Automates, langages et compilation

Introduction

Isabelle Ryl

2024 – 2025

Cours de L3 - Université Paris Dauphine-PSL

Organisation

Comment le cours va-t-il se dérouler ?

- Cours 7 semaines
  - Isabelle Ryl
  - le mercredi de 8h30 à 11h45
  - du 04/09/24 au 16/10/24 inclus
  - Attention probable déplacement du cours du 16/10
- TD 6 semaines
  - MmeAriane Ravier
  - TD le jeudi de 8h30 à 11h45
  - du 12/09 au 17/10
- TP 4 semaines du 31/10 au 21/11
  - Mme Ariane Ravier le jeudi de 8h30 à 11h45
  - M. Matthieu Hervouin (edt à confirmer)
- Évaluation
  - 30% contrôle continu
  - 70% examen

Introduction

De quoi allons nous parler ?

Le cours porte sur les langages formels, par exemple $\{a, bdad, bb, db\}$

- est un langage de 4 mots, qui est donc fini
- sur un alphabet qui comprend au moins a, b, d
- on l'a représenté par l'ensemble de ses mots

Mécanismes principaux

Le cours va s'intéresser à deux familles de mécanismes permettant de définir les langages, ceux qui :

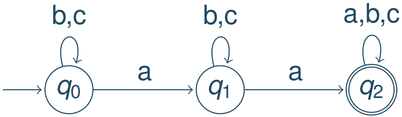
- les « génèrent » comme
  - les expressions régulières. Ex :

$$(b+c)^*a(b+c)^*a(a+b+c)^*$$

  - les grammaires. Ex :

$$\begin{aligned} S &\longrightarrow TaTaT \\ T &\longrightarrow Ta \mid Tb \mid Tc \mid \varepsilon \end{aligned}$$

  $\Rightarrow$ plutôt pour décrire les langages
- les « acceptent » ou « reconnaissent » comme les automates



$\Rightarrow$ plutôt pour décrire des algorithmes

Pourquoi ?

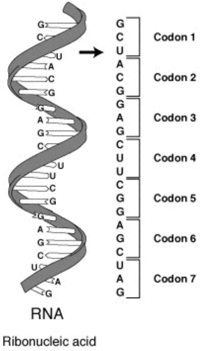
- Plusieurs courants différents ont cherché des modèles, soit des modèles de calcul ou d'encodage, soit des modèles des langues naturelles (les linguistes et en particuliers Chomsky dont nous reparlerons)
- Idée : se doter d'outils formels qui permettent de manipuler les « langages », ou des suites de symboles finies ou non qui peuvent servir à modéliser de nombreux problèmes discrets
- De nombreux autres modèles qui « ressemblent », qui produisent des sorties, qui peuvent modéliser des ressources, etc

☛ L'école française est très reconnue dans le domaine

Pour faire quoi ?

Pour décrire et modéliser de nombreux systèmes ou problèmes discrets, ex :

- Compilation
- Langage naturel
- Systèmes dynamiques discrets
- Bioinfo [1]



1. Téléversé par Sverdrup sur Wikipédia anglais. Transféré de en.wikipedia à Commons., Domaine public,https ://commons.wikimedia.org/w/index.php ?curid=1534478

La compilation

Permet de traduire un ensemble de commandes écrites dans un langage dans un autre langage, en particulier des langages de programmation vers in fine le langage machine. Étapes :

1. analyse lexicale +/- « est-ce que le mot est dans le dictionnaire »
2. analyse syntaxique +/- « est-ce que la phrase est correcte grammaticalement »
3. analyse sémantique +/- « quel est le sens de la phrase »

Rappels

Ensembles

Ensemble : collection d'« objets », *i.e.* d'éléments non typés, on s'intéresse en théorie des ensembles à la notion d'appartenance.

Ex : $\{0, 2, 4, 6, 8\}$ ou $\{0, a, \{1, E\}\}$

- ensemble vide : $\emptyset$
- singleton : ensemble contenant exactement un élément
- cardinalité de l'ensemble A : $|A|$

Parties d'un ensemble :

- ensemble des parties de A : $P(A)$ ou 2^A
- pour tout A, $\emptyset \in P(A)$ et $A \in P(A)$
- si $|A| = n$ alors $|P(A)| = 2^n$

Partition d'un ensemble A : $\{A_0, A_1, \ldots, A_n\}$ telle que :

- $\forall 0 \leq i \leq n, A_i \neq \emptyset \wedge A_i \subseteq A$
- $\forall 0 \leq i, j \leq n, A_i \cap A_j = \emptyset$
- $A_0 \cup A_1 \cup \cdots \cup A_n = A$

Fonctions

Une fonction f d'un ensemble A dans un ensemble B,

$$\begin{array}{rccc} f: & A & \longrightarrow & B \\ & x & \longmapsto & f(x) \end{array}$$

associe à un chaque élément de A un unique élément de B

- x est l'antécédent de $f(x)$
- $f(x)$ est l'image de x
- A est l'ensemble de départ
- B est l'ensemble d'arrivée (potentiellement égal à A)
- $f(A) = \{f(x) | x \in A\}$
- f est injective si $\forall x, y \in A, (f(x) = f(y) \Rightarrow (x = y)$
- f est surjective si $\forall y \in B, \exists x \in A$ tel que $f(x) = y$
- si f est injective et surjective, elle est bijective

Relations

Les différents objets mathématiques ont des **propriétés** qui les caractérisent. Les propriétés peuvent être vues comme des fonctions dont l'ensemble d'arrivée a 2 élements : $\{oui, non\}$, $\{vrai, faux\}$, $\{0, 1\}$, ...

Une **relation binaire** R entre les éléments de deux ensembles E et F est un sous-ensemble G du produit cartésien $E \times F$: $R = (E, F, G)$ et on note pour deux éléments $x \in E$ et $y \in F$, $x\ R\ y$ si $(x, y) \in G$

Une **relation d'arité** n entre les éléments des ensembles $E_1, \ldots, E_n$ est un sous-ensemble du produit cartésien fini $E_1 \times \cdots \times E_n$

Propriété des relations

Soit R une relation binaire sur X

- R est **réfléxive** ssi $\forall x \in X, x\ R\ x$
- R est **antiréfléxive** ssi $\forall x \in X, \neg(x\ R\ x)$
- R est **symétrique** ssi $\forall x, y \in X, (x\ R\ y) \Rightarrow (y\ R\ x)$
- R est **antisymétrique** ssi $\forall x, y \in X, (x\ R\ y \wedge y\ R\ x) \Rightarrow (x = y)$
- R est **transitive** ssi $\forall x, y, z \in X, (x\ R\ y \wedge y\ R\ z) \Rightarrow (x\ R\ z)$

Relation d'équivalence

Une **relation d'équivalence** R sur un ensemble E est une relation binaire réflexive, symétrique et transitive.

Exemple : l'égalité sur les entiers naturels.

La **classe d'équivalence** d'un élément x de E est alors
$Cl_R(x) = \{y \in E \mid x \; R \; y\}$

Classes d'équivalence - Partition

Lemme L'ensemble des classes d'équivalences de E pour R est une partition de E

Preuve. Soit $P = \{Cl_R(x) \mid \forall x \in E\}$

- comme R est réflexive, $\forall x \in E, x \in Cl_R(x)$ donc
  - tout élément de P est non vide
  - l'union des éléments de P est E
- reste à montrer que deux éléments de P sont disjoints

Soient $Cl_R(x), Cl_R(y) \in P$ telles que $Cl_R(x) \cap Cl_R(y) \neq \emptyset$ et $Cl_R(x) \neq Cl_R(y)$.

Soit $z \in Cl_R(x) \cap Cl_R(y)$, alors $x\ R\ z$ et $y\ R\ z$, comme R est symétrique $z\ R\ y$ et comme R est transitive, on a : $x\ R\ y$.

Pour tout $z \in Cl_R(y)$, $y\ R\ z$, et comme R est transitive $x\ R\ z$, donc $z \in Cl_R(x)$, donc $Cl_R(y) \subseteq Cl_R(x)$, de la même manière on montre que $Cl_R(x) \subseteq Cl_R(y)$, donc $Cl_R(x) = Cl_R(y)$. Contradiction.

Relation d'ordre

Une **relation d'ordre partiel** R sur un ensemble E est une relation binaire réflexive, antisymétrique et transitive.

Une relation d'ordre R sur un ensemble E est **totale** si $\forall x, y \in E$, soit $x \ R \ y$ soit $y \ R \ x$

Technique de preuve : la récurrence

Dans quels cas ? Une preuve par récurrence permet d'établir une propriété lorsque celle-ci peut être vue comme dépendante d'un entier n et vraie pour toutes les valeurs de n

Comment ?

- établir la propriété pour $n = 0$, soit $P(n)$
- hypothèse de récurrence : pour tout $i \leq n$, la propriété $P(i)$ est vraie
- à montrer :
  - $P(n) \Rightarrow P(n+1)$: récurrence « simple »
  - $(P(0), P(1), \ldots, P(n)) \Rightarrow P(n+1)$: récurrence « forte »

Idée sous-jacente Pour tout ensemble $E \subseteq \mathbb{N}$ tel que :

- $0 \in E$
- $\forall n, (n \in E) \Rightarrow (n+1 \in E)$

alors $E = \mathbb{N}$

Technique de preuve : principe des tiroirs

Idée : si on a n chaussettes à ranger dans m tiroirs avec $n > m$ alors au moins un tiroir contiendra strictement plus d'une chaussette

Définition formelle Soient E et F des ensembles finis tels que $|E| > |F|$ alors il n'existe pas de fonction injective de E dans F.

Techniques de preuve

Il existe évidemment de nombreuses autres techniques de preuve.

➥À proscrire :

- preuve par intimidation
- preuve par incantation
- …