Langages formels et automates – cours 6

Compléments d'expressions régulières

Catalin Dima

Compléments expressions régulières

- Expressions régulières étendues : inclure la complémentation dans les opérations :

$$\overline{\left(a + \overline{bc}\right)^* + ac^*} + a(c\overline{a^*b})$$

  - Sémantique de la nouvelle opération = complémentation :

$$|\overline{E}| = \Sigma^* \setminus |E|$$

  - Donc on inclut l'intersection aussi !
- Équivalence des expressions régulières étendues avec les automates finis :
  - Corollaire du théorème de Kleene ...
  - ... et de la fermeture de $Rec(\Sigma)$ par complémentation.
- Exemple :
  - $L_5 = \{w \in \{a, b, c\}^* \mid w$ ne contient pas la séquence $aab\}$.

$$L_5 = |\overline{(a + b + c)^* aab(a + b + c)^*}|$$

  - Ça suffit pour prouver que L_5 est reconnaissable !

Compléments expressions régulières

- Problème de *sémantique vide* :
  - Étant donnée une expression régulière étendue, décider si sa sémantique est un langage vide.
  - Algorithme : il faut convertir en automate fini.
  - Complexité énorme si bcp de compléments imbriqués : chaque complémentation implique une déterminisation !
  - Donc une explosion exponentielle de l'espace d'états !
  - On peut prouver que cette explosion exponentielle est inévitable !

Compléments expressions régulières

Et l'algorithme de test de sémantique vide pour des expressions régulières normales ?

- Trivial !
- Un peu (mais très peu !) plus compliqué si on accepte aussi $\emptyset$ dans les expressions atomiques.
- Exemple :

$$E_1 = (a + bc^*)^* + \emptyset^* a$$
$$E_1 = a^*\emptyset + \emptyset(b + c)$$

- Idée d'algorithme :
  - Construire l'arbre syntaxique de l'expression.
  - Vérifier qu'il existe au moins un chemin pour lequel
    1. Soit le chemin contient une étoile.
    2. Soit le chemin se termine avec une expression atomique non-vide.
  - Utiliser n'importe quel parcours (prefix, infix, postfix).

Automates déterministes à partir d'expressions régulières

- On sait construire des automates (avec ε-transitions) à partir d'expressions régulières.
- Et on sait éliminer les ε-transitions et déterminiser les automates finis.
- Mais peut-on construire des automates déterministes plus directement ?
- Petite idée : utiliser les *quotients* des langages :
  - $a\backslash L$ = ce qui reste à parcourir du mot, après avoir "avalé" un a.
- Il faudrait alors essayer de construire itérativement les quotients à gauche pour toutes les lettres de l'alphabet.
- Mais comment formaliser le résultat sur des expressions régulières ?

Dérivées de Brzozowski

- Les quotients gauche des expressions régulières – calculés itérativement à l'aide des "schémas"
  - Un peu comme les dérivées des fonctions.
- $\partial_a(E) =$ autre notation pour $a\backslash E$.
- Règles de calcul (définition par induction structurelle) :

$$\partial_a(a) = \varepsilon \qquad \partial_a(E_1 + E_2) = \partial_a(E_1) + \partial_a(E_2)$$
$$\partial_a(b) = \partial_a(\varepsilon) = \emptyset \qquad \partial_a(E_1 \cdot E_2) = \partial_a(E_1) \cdot E_2 + o(E_1) \cdot \partial_a(E_2)$$
$$\partial_a(E^*) = \partial_a(E) \cdot E^*$$

- Ici, $o(E)$ indique si ε appartient à $|E|$.
- Sa définition par induction structurelle est :

$$o(a) = \emptyset \qquad o(E_1 + E_2) = o(E_1) + o(e_2)$$
$$o(\varepsilon) = \varepsilon \qquad o(E_1 \cdot E_2) = o(E_1) \cdot o(E_2)$$
$$o(E^*) = \varepsilon$$

Dérivées d'une expressions régulière

- Étant donné une expression E, on construit toutes ses dérivées.
- Et toutes les dérivées des dérivées, et les dérivées de 3e ordre, de 4e ordre...
- Quand s'arrêter ?
- Formalisation

$$\mathcal{D}_0 = \{E\}$$
$$\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{\partial_a(E) \mid E \in \mathcal{D}_n, a \in \Sigma\}$$

- Exemple :

$$E = (ab^* + c)^* + a^*bc^*$$
$$\mathcal{D}_1 =? \quad \mathcal{D}_2 =? \quad \mathcal{D}_3 =?.....$$

- Il faut observer que $|a^*bc^*| \subseteq |(ab^* + c)^*|$.
- ... et donc on peut s'arrêter plus tot !
- Et en général, la suite $(\mathcal{D}_n)$ s'arrête toujours ?...

Théorème de Brzozowski

1. Pour toute $E \in RegExpr(\Sigma)$ et $a \in \Sigma$,

$$|\partial_a(E)| = a\backslash|E| = \left\{w \in \Sigma^* \mid aw \in |E|\right\} \tag{1}$$

$$|o(E)| = \begin{cases} \varepsilon & \text{si } \varepsilon \in |E| \\ \emptyset & \text{sinon} \end{cases} \tag{2}$$

2. La suite $(\mathcal{E}_n)_{n\geq 0}$ définie comme suit :

$$\mathcal{E}_n = \left\{|E| \mid E \in \mathcal{D}_n\right\}$$

est convergente : il existe un n tel que $\mathcal{E}_n = \mathcal{E}_{n+1} = \mathcal{E}$.

3. La structure suivante est un automate fini équivalent avec l'expression donnée :

$$\mathcal{A} = (\mathcal{E}_n, \Sigma, \delta, \{|E|\}, \mathcal{E}^f)$$

$$\delta = \left\{|E_i| \xrightarrow{a} |E_j| \mid E_i, E_j \in \mathcal{D}_n, \partial_a(E_i) = E_j\right\}$$

$$\mathcal{E}^f = \left\{|E_i| \mid E_i \in \mathcal{D}_n, o(E_i) = \varepsilon\right\}$$

Le théorème de Brzozowski dans la "pratique"

Exemple pour $E = (ab^* + c)^* + a^*bc^*$

- On calcule $\mathcal{D}_n$ itérativement.
- Mais il nous faut $\mathcal{E}_n$.
- Malheureusement cela impliquerait de construire des automates pour chacune des expressions de $\mathcal{D}_n$...
- ... et de tester l'égalité des langages ! !
- Alors on essaie de construire un automate dont les états sont étiquetés par $\mathcal{D}_n$, et non pas par $\mathcal{E}_n$.
- Exemple avec $E = (ab^* + c)^* + a^*bc^*$.
- Il nous fait toutefois des règles permettant de décider quand on ne rajoute pas de nouvel état !
- Il nous faut des identités sur les trois opérations $\cdot, +, ^*$

Algèbre de Kleene

- Est-ce qu'on peut donner un ensemble de règles sur les trois opérations $\cdot, +, ^*$?
  - On voudrait prouver que $|E_1| = |E_2|$ sans devoir construire les automates !
  - On voudrait juste modifier l'expression E_1 pour la transformer en E_2.
- On cherche en fait des axiomes !
- Et on en connait quelques-unes :
  - Associativité et commutativité de l'union.
  - Associativité de la concaténation.
  - Distributivité de $\cdot$ par rapport à $+$.
  - Idempotence de $+$: $X + X = X$.
- Et l'étoile ?
- On a une structure algébrique nommée **algèbre de Kleene**

$$(RegExpr(\Sigma), +, \cdot, ^*, \emptyset, \{\varepsilon\})$$

Axiomatization de Conway

- Toute identité de type $|E_1| = |E_2|$ peut se prouver en utilisant l'ensemble d'axiomes suivant :
  1. Associativité de l'union et de la concaténation.
  2. Commutativité et idempotence de l'union.
  3. $\emptyset$ est élément neutre pour l'union.
  4. $\{\varepsilon\}$ est élément neutre pour la concaténation.
  5. Distributivité de la concaténation par rapport à l'union.
  6. Axiomes pour l'étoile :

$$1 + X \cdot X^* \subseteq X^* \tag{3}$$

$$1 + X^* \cdot X \subseteq X^* \tag{4}$$

$$\text{si } 1 + AX \subseteq X \text{ alors } A^*X \subseteq X \tag{5}$$

$$\text{si } 1 + XA \subseteq X \text{ alors } XA^* \subseteq X \tag{6}$$

- Ça peut se traduire en algorithme.
- Mais nous on va se limiter à deviner les équivalences.
- Quelques exemples en TD.

Langages réguliers sur une seule lettre

- Qu'est-ce qu'on obtient si $\Sigma = \{a\}$?
- Peut-on décrire de manière plus simple un langage comme :

$$E = \left((aa^* + aaa)^* + aa^*aa^*\right)^*$$

- **Théorème** : La sémantique de toute expression régulière ayant un seul atome a est un langage quasi-périodique :
  - L'ensemble des longueurs de mots du langage forme un ensemble finalement périodique.
  - C.à.d. union finie de progressions arithmétiques de même ratio, plus un ensemble fini de nombres.
- Pour notre cas on retrouve tous les mots d'une seule lettre :

$$|E| = |a^*|$$

- Preuve par induction :
  - Il faut prouver que l'étoile d'un langage de type quasi-périodique est aussi quasi-périodique.
  - Ce qui revient à prouver une propriété similaire sur les ensembles de nombres naturels.

Algèbre de Kleene sur les ensembles d'entiers

- On peut organiser $\mathcal{P}(\mathbb{N})$ comme algèbre de Kleene :

$$\left(\mathcal{P}(\mathbb{N}), \cup, +, ^*, \emptyset, \mathbf{0}\right)$$

  - L'étoile :

$$A^* = \left\{a_1 + \ldots + a_n \mid n \geq 0, a_i \in A \text{ pour tout } 1 \leq i \leq n\right\}$$

- Idée de preuve : écrire comme un ensemble finalement périodique :

$$\{2,3\}^* = ? \qquad \{12,15,21\}^* = ? \qquad (2^* \cup 3^*)^* = ?$$

- On découvre que

$$\{a,b\}^* = F \cup \{dn + k \mid n \in \mathbb{N}\} = F \cup (\{d\}^* + \{k\})$$

où d est le pgcd de a et de b.

PGCD et expressions régulières sur des alphabets à une lettre

- Pourquoi a-t-on $\{a, b\}^* = F \cup (\{pgcd(a, b)\}^* + \{k\})$?
- C'est une application d'un théorème de Bézout
  - L'équation $ax + by = c$ admet des solutions si et seulement si c est un multiple du $pgcd(a, b)$.
  - Si $c > ab$ alors cette équation admet des solutions *positives*.
- Donc tout multiple du $pgcd(a, b)$ plus grand que ab appartient à $\{a, b\}^*$!
- Pour tous les autres (nombre fini), il y a certains qui le sont, certains qui ne le sont pas.
- Ceux qui le sont appartiennent au F.
- Les axiomes d'algèbre de Kleene nous assurent alors que le théorème est vrai en général :

$$\{a, b, c\}^* = \left(\{a, b\}^* \cup \{c\}\right)^*$$

- Exemples :

$$|(aa)^*| = \{a^{2n} \mid n \in \mathbb{N}\}$$
$$|(a^6 + a^4)^*| = \{\varepsilon\} \cup \{a^{2n+4} \mid n \in \mathbb{N}\}$$