

Rappel: Grammaires

Définition : $G = (\Sigma, V, \rightarrow, S)$ où

- ▶ Σ alphabet *terminal* (fini)
- ▶ V *variables* (ensemble fini)
- ▶ $\rightarrow \subseteq (\Sigma \cup V)^* \times (\Sigma \cup V)^*$ ensemble fini de *productions*
- ▶ S variable initiale

Grammaire de type 2 (*hors contexte* ou *algébrique*) : $\rightarrow \subseteq V \times (\Sigma \cup V)^*$

Exemple: $S \rightarrow aSb \mid \varepsilon$ engendre $\{a^n b^n \mid n \geq 0\}$

Intérêt des langages algébriques :

- ▶ spécification de langages au-delà des reconnaissables, analyse syntaxique
- ▶ correspondent aux *automates à pile*

Automates à pile

Définition : $\mathcal{A} = (Q, \Sigma, Z, T, q_0z_0, F)$ où

- ▶ Q ensemble fini d'états
- ▶ Σ alphabet d'entrée
- ▶ Z alphabet de pile
- ▶ $T \subseteq QZ \times (\Sigma \cup \{\varepsilon\}) \times QZ^*$ ensemble fini de transitions
- ▶ $q_0z_0 \in QZ$ configuration initiale
- ▶ $F \subseteq Q$ acceptation par état final.

De plus, \mathcal{A} est *temps-réel* s'il n'a pas d' ε -transition.

Définition : Système de transitions (infini) associé

- ▶ $\mathcal{T} = (QZ^*, T', q_0z_0, FZ^*)$
- ▶ Une configuration de \mathcal{A} est un état $ph \in QZ^*$ de \mathcal{T}
- ▶ Transitions de \mathcal{T} : $T' = \{pzh \xrightarrow{a} qgh \mid (pz, a, qg) \in T\}$.
- ▶ $\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^* \mid \exists q_0z_0 \xrightarrow{w} qh \in FZ^* \text{ dans } \mathcal{T}\}$.

Automates à pile

Exemples :

- ▶ $L_1 = \{a^n b^n c^p \mid n, p > 0\}$ et $L_2 = \{a^n b^p c^p \mid n, p > 0\}$
- ▶ $L = L_1 \cup L_2$ (non déterministe)

Exercices :

1. Montrer que le langage $\{w\tilde{w} \mid w \in \Sigma^*\}$ et son complémentaire peuvent être acceptés par un automate à pile.
2. Montrer que le complémentaire du langage $\{ww \mid w \in \Sigma^*\}$ peut être accepté par un automate à pile.
3. Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0 z_0, F)$ un automate à pile. Montrer qu'on peut construire un automate à pile équivalent \mathcal{A}' tel que $T' \subseteq Q'Z \times (\Sigma \cup \{\varepsilon\}) \times Q'Z^{\leq 2}$.

Propriétés fondamentales

Lemme : fondamental

Soient $\mathcal{A} = (Q, \Sigma, Z, T, q_0 z_0)$ un automate à pile, $p, r \in Q$, $g, h \in Z^*$, $w \in \Sigma^*$ et $n \geq 0$. Les conditions suivantes sont équivalentes:

1. $pgh \xrightarrow[n]{w} r$ est un calcul de \mathcal{A} ,
2. il existe deux calculs $pg \xrightarrow[n_1]{w_1} q$ et $qh \xrightarrow[n_2]{w_2} r$ de \mathcal{A} avec $q \in Q$, $w = w_1 w_2$ et $n = n_1 + n_2$.

Preuve

\implies : Dans le calcul $pgh \xrightarrow[n]{w} r$, on considère **la première fois** que le contenu de la pile est h (possible car initialement gh , finalement ε). Soit qh la configuration correspondante après n_1 étapes. Le calcul s'écrit $pgh \xrightarrow[n_1]{w_1} qh \xrightarrow[n_2]{w_2} r$.

Si $p_0 g_0 h \xrightarrow{a_1} p_1 g_1 h \cdots \xrightarrow{a_k} p_k g_k h$ est un calcul de \mathcal{A} tel que $g_i \neq \varepsilon$ pour $0 \leq i < n$ alors $p_0 g_0 \xrightarrow{a_1} p_1 g_1 \cdots \xrightarrow{a_k} p_k g_k$ est un calcul de \mathcal{A} .

\impliedby : On utilise la remarque suivante: Si $sf \xrightarrow[k]{v} s'f'$ est un calcul de \mathcal{A} avec $s \in Q$, $f, f', f'' \in Z^*$ alors $sf f'' \xrightarrow[k]{v} s'f' f''$ est aussi un calcul de \mathcal{A} .

Acceptation généralisée

Définition :

Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0 z_0)$ un automate à pile et $K \subseteq QZ^*$ un langage reconnaissable. Le langage reconnu par \mathcal{A} avec *acceptation généralisée* K est

$$\mathcal{L}_K(\mathcal{A}) = \{w \in \Sigma^* \mid \exists q_0 z_0 \xrightarrow{w} qh \in K \text{ dans } \mathcal{T}\}$$

Cas particuliers :

- ▶ $K = FZ^*$: acceptation classique **par état final**.
- ▶ $K = Q$: acceptation **par pile vide**.
- ▶ $K = F$: acceptation **par pile vide et état final**.
- ▶ $K = QZ'Z^*$ avec $Z' \subseteq Z$: acceptation **par sommet de pile**.

Exemple :

$L = \{a^n b^n \mid n \geq 0\}$ peut être accepté par pile vide ou par sommet de pile.

Proposition : Acceptation généralisée

Soit \mathcal{A} un automate à pile avec acceptation généralisée K , on peut effectivement construire un automate à pile \mathcal{A}' acceptant par état final tel que $\mathcal{L}_K(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.
Donc, tous les modes d'acceptation ci-dessus sont équivalents.

Automates à pile et grammaires

Proposition : Grammaire vers automate à pile

Soit $G = (\Sigma, V, P, S)$ une grammaire. On peut construire un automate à pile simple (un seul état) \mathcal{A} qui accepte $L_G(S)$ par pile vide.

On construit l'automate à pile simple non déterministe acceptant par pile vide : $\mathcal{A} = (\Sigma, \Sigma \cup V, T, S)$ dont les transitions sont

- ▶ des expansions : $x \xrightarrow{\varepsilon} \alpha$ avec $(x, \alpha) \in P$,
- ▶ ou des vérifications : $a \xrightarrow{a} \varepsilon$ avec $a \in \Sigma$.

Remarques :

- ▶ AAP simple: un seul état (non noté dans les transitions)
- ▶ dérivation gauche \rightarrow_g^* : remplacer toujours la variable la plus à gauche
- ▶ dérivation droite \rightarrow_d^* : analogue

Preuve (idée) : pour tout $\alpha \in (\Sigma \cup V)^*$ et $w \in \Sigma^*$, prouver $\alpha \rightarrow_g^* w$ ssi $\alpha \xrightarrow{\mathcal{A}}^* \varepsilon$

- ▶ \rightarrow : récurrence sur longueur de dérivation
- ▶ \leftarrow : récurrence sur longueur du calcul

Automates à pile et grammaires

Proposition : Automate à pile vers grammaire

Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0 z_0)$ un automate à pile reconnaissant par pile vide. On peut construire une grammaire G qui engendre $\mathcal{L}(\mathcal{A})$.

Construction:

- ▶ $G = \langle \Sigma, \{S\} \cup (Q \times Z \times Q), \rightarrow, S \rangle$
- ▶ pour tout $q \in Q$, $S \rightarrow \langle q_0, z_0, q \rangle$;
- ▶ pour tout $a \in \Sigma \cup \{\varepsilon\}$ et $\langle qz, a, q' \rangle \in T$, $\langle q, z, q' \rangle \rightarrow a$;
- ▶ pour tout $a \in \Sigma \cup \{\varepsilon\}$, $n \geq 1$, $\langle qz, a, q' a_1 \cdots a_n \rangle \in T$ et $q_1, \dots, q_n \in Q$,

$$\langle q, z, q_n \rangle \rightarrow a \langle q', a_1, q_1 \rangle \langle q_1, a_2, q_2 \rangle \cdots \langle q_{n-1}, a_n, q_n \rangle$$

Preuve (idée): $\langle q, z, q' \rangle \rightarrow w$ dans G ssi $qz \xrightarrow{w^*} q'$ dans \mathcal{A}
(par récurrence sur longueur de dérivation resp. calcul)

Configurations accessibles (mots de pile)

Proposition : Reconnaissabilité des configurations accessibles

Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0z_0)$ un automate à pile.

Pour $p \in Q$ et $g \in Z^*$, on note

$$\mathcal{C}(pg) = \{qh \in QZ^* \mid \exists pg \xrightarrow{*} qh \text{ dans } \mathcal{T}\}$$

l'ensemble des configurations accessibles à partir de pg .

On peut effectivement construire un automate fini \mathcal{B} qui reconnaît $\mathcal{C}(pg)$.

Preuve : Voir plus loin.

Corollaire : Décidabilité du vide

Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0z_0, F)$ un automate à pile.

On peut décider si $\mathcal{L}(\mathcal{A}) = \emptyset$.

Preuve

Il suffit de tester si $q_0 \in F$ ou $\mathcal{C}(q_0z_0) \cap FZ^* \neq \emptyset$.

Calculs d'accessibilité

Corollaire :

Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0, z_0)$ un automate à pile.

On peut effectivement calculer les ensembles suivants :

1. $X = \{(p, x, q) \in Q \times Z \times Q \mid \exists px \xrightarrow{\neq} q \text{ dans } \mathcal{T}\}$
2. $Y = \{(p, x, q, y) \in Q \times Z \times Q \times Z \mid \exists px \xrightarrow{\neq} qy \text{ dans } \mathcal{T}\}$
3. $W = \{(p, x, q, y) \in Q \times Z \times Q \times Z \mid \exists px \xrightarrow{\neq} qyh \text{ dans } \mathcal{T}\}$
4. $X' = \{(p, x, q) \in Q \times Z \times Q \mid \exists px \xrightarrow{\varepsilon} q \text{ dans } \mathcal{T}\}$
5. $Y' = \{(p, x, q, y) \in Q \times Z \times Q \times Z \mid \exists px \xrightarrow{\varepsilon} qy \text{ dans } \mathcal{T}\}$
6. $W' = \{(p, x, q, y) \in Q \times Z \times Q \times Z \mid \exists px \xrightarrow{\varepsilon} qyh \text{ dans } \mathcal{T}\}$

Preuve

1. $(p, x, q) \in X$ ssi $q \in \mathcal{C}(px)$.
2. $(p, x, q, y) \in Y$ ssi $qy \in \mathcal{C}(px)$.
3. $(p, x, q, y) \in W$ ssi $\mathcal{C}(px) \cap qyZ^* \neq \emptyset$.

On applique ce qui précède à l'automate \mathcal{A}' obtenu à partir de \mathcal{A} en ne conservant que les ε -transitions.

Clôture et réduction.

Soit Γ un alphabet, $\bar{\Gamma} = \{\bar{a} \mid a \in \Gamma\}$ une copie de Γ et $\tilde{\Gamma} = \Gamma \uplus \bar{\Gamma}$.

On définit la réduction sur $\tilde{\Gamma}^*$ par $\bar{a}a \xrightarrow{\text{red}} \varepsilon$ pour $a \in \Gamma$.

Remarque : Soit $D = \{w \in \tilde{\Gamma}^* \mid w \xrightarrow[*]{\text{red}} \varepsilon\}$.

On peut montrer que D est engendré par la grammaire $S \rightarrow \varepsilon + \sum_{a \in \Gamma} \bar{a}SaS$.

Il s'agit donc du langage de Dyck en considérant \bar{a} comme une parenthèse ouvrante et a comme la parenthèse fermante correspondante.

Pour $L \subseteq \tilde{\Gamma}^*$ on pose $\text{Clot}(L) = \{w \in \tilde{\Gamma}^* \mid \exists v \in L, v \xrightarrow[*]{\text{red}} w\}$.

Lemme : Reconnaissabilité de la clôture

Si $L \subseteq \tilde{\Gamma}^*$ est un langage reconnaissable alors $\text{Clot}(L) \subseteq \tilde{\Gamma}^*$ aussi.

On peut construire un automate pour $\text{Clot}(L)$ à partir d'un automate pour L (PTIME).

Configurations accessibles (Preuve)

Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0 z_0)$ un automate à pile.

On définit $\Gamma = Q \uplus Z$ et le langage fini $K = \{qh\bar{x}\bar{p} \mid \exists (px, a, qh) \in T\} \subseteq \tilde{\Gamma}^+$.

Lemme : Soit $n \geq 0$

il existe un calcul $pg \xrightarrow{n} qh$ dans \mathcal{A} ssi il existe $w \in K^n$ tel que $wpg \xrightarrow{\text{red}_{2n}} qh$

Corollaire : $\mathcal{C}(pg) = \text{Clot}(K^+pg) \cap QZ^*$.

Puisque K est un langage fini, le langage K^+pg est reconnaissable et on peut construire (PTIME) un automate \mathcal{B} qui reconnaît $\text{Clot}(K^+pg) \cap QZ^*$.

AAP régulier

Remarques: Dans ce transparent et le suivant seulement:

- ▶ Σ' note $\Sigma \cup \{\varepsilon\}$;
- ▶ l'état et sommet de pile sont notés à droite !

Définition : AAP régulier

$\mathcal{A} = \langle Q, \Sigma, Z, T, q_0, F \rangle$, avec

$$T \subseteq (\text{Rec}(Z^*) \times Q \times \Sigma' \times Z \times Q) \cup (\text{Rec}(Z^*) \times Q \times \Sigma' \times Q)$$

1. $wq \xrightarrow{a} wzq'$ si $\langle L, q, a, z, q' \rangle \in T$ et $w \in L$ (push)
2. $wzq \xrightarrow{a} wq'$ si $\langle L, q, a, q' \rangle \in T$ et $wz \in L$ (pop)

AAP régulier \rightarrow AAP ordinaire

Soit $\mathcal{A} = \langle Q, \Sigma, Z, T, q_0, F \rangle$ un AAP régulier avec $k := |T|$
et $\forall i : \mathcal{A}_i = \langle Q_i, Z, \delta_i, \iota_i, F_i \rangle$ DCA acceptant les langages dans T . Définissons:

- ▶ $\mathcal{Q} := Q_1 \times \dots \times Q_k, \quad \iota := \langle \iota_1, \dots, \iota_k \rangle$
- ▶ $\mathcal{F}_i := \{ \langle q_1, \dots, q_k \rangle \in \mathcal{Q} \mid q_i \in F_i \}$
- ▶ $\delta : \mathcal{Q} \times Z \rightarrow \mathcal{Q}$ avec $\delta(\langle q_1, \dots, q_k \rangle, z) := \langle \delta_1(q_1, z), \dots, \delta_k(q_k, z) \rangle$.

Construction d'un AAP ordinaire équivalent à \mathcal{A}

$\mathcal{A}' := \langle \mathcal{Q} \times \mathcal{Q}, \Sigma, \mathcal{Q} \times Z, T', \langle \iota, q_0 \rangle, \mathcal{Q} \times F \rangle$, avec:

- ▶ (push) pour tout $\langle L_i, q, a, z, q' \rangle \in T$ et $f \in \mathcal{F}_i$, on a $\langle \langle f, q \rangle, a, \langle f, z \rangle, \langle \delta(f, z), q' \rangle \rangle \in T'$;
- ▶ (pop) pour tout $\langle L_i, q, a, q' \rangle \in T, z \in Z, q'' \in \mathcal{Q}$ et $f \in F_i$, on a $\langle \langle q'', z \rangle, \langle f, q \rangle, a, \langle q'', q' \rangle \rangle \in T'$.

Invariante: Si \mathcal{A} accède à une configuration $z_1 \dots z_n q$, alors \mathcal{A}' accède à $\langle q'_0, z_1 \rangle \dots \langle q'_{n-1}, z_n \rangle \langle q'_n, q \rangle$, avec $q'_0 = \iota$ et $q'_{i+1} = \delta(q'_i, z_{i+1})$ pour $i = 0, \dots, n-1$.

Calculs d'accessibilité dans une grammaire

Proposition : Reconnaissabilité des configurations précédentes

Soit $G = \langle \Sigma, V, \rightarrow, S \rangle$ une grammaire (algébrique).

Soit $L \subseteq (\Sigma \cup V)^*$ reconnaissable (par un automate \mathcal{A}).

On note $pre(L) = \{ \alpha \in (\Sigma \cup V)^* \mid \exists \beta \in L : \alpha \rightarrow_G^* \beta \}$.

On peut construire (en PTIME) un automate fini \mathcal{B} qui reconnaît $pre(L)$.

Preuve : Voir transparent suivant.

Corollaire : Corollaires

Les problèmes suivants sont décidables en PTIME:

- ▶ mot: pour $w \in \Sigma$, $w \in \mathcal{L}(G)$?
- ▶ variable productive: pour $A \in V$, existe-il $w \in \Sigma^*$ t.q. $A \rightarrow_G^* w$?
- ▶ $\mathcal{L}(G)$ est-il fini ?

Automate pour $pre(L)$

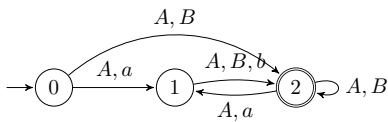
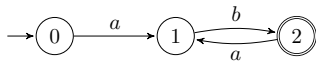
Proposition :

Soit $\mathcal{A} = \langle Q, \Sigma \cup V, \delta, q_0, F \rangle$ un automate reconnaissant L .

Construire \mathcal{B} en ajoutant (itérativement) des transitions selon la règle suivante :

Si $\langle A, \beta \rangle \in \rightarrow$ et $q \xrightarrow{\beta}_{\mathcal{B}} q'$, ajouter $\langle q, A, q' \rangle$ dans \mathcal{B} .

Exemple: $A \rightarrow a \mid BB, B \rightarrow AB \mid b$



Preuve (idée): $L(\mathcal{B}) = pre(\mathcal{L}(\mathcal{A}))$

- ▶ \subseteq : récurrence sur nombre de transitions ajoutées
- ▶ \supseteq : récurrence sur longueur de dérivation

Arbres de dérivation

Définition :

Soit $G = (\Sigma, V, P, S)$ une grammaire.

Un arbre de dérivation pour G est un arbre t étiqueté dans $V \cup \Sigma \cup \{\varepsilon\}$ tel que chaque nœud interne u est étiqueté par une variable $x \in V$ et si les fils de u portent les étiquettes $\alpha_1, \dots, \alpha_k$ alors $(x, \alpha_1 \cdots \alpha_k) \in P$.

De plus, si $k \neq 1$, on peut supposer $\alpha_1, \dots, \alpha_k \neq \varepsilon$.

Exemple :

Arbres de dérivation pour les expressions.

Mise en évidence des priorités ou de l'associativité G ou D.

Définition : Ambiguïté

- ▶ Une grammaire est ambiguë s'il existe deux arbres de dérivations (distincts) de même racine et de même frontière.
- ▶ Un langage algébrique est *non ambigu* s'il existe une grammaire non ambiguë qui l'engendre. Dans le cas contraire, on dit qu'il est *inhéremment ambigu*.

Lemme d'itération

Théorème : Bar-Hillel, Perles, Shamir ou Lemme d'itération

Soit $L \subseteq \Sigma^*$ algébrique. Il existe $N \geq 0$ tel que pour tout $w \in L$, si $|w| \geq N$ alors on peut trouver une factorisation $w = \alpha u \beta v \gamma$ avec $|uv| > 0$ et $|u\beta v| < N$ et $\alpha u^n \beta v^n \gamma \in L$ pour tout $n \geq 0$.

Exemple :

Le langage $L_1 = \{a^n b^n c^n \mid n \geq 0\}$ n'est pas algébrique.

Corollaire :

Le langages algébriques ne sont pas fermés par intersection ou complémentaire.

Langages déterministes

Définition : Automate à pile déterministe

$\mathcal{A} = (Q, \Sigma, Z, T, q_0 z_0, F)$ est *déterministe* si

- ▶ $\forall (pz, a) \in QZ \times (\Sigma \cup \{\varepsilon\}), \quad |T(pz, a)| \leq 1,$
- ▶ $\forall pz \in QZ, \quad T(pz, \varepsilon) \neq \emptyset \implies \forall a \in \Sigma, T(pz, a) = \emptyset$

Un langage $L \subseteq \Sigma^*$ est *déterministe* s'il existe un automate à pile déterministe qui accepte L par état final.

Exemples :

1. Le langage $L_4 = \{a^n b^p c^n \mid n, p > 0\} \cup \{a^n b^p d^p \mid n, p > 0\}$ est déterministe mais pas D+TR.
2. $L_5 = \{a^n b a^n \mid n > 0\}$ peut être accepté par un automate D+TR mais pas par un automate D+S car il n'est pas fermé par préfixe.

Acceptation par pile vide

Exemples :

1. Le langage $L_5 = \{a^n b a^n \mid n \geq 0\}$ peut être accepté par *pile vide* par un automate D+TR+S.
2. Le langage $L_4 = \{a^n b^p c^n \mid n, p > 0\} \cup \{a^n b^p d^p \mid n, p > 0\}$ peut être accepté par *pile vide* par un automate D.

Exercices :

1. Montrer qu'un langage L est déterministe et préfixe ($L \cap L\Sigma^+ = \emptyset$) ssi il existe un automate déterministe qui accepte L par pile vide.
2. Montrer que pour les automates à pile déterministes, l'acceptation par pile vide est équivalente à l'acceptation par pile vide ET état final.

Exercice :

Montrer que D_n^* peut être accepté par sommet de pile par un automate D+TR+S.

Complémentaire

Théorème : Les déterministes sont fermés par complémentaire.

Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0, z_0, F)$ un automate à pile déterministe, on peut effectivement construire un automate à pile déterministe \mathcal{A}' qui reconnaît $\Sigma^* \setminus \mathcal{L}(\mathcal{A})$.

Il y a deux difficultés principales :

1. Un automate déterministe peut se bloquer (deadlock) ou entrer dans un ε -calcul infini (livelock). Dans ce cas il y a des mots qui n'admettent aucun calcul dans l'automate.
2. Même avec un automate déterministe, un mot peut avoir plusieurs calculs (ε -transitions à la fin) certains réussis et d'autres non.

Blocage

Définition : Blocage

Un automate à pile $\mathcal{A} = (Q, \Sigma, Z, T, q_0 z_0)$ est sans blocage si pour toute configuration accessible $p\alpha$ et pour toute lettre $a \in \Sigma$ il existe un calcul $p\alpha \xrightarrow[*]{\varepsilon} \xrightarrow{a}$.

Proposition : Critère d'absence de blocage

Un automate *déterministe* est sans blocage si et seulement si pour toute configuration accessible $p\alpha$ on a

1. $\alpha \neq \varepsilon$, et donc on peut écrire $\alpha = x\beta$ avec $x \in Z$,
2. $px \xrightarrow{\varepsilon}$ ou $\forall a \in \Sigma, px \xrightarrow{a}$,
3. $px \not\xrightarrow{\varepsilon}$.

De plus, ce critère est décidable.

Remarque :

Si \mathcal{A} est sans blocage alors chaque mot $w \in \Sigma^*$ a un unique calcul maximal (et fini) $q_0 z_0 \xrightarrow[*]{w} p\alpha \not\xrightarrow{\varepsilon}$ dans \mathcal{A} (avec $\alpha \neq \varepsilon$).

Blocage

Proposition : Suppression des blocages

Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0 z_0, F)$ un automate à pile déterministe, on peut effectivement construire un automate à pile déterministe *sans blocage* $\mathcal{A}' = (Q', \Sigma, Z', T', q'_0 z'_0, F')$ qui reconnaît le même langage.

Preuve

$Q' = Q \uplus \{q'_0, d, f\}$, $F' = F \uplus \{f\}$, $Z' = Z \uplus \{\perp\}$, $z'_0 = \perp$ et pour $p \in Q$, $a \in \Sigma$ et $x \in Z$

1. $q'_0 \perp \xrightarrow{\varepsilon} q_0 z_0 \perp$,
2. Si $px \xrightarrow{a} q\alpha \in T$ alors $px \xrightarrow{a} q\alpha \in T'$,
3. Si $px \not\xrightarrow{a}$ et $px \xrightarrow{f}$ dans \mathcal{A} alors $px \xrightarrow{a} dx \in T'$,
4. Si $px \not\xrightarrow{\varepsilon}$ dans \mathcal{A} et $px \xrightarrow{\varepsilon} q\alpha \in T$ alors $px \xrightarrow{\varepsilon} q\alpha \in T'$,
5. Si $px \xrightarrow{\varepsilon} \omega$ dans \mathcal{A} et $\exists px \xrightarrow{\varepsilon} q\alpha$ avec $q \in F$ alors $px \xrightarrow{\varepsilon} fx \in T'$,
6. Si $px \xrightarrow{\varepsilon} \omega$ dans \mathcal{A} et $\forall px \xrightarrow{\varepsilon} q\alpha$ on a $q \notin F$ alors $px \xrightarrow{\varepsilon} dx \in T'$,
7. $p\perp \xrightarrow{\varepsilon} d\perp$, $d\perp \xrightarrow{a} d\perp$, $dx \xrightarrow{a} dx$ et $fx \xrightarrow{a} dx$.

Cette construction est effective.

Blocage

Preuve : $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}')$

Soit $q_0z_0 \xrightarrow{w} q\beta$ un calcul acceptant de \mathcal{A} pour $w \in \Sigma^*$.

Si le calcul ne passe pas par une configuration $px\alpha$ avec $px \xrightarrow{\varepsilon} ((p, x) \in V')$ alors

$$q'_0 \perp \xrightarrow{\varepsilon} q_0z_0 \perp \xrightarrow{(2,4)} q\beta \perp$$

est un calcul acceptant de \mathcal{A}' .

Sinon, le calcul s'écrit $q_0z_0 \xrightarrow{w} px\gamma \xrightarrow{*} q\beta$ avec $px \xrightarrow{\varepsilon}$

Donc $\beta = \alpha\gamma$ avec $px \xrightarrow{*} q\alpha$ et $q \in F$. On obtient un calcul acceptant dans \mathcal{A}' :

$$q'_0 \perp \xrightarrow{\varepsilon} q_0z_0 \perp \xrightarrow{(2,4)} px\gamma \perp \xrightarrow{\varepsilon} fx\gamma \perp$$

Preuve : $\mathcal{L}(\mathcal{A}') \subseteq \mathcal{L}(\mathcal{A})$

Un calcul acceptant de \mathcal{A}' ne peut pas atteindre l'état puits d . Deux possibilités:

Si le calcul accepte grâce à l'état f , il s'écrit:

$$q'_0 \perp \xrightarrow{\varepsilon} q_0z_0 \perp \xrightarrow{(2,4)} px\gamma \perp \xrightarrow{\varepsilon} fx\gamma \perp$$

On en déduit $q_0z_0 \xrightarrow{w} px\gamma$ dans \mathcal{A} et il existe $px \xrightarrow{*} q\alpha$ dans \mathcal{A} avec $q \in F$.

Donc w est accepté par le calcul $q_0z_0 \xrightarrow{w} px\gamma \xrightarrow{*} q\alpha\gamma$ de \mathcal{A} .

Sinon, le calcul de \mathcal{A}' s'écrit $q'_0 \perp \xrightarrow{\varepsilon} q_0z_0 \perp \xrightarrow{(2,4)} q\beta \perp$ avec $q \in F$

et on obtient le calcul acceptant $q_0z_0 \xrightarrow{w} q\beta$ dans \mathcal{A} pour w .

Complémentaire

Proposition :

Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0z_0, F)$ un automate à pile déterministe, on peut effectivement construire un automate à pile déterministe \mathcal{A}' qui reconnaît $\Sigma^* \setminus \mathcal{L}(\mathcal{A})$.

Proposition :

Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0z_0, F)$ un automate à pile déterministe, on peut effectivement construire un automate à pile déterministe équivalent \mathcal{A}' tel qu'on ne puisse pas faire d' ε -transition à partir d'un état final de \mathcal{A}' .

Exercice :

Montrer que tout langage déterministe est non ambigu.

Complémentaire

Preuve : Complémentaire

On suppose \mathcal{A} déterministe et sans blocage. On pose $Q' = Q \times \{1, 2, 3, 4\}$. L'état initial est $q'_0 = (q_0, 1)$ si $q_0 \notin F$ et $q'_0 = (q_0, 2)$ sinon.

1. Si $px \xrightarrow{\varepsilon} q\alpha \in T$ et $i \in \{1, 2\}$ alors
$$(p, i)x \xrightarrow{\varepsilon} (q, j)\alpha \in T' \text{ avec } j = \begin{cases} 1 & \text{si } i = 1 \wedge q \notin F \\ 2 & \text{sinon.} \end{cases}$$
2. Si $px \xrightarrow{a} q\alpha \in T$ et $i \in \{1, 2\}$ alors $(p, i)x \xrightarrow{\varepsilon} (p, i+2)x \in T'$ et
$$(p, i+2)x \xrightarrow{a} (q, j)\alpha \in T' \text{ avec } j = \begin{cases} 1 & \text{si } q \notin F \\ 2 & \text{sinon.} \end{cases}$$

L'automate \mathcal{A}' est déterministe et sans blocage.

Soit $w \in \Sigma^*$ et $q_0 z_0 \xrightarrow{w} p\alpha$ l'unique calcul *maximal* de \mathcal{A} sur w . On a $p\alpha \xrightarrow{\varepsilon}$.

L'unique calcul *maximal* de \mathcal{A}' sur w s'écrit $q'_0 z_0 \xrightarrow{w} (p, j)\alpha$ avec $j = 3$ si le calcul de \mathcal{A} n'a pas visité F depuis la dernière transition visible ($\neq \varepsilon$), et $j = 4$ sinon.

- ▶ Avec $F' = Q \times \{3\}$ on obtient $\mathcal{L}(\mathcal{A}') = \overline{\mathcal{L}(\mathcal{A})}$.
- ▶ Avec $F' = Q \times \{4\}$ on obtient $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.

Dans les deux cas, à partir d'un état de F' il n'y a pas d' ε -transition.

De plus, chaque mot $w \in \mathcal{L}(\mathcal{A}')$ a un unique calcul acceptant dans \mathcal{A}' .

Langages déterministes

Exercice :

Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0 z_0, K)$ un automate à pile déterministe avec acceptation généralisée par le langage rationnel $K \subseteq QZ^*$.

Montrer qu'on peut effectivement construire un automate à pile déterministe équivalent reconnaissant par état final.

Exercice :

Soit \mathcal{A} un automate à pile déterministe. Montrer qu'on peut effectivement construire un automate à pile déterministe qui reconnaît le même langage et dont les ε -transitions sont uniquement effaçantes : $px \xrightarrow{\varepsilon} q$.

Lemme d'itération pour les déterministes

Lemme : Itération

Soit $L \subseteq \Sigma^*$ un langage déterministe. Il existe un entier $N \in \mathbb{N}$ tel que tout mot $w \in L$ contenant au moins N lettres distinguées se factorise en $w = \alpha u \beta v \gamma$ avec

1. $\forall p \geq 0 : w = \alpha u^p \beta v^p \gamma \in \mathcal{L}(\mathcal{A})$,
2. $u \beta v$ contient moins de N lettres distinguées,
3. soit α, u, β soit β, v, γ contiennent des lettres distinguées,
4. pour tout $\gamma' \in \Sigma^*$,

$$\exists p : \alpha u^p \beta v^p \gamma' \in L \implies \forall p : \alpha u^p \beta v^p \gamma' \in L$$

Preuve Admis. Voir [?]

Rappels: Propriétés de clôture

Proposition :

Les langages algébriques :

- ▶ sont clôturés par union ;
- ▶ ne sont pas clôturés par complément (voir $\{ ww \mid w \in \Sigma^* \}$) ni intersection.

Proposition :

Les langages déterministes :

- ▶ sont clôturés par complément ;
- ▶ ne sont pas clôturés par intersection (voir $\{ a^n b^p c^n \mid n, p \geq 1 \}$, $\{ a^n b^n c^p \mid n, p \geq 1 \}$) ni intersection ;
- ▶ sont strictement moins expressif que les algébriques en général.

Problèmes indécidables

Proposition :

Soient L, L' deux langages algébriques et R un langage rationnel.

Les problèmes suivants sont indécidables :

- ▶ $L \cap L' = \emptyset$?
- ▶ $L = \Sigma^*$?
- ▶ $L = L'$?
- ▶ $L \subseteq L'$?
- ▶ $R \subseteq L$?
- ▶ L est-il rationnel ?
- ▶ L est-il déterministe ?
- ▶ L est-il ambigu ?
- ▶ \overline{L} est-il algébrique ?
- ▶ $L \cap L'$ est-il algébrique ?

Langages déterministes

Proposition : Décidabilité et indécidabilité

On ne peut pas décider si un langage algébrique est déterministe.

Soient L, L' deux langages déterministes et R un langage rationnel.

Les problèmes suivants sont décidables :

▶ $R \subseteq L$?

$$R \subseteq L \iff R \cap \bar{L} = \emptyset$$

▶ $L = R$?

$$R = L \iff R \cap \bar{L} = \emptyset = \bar{R} \cap L$$

▶ L est-il rationnel ?

▶ $L = L'$?

Géraud Sénizergues, prix Gödel 2002

Les problèmes suivants sont indécidables :

▶ $L \cap L' = \emptyset$?

▶ $L \subseteq L'$?

▶ $L \cap L'$ est-il algébrique ?

▶ $L \cap L'$ est-il déterministe ?

▶ $L \cup L'$ est-il déterministe ?