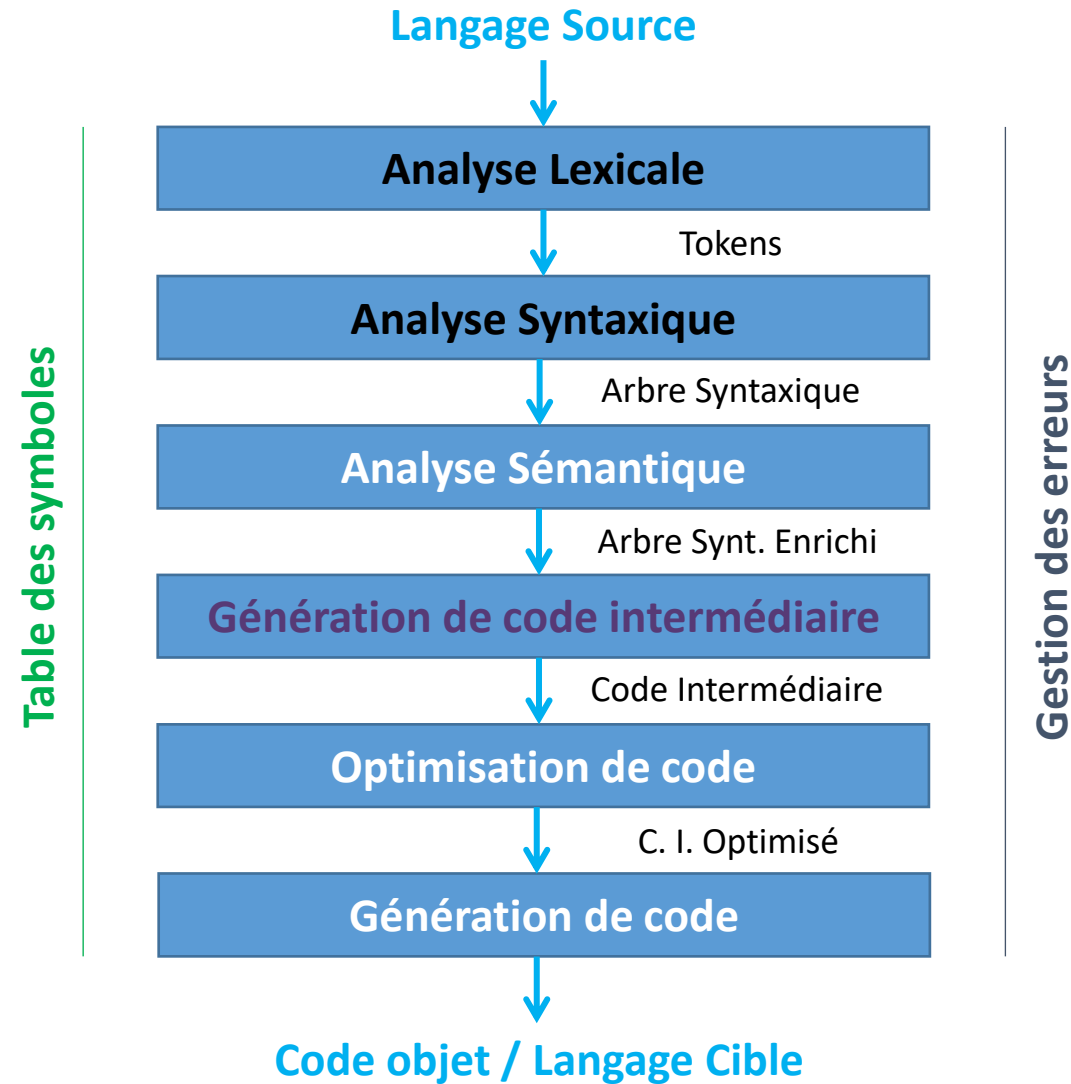

- Un **langage formel** est un ensemble de mots sur un alphabet fini.
- Exemples:
 - Les nombres entiers, réels, ...
 - Le codage binaire, hexadécimal, ...
 - Les langages de programmation, les langages machines, ...
- Peut-on décider par algorithme **si un mot donné appartient à ce langage ?**
 - Peut-on décider si une phrase est correcte en français ?
 - Si un email, un numéro de téléphone ou un matricule sont corrects ?
 - Peut-on vérifier si un code source est correct selon le langage de programmation ?
- Si oui, quelle est la complexité algorithmique d'une telle réponse ?

- Transformer à l'aide d'un programme informatique, un code source écrit dans un langage de programmation en un autre langage informatique (le langage cible).

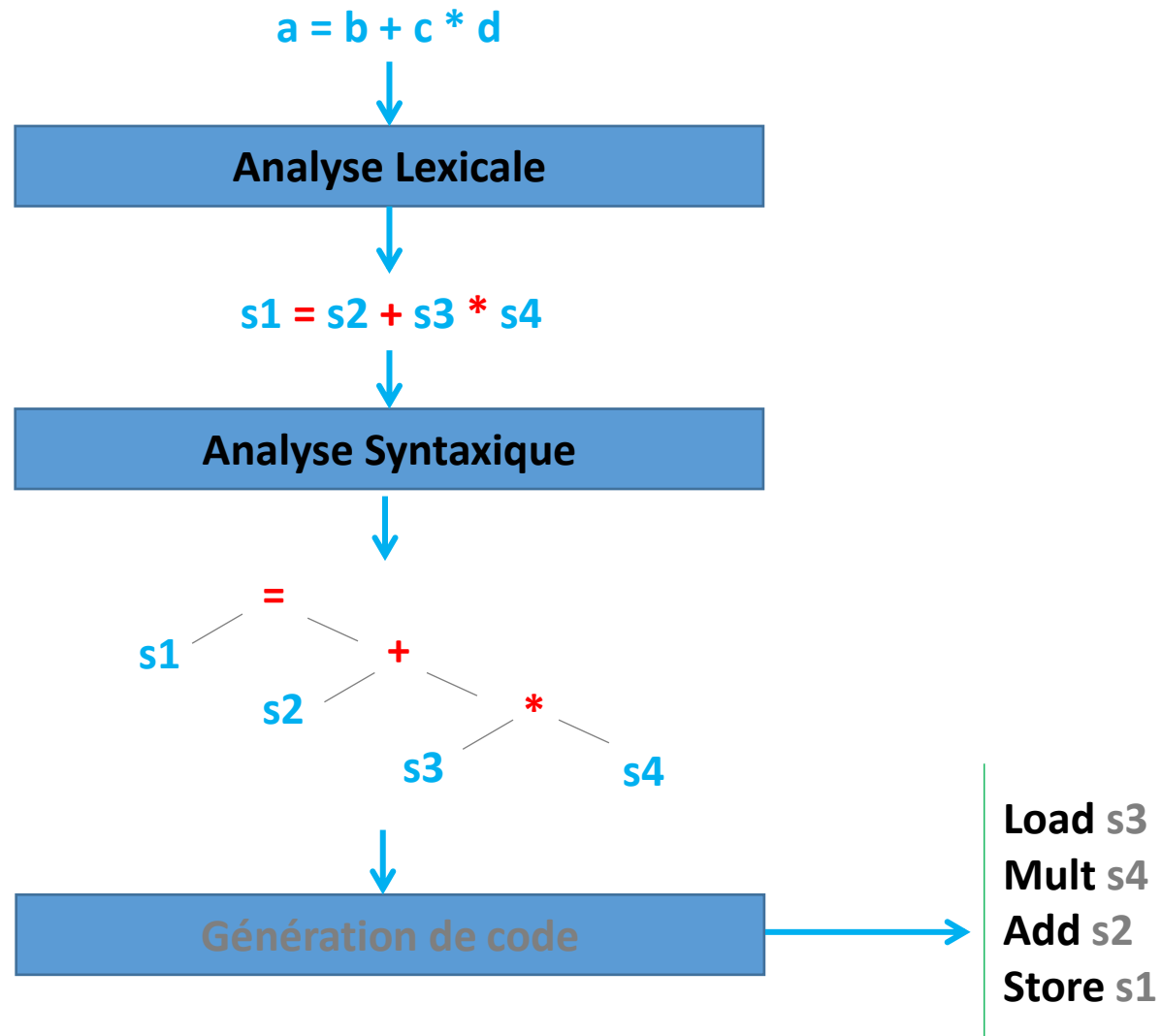


- Compilation anticipée (Ahead-of-Time)
Le binaire est produit une seule fois
 - Interprétation
Le code source est analysé à chaque exécution
 - Compilation à la volée (Just-in-Time)
Un code intermédiaire est produit
 - Compilation source-à-source
D'un langage vers un autre langage
 - Recompilation
Exécuter un code machine sur une autre machine
- C, C++, ...
 - PHP, Perl, Python, lignes de commandes, ...
 - JAVA, .NET, ...
 - Transcompilateurs
 - Les machines virtuelles, les émulateurs.

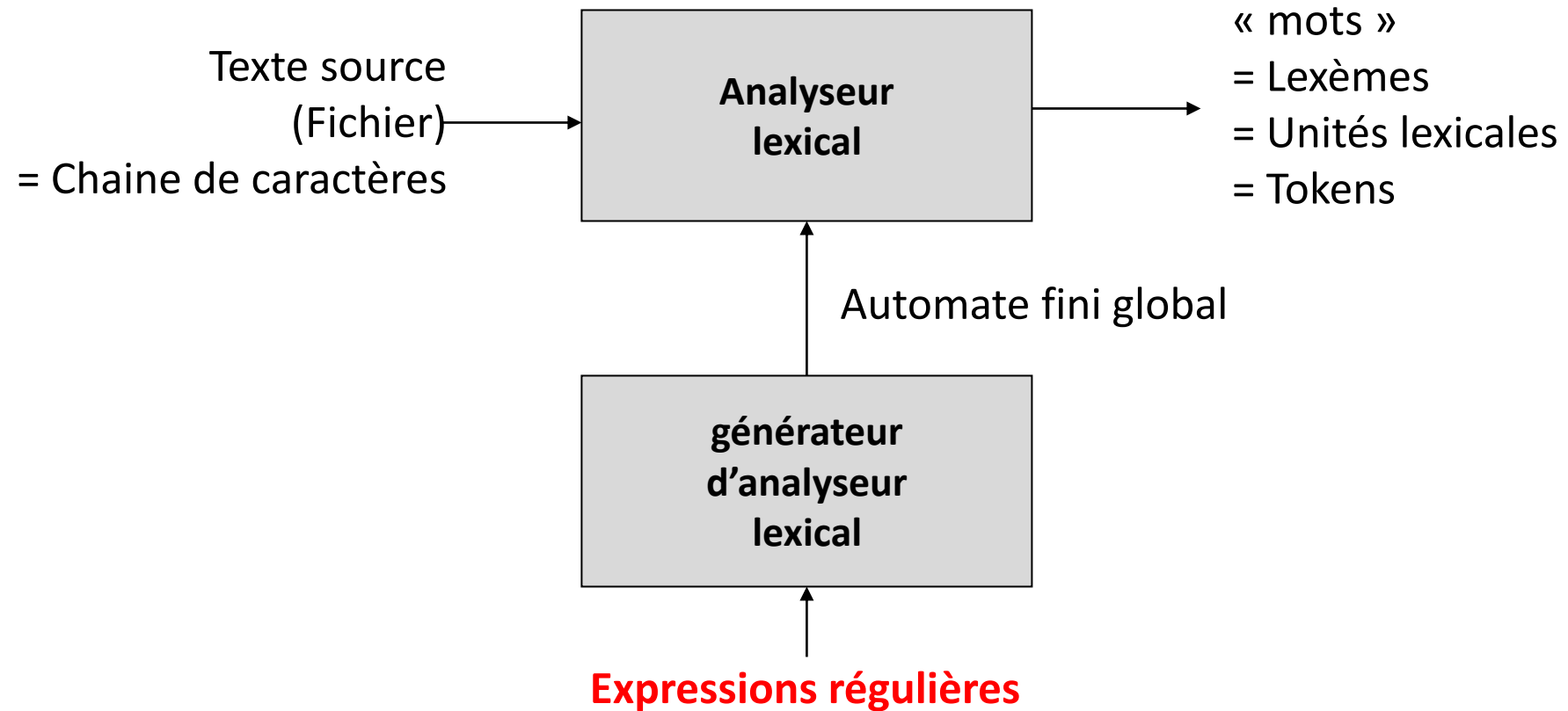
- | | |
|-------------------------------------|----------------------------------------|
| 1. Analyse lexicale | Reconnaître les mots et les symboles |
| 2. Analyse syntaxique | reconnaître les « phrases » |
| 3. Analyse sémantique | vérifier les types, etc. |
| 4. Optimisation | élimination de code inutile, etc. |
| 5. Allocation mémoire | choisir la place des variables |
| 6. Génération de code | instructions machine |
| 7. Optimisations propres à la cible | séquences d'instruction plus efficaces |



s1	a	
s2	b	
s3	c	
s4	d	



[Exemples](#)



- Un projet à réaliser par binômes.
- Sur 5 séances
- Les détails du projet seront partagées très bientôt dans un document à part.

- Alphabets et langages
- Grammaires formelles
- Automates d'états fini
- Expression régulières
- Analyse lexicale
- Analyse syntaxique
- Outils de génération d'analyseurs
- Génération de code
- ...