Informatique Théorique 3
2004/05

Semaine 4

Définition par des règles d'inférence

Exemple (juste pour motiver les formes):

X : le plus petit ensemble qui satisfait des règles des formes

$$\frac{c \text{ in } C}{c \text{ in } X} \qquad \frac{x_1 \text{ in } X \quad \ldots \quad x_n \text{ in } X}{f(x_1, \ldots, x_n) \text{ in } X} \qquad \frac{y \text{ in } X}{x \text{ in } X}\, P(x,y)$$

où C ensemble, f opération n-aires, P prédicat.

Alors, on est sur qu'il n'y pas d'autres éléments dans X que ceux déduisable par ce règles.

Ceci nous offre aussi un ***principe d'induction*** sur X : il suffit de définir/vérifier tous les cas de ces règles.

Noter: les *relations* et *fonctions* sont aussi des ensembles!

Expressions régulières

2.4.1 Définition (Expressions régulières) Soit Σ un alphabet. Pour chaque $a \in \Sigma$, on se donne un nouveau symbole que l'on note $\mathbf{a}$. On se donne également les symboles $\boldsymbol{\epsilon}$, $\boldsymbol{ø}$, l'opérateur unaire *, les opérateurs binaires $+, \cdot$. On définit alors l'ensemble $\mathbf{RE}_\Sigma$ des *expressions régulières sur* Σ comme étant le plus petit ensemble satisfaisant les règles :

$$\frac{a \in \Sigma}{\mathbf{a} \in \mathbf{RE}_\Sigma} \qquad \frac{}{\boldsymbol{\epsilon} \in \mathbf{RE}_\Sigma} \qquad \frac{}{\boldsymbol{ø} \in \mathbf{RE}_\Sigma}$$

$$\frac{\mathbf{x} \in \mathbf{RE}_\Sigma}{(\mathbf{x}^*) \in \mathbf{RE}_\Sigma} \qquad \frac{\mathbf{x} \in \mathbf{RE}_\Sigma \quad \mathbf{y} \in \mathbf{RE}_\Sigma}{(\mathbf{x} + \mathbf{y}) \in \mathbf{RE}_\Sigma} \qquad \frac{\mathbf{x} \in \mathbf{RE}_\Sigma \quad \mathbf{y} \in \mathbf{RE}_\Sigma}{(\mathbf{x} \cdot \mathbf{y}) \in \mathbf{RE}_\Sigma}$$

Sémantique

2.4.3 Définition (Langage dénoté par une expression régulière) Soit Σ un alphabet. Le *langage* $\mathrm{L}(\mathbf{x})$ *denoté par une expression régulière* $\mathbf{x} \in \mathbf{RE}_\Sigma$ est défini par induction structurelle sur l'expression $\mathbf{x}$ comme suit :

$$\begin{aligned} \mathrm{L}(\mathbf{a}) &\triangleq \{\, a \,\} & \mathrm{L}(\mathbf{x}+\mathbf{y}) &\triangleq \mathrm{L}(\mathbf{x}) \cup \mathrm{L}(\mathbf{y}) \\ \mathrm{L}(\boldsymbol{\epsilon}) &\triangleq \{\, \epsilon \,\} & \mathrm{L}(\mathbf{x}\cdot\mathbf{y}) &\triangleq \mathrm{L}(\mathbf{x})\,\mathrm{L}(\mathbf{y}) \\ \mathrm{L}(\boldsymbol{ø}) &\triangleq \emptyset & \mathrm{L}(\mathbf{x}^*) &\triangleq \mathrm{L}(\mathbf{x})^* \end{aligned}$$

2.4.5 Théorème Soit $\mathbf{x}$ une expression régulière sur un alphabet Σ. Alors, il existe un $\mathrm{AFN}_{\mathbf{e}}$ M sur Σ tel que $\mathrm{L}_{\mathbf{e}}(M) = \mathrm{L}(\mathbf{x})$.

Équivalences et ordres

2.4.6 Définition (Expressions régulières équivalentes et ordonnées) Deux expressions régulières $\mathbf{x}$ et $\mathbf{y}$ sont équivalentes, noté $\mathbf{x} \approx \mathbf{y}$, si les langages qu'elles dénotent sont égaux :

$$\boxed{\mathbf{x} \approx \mathbf{y} \Leftrightarrow \mathrm{L}(\mathbf{x}) = \mathrm{L}(\mathbf{y})}$$

Similairement, $\mathbf{x}$ et $\mathbf{y}$ sont ordonnées, noté $\mathbf{x} \lesssim \mathbf{y}$, si le langage dénoté par $\mathbf{x}$ est inclus dans le langage dénoté par $\mathbf{y}$, i.e.

$$\boxed{\mathbf{x} \lesssim \mathbf{y} \Leftrightarrow \mathrm{L}(\mathbf{x}) \subseteq \mathrm{L}(\mathbf{y})}$$

2.4.7 Lemme Soit $\mathbf{x}$ et $\mathbf{y}$ deux expressions régulières. Alors :

1. $\approx$ est une équivalence.
2. $\mathbf{x} \lesssim \mathbf{y} \Leftrightarrow \mathbf{x} + \mathbf{y} \approx \mathbf{y}$
3. $\mathbf{x} \lesssim \mathbf{y} \wedge \mathbf{y} \lesssim \mathbf{x} \Rightarrow \mathbf{x} \approx \mathbf{y}$

Lois de Kleene (I)

2.4.11 Proposition

$$\mathbf{x} + (\mathbf{y} + \mathbf{z}) \approx (\mathbf{x} + \mathbf{y}) + \mathbf{z} \tag{2.1}$$
$$\mathbf{x} + \mathbf{y} \approx \mathbf{y} + \mathbf{x} \tag{2.2}$$
$$\mathbf{x} + \boldsymbol{\emptyset} \approx \mathbf{x} \tag{2.3}$$
$$\mathbf{x} + \mathbf{x} \approx \mathbf{x} \tag{2.4}$$
$$\mathbf{x}(\mathbf{y}\mathbf{z}) \approx (\mathbf{x}\mathbf{y})\mathbf{z} \tag{2.5}$$
$$\mathbf{x}\boldsymbol{\epsilon} \approx \mathbf{x} \tag{2.6}$$
$$\boldsymbol{\epsilon}\mathbf{x} \approx \mathbf{x} \tag{2.7}$$
$$\mathbf{x}\boldsymbol{\emptyset} \approx \boldsymbol{\emptyset} \tag{2.8}$$
$$\boldsymbol{\emptyset}\mathbf{x} \approx \boldsymbol{\emptyset} \tag{2.9}$$

Lois de Kleene (II)

$$\mathbf{x}(\mathbf{y} + \mathbf{z}) \approx \mathbf{x}\mathbf{y} + \mathbf{x}\mathbf{z} \tag{2.10}$$
$$(\mathbf{x} + \mathbf{y})\mathbf{z} \approx \mathbf{x}\mathbf{z} + \mathbf{y}\mathbf{z} \tag{2.11}$$
$$\boxed{\boldsymbol{\epsilon} + \mathbf{x}\mathbf{x}^* \approx \mathbf{x}^*} \tag{2.12}$$
$$\boldsymbol{\epsilon} + \mathbf{x}^*\mathbf{x} \approx \mathbf{x}^* \tag{2.13}$$
$$\mathbf{y} + \mathbf{x}\mathbf{z} \lesssim \mathbf{z} \Rightarrow \mathbf{x}^*\mathbf{y} \lesssim \mathbf{z} \tag{2.14}$$
$$\mathbf{y} + \mathbf{z}\mathbf{x} \lesssim \mathbf{z} \Rightarrow \mathbf{y}\mathbf{x}^* \lesssim \mathbf{z} \tag{2.15}$$

Contextes

2.4.8 Définition (Contextes des expressions régulières) Soit Σ un alphabet. On définit l'ensemble $\mathbf{CRE}_\Sigma$ des *contextes des expressions régulières sur* Σ comme étant le plus petit ensemble satisfaisant les règles :

$$\frac{}{[\cdot] \in \mathbf{CRE}_\Sigma} \qquad \frac{\mathbf{c} \in \mathbf{CRE}_\Sigma}{(\mathbf{c}^*) \in \mathbf{CRE}_\Sigma}$$

$$\frac{\mathbf{x} \in \mathbf{RE}_\Sigma \quad \mathbf{c} \in \mathbf{CRE}_\Sigma}{(\mathbf{x} + \mathbf{c}) \in \mathbf{CRE}_\Sigma} \qquad \frac{\mathbf{x} \in \mathbf{RE}_\Sigma \quad \mathbf{c} \in \mathbf{CRE}_\Sigma}{(\mathbf{c} + \mathbf{x}) \in \mathbf{CRE}_\Sigma}$$

$$\frac{\mathbf{x} \in \mathbf{CRE}_\Sigma \quad \mathbf{c} \in \mathbf{CRE}_\Sigma}{(\mathbf{x} \cdot \mathbf{c}) \in \mathbf{CRE}_\Sigma} \qquad \frac{\mathbf{x} \in \mathbf{CRE}_\Sigma \quad \mathbf{c} \in \mathbf{CRE}_\Sigma}{(\mathbf{c} \cdot \mathbf{x}) \in \mathbf{CRE}_\Sigma}$$

$[\cdot]$ est appelé le *trou* d'un contexte. Si $\mathbf{c} \in \mathbf{CRE}_\Sigma$ et $\mathbf{x} \in \mathbf{RE}_\Sigma$, alors $\mathbf{c}[\mathbf{x}]$ denote l'expression régulière que l'on obtient comme résultat de l'*application* de $\mathbf{c}$ à $\mathbf{x}$, par laquelle le trou est remplacée par l'expression $\mathbf{x}$.

Congruences

2.4.9 Définition (Congruence) Soit R une équivalence sur $\mathbf{RE}_\Sigma$. R est une *congruence* si elle est préservée par tous les contextes $\mathbf{c} \in \mathbf{CRE}_\Sigma$, donc, si elle satisfait la régle :

$$\frac{\mathbf{x}\ R\ \mathbf{y} \qquad \mathbf{c} \in \mathbf{CRE}_\Sigma}{\mathbf{c}[\mathbf{x}]\ R\ \mathbf{c}[\mathbf{y}]}$$

2.4.10 Lemme La relation $\approx$ est une congruence.

C'est seulement par le Lemme 2.4.10,
que nous avons le droit de remplacer
un sous-terme x d'une expression régulière c[x]
par un autre terme x' égal à x
étant sur que le langage du terme ne change pas.

Langages des états d'un AFD

2.4.12 Définition Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD.
Pour tout $q \in Q$, on définit le langage L_q par :

$$\boxed{L_q \triangleq \mathrm{L}(\,(Q, \Sigma, \delta, q, F)\,)}$$

Cas spécifique:
L(M) =L_s

2.4.13 Lemme Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD. Alors, pour tout $q \in Q$:

$$L_q = \boxed{\left(\bigcup_{a\in\Sigma} \{a\} \cdot L_{\delta(q,a)}\right)} \cup \boxed{\{\epsilon \mid q \in F\}} \qquad \text{(LIN)}$$

2.4.14 Lemme (Arden) Soit Σ un alphabet. Soit $X, A, B \subseteq \Sigma^*$ des langages sur Σ tels que $\epsilon \notin A$ et satisfaisant l'équation :

$$\boxed{X = AX + B}$$

Alors $\boxed{X = A^*B}$.

Conversion AFD > RE

2.4.15 Définition (Conversion de AFD en RE) Soit $(Q, \Sigma, \delta, s, F)$.

1. Établir un système de $\#(Q)$ équations linéaires (du format (LIN)) qui définissent les « variables » L_q correspondants aux états $q \in Q$.
2. Par abus de notation, nous écrivons $\mathbf{a}$ à la place de $\{a\}$ et $+$ à la place de $\cup$; ceci est justifié par Définition 2.4.3.
3. Résoudre ce système de manière récursive par élimination des variables (sauf L_s) l'une après l'autre en utilisant le Lemme 2.4.14 d'Arden et la Proposition 2.4.11.
4. On résout ce système afin de trouver une expression régulière pour L_s.

2.4.16 Théorème Soit M un AFD et $\mathbf{x}$ le résultat du système d'équations de la Définition 2.4.15 appliqué à M. Alors, $\mathrm{L}(\mathbf{x}) = \mathrm{L}(M)$.

AFNG

2.4.17 Définition (AFNG) Un *automate fini non déterministe généralisé* (AFNG) est un quintuplet

$$M = (Q, \Sigma, \mathbf{\Delta}, s, f)$$

où

- Q est un ensemble fini d'*états*,
- Σ est un ensemble fini de symboles, *l'alphabet (d'entrée)*,
- $\boxed{\mathbf{\Delta} : (Q \setminus \{f\}) \times (Q \setminus \{s\}) \to \mathbf{RE}_\Sigma}$ est la *fonction (totale) de transition*,
- $s \in Q$ est l'*état initial* (ou de *départ*),
- $f \in Q$ est l'*état accepteur* (ou *final*).

Noter:

(presque) toute paire d'états est lié par une flèche
s et *f* sont différents dans ce sens

Langage d'un AFNG

2.4.18 Définition Soit $M = (Q, \Sigma, \boldsymbol{\Delta}, s, f)$ un AFNG.

1. M *accepte* un mot $w \in \Sigma^*$
   s'il existe $w_1, \ldots, w_k \in \Sigma^*$ et $q_0, \ldots, q_k \in Q$ tels que
   (a) $w = w_1 \cdot \ldots \cdot w_k$
   (b) $q_0 = s$ et $q_k = f$
   (c) $\forall i \in [1, k] : w_i \in \mathrm{L}(\boldsymbol{\Delta}(q_{i-1}, q_i))$
2. Le *langage accepté* par M est défini par :

$$\mathrm{L}(M) \triangleq \{\, w \in \Sigma^* \mid w \text{ accepté par } M \,\}$$

Conversion AFD > AFNG

2.4.19 Définition $\quad \mathrm{G} : \quad \text{AFD} \to \text{AFNG}$

$$(Q, \Sigma, \delta, s, F) \mapsto (Q_{\mathrm{G}}, \Sigma, \boldsymbol{\Delta}, s_{\mathrm{G}}, f_{\mathrm{G}})$$

avec $Q_{\mathrm{G}} \triangleq Q \cup \{\, s_{\mathrm{G}}, f_{\mathrm{G}} \,\}$ où $Q \cap \{\, s_{\mathrm{G}}, f_{\mathrm{G}} \,\} = \emptyset$ et $\boldsymbol{\Delta}$ telle que

$$\frac{}{\boldsymbol{\Delta}(s_{\mathrm{G}}, s) = \boldsymbol{\epsilon}} \qquad \frac{q \in Q \setminus \{s\} \cup \{\, f_{\mathrm{G}} \,\}}{\boldsymbol{\Delta}(s_{\mathrm{G}}, q) = \boldsymbol{\varnothing}}$$

$$\frac{q \in F}{\boldsymbol{\Delta}(q, f_{\mathrm{G}}) = \boldsymbol{\epsilon}} \qquad \frac{q \in Q \setminus F}{\boldsymbol{\Delta}(q, f_{\mathrm{G}}) = \boldsymbol{\varnothing}}$$

$$\frac{q, q' \in Q \qquad \{\, a \mid \delta(q, a) = q' \,\} = \{\, a_1 \ldots, a_n \,\} \neq \emptyset}{\boldsymbol{\Delta}(q, q') = \mathbf{a}_1 + \cdots \mathbf{a}_n}$$

$$\frac{q, q' \in Q \qquad \{\, a \mid \delta(q, a) = q' \,\} = \emptyset}{\boldsymbol{\Delta}(q, q') = \boldsymbol{\varnothing}}$$

Élimination des états d'un AFNG

2.4.20 Théorème Soit M un AFD. Alors $\mathrm{L}(M) = \mathrm{L}(\mathrm{G}(M))$

2.4.21 Définition (Réduction de AFNG)
Soit $M = (Q, \Sigma, \boldsymbol{\Delta}, s, f)$ un AFNG et $\widehat{q} \in Q \setminus \{\, s, f \,\}$
La *réduction* $\mathrm{R}(M, \widehat{q})$ de M par $\widehat{q}$ est définie par

$$\mathrm{R}(M, \widehat{q}) \triangleq (Q \setminus \{\, \widehat{q} \,\}, \Sigma, \boldsymbol{\Delta}', s, f)$$

avec $\boldsymbol{\Delta}'$ étant définie par la règle :

$$\frac{q \in Q \setminus \{\, f \,\} \qquad q' \in Q \setminus \{\, s \,\}}{\boldsymbol{\Delta}'(q, q') = \boldsymbol{\Delta}(q, q') + \boldsymbol{\Delta}(q, \widehat{q}) \cdot (\boldsymbol{\Delta}(\widehat{q}, \widehat{q}))^* \cdot \boldsymbol{\Delta}(\widehat{q}, q')}$$

2.4.22 Lemme Soit M un AFNG. Alors $\mathrm{L}(M) = \mathrm{L}(\mathrm{R}(M))$

Conversion AFD > RE

2.4.23 Définition (Algorithme de conversion de AFD en RE)

reduce$(\,(Q, \Sigma, \boldsymbol{\Delta}, s, f)\,)$:=
  si $\#(Q) > 2$
    choisir $q \in Q \setminus \{s, f\}$
    retourner *reduce*$(\,\mathrm{R}(\,(Q, \Sigma, \boldsymbol{\Delta}, s, f), q\,)\,)$
  sinon
    retourner $(Q, \Sigma, \boldsymbol{\Delta}, s, f)$

afd2re$(Q, \Sigma, \delta, s, F)$:=
  calculer $(Q', \Sigma, \boldsymbol{\Delta}, s', f')$=*reduce*$(\,\mathrm{G}((Q, \Sigma, \delta, s, F))\,)$
  retourner $\boldsymbol{\Delta}(s', f')$

2.4.24 Théorème (Correction de l'algorithme de conversion) Soit M un AFD et $\mathbf{x}$ le résultat de l'exécution de *afd2re*(M). Alors $\mathrm{L}(M) = \mathrm{L}(\mathbf{x})$