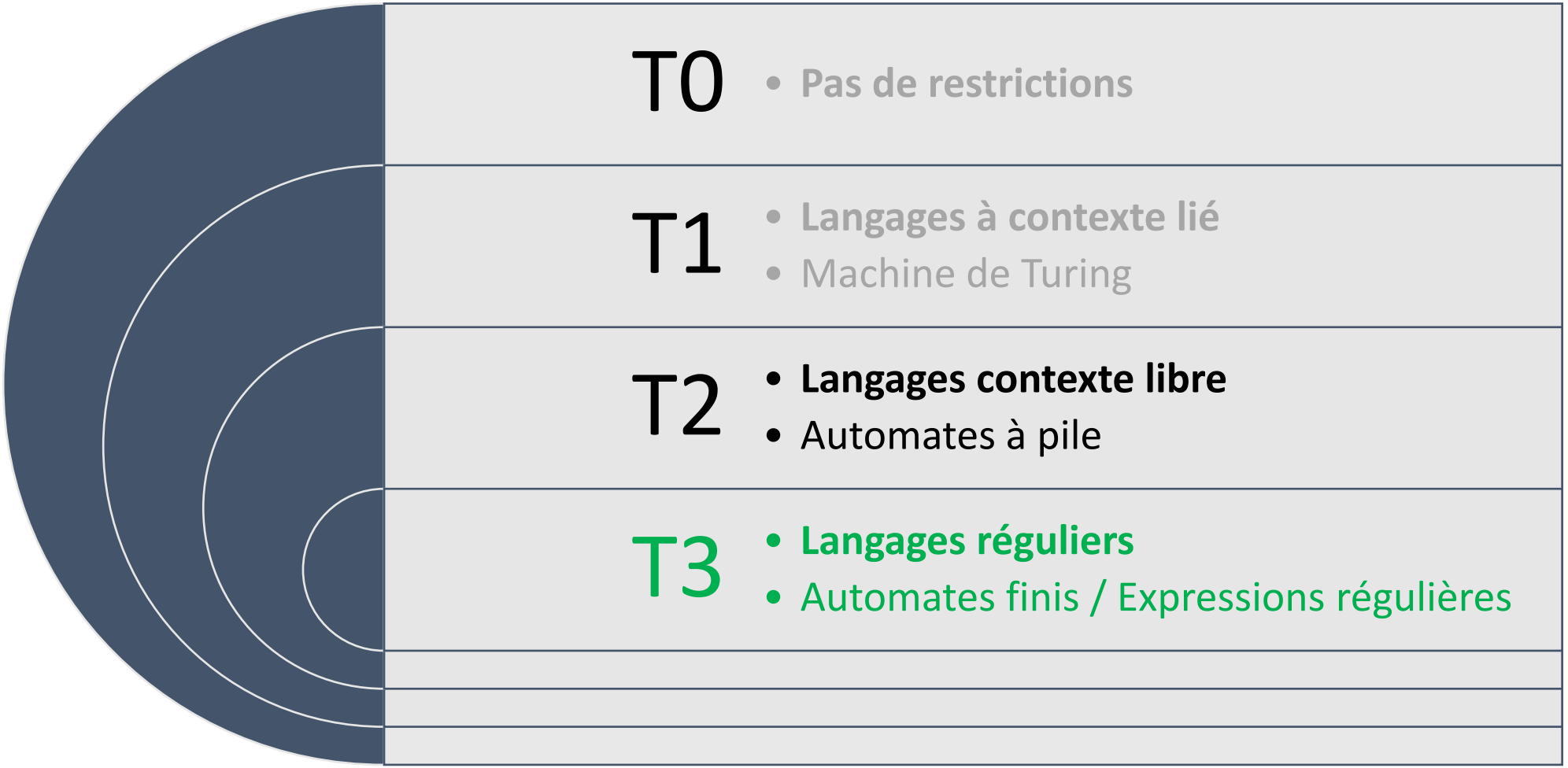
ISEN CIR ^{3}

Théorie des langages

1. Expressions régulières

2021-2022

Rappel : Types de grammaires



Rappel : Grammaires régulières

- **Grammaire de type 3 :** Dans toutes les productions :
  - La partie gauche est composée d'un seul non-terminal
  - La partie droite est soit un terminal, soit une paire terminal/non-terminal,
  - en respectant la même position du non-terminal pour toutes les règles (à gauche ou à droite).

- Exemple d'une grammaire régulière avec les non-terminaux <u>à droite</u>:

S → Aa | Sb | ε

A → Sa | Ab

Exemple

- Autre exemple, mais cette fois d'une grammaire régulière gauche:

$$S \rightarrow 0A \mid 1B \mid \varepsilon$$
$$A \rightarrow 0S \mid 1C$$
$$B \rightarrow 0C \mid 1S$$
$$C \rightarrow 0B \mid 1A$$

- Rappel: Cette grammaire permet de reconnaitre des nombres binaires ayant un nombre paire de 0 et un nombre pair de 1

Les langages réguliers

- Les langages réguliers sont la forme la plus simple des langages.

- Leur vérification par des outils informatiques est très simple et très rapide.

Autres appellations

- Appelés aussi:

Expressions rationnelles

Motifs

Expressions normales

Regex

Regexp

Définition : Ensemble des langages réguliers

- **L'ensemble des langages réguliers** (ou **langages rationnels**), sur un alphabet **A** est le plus petit ensemble **R** des langages satisfaisant les conditions suivantes :

  - $\emptyset \subset R$
  - $\{\varepsilon\} \subset R$
  - $\forall a \in A : \{a\} \subset R$
  - Si **A** et $B \subset R$ alors :

    $A.B \subset R$ (La concaténation)

    $A \cup B \subset R$ (Le choix)

    $A^* \subset R$ (La répétition)

En d’autres termes

- Les Langages réguliers sont les langages définis en utilisant une association de ces trois opérateurs :

La concaténation

Le choix

La répétition

Lien Grammaire / Automates / Exp. Reg.

- Les langages réguliers peuvent être décrits de 3 façons:
  1. Une **grammaire régulière** (linéaire à droite ou à gauche).
  2. Une **expression régulière**.
  3. Un **automate fini** (Prochain cours).

- Pour démontrer qu'un langage est régulier, il suffit de le décrire par une des 3 façons.
- Il existe des méthodes (algorithmes) pour passer d'une forme à une autre.

Expressions régulières - Définition

- Une expression régulière est une formule qui combine les éléments de l'alphabet et un ensemble d'opérateurs :
  { | (choix), . (concaténation), * (répétition), (,), ε}.
- Une expression régulière exprime un langage régulier d'une manière plus synthétique.
- Les opérations sur les expressions correspondent aux opérations sur les langages

| (Choix)	∪ (union)
. (Concaténation)	. (Produit de langages)
* (Répétition 0 à n fois)	* (Etoile, Fermeture de Kleene)

Exemples

- Constante naturelle en décimal

  0 | (1|2|3|4|6|7|8|9) . (0|1|2|3|4|5|6|7|8|9)*

  Simplifiée (dans certains langages) par : **0|[1-9][0-9]***

- Binaires multiples de 2

  (0|1)*.0

- Identificateurs en C++

  (a| … |z|A| … |Z|_).(a| … |z|A| … |Z|_|0| … |9)*

Outils informatiques

- Langages de programmation

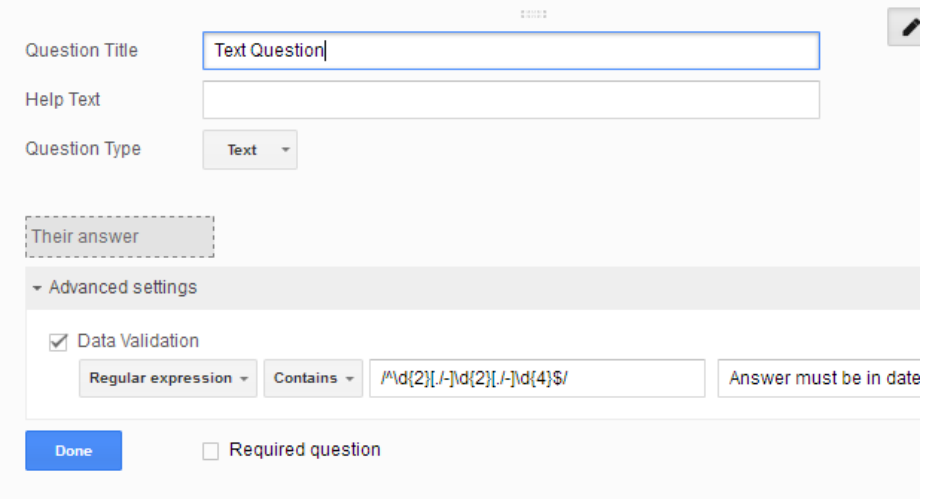
  C++, Javascript, Perl, PHP, …

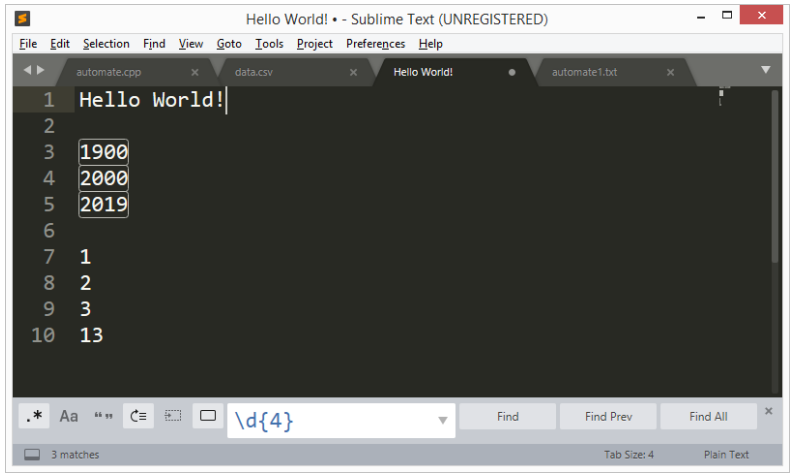
- Editeurs de texte

  Word, Notepad++, Sublime, …

- Environnement de commande

  grep






```
alw@DESKTOP-FDCJ5J0:~$ egrep d\.g
He was a good dog
He was a good dog
alw@DESKTOP-FDCJ5J0:~$ egrep d\\.g
He was a good dog
This is the match string d.g
This is the match string d.g
alw@DESKTOP-FDCJ5J0:~$
```

Plusieurs normes, notations et variantes

- Il existes plusieurs variantes et de notations pour les expressions régulières.
- le standard C++2011 énumère les différentes notations:

Ecmascript, POSIX, awk, grep, …

https://fr.wikipedia.org/wiki/Expression_r%C3%A9guli%C3%A8re

- Chacune de ses normes ajoute, en plus des trois opérateurs de base, des opérateurs relatifs aux classes de caractères (Chiffres, majuscules, minuscules, …), des opérateurs relatifs aux répétitions et autres.

Exemples

[]	Classe de caractères. Ex : [aeyuio] [0-9] [A-Z] [a-z]
.	Tout caractère sauf le caractère fin de ligne
*****	Quantificateur : 0 ou *n* fois
+	Quantificateur : 1 ou *n* fois (au moins une fois)
?	Quantificateur : Au plus une occurrence (0 ou une fois)
{m, n}	Quantificateur : Entre *n* et *m* fois. Ex : {4, 8} : répétition entre 4 et 8 fois
^	Début de ligne
$	Fin de ligne

REGEX SYNTAX	MEANING	EXAMPLE	MATCHES	DOES NOT MATCH
.	Any single character	go.gle	google, goggle	gogle
[abc]	Any of these character	analy[zs]e	analyse, analyze	analyxe
[a-z]	Any character in this range	demo[2-4]	demo2, demo3	demo1, demo5
[^abc]	None of these characters	analy[^zs]e	analyxe	analyse, analyze
[^a-z]	Not a character in this range	demo[^2-4]	demo1, demo5	demo2, demo3
|	Or	demo|example	demo, demos, example	test
^	Starts with	^demo	demos, demonstration	my demo
$	Ends with	demo$	my demo	demonstration
?	Zero or one times (greedy)	demos?123	demo123, demos123	demoA123
??	Zero or one times (lazy)			
*	Zero or more times (greedy)	goo*gle	gogle, goooogle	goggle
*?	Zero or more times (lazy)			
+	One or more times (greedy)	goo+gle	google, goooogle	gogle, goggle
+?	One or more times (lazy)			
{n}	n times exactly	w{3}	www	w, ww
{n,m}	from n to m times	a{4, 7}	aaaa, aaaaa, aaaaaa, aaaaaaa	aaaaaaaa, aaa, a
{n,}	at least n times	go{2,}gle	google, gooogle, goooogle	ggle, gogle
()	Group	^(demo|example)[0-9]+	demo1, example4	demoexample2
(?:)	Passive group (Useful for filters)			
\	Escape	AU\$10	AU$10, AU$100	AU10, 10
\s	White space			
\S	Non-white space			
\d	Digit character			
\D	Non-digit character			
\w	Word			
\W	Non-word (e.g. punctuation, spaces)			

Quelques motifs célèbres 1/2

- **Username**

  [a-z0-9_-]{3,16}

- **Password**

  [a-zA-Z0-9_-]{6,18}

- **Couleurs hexadécimales en HTML/CSS**

  #?([a-f0-9]{6}|[a-f0-9]{3})

- **Adresse e-mail**

  ([a-z0-9_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})

Quelques motifs célèbres 2/2

- **URL**

(https?:\/\/)?([\da-z\.-]+)\.([a-z\.]{2,6})([\/\w \.-]*)*\/?

- **Adresse IP**

((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)

- **Tag HTML**

<([a-z]+)([^<]+)*(?:> (.*) <\/\1>|\s*\/>)

Aller plus loin

https://regex101.com/

http://regexr.com/

https://www.regexpal.com/