Fiches d'info

Samuel MIMRAM

2001-2002

Table des matières

Caml

`assoc : 'a -> ('a * 'b) list -> 'b = <fun>` renvoie le premier b tq (a, b) figure ds la liste
`rev : ('a list) -> ('a list) = <fun>` renvoie le mirroir d'une liste

Ne pas oublier les `!` pour les `ref`, les `:=`, etc...

1 Automates

Automate : $\mathcal{A} = (E, X, \delta, e_0, F)$:
– E : ensemble des états
– X : alphabet
– δ : fonction de transition : d'une partie de $E \times X \to X$
– e_0 : état initial
– F : ensemble des états finaux ($\subset E$)

Automate complet : $\delta : E \times X \to X$
Loi de composition externe ($\delta(e, x) = e \cdot x$) : $e \boxdot \varepsilon = e$ $\quad e \boxdot a = e \cdot a$ $\quad$ par réc : $e \boxdot (wa) = (e \boxdot w) \boxdot a$
Alors on a (par réc sur $|w_2|$) : $e \boxdot (w_1 w_2) = (e \boxdot w_1) \boxdot w_2$

Langage régulier (ou rationel) : il existe un automate qui le reconnaisse
Si un langage est régulier, il existe un automate complet qui le reconnaisse (rajouter un puits)
Si L est un langage régulier, il existe un automate accessible le reconnaissant
Si L est régulier alors L^c et mirroir(L) aussi
Si L_1 et L_2 sont réguliers alors $L_1 \cap L_2$ et $L_1 \cup L_2$ aussi
Si L et M sont réguliers alors $L \cdot M$ et L^* aussi
Si L est régulier et f est un morphisme de monoïdes alors $f(L)$ et $f^{<-1>}(L)$ sont aussi réguliers

Pumping lemma (lemme de l'étoile) : L langage régulier alors
$\exists n_0 \in \mathbb{N}^*, \forall w \in L, |w| \geq n_0 \Rightarrow \exists (x, y, z) \in (X^*)^3,$

1. $w = xyz$
2. $|y| \geq 1$
3. $|xy| \leq n_0$

et $\forall n \in \mathbb{N}, xy^n z \in L$

Ne sont pas réguliers : $a^n b^n$, $\{w \ / \ |w|_a = |w|_b\}$, $\{a^{(n^2)}\}_{n \in \mathbb{N}}$, $\{w \ / \ |w|$ est un carré$\}$, $\{a^p \ / \ p$ premier$\}$

Résiduel de u_0 : $R_{u_0} = \{v \ / \ u_0 v \in L\}$
Automate des résiduels : $e_0 = R_\varepsilon$, $R_u \cdot a = R_{ua}$ (on montre que $R_u = R_v \Rightarrow R_{ua} = R_{va}$), $F = \{R_w \ / \ w \in L\}$
Théorème de Myhill-Nerod :
– un langage est régulier ssi il admet un nb fini de résiduels
– il y a unicité (à un isom d'automates près) de l'automate minimal : l'automate des résiduels
  (avec $\phi : e \mapsto R_u$ tq $e_0 \cdot u = e$: surjective)

AFND : δ : d'une partie de $E \times X \to \mathcal{P}(E)$
Détermination : $E' = \mathcal{P}(E)$ et $U \cdot \varepsilon = U$, $U \cdot a = \bigcup_{u \in U} \delta(u, a)$

AFND$_\eta$: avec η-transitions
Détermination : E' = ensemble des clôtures, $G \cdot \varepsilon = \overline{G}$, $G \cdot a = \overline{G \cdot a}$

Minimal : aucun plus petit minimum : plus petit que les autres
Assez bon ordre $\leq$: $\forall P \neq \emptyset$, P admet un élément minimal bon ordre : P admet un élément minimum

Récurrence généralisée sur $(E, \leq)$ avec $\leq$ ABO (induction structurelle) : $\mathcal{P}$ prédicat ($E \to \{true, false\}$) ; si
– $\mathcal{P}$ est vraie sur les minimaux
– $[\forall x \leq y, \mathcal{P}(x)] \Rightarrow \mathcal{P}(y)$
alors $\mathcal{P}$ est vraie sur E

$\leq$ est un ABO $\Leftrightarrow$ il n'existe pas de suite infinie strictement décroissante

Soit E un ensb, $B \subset E$ (base) et $\mathcal{F}_0 \subset \mathcal{F}$; on appelle partie définie inductivement par B et $\mathcal{F}_0$ la plus petite partie F de E qui vérifie :
– $B \subset F$
– $\forall f \in \mathcal{F}_0, \forall (x_1, \ldots, x_{a(f)}) \in F^{a(f)}, f(x_1, \ldots, x_{a(f)}) \in F$

Si $B_0 = B$ et $B_{k+1} = B_k \cup \bigcup_{f \in \mathcal{F}_0} f(B_k^{a(f)})$ alors $F = \bigcup_{k \leq 0} B_k$
Si $\mathcal{P}$ est vrai sur la base B et $\forall f \in \mathcal{F}_0, \forall (x_1, \ldots, x_n)$ vérifiant $\mathcal{P}$ $f(x_1, \ldots, x_n)$ vérifie $\mathcal{P}$ alors $\mathcal{P}$ est vrai sur tt l'ensb défini inductivement par $(B, \mathcal{F}_0)$

Le langage associé à un expression régulière est régulier (démo : par induction structurelle)
Lemme d'Arden : si A et B st langages tq $\varepsilon \notin A$ l'éq $X = AX + B$ admet pour unique sol A^*B
Th de Kleene : L est régulier $\Leftrightarrow$ L est associé à une expression régulière

2 Arbres

Arbre binaire : `type 'a bintree = Nil | BinNode of 'a bintree * 'a * 'a bintree;;`
Arbre général : `type 'a gentree = GenNode of 'a * 'a gentree list;;`

`map : ('a list -> 'b list) -> 'a list -> 'b list = <fun>` retourne la liste $[f(x_i)]$ $O(n)$
`it_list : ('a -> 'b -> 'a) -> 'a -> 'b list -> 'a = <fun>` itère : $f(f(x_{i-1}), x_i)$ att : paramètre initial
`do_list : ('a -> unit) -> 'a list -> unit = <fun>` applique une procédure aux x_i
`flat_map : ('a -> 'b list) -> 'a list -> 'b list = <fun>` retourne la liste des éléments des listes retournées par les $f(x_i)$ $O(\sum_{i=1}^n |l_i|)$
concaténation : $O(n)$ n : lg de la liste de gauche

Parcours en profondeur :

```
let rec depth_of_gentree (GenNode(a,l)) = a::(flat_map depth_of_gentree l);;
```

Parcours en largeur :

```
let rec breadth_of_forest = fun
    | [] -> []
    | l  -> (map gentree_root l)@(map breadth_of_forest (flat_map gentree_sons l))
and breadth_of_gentree t = breadth_of_forest [t];;
```

Bijection entre les bintree et les forest ("fils gauche – frère droit") :

```
let rec bintree_of_forest = fun
    | [] -> Nil
    | (GenNode(a, l)::lt) -> BinNode(bintree_of_forest l, a, bintree_of_forest lt);;
let rec forest_of_bintree = fun
    | Nil -> []
    | BinNode(l, a, r) -> GenNode(a, forest_of_bintree l)::(forest_of_bintree r);;
```

.
Pile LIFO (stack) : `let sk=new()` `push x sk` `pop x sk`

Langage algébrique :
– les non terminaux : A. . . Z

– les terminaux : a. . . z, +, -, *, /, (,)
– des règles : NT → $(N \cup T)^*$

Mots bien parenthèsés : $S \rightarrow (S)S|\varepsilon$
Mini-arithmétique (grammaire non ambiguë) : $S \rightarrow T + S|T \qquad T \rightarrow U * T|U \qquad U \rightarrow a \ldots z|(S)$
Si L_1 et L_2 st algébriques alors $L_1 \cup L_2$, L_1^* et $\widetilde{L_1}$ st algébriques
Rationnel ⇒ algébrique

3 Arbres binaires

```
type 'a bintree = Nil | BinNode of 'a * 'a bintree * 'a;;
```

Longueur moyenne des chemins : $l = \frac{\sum \text{profondeur des nœuds}}{\text{nb nœuds}}$
arbre homogène de hauteur h : $l = \frac{(h-2)2^{h-1}+1}{2^h-1}$
Nb de Catalan C_n : nb d'arbres binaires à n nœuds : $C_{n+1} = \sum_{k=0}^{n} C_k C_{n-k}$ considérer la stg $C_n x^n$ $\quad C_n = \frac{C_{2n}^n}{n+1}$
$C_n \sim \frac{4^n}{n^{3/2}\sqrt{\pi}}$ hauteur moyenne : $h_n \sim \sqrt{n\pi}$ lg moyenne : $l_n \sim \sqrt{n\pi}$