Automates d'arbres

X-2023

Q1 $\Sigma = \{0,1\}$. $Q = \underset{\{V,F\}}{\mathbb{B}}$ $I = \{F\}$ $F = \{V\}$

$$\delta = \left(\begin{array}{lcl} \Sigma \times Q \times Q & \to & \mathcal{P}(Q) \\ (1, x, y) & \mapsto & \{V\} \\ (0, x, y) & \mapsto & \{x+y\} \end{array}\right)$$

$\mathcal{A} = (\Sigma, Q, I, F, \delta)$ reconnait l'ensemble des arbres ayant au moins une feuille étiquetée par 1.

Bonus

NB : $\mathcal{A}$ est un automate déterministe.
On peut alors voir δ comme f° $\Sigma \times Q \times Q \to Q$.

On note alors δ^* la f° étendue aux arbres.
Elle est définie par induc° comme suit.

$$\delta^* = \left(\begin{array}{lcl} \mathcal{A}_{BNV}(\Sigma) & \longrightarrow & Q \\ \square & \longmapsto & F = q_{init} \\ \overset{(a)}{g \quad d} & \longmapsto & \delta(a, \delta^*(g), \delta^*(d)) \end{array}\right)$$

MQ $\forall T \in \mathcal{A}_{BNV}(\Sigma)$ $\underset{\text{ie } \in F}{\delta^*(T) = V}$ ssi T présente au - une étiquette 1.
par induc° sur $\mathcal{A}_{BNV}(\Sigma)$

- Si $T = \square$, on a $\delta^*(T) = F$ et T ne présente aucune étiquette donc la pté est vraie pour T.

- Si $T = \overset{(a)}{g \quad d}$ où g et d vérifient cette propriété.
$a \in \Sigma = \{0,1\}$

$$\delta^*(T) = V \overset{\text{déf de } \delta^*}{\Leftrightarrow} \delta(a, \delta^*(g), \delta^*(d))$$

$$\overset{\text{déf de } \delta}{\Leftrightarrow} \begin{cases} V & \text{si } a = 1 \\ \delta^*(g) + \delta^*(d) & \text{si } a \neq 1 \end{cases}$$

$$\Leftrightarrow \begin{cases} V & \text{si la racine de T a une étiquette 1} \\ V & \text{si } \delta^*(g) = V \text{ ou } \delta^*(d) = V \\ F & \text{sinon} \end{cases}$$

$\delta^*(g) = V \Updownarrow$ par HI g a au - une étiquette 1 ; $\delta^*(d) = V \Updownarrow$ d a au - 1 étiquette 1

$$\Leftrightarrow \begin{cases} V & \text{si T présente au - 1 étiquette 1} \\ F & \text{sinon} \end{cases}$$

Dans la suite, on note $\mathcal{A}_{BNV}(\Sigma)$ l'ensemble des arbres binaires non vides | aux feuilles non étiquetées
| aux nds. internes étiquetés par Σ.

= l'ensemble construit par induc° par $\square|^0_-$, $\bigcirc|^2_\Sigma$

Q2 Pour déterminiser un automate d'arbre montant $\mathcal{A}$ on définit l'automate des parties associés.

$$\hat{\mathcal{A}} = (\Sigma, \hat{Q}, \hat{I}, \hat{F}, \hat{\delta}) \text{ où } \begin{array}{l} \hat{Q} = \mathcal{P}(Q) \\ \hat{I} = \{I\} \\ \hat{F} = \{X \in \hat{Q} \mid X \cap F \neq \emptyset\} \end{array}$$

$$\text{et } \hat{\delta} = \left(\begin{array}{lcl} \Sigma \times \hat{Q} \times \hat{Q} & \to & \mathcal{P}(Q) \\ (a, \hat{q_1}, \hat{q_2}) & \mapsto & \bigcup_{(q_1,q_2) \in \hat{q_1} \times \hat{q_2}} \delta(a, q_1, q_2) \end{array} \right)$$

- $\hat{\mathcal{A}}$ est déterministe, par construction.
- Soit $T \in \mathcal{A}_{BNV}(\Sigma)$ un arbre.
On note $\hat{e}$ l'unique execution possible de $\hat{\mathcal{A}}$ sur T.
Elle peut ê définie par $\hat{e} = \left(\begin{array}{l} \mathcal{N}d(T) \to \hat{Q} \\ n \mapsto \begin{cases} I & \text{si } n \text{ est une feuille de } T \\ \hat{\delta}(a, \hat{e}(n.g), \hat{e}(n.d)) & \text{si } n \text{ est un nœud d'étiquette } a \end{cases} \end{array} \right)$

- <u>MQ $\mathcal{A}$ accepte $T \Rightarrow \hat{\mathcal{A}}$ accepte T.</u>

Supposons que $\mathcal{A}$ accepte T. Notons alors e une execution acceptante de $\mathcal{A}$ sur T.

On peut alors montrer par "induc° finie" $\boxed{\forall n \in \mathcal{N}d(T), e(n) \in \hat{e}(n)}$

- Si $n \in \mathcal{N}d(T)$ est une feuille
alors $e(n) \in I$ par déf de e une execution
or $\hat{e}(n) = I$ par def de $\hat{e}$, d'où $e(n) \in \hat{e}(n)$.

- Si n est un noeud interne d'étiquette a, et de fils qui vérifient déjà la pté, on a

$e(n) \in \delta(a, e(n.g), e(n.d))$ par déf de e une exec.

or $e(n.g) \in \hat{e}(n.g)$ et $e(n.d) \in \hat{e}(n.d)$ par hyp,

donc $e(n) \in \bigcup_{(q_1,q_2) \in \hat{e}(n.g) \times \hat{e}(n.d)} \delta(a, q_1, q_2) \underset{\text{def de } \hat{\delta}}{=} \hat{\delta}(a, \hat{e}(n.g), \hat{e}(n.d)) \underset{\text{def de } \hat{e}}{=} \hat{e}(n)$

Donc $\forall n \in \mathcal{N}d(T), e(n) \in \hat{e}(n)$.

En particulier pour $n = rac(T)$ on a $e(rac(T)) \in \hat{e}(rac(T))$.
Or puisque e est acceptance, $e(rac(T)) \in F$, donc $\hat{e}(rac(T)) \in \hat{F}$ par déf de $\hat{F}$, ce qui fait de $\hat{e}$ un exéc. acceptante. Donc $\hat{\mathcal{A}}$ accepte T.

- MQ $\hat{A}$ accepte T $\Rightarrow$ A accepte T

Supposons cette fois que $\hat{A}$ accepte T, Nééc $\hat{e}$ est une exéc. acceptante, donc $\hat{e}(\mathrm{rac}(T)) \in \hat{F}$.

Par déf de $\hat{F}$, il existe donc $q_f \in F$ tq $q_f \in \hat{e}(\mathrm{rac}(T))$

On peut ... montrer par induc° finie $\boxed{\forall n \in Nd(T), \forall q \in \hat{e}(n), \exists e \text{ exécu° de A sur* tq } e(\varepsilon) = q}$

* le ss-arbre issus de n

- Si n est une feuille de T, $\hat{e}(n) \in \hat{I}$, or $\hat{I} = \{I\}$ donc néc $\hat{e}(n) = I$, et pour tout $q \in I$ on peut bien construire e exec de A sur $\boxed{\ }_{\hat{}n}$ tq $e(\underset{\varepsilon \text{ racine } = n}{\varepsilon}) = q$.
- Si n est un noeud interne d'étiquette a et dont les fils vérifient la pté, on a $\hat{e}(n) = \hat{\delta}(a, \hat{e}(n.g), \hat{e}(n.d))$

Si $q \in \hat{e}(n)$, par déf de $\hat{\delta}$, il existe $(q_1, q_2) \in \hat{e}(n.g) \times \hat{e}(n.d)$ tq $q \in \delta(a, q_1, q_2)$.

Par hyp. d'induc°, il existe e_1, e_2 définie sur le ss-arbre issus de n.g S_g, S_d telles que $e_1(\varepsilon) = q_1$, $e_2(\varepsilon) = q_2$. On définit alors e sur le ss-arbre S issus de n en fusionnant e_1, e_2 et q à la racine.

On a $Nd(S) = \varepsilon \cup g.Nd(S_g) \cup d.Nd(S_d)$ donc formellement

$$e = \begin{pmatrix} \varepsilon \mapsto q \\ g.n_1 \mapsto e_1(n_1) \\ d.n_2 \mapsto e_2(n_2) \end{pmatrix}$$

C'est bien une exécution car $e(\varepsilon) = q \in \delta(a, \underset{\substack{e_1(\varepsilon) \\ \| \\ e(g)}}{\overset{}{q_1}}, \underset{\substack{e_2(\varepsilon) \\ \| \\ e(d)}}{q_2})$

d'où la pté pour n.

Ainsi on a, par induction finie, qu'il existe en particulier e une exéc de A sur T (= l'arbre issus de racine de T) telle que $e(\varepsilon) = q_f$, autrement dit tq $e(\mathrm{rac}(T)) = q_f$, et comme $q_f \in F$, on en déduit que celle-ci est acceptante.

Ainsi A accepte T.

Q3 Soit $\mathcal{A}$ un automate montant, disons $\mathcal{A} = (\Sigma, Q, I, F, \mu)$.
On considère l'automate descendant $\mathcal{D} = (\Sigma, Q, \hat{I}, \hat{F}, \hat{\delta})$
défini par $\hat{I} = F$ et $\hat{F} = I$ et $\boxed{\hat{\delta} = \begin{pmatrix} Q \times \Sigma \to \mathcal{P}(Q \times Q) \\ (q, a) \mapsto \{(q_1, q_2) \in Q \times Q \mid q \in \mu(q_1, q_2, a)\} \end{pmatrix}}$

Montrons que $\mathcal{D}$ et $\mathcal{A}$ sont équivalents.
Soit $T \in \mathcal{A}_{B}^{NV}(\Sigma)$. Soit $e \in \mathcal{F}(\mathcal{N}d(T), Q)$

a) $e(rac(T)) \in F \iff e(rac(T)) \in \hat{I}$ car $\hat{I} = F$

b) $\forall x \in feuille(T)$, $e(x) \in I \Rightarrow e(x) \in \hat{F}$ car $\hat{F} = I$.

c) $\forall x$ nd interne de T d'étiquette a,
$e(x) \in \mu(a, e(x.0), e(x.1))$
$\iff (e(x), e(x.1)) \in \hat{\delta}(e(x), a)$ par déf de $\hat{\delta}$

Avec a), b) et c) on a donc e exéc. accept de $\mathcal{A}$ sur T
$\iff e$ exéc. accept. de $\mathcal{D}$ sur T

D'où $\underline{\mathcal{A} \text{ accepte } T \text{ ssi } \mathcal{D} \text{ accepte } T}$ □

Q4 Soit $\mathcal{D}$ un automate descendant, disons $\mathcal{D} = (\Sigma, Q, I, F, \delta)$.
On considère alors l'automate montant $\mathcal{A} = (\Sigma, Q, \hat{I}, \hat{F}, \hat{\mu})$
défini par $\hat{I} = F$, $\hat{F} = I$ $\boxed{\hat{\mu} = \begin{pmatrix} Q \times Q \times \Sigma \to \mathcal{P}(Q) \\ (q_1, q_2, a) \mapsto \{q \in Q \mid \delta(q, a) \ni (q_1, q_2)\} \end{pmatrix}}$

Montrons que $\mathcal{D}$ et $\mathcal{A}$ sont équivalents.
Soit $T \in \mathcal{A}_{B}^{NV}(\Sigma)$. Soit $e \in \mathcal{F}(\mathcal{N}d(T), Q)$.

☆ a) $e(rac(T)) \in F \iff e(rac(T)) \in \hat{I}$ car $\hat{I} = F$

b) $\forall x \in feuilles(T)$, $e(x) \in I \iff e(x) \in \hat{F}$ car $\hat{F} = I$

c) $\forall x$ nd. interne de T d'étiquette a,
$(e(x.0), e(x.1)) \in \delta(a, e(x))$
$\iff e(x) \in \hat{\mu}(a, e(x.0), e(x.1))$ par déf de $\hat{\mu}$

Avec a, b et c, on a e exécution acceptante de $\mathcal{D}$ sur $T \iff$
e ——— $\mathcal{A}$ sur T

D'où $\underline{\mathcal{D} \text{ accepte } T \iff \mathcal{A} \text{ accepte } T}$ □

CCL Un automate descendant est équiv à un automate montant, lui m̂ équiv à un automate montant déterministe !

Q5

Montrons qu'un automate descendant n'est pas néc. équivalent à un automate descendant déterministe.

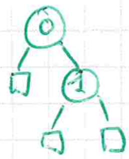
Autr. dit, on ne peut pas toujours déterminiser un automate descendant.

Considérons par exemple l'automate desc. suivant qui reconnaît les arbres ayant au moins un 1.
↖ étiquetés par $\{0,1\}$.

$A = (\Sigma, Q, I, F, \delta)$ avec $Q = \mathbb{B}$, $I = \{V\}$, $F = \{F\}$ (faux)

$$\delta = \left(\begin{array}{l} \Sigma \times Q \to \mathcal{P}(Q \times Q) \\ (1, V) \mapsto \{(V,F), (V,V), (F,F), (F,V)\} \\ (0, V) \mapsto \{(V,F), (V,V), (F,V)\} \\ (0, F) \mapsto \{(F,F)\} \\ (1, F) \mapsto \emptyset \end{array} \right)$$

On suppose qu'il existe $\hat{A} = (\Sigma, \hat{Q}, \hat{I}, \hat{F}, \hat{\delta})$ un automate desc. déterministe équivalent à A. Notons q_{init} l'unique état de $\hat{I}$.

$\hat{A}$ accepte les arbres T =  et T' = 

Notons e (resp e') l'exécution de $\hat{A}$ sur T (resp T').
On a néc $e(\varepsilon) = q_{init}$ $(e'(\varepsilon) = q_{init})$.
En notant de plus $q_g = e(0)$ (= fils gauche de la racine) et $q_d = e(1)$ (fils droit de la racine) $\left(\begin{array}{l} q_g' = e'(0) \\ q_d' = e'(1) \end{array} \right)$

on a aussi $(q_g, q_d) = \delta(0, e(\varepsilon)) = \delta(0, q_{init}) = \delta(0, e'(\varepsilon)) = (q_g', q_d')$.

Enfin e (resp e') étant acceptante on a $\begin{cases} q_d \in F & \text{(car = el feuille)} \\ q_g' \in F & \text{(idem ...)} \end{cases}$

On considère alors l'exécution f de $\hat{A}$ sur T'' = 

On a néc $f(\varepsilon) = q_{init}$

$$(f(0), f(1)) = \delta(0, q_0) = (q_d, q_g') \in F \times F$$

Ainsi les 2 feuilles de T'' sont associées à un état final, donc T'' est accepté par $\hat{A}$. ABS car T'' n'a pas au moins une étiquette 1.