Automates et Calculabilité

– Expressions régulières & Algèbres de Kleene –

Leçon 6 : 24 nov 2003

Uwe Nestmann

EPFL-I&C

Equivalence

Théorème:

Soit $A \subseteq \Sigma^*$. Les propositions suivantes sont équivalentes :

(0) A est un ***langage régulier***.

(1) Il existe un AFD/AFN/AFN$_\epsilon$ M tel que $L(M) = A$.

(2) Il existe une ***expression régulière*** α tel que $L(\alpha) = A$.

(3) Il existe un motif α tel que $L(\alpha) = A$.

Preuve :

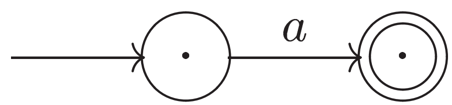
- (0) $\Leftrightarrow$ (1) : par définition.
- (2) $\Leftarrow$ (3) : par inclusion.
- (2) $\Rightarrow$ (3) : par “simulation”, voir Exemple (III) de leçon 5.
- (2) $\Rightarrow$ (1) : par induction sur la définition de $L(\alpha)$.
- (1) $\Rightarrow$ (2) : ???

$(2) \Rightarrow (1)$

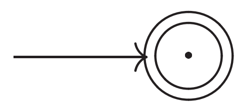
Il faut démontrer que pour tout expression régulière α, il existe un AFD (ou un AFN$_{\boldsymbol{\epsilon}}$) M_α tel que $L(M_\alpha) = L(\alpha)$.

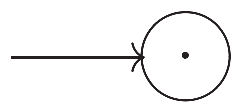
La démonstration se fait par induction sur la définition de $L(\alpha)$:

- 3 cas de base :

cas $a \in \Sigma$: $L\Bigg($ 

$\Bigg) = \{\, a \,\} = L(a)$

cas $\boldsymbol{\epsilon}$: $L\Bigg($  $\Bigg) = \{\, \epsilon \,\} = L(\boldsymbol{\epsilon})$

cas ø : $L\Bigg($  $\Bigg) = \emptyset = L(\text{ø})$

$(2) \Rightarrow (1)$

- 3 cas d'induction.
  Nous supposons que l'hypothèse est vraie
  pour les expressions régulières α et β :
  il existe M_α et M_β tel que $L(M_\alpha) = L(\alpha)$ et $L(M_\beta) = L(\beta)$.

  cas α^* : Par la propriété de clôture de la $*$-fermeture des langages réguliers, nous construisons $(M_\alpha)^*$ qui satisfait $L((M_\alpha)^*) = L(\alpha)^* = L(\alpha^*)$.

  cas $\alpha + \beta$: Par la propriété de clôture de l'union des langages réguliers, nous construisons $M_{\alpha\cup\beta}$ qui satisfait $L(M_{\alpha\cup\beta}) = L(\alpha) \cup L(\beta) = L(\alpha + \beta)$.

  cas $\alpha \cdot \beta$: Par la propriété de clôture de la concaténation des langages réguliers, nous construisons $M_{\alpha\beta}$ qui satisfait $L(M_{\alpha\beta}) = L(\alpha)L(\beta) = L(\alpha\beta)$.

$(2) \Rightarrow (1)$

Il faut démontrer que pour tout AFN$_\epsilon$ N,
il existe une expression régulière α telle que $L(M) = L(\alpha)$.

Comment peut-on — de manière générique — construire un automate à partir d'une expression regulière?

Il existe plusieurs méthodes.
Nous allons étudier *une* méthode qui se base sur la théorie des ***Algèbres de*** (Stephen C.) ***Kleene***.

Algèbres / Structures algébriques

Les algèbres sont des structures $(X, c_1 \ldots, c_n, f_1 \ldots, f_m)$ dont

- $c_1 \ldots, c_n$ sont des symboles de constante (type X),
- $f_1 \ldots, f_m$ sont des symboles de fonction (type $X^k \rightarrow X$),

Dans l'interprétation *syntaxique* générique,
X est exactement l'ensemble de tous les termes proprement formé selon l'arité (type) des symboles.

Dans l'interprétation *sémantique*,
X peut-être un ensemble d'éléments, et les symboles sont associés avec des opérations et relations de la propre arité sur cet ensemble X.

Il existe aussi des *structures algébriques relationnels*
$(X, c_1 \ldots, c_n, f_1 \ldots, f_m, r_1 \ldots, r_o)$
où $r_1 \ldots, r_o$ sont des symboles de relations.

Raisonnement équationnel (I)

Nous considérons des algèbres $(X, c_1 \ldots, c_n, f_1 \ldots, f_m, e)$ qui sont équipées avec une relation e d'***égalité***, une relation binaire définie et contrainte par un ensemble de lois qui nous permet de raisonner de manière ***équationnelle***, c'est-à-dire, de *calculer* avec les termes en appliquant les ***lois*** de l'algèbre.

Le principe de base est la substitution d'un terme à un autre. Ce mécanisme doit être appliquable

- □ plusieures fois de suite, et
- □ à tous les sous-termes.

Souvent, le but du raisonnement équationnel est de *simplifier* les termes pour arriver à une *forme standard*. ($\Rightarrow$ *programmation fonctionnelle*)

Nombres naturels (I)

Soit la structure $(\text{NAT}, \text{zéro}, \text{succ}, \text{plus})$ définie par :

$$\frac{}{\text{zéro} \in \text{NAT}} \qquad \frac{n \in \text{NAT}}{\text{succ}(n) \in \text{NAT}} \qquad \frac{n_1 \in \text{NAT} \qquad n_2 \in \text{NAT}}{\text{plus}(n_1, n_2) \in \text{NAT}}$$

Une relation binaire $\equiv$ sur NAT est définie sur cette structure.
Elle est donnée par les lois :

$$\begin{aligned} \text{plus}(\text{zéro}, m) &\equiv m \\ \text{plus}(\text{succ}(n), m) &\equiv \text{succ}(\text{plus}(n, m)) \end{aligned}$$

L'utilisation de *schémas* (comme dans cet exemple où les metavariables m, n représentent les éléments de NAT) nous permet d'éviter de devoir écrire un nombre infini de règles.

Raisonnement équationnel (II)

Soit X un ensemble.
Soit x, y, z des metavariables représentant les éléments de X.

Une relation $\mathcal{B}$, binaire sur X, est une ***équivalence***
si elle satisfait les trois conditions suivantes :

$$(\text{REFL})\ \frac{}{x\ \mathcal{B}\ x} \qquad (\text{SYM})\ \frac{x\ \mathcal{B}\ y}{y\ \mathcal{B}\ x} \qquad (\text{TRANS})\ \frac{x\ \mathcal{B}\ y \qquad y\ \mathcal{B}\ z}{x\ \mathcal{B}\ z}$$

réflexivité : doit trivialement être satisfaite

symétrie : permet de lire les lois dans les deux sens

transitivité : permet l'application répétitive des lois

Nombres naturels (II)

Soit la relation binaire $\equiv$ sur NAT redéfinie comme étant *la plus petite **équivalence** sur NAT qui satisfait les lois* :

$$\begin{aligned} \text{plus}(\text{zéro}, m) &\equiv m \\ \text{plus}(\text{succ}(n), m) &\equiv \text{succ}(\text{plus}(n, m)) \end{aligned}$$

Exemple:

Simplifier autant que possible avec la relation $\equiv$.

1. $\text{plus}(\text{succ}(\text{succ}(\text{zéro})), \text{succ}(\text{zéro}))$
2. $\text{plus}(\text{plus}(\text{succ}(\text{zéro}), \text{succ}(\text{zéro})), \text{zéro})$

Raisonnement équationnel (III)

Soit X un ensemble.
Soit x, y, z des metavariables représentant les termes de X.

Si un terme y est un ***sous-terme*** du terme z, on écrit $z = C[y]$ où $C[\cdot]$ dénote un ***contexte*** : « tout le reste du terme z autour de y ». Un contexte $C[\cdot]$ est un terme avec un « trou ».
$C[y]$ est le *terme* dans lequel le trou a été remplacé par y.
La formalisation précise de $C[\cdot]$ dépend de la construction (la structure) des éléments de X.

Une équivalence $\mathcal{B}$ est une ***congruence*** si elle satisfait

$$\text{(CONG)} \ \frac{x \ \mathcal{B} \ y}{C[x] \ \mathcal{B} \ C[y]}$$

pour tout contexte $C[\cdot]$

Nombres naturels (III)

L'ensemble des contextes sur NAT est défini par :

$$C[\cdot] \quad ::= \quad [\cdot] \quad \Big| \quad \text{succ}(C[\cdot]) \quad \Big| \quad \text{plus}(n, C[\cdot]) \quad \Big| \quad \text{plus}(C[\cdot], n)$$

Soit la relation binaire $\equiv$ sur $\mathbb{N}$ redéfinie comme étant
la plus petite ***congruence*** *sur* $\mathbb{N}$ *qui satisfait les lois* :

$$\begin{aligned} \text{plus}(\text{zéro}, m) &\equiv m \\ \text{plus}(\text{succ}(n), m) &\equiv \text{succ}(\text{plus}(n, m)) \end{aligned}$$

Exemple:

Démontrer que
$\text{succ}(\text{succ}(\text{zéro})) \equiv \text{plus}(\text{plus}(\text{succ}(\text{zéro}), \text{succ}(\text{zéro})), \text{zéro})$.

Expressions régulières

Soit $\Sigma = \{ a_1 \ldots, a_n \}$ un alphabet fini.
Considérons la structure algébrique $(\mathbf{R}_\Sigma, \boldsymbol{\emptyset}, \boldsymbol{\epsilon}, a_1 \ldots, a_n, \cdot, +, {}^*)$.

Théorème: La relation $\stackrel{L}{=}$ est une congruence sur $\mathbf{R}_\Sigma$.

Théorème: Il existe un ensemble de lois qui défini (de manière syntaxique) une congruence $\equiv$ sur $\mathbf{R}_\Sigma$ telle que

$$\alpha \equiv \beta \qquad \text{ssi} \qquad \alpha \stackrel{L}{=} \beta.$$

Regardons cet ensemble de lois d'un peu plus près . . .

« Kleene »

$$\begin{aligned}
\alpha + (\beta + \gamma) &\equiv (\alpha + \beta) + \gamma \\
\alpha + \beta &\equiv \beta + \alpha \\
\alpha + \emptyset &\equiv \alpha \\
\alpha + \alpha &\equiv \alpha
\end{aligned}$$

$$\begin{aligned}
\alpha(\beta\gamma) &\equiv (\alpha\beta)\gamma \\
\alpha\epsilon &\equiv \alpha \\
\epsilon\alpha &\equiv \alpha \\
\alpha\emptyset &\equiv \emptyset \\
\emptyset\alpha &\equiv \emptyset
\end{aligned}$$

$$\begin{aligned}
\alpha(\beta + \gamma) &\equiv \alpha\beta + \alpha\gamma \\
(\alpha + \beta)\gamma &\equiv \alpha\gamma + \beta\gamma
\end{aligned}$$

$$\begin{aligned}
\epsilon + \alpha\alpha^* &\equiv \alpha^* \\
\epsilon + \alpha^*\alpha &\equiv \alpha^*
\end{aligned}$$

Définissons un ordre (partiel) : $\alpha \leq \beta \stackrel{\text{def}}{\iff} \alpha + \beta \equiv \beta$
Comparer avec : $L(\alpha) \subseteq L(\beta) \iff L(\alpha) \cup L(\beta) = L(\beta)$

$$\frac{\beta + \alpha\gamma \leq \gamma}{\alpha^*\beta \leq \gamma} \qquad \frac{\beta + \gamma\alpha \leq \gamma}{\beta\alpha^* \leq \gamma}$$

Algèbres de Kleene

Soit $\mathbf{K} \stackrel{\text{def}}{=} (X, 0, 1, ^*, \cdot, +, =)$ une structure dont

- □ 0 et 1 sont des éléments constants dans X,
- □ * est une opération unaire sur X, et
- □ $\cdot, +$ sont des opérations binaires sur X,
- □ $=$ est une relation binaire sur X.

Si $=$ est une congruence qui satisfait
exactement les lois de « Kleene » du transparent précédent
(où nous remplaçons **ø** par 0, ϵ par 1, et α, β, γ par x, y, z)
alors $\mathbf{K}$ est appelée une *algèbre de Kleene*.

Lemme: (***Arden***)
Soit $(X, 0, 1, \cdot, +, ^*, =)$ une algèbre de Kleene.
Alors, $y^* z$ est une solution de l'équation $x = yx + z$.
En plus, $y^* z$ est la solution la plus petite par rapport à l'ordre $\leq$.

Autres algèbres de Kleene

1. $(2^{\Sigma^*}, \emptyset, \{\,\epsilon\,\}, {}^*, \cdot, \cup, =)$

2. $(X \times X, \emptyset, \text{id}, \circ, \cup, =)$
   où
   - id $\stackrel{\text{def}}{=} \{\,(u,u) \mid u \in X\,\}$, et
   - $R^0 \stackrel{\text{def}}{=}$ id et $R^{n+1} \stackrel{\text{def}}{=} R^n \circ R$.
   - $R^* \stackrel{\text{def}}{=} \bigcup_{n\geq 0} R^n$

3. ...

Quelques lois pratiques . . .

$$\begin{array}{rcll}
\alpha^*\alpha^* & \equiv & \alpha^* & \\
\alpha^{**} & \equiv & \alpha^* & \\
\alpha^* & \equiv & (\alpha\alpha)^* + \alpha(\alpha\alpha)^* & \\
\alpha\alpha^* & \equiv & \alpha^*\alpha & \\
\alpha(\beta\alpha)^* & \equiv & (\alpha\beta)^*\alpha & \text{« shifting »} \\
(\alpha^*\beta)^*\alpha^* & \equiv & (\alpha+\beta)^* & \text{« denesting »} \\
\alpha^*(\beta\alpha^*)^* & \equiv & (\alpha+\beta)^* &
\end{array}$$

A utiliser dans les solutions des exercices d'aujourd'hui.

Il y a deux façons de démontrer la validité d'une loi $\alpha_1 \equiv \alpha_2$:

1. « sémantiquement » par $L(\alpha_1) = L(\alpha_2)$;
2. « syntaxiquement » par raisonnement équationnel à partir des lois de l'algèbre de Kleene.

Exemple

Démontrer que

$(b + ab + aab)^*(\boldsymbol{\epsilon} + a + aa) \equiv ((\boldsymbol{\epsilon} + a)(\boldsymbol{\epsilon} + a)b)^*(\boldsymbol{\epsilon} + a)(\boldsymbol{\epsilon} + a)$

Convertir un automate en expression

Soit $(\mathcal{Q}, \{\, a_1 \ldots, a_n, \boldsymbol{\epsilon}\,\}, \Delta, S, F)$ l'automate à convertir.

1. Pout tout état q_i, établir une équation

   $$q_i \equiv \alpha_{i1} \cdot q_1 + \cdots + \alpha_{in} \cdot q_n \ \ [+\boldsymbol{\epsilon}]$$

   où $\alpha_{ij} \ \stackrel{\text{def}}{=} \ b_1 + \cdots + b_n$ est défini pour $1 {\leq} k {\leq} n$ par

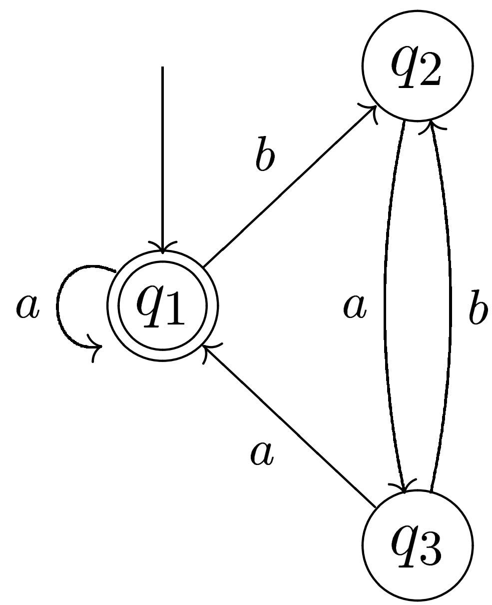
   $$b_k \ \stackrel{\text{def}}{=} \ \begin{cases} a_k & \text{si } q_j \in \Delta(q_i, a_k) \\ \boldsymbol{\varnothing} & \text{si } q_j \notin \Delta(q_i, a_k) \end{cases}$$

   et en ajoutant la partie $[+\boldsymbol{\epsilon}]$ uniquement si $q_i \in F$.

2. On arrive à un système de $|\mathcal{Q}|$ équations linéaires qui définissent les $|\mathcal{Q}|$ « variables » q_i.
   On peut résoudre ce système en utilisant le lemme d'Arden et les lois de l'algèbre de Kleene.

3. La solution est l'union des résultats des états initiaux.

Exemple



Attention:
En fonction de l'ordre dans lequel on résoud les équations, on arrive à des solutions « différentes ».
Cependant, ceci n'est pas un problème parce que . . .