Cours de Théorie des Langages
Partie 1

Anne Berry

11 septembre 2015

Anne.BERRY@univ-bpclermont.fr
http ://www.isima.fr/ berry/

Organisation du cours de Théorie des Langages

- Site du cours sur l'ENT (transparents, feuilles d'exercices)
- Cours en 2 parties
  - Partie 1 (Anne Berry) : septembre 2014 - octobre
  - Partie 2 (Olivier Raynaud) : noevmbre - décembre
- Modalités de contrôle des connaissances : 1e session :
  2 contrôles continus (un à la fin de chaque partie)
  2e session : examen écrit (2 heures)

Contenu de ce cours

- Introduction à la théorie des langages
- Machines à états finis, grammaires, expressions régulières
- Etude des langages rationnels

Bibliographie

- Introduction to Automata Theory, Languages, and Computation
  John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman
- Introduction to Computer Theory
  Daniel I. A. Cohen
- Wikipedia

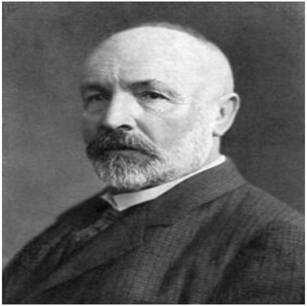
Historique

Théorie des langages : informatique fondamentale
Révolutions théoriques du 20e siècle + heureuses coïncidences !

- Georg Canter (1845-1918) (Allemand)
- David Hilbert (1862-1943) (Allemand)
- Kurt Gödel (1906-1978) (Autrichien, US)
- Alonzo Church (1903-1995) (US)
- Alan Turing (1903-1995) (Anglais)
- John Von Neumann (1903-1957) (Hongrois, US)
- Noam Chomsky (1928-) (US)
- John Backus (1924-2007) (US)
- Peter Naur (1928-) (Danois)

Historique

Georg Cantor (1845-1918) (Allemand)



Théorie des ensembles
Paradoxes sur l'infini

Historique

Bertrand Russel (1872 – 1970) (Anglais)

Philosophe et logicien

Paradoxe (antinomie) de Russel (1902) :

l'ensemble des ensembles qui ne se contiennent pas eux-mêmes....

Historique

David Hilbert (1862-1943) (Allemand)

Technique de preuve, algorithme de preuve.

Historique

Kurt Gödel (1906-1978) (Autrichien, US)

Théorème d'incomplétude (1931).

Alonzo Church (1903-1995) (US)

Existance de problème non résolubles par un algorithme.

Alan Turing (1903-1995) (Anglais)

Machine algorithmique universelle.
Programme de décodage (guerre).

Historique

John Von Neumann (1903-1957) (Hongrois, US)

De la calculette à l'ordinateur : concept du programme stocké.
Avant :

- Blaise Pascal (1623-1662)
- Ada lovelace (1815-1852)
- Charles Babbage (1791-1871)

Historique

Noam Chomsky (1928-) (US)

Linguiste, formalisation des langages.

Historique

John Backus (1924-2007) (US) Peter Naur (1928-) (Danois)

Langages de haut niveau, Algol 60

Plan du cours

1. Langages
2. Grammaires
3. Langages rationnels
4. Automates d'états finis

2. Langages

2.1. Définition d'un langage

Un langage est un ensemble de mots, qui peut être défini :

- En **extension** : liste exhaustive de tous les mots du langage.
  Exemple : un dictionnaire
- En **compréhension** : on commence une énumération.
  Exemple : $L = \{ab, aabb, aaabbb, ...\}$
- En **intension** : on se donne des 'règles'
  Exemple : Tous les mots formés de 'a' et de 'b' qui comportent autant d'occurences de 'a' que d'occurences de 'b' et dont tous les 'a' sont en début de mot.
- **Inductivement**.

$\rightarrow$ Outils de définitions compacts des langages.

Les définitions inductives (en anglais : recursive definitions)
Idée : on procède en 3 étapes

1. On se donne une **base** d'objets appartenant à l'ensemble que l'on veut définir
2. On se donne des **règles** pour construire d'autres objets de l'ensemble à partir d'objets de la base ou d'objets déjà construits.
3. On déclare que les seuls objets de l'ensemble sont ceux construits en **appliquant** un nombre **fini** de fois les règles.

- on n'exige pas que la base soit minimale
- souvent on donne une définition inductive sous la forme base+règles

2.2. Définitions inductives

Exemple :

Définissons l'ensemble PAIR des entiers pairs positifs

1. Base : 2 appartient à PAIR
2. Règle : Si x est dans PAIR, alors x+2 est dans PAIR

Pour montrer qu'un nombre est dans PAIR, on exhibera une suite d'application des règles.

2.2. Définitions inductives

La définition inductive d'un ensemble n'est pas unique.

Exemple On peut définir PAIR par :

1. Base : 2 est dans PAIR
2. Règle : si x et y sont dans PAIR, alors x+y est dans PAIR

Avantage de la 2e définition : les preuves qu'un nombre appartient à PAIR sont plus courtes.

Définition formelle de la fermeture inductive

Définition

Soit U un univers et $B \subset U$ une base, soit Ω une famille d'opérations sur U.
On appelle **fermeture inductive** *de B par Ω la partie E de U définie par :*

- *initialisation : $B \subset E$*
- *construction : $\forall f \in \Omega$ et $\forall x_1, x_2, ...x_n \in E$, si n est l'arité de f, si $x = f(x_1, x_2, ...x_n)$ est défini, alors $x \in E$*
- *fermeture : E est la plus petite partie de U qui contienne B et qui soit stable par Ω.*

2.2. Définitions inductives

Principe pour décrire des langages par une **grammaire**
(procédé formel de construction inductive du langage)
sous la forme d'un axiome (la base) et d'un ensemble de règles de production.

2.3. Définitions sur les langages

Définition

Un alphabet Σ est un ensemble fini de caractères.

Exemple : $\Sigma = \{a, b\}$

Définition

- *Un* **mot** *(appelé aussi une* **chaîne***)* ω *sur* Σ *est une suite finie de symboles de* Σ *juxtaposés.*
- *Sa* **longueur** *(= nombre de caractères) est notée* $|\omega|$.
- $|\omega|_a$ *dénote le nombre d'occurences de la lettre 'a'*$\in \Sigma$ *dans le mot* ω.

Exemple : $\Sigma = \{a, b\}$, $\omega = aabaa$ *est un mot sur* Σ *;*
$|\omega| = 5$, $|\omega|_a = 4$.

Le **mot vide**, ne contenant aucun symbole, est noté ϵ.

$|\epsilon|$=0.

ϵ peut être un mot d'un langage, mais n'est pas une lettre de l'alphabet.

Définition

Si α et β sont 2 mots sur Σ, on appelle **concaténation** *de α et β le mot $\alpha\beta$,*
noté $\alpha \circ \beta$, $\alpha \bullet \beta$, $\alpha.\beta$, $\alpha\beta$

Exemple :
$\alpha = ab$, $\beta = cd$
$\alpha \bullet \beta = abcd$

Concaténation avec le mot vide :
$\alpha \bullet \epsilon = \epsilon \bullet \alpha = \alpha$.

On notera a^n la concaténation de n occurences de a (n un nombre fini).

a^0 dénotera le mot vide ϵ.

Définition

On appelle **facteur** *ou* **sous-mot** *d'un mot* ω
un mot α *tel qu'il existe 2 mots* β *et* γ, *avec* $\omega = \beta\alpha\gamma$.

Exemple :
$\omega = abccdx$
cc est un facteur de ω.

Définition

Si $\omega = \alpha.\beta$

on dira que α *est un* **préfixe** *de* ω, *et que* β *est un* **suffixe** *de* ω.

Exemple :

$\omega =$*abccdx*

abc est un préfixe (propre).

abccdx est préfixe (mais pas propre).

dx est suffixe.

Définition

Soit Σ un alphabet, on appelle **fermeture de Kleene** *de Σ, noté Σ^*, l'ensemble défini inductivement de la façon suivante :*

base : tous les caractères de Σ ainsi que le mot vide ϵ sont dans Σ^*.

règle : si x et y sont dans Σ^*, alors xy est dans Σ^*.

Σ^* est l'ensemble des mots sur Σ, de longueur finie, plus le mot vide ϵ.

Définition

Soit Σ un alphabet, Σ^+ est l'ensemble défini inductivement de la façon suivante :

base : tous les caractères de Σ sont dans Σ^+.

règle : si x et y sont dans Σ^+, alors xy est dans Σ^+.

Σ^+ est l'ensemble des mots sur Σ, de longueur finie.

2.3. Définitions sur les langages

Définition

On appelle **langage** *(souvent noté L) sur un alphabet Σ un sous-ensemble de Σ^*.*

Définition

$\overline{L_1}$ *est l'ensemble des mots de* Σ^* *qui ne sont pas dans* L_1*.*
$\overline{L_1}$ *s'appelle le* **complémentaire** *de* L_1*.*

2.4. Opérations sur les langages

$\cup$,$\cap$

Soient L_1 et L_2 2 langages sur l'alphabet Σ

$L_1 \cup L_2 = \{\omega \mid \omega \in L_1 \text{ ou } \omega \in L_2\}$

$L_1 \cap L_2 = \{\omega \mid \omega \in L_1 \text{ et } \omega \in L_2\}$

2.4. Opérations sur les langages

Produit de 2 langages
Soient L_1 un langage sur l'alphabet Σ_1
et L_2 un langage sur l'a lphabet Σ_2
$L_1 \bullet L2 = \{\omega_1\omega_2 \in (\Sigma_1 \cup \Sigma_2)^*, \omega_1 \in L_1$ et $\omega_2 \in L_2\}$

Fermeture de Kleene d'un langage :

1 - $L^0 = \{\epsilon\}$

2 - $L^n = LL^{n-1}$, $\forall n \geq 1$

3 - $L^* = \cup\ L^n$, $n \geq 0$

4 - $L^+ = \cup\ L^n$, $n > 0$

2.5. L'ambiguïté

Définition

Un phrase ***ambigüe*** *est une phrase à laquelle on peut attribuer plusieurs sens.*

En informatique : **conflit**.

Définition

Une ***interprétation*** *d'une phrase ambigüe est un sens que l'on attribue à cette phrase.*

Exemple 1 : C'est la voiture de l'étudiant qui a coulé une bielle.

*Exemple 2 : L'expression 2 + 3 * 4 est ambigüe :*

- Interprétation 1 : (2+3)*4 (le résultat est 20)
- Interprétation 2 : 2+(3*4) (le résultat est 14)

2.5. Une hiérarchie de langages

Chomsky a défini une hiérarchie des langages (la hiérarchie de Chomsky) en 4 grandes classes.

- Type 0 : Langages récursivement énumérables.
- Type 1 : Langages contextuels.
- Type 2 : Langages algébriques (context-free).
- Type 3 : Langages rationnels (réguliers).

Propriété

On a : Type 3 $\subsetneq$ *Type 2* $\subsetneq$ *Type 1* $\subsetneq$ *Type 0*

Dans cette 1e partie, nous étudierons les langages de type 3 (les plus simples).