De Turing à Chomsky

Langues, langages, machines et monoïdes

22 octobre 2012

Pierre Senellart

Plan

Mots et langages

Machines de Turing

Équivalence syntaxique et automates

Automates à pile

Hiérarchie de Chomsky

Alphabet, mot, langage

Définition

Un alphabet est un ensemble fini de symboles.

Alphabet, mot, langage

Définition

Un alphabet est un ensemble fini de symboles.

Définition

Un mot sur un alphabet Σ est une suite finie de symboles de Σ. La suite vide (le mot vide) est noté ε, l'ensemble de tous les mots sur Σ, Σ^*.

Alphabet, mot, langage

Définition

Un alphabet est un ensemble fini de symboles.

Définition

Un mot sur un alphabet Σ est une suite finie de symboles de Σ. La suite vide (le mot vide) est noté ε, l'ensemble de tous les mots sur Σ, Σ^*.

Exemple

$\Sigma = \{0, 1\}$. Des mots sur Σ sont ε, 0, 1, 0001, 10101, etc.

Alphabet, mot, langage

Définition

Un alphabet est un ensemble fini de symboles.

Définition

Un mot sur un alphabet Σ est une suite finie de symboles de Σ. La suite vide (le mot vide) est noté ε, l'ensemble de tous les mots sur Σ, Σ^*.

Exemple

$\Sigma = \{0, 1\}$. Des mots sur Σ sont ε, 0, 1, 0001, 10101, etc.

Définition

Un langage sur un alphabet Σ est un ensemble de mots sur Σ, c.-à-d., une partie de Σ^*.

Adresses e-mail

Alphabet : ensemble des caractères autorisés (voir la norme !)

a . . . z 0 . . . 9 - @. . .

Mots du langage

pierre.senellart@telecom-paristech.fr
toto@titi.com
bu@bi.bo.ba

Mots mal formés

a@b..com
blah
a@@b

Phrases du français

Alphabet : mots du dictionnaires (et leur formes fléchies)

chat chats manger mange manges mangeons mangerais...

Mots (phrases) du langage

le chat mange la souris
la souris mange le chat
le chat et la souris mangent le fromage

Mots (phrases) mal formés

le chat mange le souris
le chat mangent le fromage
la mange chat le

Expressions arithmétiques

Alphabet : ensemble des caractères autorisés

$$0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ +\ -\ \times\ /\ (\ )$$

Mots du langage

$$1 + 2 \times (3 - 4)$$
$$7 \times 3$$
$$1 + (2 + (3 + (4 + (5 + (6)))))$$

Mots mal formés

$$+$$
$$1 + (3\times)$$
$$1 + (2 + (3 + (4 + (5 + (6))))$$

Fragments de code C

Alphabet : mots-clefs, caractères spéciaux, caractères formant les noms de variable, opérateurs, etc.

$$for\ \{\ \}\ x <= \ldots$$

Mots du langage

```
for(i=0;i<3;++i) { printf("hello"); }
do_something();
while(1);
```

Mots mal formés

```
printf("hello")
for(;){}
"blah
```

Structure de monoïde

Soit Σ un alphabet.

Définition

La concaténation de deux mots $u = u_1 \dots u_n$ et $v = v_1 \dots v_m$ de Σ^* est le mot $u \cdot v$ (ou, plus simplement, uv) formé de $u_1 \dots u_n v_1 \dots v_m$.

Structure de monoïde

Soit Σ un alphabet.

Définition

La concaténation de deux mots $u = u_1 \dots u_n$ et $v = v_1 \dots v_m$ de Σ^* est le mot $u \cdot v$ (ou, plus simplement, uv) formé de $u_1 \dots u_n v_1 \dots v_m$.

Proposition

$(\Sigma^*, \cdot)$ *est un monoïde non commutatif.*

Structure de monoïde

Soit Σ un alphabet.

Définition

La concaténation de deux mots $u = u_1 \dots u_n$ et $v = v_1 \dots v_m$ de Σ^* est le mot $u \cdot v$ (ou, plus simplement, uv) formé de $u_1 \dots u_n v_1 \dots v_m$.

Proposition

$(\Sigma^*, \cdot)$ *est un monoïde non commutatif.*

Démonstration.

- $\cdot$ est associatif : $u(vw) = (uv)w$
- ε est neutre pour $\cdot$: $u\varepsilon = \varepsilon u = u$ $\square$

Classification des langages

But : formaliser les problèmes suivants et y répondre

- Comment donner une description compacte d'un langage ?
- Comment reconnaître efficacement si un mot est dans un langage ?
- Quelle est la complexité intrinsèque d'un langage ?

Plan

Mots et langages

Machines de Turing

Équivalence syntaxique et automates

Automates à pile

Hiérarchie de Chomsky

Modèle de calcul

- Modèle de calcul : formalisme pour décrire un langage de manière compacte
- La description d'un langage donné doit être :
  - finie
  - « implémentable », formée de règles de calcul que l'on peut appliquer sans réfléchir pour déterminer si un mot est dans le langage

Modèle de calcul

- Modèle de calcul : formalisme pour décrire un langage de manière compacte
- La description d'un langage donné doit être :
  - finie
  - « implémentable », formée de règles de calcul que l'on peut appliquer sans réfléchir pour déterminer si un mot est dans le langage

Théorème

Un modèle de calcul donné ne permet pas de décrire tous les langages possibles.

Modèle de calcul

- Modèle de calcul : formalisme pour décrire un langage de manière compacte
- La description d'un langage donné doit être :
  - finie
  - « implémentable », formée de règles de calcul que l'on peut appliquer sans réfléchir pour déterminer si un mot est dans le langage

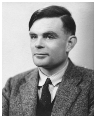
Théorème

Un modèle de calcul donné ne permet pas de décrire tous les langages possibles.

Démonstration.

- Nombre dénombrable de descriptions finies
- Nombre non dénombrable de langages possibles

Modèles de calcul universels

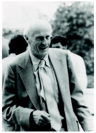


(Alan M. Turing)

Machine de Turing : transitions décrivant les actions d'une tête de lecture/écriture sur une bande infinie

(Alonzo Church)

λ-calcul : définition et application de fonctions récursives (inspiration des langages de programmation fonctionnels)



(Stephen C. Kleene)

Fonctions récursives : fonctions définissables par un ensemble d'opérations (constantes, récursion, composition, etc.)

(John von Neumann)

Machines à registres : code lisant et écrivant la valeur de registres (variables) dans un ordre quelconque

Thèse de Church–Turing

Théorème

Tous les modèles de calcul du transparent précédent sont équivalents, c.-à-d., permettent de définir les mêmes langages.

Thèse de Church–Turing

Théorème

Tous les modèles de calcul du transparent précédent sont équivalents, c.-à-d., permettent de définir les mêmes langages.

Thèse (Church–Turing)

Ces modèles de calcul définissent exactement les langages qu'un cerveau humain, ou que la nature, sont capables de calculer.

Machine de Turing

Une machine de Turing (déterministe) est définie par :

- un alphabet Σ et un alphabet de travail $\Gamma \supseteq \Sigma$
- un symbole blanc $\perp \in \Gamma \setminus \Sigma$
- un ensemble fini d'états Q
- un état initial $q_0 \in Q$ et un ensemble d'états finaux $F \subseteq Q$
- une fonction de transition (partielle) :
  $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \to\}$

Machine de Turing

Une machine de Turing (déterministe) est définie par :

- un alphabet Σ et un alphabet de travail $\Gamma \supseteq \Sigma$
- un symbole blanc $\perp \in \Gamma \setminus \Sigma$
- un ensemble fini d'états Q
- un état initial $q_0 \in Q$ et un ensemble d'états finaux $F \subseteq Q$
- une fonction de transition (partielle) :
  $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$

Une transition $(q, \alpha) \mapsto (q', \alpha', \leftarrow)$ se lit :

> *Quand la machine se trouve dans l'état q et lit le symbole α sous la tête de lecture/écriture, elle passe dans l'état q', écrit le symbole α' et déplace la tête de lecture/écriture vers la gauche.*

Langage accepté par une machine de Turing

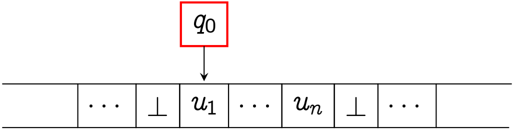
Une configuration d'une machine de Turing est :

- le contenu d'une bande infinie (avec une infinité de $\perp$ à gauche et à droite)
- la position de la tête sur la bande
- un état courant $q \in Q$

Langage accepté par une machine de Turing

Une configuration d'une machine de Turing est :

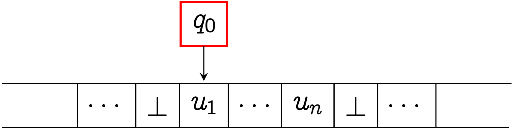
- le contenu d'une bande infinie (avec une infinité de $\perp$ à gauche et à droite)
- la position de la tête sur la bande
- un état courant $q \in Q$



Configuration initiale pour $u = u_1 \dots u_n$

Une configuration d'une machine de Turing est :

- le contenu d'une bande infinie (avec une infinité de $\perp$ à gauche et à droite)
- la position de la tête sur la bande
- un état courant $q \in Q$



Configuration initiale pour $u = u_1 \dots u_n$

Le langage accepté par une machine de Turing est l'ensemble des mots u tels que, si on part de la configuration finale pour u, on atteint un état final en suivant les transitions.

Exemple de machine de Turing

$\Sigma = \{a, b\}$, $\Gamma = \{a, b, \perp\}$, $Q = \{0 \dots 7\}$, $q_0 = 0$, $F = \{7\}$

Exemple de machine de Turing

$\Sigma = \{a, b\}$, $\Gamma = \{a, b, \perp\}$, $Q = \{0 \dots 7\}$, $q_0 = 0$, $F = \{7\}$

δ :

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$

Exemple de machine de Turing

$\Sigma = \{a, b\}$, $\Gamma = \{a, b, \perp\}$, $Q = \{0 \dots 7\}$, $q_0 = 0$, $F = \{7\}$

δ :

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$

$\boxed{0}$

$\cdots$	$\perp$	a	b	a	$\perp$	$\cdots$

Exemple de machine de Turing

$$\Sigma = \{a, b\},\ \Gamma = \{a, b, \perp\},\ Q = \{0 \dots 7\},\ q_0 = 0,\ F = \{7\}$$

δ :

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$

$\boxed{0}$

$\cdots$	$\perp$	a	b	a	$\perp$	$\cdots$

Exemple de machine de Turing

$\Sigma = \{a, b\}$, $\Gamma = \{a, b, \perp\}$, $Q = \{0 \dots 7\}$, $q_0 = 0$, $F = \{7\}$

δ :

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$

$\boxed{1}$

$\cdots$	$\perp$	$\perp$	b	a	$\perp$	$\cdots$

Exemple de machine de Turing

$\Sigma = \{a, b\}$, $\Gamma = \{a, b, \perp\}$, $Q = \{0 \dots 7\}$, $q_0 = 0$, $F = \{7\}$

δ :

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$

$\boxed{1}$

$\downarrow$

$\cdots$	$\perp$	$\perp$	b	a	$\perp$	$\cdots$

Exemple de machine de Turing

$$\Sigma = \{a, b\},\ \Gamma = \{a, b, \perp\},\ Q = \{0 \dots 7\},\ q_0 = 0,\ F = \{7\}$$

δ :

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$

$\boxed{1}$

$\cdots$	$\perp$	$\perp$	b	a	$\perp$	$\cdots$

Exemple de machine de Turing

$$\Sigma = \{a, b\},\ \Gamma = \{a, b, \perp\},\ Q = \{0 \dots 7\},\ q_0 = 0,\ F = \{7\}$$

δ :

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$

$\boxed{3}$

$\cdots$	$\perp$	$\perp$	b	a	$\perp$	$\cdots$

Exemple de machine de Turing

$\Sigma = \{a, b\}$, $\Gamma = \{a, b, \perp\}$, $Q = \{0 \ldots 7\}$, $q_0 = 0$, $F = \{7\}$

δ :

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$

$\boxed{5}$

$\downarrow$

	$\cdots$	$\perp$	$\perp$	b	$\perp$	$\perp$	$\cdots$	

Exemple de machine de Turing

$\Sigma = \{a, b\}$, $\Gamma = \{a, b, \perp\}$, $Q = \{0 \ldots 7\}$, $q_0 = 0$, $F = \{7\}$

δ :

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$

$\boxed{5}$

$\cdots$	$\perp$	$\perp$	b	$\perp$	$\perp$	$\cdots$

Exemple de machine de Turing

$$\Sigma = \{a, b\},\ \Gamma = \{a, b, \perp\},\ Q = \{0 \dots 7\},\ q_0 = 0,\ F = \{7\}$$

$\delta :$

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$

$\boxed{0}$

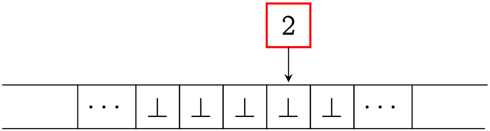
$\cdots$	$\perp$	$\perp$	b	$\perp$	$\perp$	$\cdots$

Exemple de machine de Turing

$\Sigma = \{a, b\}$, $\Gamma = \{a, b, \perp\}$, $Q = \{0 \dots 7\}$, $q_0 = 0$, $F = \{7\}$

δ :

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$



Exemple de machine de Turing

$$\Sigma = \{a, b\},\ \Gamma = \{a, b, \perp\},\ Q = \{0 \dots 7\},\ q_0 = 0,\ F = \{7\}$$

δ :

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$

$\boxed{4}$ $\downarrow$

$\cdots$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\cdots$

Exemple de machine de Turing

$\Sigma = \{a, b\}$, $\Gamma = \{a, b, \perp\}$, $Q = \{0 \ldots 7\}$, $q_0 = 0$, $F = \{7\}$

δ :

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$

$\boxed{7}$

$\cdots$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\cdots$

Accepté !

Exemple de machine de Turing

$$\Sigma = \{a, b\},\ \Gamma = \{a, b, \perp\},\ Q = \{0 \dots 7\},\ q_0 = 0,\ F = \{7\}$$

δ :

	a	b	$\perp$
0	$1, \perp, \rightarrow$	$2, \perp, \rightarrow$	$5, \perp, \rightarrow$
1	$1, a, \rightarrow$	$1, b, \rightarrow$	$3, \perp, \leftarrow$
2	$2, a, \rightarrow$	$2, b, \rightarrow$	$4, \perp, \leftarrow$
3	$5, \perp, \leftarrow$	$6, b, \leftarrow$	$7, \perp, \leftarrow$
4	$6, a, \leftarrow$	$5, \perp, \leftarrow$	$7, \perp, \leftarrow$
5	$5, a, \leftarrow$	$5, b, \leftarrow$	$0, \perp, \rightarrow$

$\boxed{7}$

$\cdots$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\cdots$

Accepté ! (palindromes)

Avantages et inconvénients des machines de Turing

Théorème (Church–Turing)

Il n'y a pas de machine de Turing qui accepte le langage des machines de Turing dont le calcul termine sur toute entrée.

- Modèle le plus général possible (thèse de Church–Turing)
- Équivalence avec machines non-déterministes (plusieurs choix possible par transition)
- Même s'il s'arrête, le calcul d'une machine de Turing peut prendre un temps arbitrairement long
- Modèle trop général pour les langages simples !

Plan

Automate déterministe

On cherche à explorer la structure des langages, pour trouver une classe de langages plus simple que ceux acceptés par une machine de Turing.

Automate déterministe

On cherche à explorer la structure des langages, pour trouver une classe de langages plus simple que ceux acceptés par une machine de Turing.
Un automate déterministe est défini par :

- un alphabet Σ
- un ensemble fini d'états Q
- un état initial $q_0 \in Q$ et un ensemble d'états finaux $F \subseteq Q$
- une fonction de transition (partielle) : $\delta : Q \times \Sigma \to Q$

Automate déterministe

On cherche à explorer la structure des langages, pour trouver une classe de langages plus simple que ceux acceptés par une machine de Turing.
Un automate déterministe est défini par :

- un alphabet Σ
- un ensemble fini d'états Q
- un état initial $q_0 \in Q$ et un ensemble d'états finaux $F \subseteq Q$
- une fonction de transition (partielle) : $\delta : Q \times \Sigma \to Q$

Une transition $(q, \alpha) \mapsto q'$ se lit :

> *Quand l'automate se trouve dans l'état q et lit le symbole α, il passe dans l'état q'.*

Automate déterministe

On cherche à explorer la structure des langages, pour trouver une classe de langages plus simple que ceux acceptés par une machine de Turing.

Un automate déterministe est défini par :

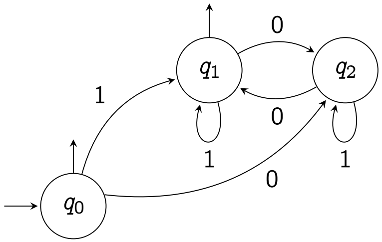
- un alphabet Σ
- un ensemble fini d'états Q
- un état initial $q_0 \in Q$ et un ensemble d'états finaux $F \subseteq Q$
- une fonction de transition (partielle) : $\delta : Q \times \Sigma \rightarrow Q$

Une transition $(q, \alpha) \mapsto q'$ se lit :

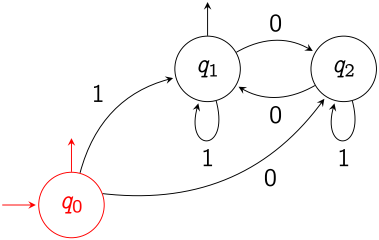
> *Quand l'automate se trouve dans l'état q et lit le symbole α, il passe dans l'état q'.*

Le langage accepté par un automate déterministe est l'ensemble des mots tels que la machine atteint, en suivant les transitions étiquetées par u, un état final en partant de l'état initial.

Exemple d'automate

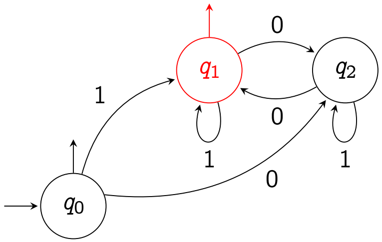


Exemple d'automate



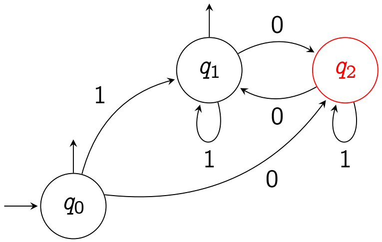
1001100

Exemple d'automate



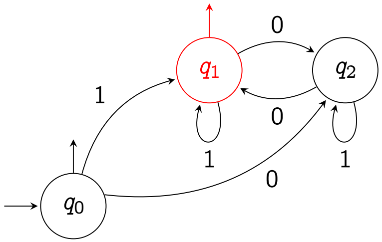
1001100

Exemple d'automate



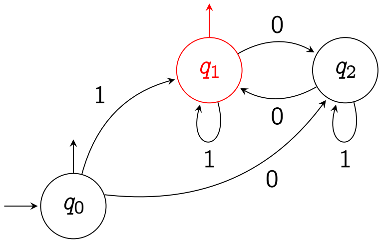
1001100

Exemple d'automate



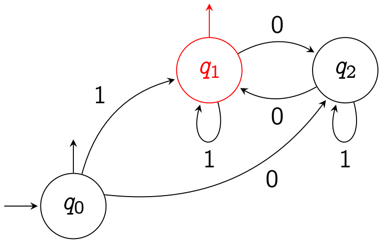
1001100

Exemple d'automate



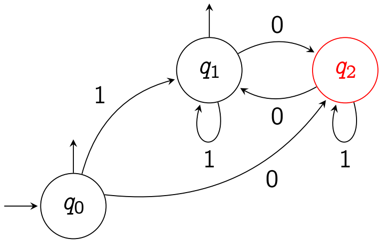
1001100

Exemple d'automate



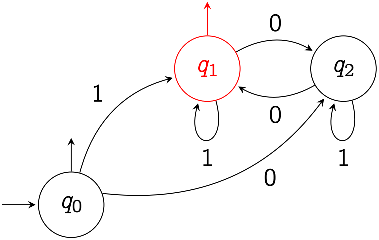
1001100

Exemple d'automate



1001100

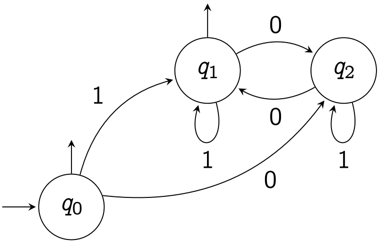
Exemple d'automate



1001100

Accepté !

Exemple d'automate



1001100

Accepté !

Langage : mots ayant un nombre pair de 0

Équivalence syntaxique

Définition

Deux mots u, v de Σ^* sont syntaxiquement équivalents à droite par rapport à un langage L s'ils ont les mêmes continuations dans L :

$$\forall w \in \Sigma^*, uw \in L \iff vw \in L.$$

On note : $u \sim_L v$ (clairement une relation d'équivalence).

Équivalence syntaxique

Définition

Deux mots u, v de Σ^* sont syntaxiquement équivalents à droite par rapport à un langage L s'ils ont les mêmes continuations dans L :

$$\forall w \in \Sigma^*, uw \in L \iff vw \in L.$$

On note : $u \sim_L v$ (clairement une relation d'équivalence).

Exemple

Si L est le langage des expressions arithmétiques bien parenthésées :

$$(1+2)+(\quad \sim_L \quad (3\times$$

$$\varepsilon \sim_L 1+$$

$$\varepsilon \not\sim_L 1$$

Langages rationnels

Définition

Un langage L est rationnel si $\Sigma^*/\sim_L$ (l'ensemble des classes d'équivalence de $\sim_L$) est fini.

Proposition

$\sim_L$ est compatible à droite avec la concaténation : si $u \sim_L v$ alors pour tout $w \in \Sigma^*$ $uw \sim_L vw$.

Corollaire

La concaténation à droite définit une action de monoïde de Σ^* sur $\Sigma^*/\sim_L$.

Automate minimal

Définition

L'automate minimal d'un langage rationnel L est défini comme suit :

- Σ est l'alphabet de L
- $Q = \Sigma^* / \sim_L$
- q_0 est la classe d'équivalence de ε pour $\sim_L$
- F est l'ensemble des classes d'équivalence de mots de L
- δ est l'ajout d'un symbole à droite sur $\Sigma^* / \sim_L$

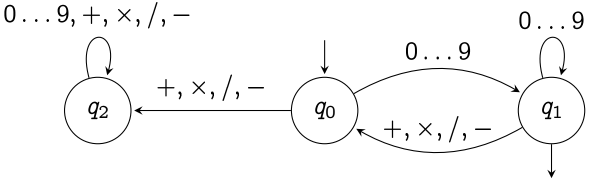
Exemple d'automate minimal

L est le langage des expressions arithmétiques sans parenthèses (p. ex., 1, $1 + 2$, $1 + 3 \times 4 \ldots$). Classes d'équivalence pour $\sim_L$:

q_0 : mot vide, mots de L suffixés d'un opérateur

q_1 : les mots de L (final)

q_2 : tous les mots sans continuation dans L ($1 + +$, $+\times$, etc.)



Minimalité

Théorème

Un langage L est rationnel si et seulement s'il est accepté par un automate déterministe. De plus, l'automate minimal de L est l'unique automate acceptant L qui soit de taille minimale et dont la fonction de transition est totale.

Minimalité

Théorème

Un langage L est rationnel si et seulement s'il est accepté par un automate déterministe. De plus, l'automate minimal de L est l'unique automate acceptant L qui soit de taille minimale et dont la fonction de transition est totale.

Démonstration.

- $\Rightarrow$ L'automate minimal est un automate déterministe.
- $\Leftarrow$ Soit A un automate pour L avec fonction de transition complète ; on suppose sans perte de généralité que tous ses états sont accessibles. On note $u \sim_A v$ si l'état atteint par A sur u et v est le même. Il est clair que $u \sim_A v \Rightarrow u \sim_L v$ et le nombre de classes d'équivalences de $\sim_A$ est exactement le nombre d'états de l'automate.

Langage non rationnel

Proposition

Le langage L des expressions arithmétiques bien parenthésées n'est pas rationnel.

Langage non rationnel

Proposition

Le langage L des expressions arithmétiques bien parenthésées n'est pas rationnel.

Démonstration.

Supposons que L soit rationnel et soit A un automate déterministe acceptant L. Soit k le nombre d'états de L.

On considère le mot $u = ({}^k0)^k \in L$. Nécessairement, il existe un $x, y, z \in \Sigma^*$ tels que $u = xyz$, $y \neq \varepsilon$, $|xy| \leq k$, et x et xy conduisent au même état dans A : comme il n'y a que k états, dans les k premières transitions, on passe nécessairement deux fois par le même état.

Alors le mot xz est accepté par A ce qui est absurde, puisqu'il contient moins de parenthèses ouvrantes que de parenthèses fermantes. $\square$

Avantages et inconvénients des automates

- Modèle très simple, beaucoup de « bonnes » propriétés :
  - Existence d'un automate minimal
  - Clôture des rationnels par union, intersection, concaténation, complémentation, itération
- Équivalence avec machines non-déterministes (plusieurs choix possibles par transition)
- Si l'automate est donné, on peut déterminer si un mot est accepté en temps linéaire
- Beaucoup de langages que l'on veut pouvoir décrire ne sont pas rationnels (expressions arithmétiques, palindromes)

Intermédiaire entre automates et machines de Turing ?

Plan

Mots et langages

Machines de Turing

Équivalence syntaxique et automates

Automates à pile

Hiérarchie de Chomsky

Automate à pile

Un automate à pile (non-déterministe) est défini par :

- un alphabet Σ et un alphabet de pile Γ
- un ensemble fini d'états Q
- un état initial $q_0 \in Q$ et un ensemble d'états finaux $F \subseteq Q$
- une relation de transition : $\Delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\perp\}) \times Q \times \Gamma^*$

Automate à pile

Un automate à pile (non-déterministe) est défini par :

- un alphabet Σ et un alphabet de pile Γ
- un ensemble fini d'états Q
- un état initial $q_0 \in Q$ et un ensemble d'états finaux $F \subseteq Q$
- une relation de transition : $\Delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\perp\}) \times Q \times \Gamma^*$

Une transition $(q, \alpha, \beta) \mapsto (q', u)$ se lit :

> *Quand l'automate se trouve dans l'état q, lit le symbole α (ou ne lit rien si $\alpha = \varepsilon$), et le symbole β est en haut de la pile (ou la pile est vide si $\beta = \perp$), il passe dans l'état q' et remplace β par u.*

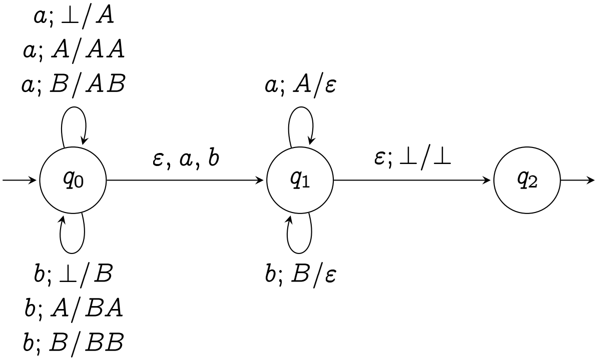
Acceptation par un automate à pile

Chaque configuration successive dans un calcul d'un automate à pile est la donnée d'un état et d'une pile de symboles.

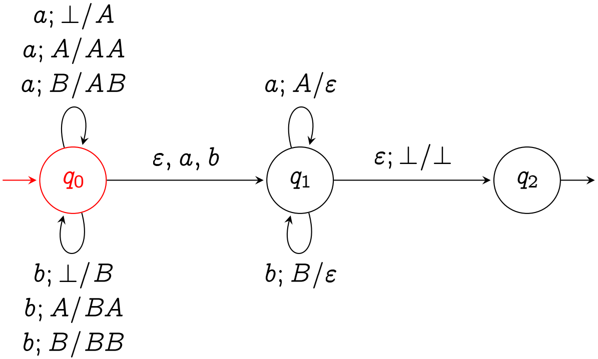
La configuration initiale est formée de l'état initial et de la pile vide.

Le langage accepté par un automate à pile est l'ensemble des mots u tels que la machine peut atteindre un état final en suivant les transitions étiquetées par u à partir de la configuration initiale.

Exemple d'automates à pile



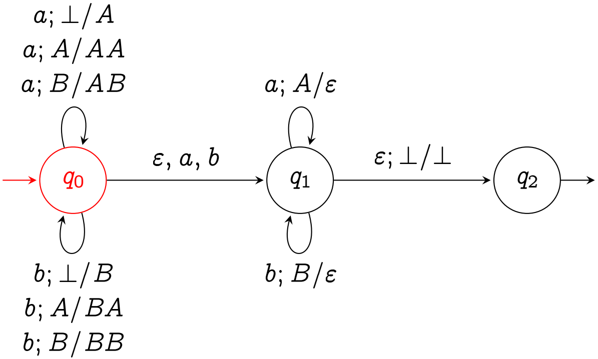
Exemple d’automates à pile



Mot : ababa

Pile

Exemple d'automates à pile



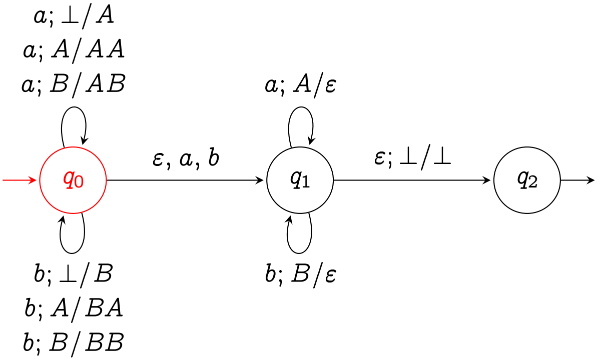
Mot : ababa

A

Pile

TELECOM ParisTech

Exemple d'automates à pile



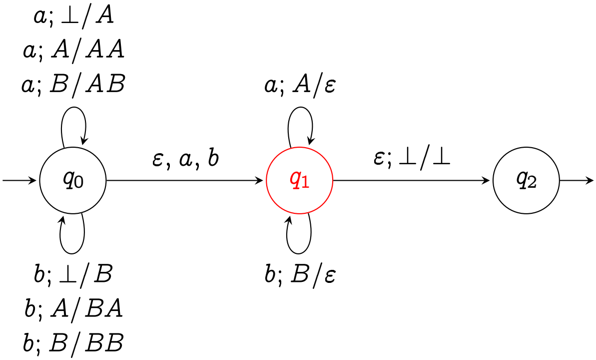
Mot : ababa

B

A

Pile

Exemple d'automates à pile



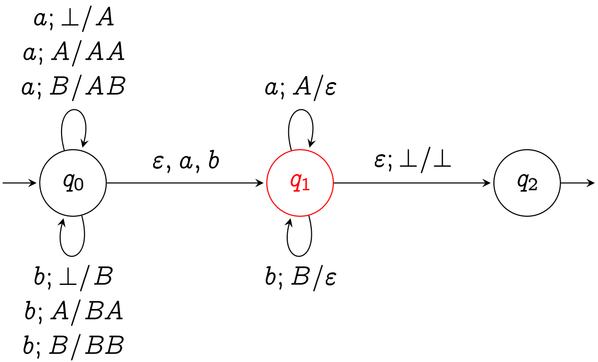
Mot : ababa

B

A

Pile

Exemple d'automates à pile

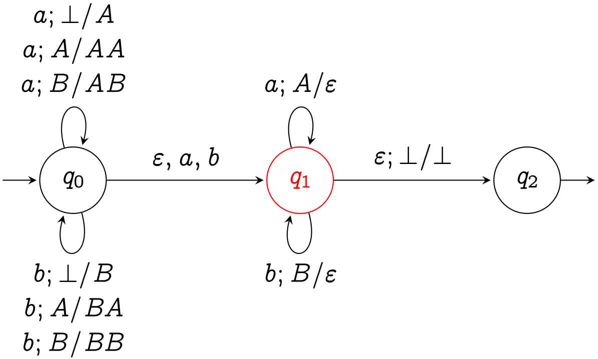


Mot : ababa

A

Pile

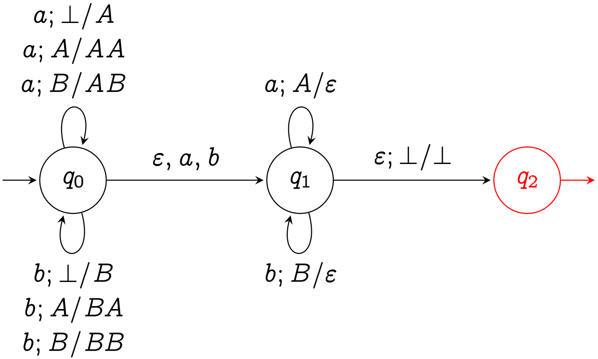
Exemple d'automates à pile



Mot : ababa

Pile

Exemple d'automates à pile



Mot : ababa

Accepté !

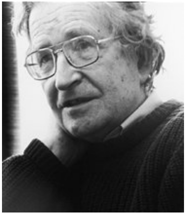
Pile

Avantages et inconvénients des automates à pile

- Plus expressif que les langages rationnels : palindromes, expressions arithmétiques, syntaxe d'un langage de programmation (sans vérifier la déclaration des variables)
- Déterminer si un mot est accepté est faisable en temps polynomial
- Non-équivalence avec machines déterministes (un seul choix possible par transition)
- Reste beaucoup de langages non exprimables : $\{a^n b^n c^n \mid n \geq 0\}$, langages de programmation avec variables. . .

Plan

Hiérarchie de Chomsky



- Noam Chomsky (né en 1928), fondateur de la linguistique moderne
- A développé la notion de grammaire générative pour les langages formels, et pour la langue naturelle
- Hypothèse d'une grammaire universelle, pré-câblée dans le cerveau humain, indépendante de la langue
- Impact majeur en informatique théorique et pratique, linguistique, traitement automatique de la langue
- Également philosophe et critique politique

Hiérarchie de Chomsky : classification des classes de langages formels, par pouvoir d'expression et complexité de reconnaissance croissants, avec modèles de calcul associés.

Type 4 : langages finis

Définition : langages finis

Exemples : $\{a, aa\}$, noms de pays du monde en français, numéros de téléphone français, mots du dictionnaire...

Complexité de reconnaissance : linéaire en la taille du mot

Machine : trie (automate déterministe acyclique)

Type 3 : Langages rationnels

Définition : langages reconnus par un automate déterministe (ou ayant un nombre fini de classes d'équivalence structurelle)

Exemples : tous les langages de type 4, adresses e-mail, ensemble des nombres pairs en base 10, lignes d'un fichier au format CSV...

Complexité de reconnaissance : linéaire en la taille du mot

Machine : automate déterministe, automate non-déterministe

Type 2 : Langages hors-contexte

Définition : langages reconnus par un automate à pile non-déterministe (ou définis par une grammaire hors contexte)

Exemples : tous les langages de type 3, palindromes, expressions arithmétiques bien parenthésées, syntaxe de certains langages de programmation sans variables, documents XML...

Complexité de reconnaissance : polynomial en la taille du mot

Machine : automate à pile non-déterministe

Type 1 : Langages contextuels

Définition : langages décrits par une grammaire générative contextuelle (ou monotone)

Exemples : tous les langages de type 2, la plupart des phénomènes des langues naturelles, $\{a^n b^n c^n \mid n \geq 1\}$, syntaxe des langages de programmation avec déclaration de variables...

Complexité de reconnaissance : exponentiel en la taille du mot

Machine : machines de Turing linéairement bornées (interdiction de dépasser les limites du mot initial sur la bande)

Type 0 : Langages semi-décidables

Définition : langages acceptés par un des modèles de calcul précédemment présenté

Exemples : tous les langages de type 1, le langage des machines de Turing qui s'arrêtent sur toute entrée, les requêtes SQL renvoyant un résultat sur au moins une base de données, le langage de n'importe quel problème dont la résolution nécessite un espace exponentiel en la taille de l'entrée...

Complexité de reconnaissance : non bornée (et pas de garantie de terminaison)

Machine : machines de Turing déterministes, machines de Turing non-déterministes, machines à registre...

Et au-delà. . .

Une infinité non dénombrable de langages non semi-décidables :

- le complément de n'importe quel langage semi-décidable mais non décidable (p. ex., machines de Turing qui ne s'arrêtent pas)
- beaucoup de langages qui n'ont pas de description intéressante (quand ils en ont une, cette description peut souvent être utilisée pour coder une machine de Turing !)

Références I

Robert F. Barsky.
Noam Chomsky: A Life of Dissent.
MIT Press, 1997.
Biographie de Noam Chomsky, également disponible en ligne sur http://cognet.mit.edu/library/books/chomsky/chomsky/.

Olivier Carton.
Langages formels, calculabilité et complexité.
Vuibert, 2008.
Cours de langages formels.

Gilles Dowek.
Les métamorphoses du calcul. Une étonnante histoire des mathématiques.
Le Pommier, 2007.

Références II

L'histoire du calcul et du raisonnement, des mathématiciens grecs aux progès récents en preuve automatique. Grand prix de philosophie 2007 de l'Académie française.

Andrew Hodges.
Alan Turing. The Enigma.
Random House, 1992.
Biographie d'Alan Turing, traduit en français sous le nom de *Alan Turing ou l'énigme de l'intelligence.*

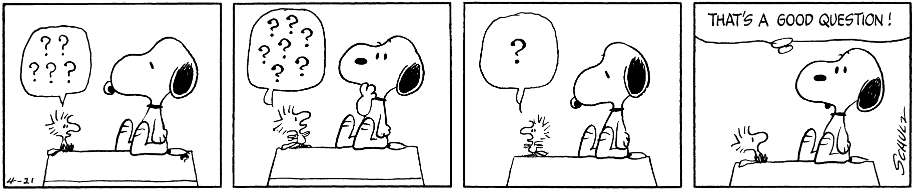
Dominique Perrin.
Les débuts de la théorie des automates.
Technique et science informatique, 14(4), 1995.
Un historique de l'émergence de la théorie des automates.

Références III

Jacques Sakarovitch.
Elements of Automata Theory.
Cambridge University Press, 2009.
Référence en théorie des automates, version française épuisée.

Alan M. Turing.
On Computable Numbers, with an Application to the Entscheidungsproblem.
Proceedings of the London Mathematical Society, 2(42), 1936.
L'article introduisant la notion de machine de Turing, un des articles fondateurs de la science informatique. Intéressant à la fois par son contenu et par son aspect historique.

Questions ?



Licence de droits d'usage

Contexte public } avec modifications

Par le téléchargement ou la consultation de ce document, l'utilisateur accepte la licence d'utilisation qui y est attachée, telle que détaillée dans les dispositions suivantes, et s'engage à la respecter intégralement.