Équivalence des langages rationnels et reconnaissables

MP/MP* Option info

Équivalence

But du cours:

Théorème

Soit L un langage. Alors:

$$L \text{ est rationnel} \iff L \text{ est reconnaissable}$$

Rationnel $\Longrightarrow$ reconnaissable

Preuve de « rationnel $\Longrightarrow$ reconnaissable »:

1. Un langage rationnel L peut être linéarisé (chaque lettre n'est alors utilisée qu'une seule fois)

Rationnel $\Longrightarrow$ reconnaissable

Preuve de « rationnel $\Longrightarrow$ reconnaissable »:

1. Un langage rationnel L peut être linéarisé (chaque lettre n'est alors utilisée qu'une seule fois)
2. Un langage linéaire est local

Rationnel $\Longrightarrow$ reconnaissable

Preuve de « rationnel $\Longrightarrow$ reconnaissable »:

1. Un langage rationnel L peut être linéarisé (chaque lettre n'est alors utilisée qu'une seule fois)
2. Un langage linéaire est local
3. Un langage local est reconnu par un automate local

Rationnel $\Longrightarrow$ reconnaissable

Preuve de « rationnel $\Longrightarrow$ reconnaissable »:

1. Un langage rationnel L peut être linéarisé (chaque lettre n'est alors utilisée qu'une seule fois)
2. Un langage linéaire est local
3. Un langage local est reconnu par un automate local
4. Cet automate local peut être « délinéarisé » pour reconnaître L

Linéarisation

Définition

Une expression rationnelle est **linéaire** si chaque lettre y apparaît au plus une fois.

Linéarisation

Définition

Une expression rationnelle est **linéaire** si chaque lettre y apparaît au plus une fois.

Définition

Soit e une expression rationnelle sur un alphabet Σ.
Soit k le nombre de lettres (avec multiplicité) apparaissant dans e.
Soit Σ' un alphabet de taille k.
Linéariser e consiste à remplacer chaque occurrence de lettre apparaissant dans e par une lettre différente de Σ'.

Linéarisation

Définition

Une expression rationnelle est **linéaire** si chaque lettre y apparaît au plus une fois.

Définition

Soit e une expression rationnelle sur un alphabet Σ.
Soit k le nombre de lettres (avec multiplicité) apparaissant dans e.
Soit Σ' un alphabet de taille k.
Linéariser e consiste à remplacer chaque occurrence de lettre apparaissant dans e par une lettre différente de Σ'.

Exemple: soit $e = \varepsilon + b(a + bb)^* b$. En prenant $\Sigma' = \{c_0, c_1, c_2, c_3, c_4\}$, on peut linéariser e en $e' = \varepsilon + c_0(c_1 + c_2 c_3)^* c_4$.

Langage local

Définition

Soit L un langage. On définit:

- $P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\}$ (premières lettres des mots de L)
- $S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\}$ (dernières lettres des mots de L)
- $F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\}$ (facteurs de longueur 2 des mots de L)
- $N(L) = \Sigma^2 \backslash F(L)$

Langage local

Définition

Soit L un langage. On définit:

- $P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\}$ (premières lettres des mots de L)
- $S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\}$ (dernières lettres des mots de L)
- $F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\}$ (facteurs de longueur 2 des mots de L)
- $N(L) = \Sigma^2 \backslash F(L)$

Question

Quel est l'ensemble des mots dont tous les facteurs de longueur 2 appartiennent à $F(L)$?

Langage local

Définition

Soit L un langage. On définit:

- $P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\}$ (premières lettres des mots de L)
- $S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\}$ (dernières lettres des mots de L)
- $F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\}$ (facteurs de longueur 2 des mots de L)
- $N(L) = \Sigma^2 \backslash F(L)$

Question

Quel est l'ensemble des mots dont tous les facteurs de longueur 2 appartiennent à $F(L)$?

$\Sigma^* \backslash (\Sigma^* N(L) \Sigma^*)$

Langage local

Définition

Soit L un langage. On définit:

- $P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\}$ (premières lettres des mots de L)
- $S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\}$ (dernières lettres des mots de L)
- $F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\}$ (facteurs de longueur 2 des mots de L)
- $N(L) = \Sigma^2 \backslash F(L)$

Question

Quel est l'ensemble des mots dont tous les facteurs de longueur 2 appartiennent à $F(L)$?

$\Sigma^* \backslash (\Sigma^* N(L) \Sigma^*)$ $(\neq \Sigma^* F(L) \Sigma^*)$

Langage local

Définition

Soit L un langage. On définit:

- $P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\}$ (premières lettres des mots de L)
- $S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\}$ (dernières lettres des mots de L)
- $F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\}$ (facteurs de longueur 2 des mots de L)
- $N(L) = \Sigma^2 \backslash F(L)$

Question

Écrire des fonctions `prefixe`, `suffixe`, `facteur` de type `'a regexp -> 'a list` pour déterminer $P(L)$, $S(L)$, $F(L)$.

Langage local

Définition

Soit L un langage. On définit:

- $P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\}$ (premières lettres des mots de L)
- $S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\}$ (dernières lettres des mots de L)
- $F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\}$ (facteurs de longueur 2 des mots de L)
- $N(L) = \Sigma^2 \backslash F(L)$

Définition

Un langage L est **local** si:

$$L \backslash \varepsilon = (P(L)\Sigma^* \cap \Sigma^* S(L)) \setminus \Sigma^* N(L) \Sigma^*$$

Langage local

Définition

Soit L un langage. On définit:

- $P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\}$ (premières lettres des mots de L)
- $S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\}$ (dernières lettres des mots de L)
- $F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\}$ (facteurs de longueur 2 des mots de L)
- $N(L) = \Sigma^2 \backslash F(L)$

Définition

Un langage L est **local** si:

$$L \backslash \varepsilon = (P(L)\Sigma^* \cap \Sigma^* S(L)) \setminus \Sigma^* N(L) \Sigma^*$$

(L est déterminé par ses préfixes, suffixes, et facteurs de tailles 2)

Langage local

Définition

Soit L un langage. On définit:

- $P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\}$ (premières lettres des mots de L)
- $S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\}$ (dernières lettres des mots de L)
- $F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\}$ (facteurs de longueur 2 des mots de L)
- $N(L) = \Sigma^2 \backslash F(L)$

Définition

Un langage L est **local** si:

$$L \backslash \varepsilon = (P(L)\Sigma^* \cap \Sigma^* S(L)) \setminus \Sigma^* N(L) \Sigma^*$$

(L est déterminé par ses préfixes, suffixes, et facteurs de tailles 2)

Remarque: $L \backslash \varepsilon \subseteq (P(L)\Sigma^* \cap \Sigma^* S(L)) \setminus \Sigma^* N(L) \Sigma^*$ est toujours vraie

Langage local

Définition

Soit L un langage. On définit:

- $P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\}$ (premières lettres des mots de L)
- $S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\}$ (dernières lettres des mots de L)
- $F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\}$ (facteurs de longueur 2 des mots de L)

Dit autrement:

Définition

Un langage L est **local** si, pour tout mot $u = u_1 u_2 ... u_n \neq \varepsilon$:

$$u \in L \iff u_1 \in P(L) \wedge u_n \in S(L) \wedge \forall k, u_k u_{k+1} \in F(L)$$

Langage local

Définition

Soit L un langage. On définit:

- $P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\}$ (premières lettres des mots de L)
- $S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\}$ (dernières lettres des mots de L)
- $F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\}$ (facteurs de longueur 2 des mots de L)

Dit autrement:

Définition

Un langage L est **local** si, pour tout mot $u = u_1 u_2 ... u_n \neq \varepsilon$:

$$u \in L \iff u_1 \in P(L) \land u_n \in S(L) \land \forall k, u_k u_{k+1} \in F(L)$$

Exemples: a^*, $(ab)^*$, $a^* + (ab)^*$, $a^*(ab)^*$ sont-ils locaux?

Linéaire $\Longrightarrow$ local

Linéaire $\implies$ local

Stabilité des langages locaux

Soient L_1 et L_2 des langages locaux sur des alphabets disjoints Σ_1 et Σ_2. Alors:

- $L_1 \cup L_2$ est local sur $\Sigma_1 \cup \Sigma_2$
- $L_1 L_2$ est local sur $\Sigma_1 \cup \Sigma_2$
- L_1^* est local sur Σ_1

Linéaire $\implies$ local

Stabilité des langages locaux

Soient L_1 et L_2 des langages locaux sur des alphabets disjoints Σ_1 et Σ_2. Alors:

- $L_1 \cup L_2$ est local sur $\Sigma_1 \cup \Sigma_2$
- $L_1 L_2$ est local sur $\Sigma_1 \cup \Sigma_2$
- L_1^* est local sur Σ_1

Théorème

Tout langage linéaire est local.

Preuves: en TD.

Automate local

Définition

Un automate déterministe (Σ, Q, q_0, F, E) est **local** si toutes les transitions étiquetées par la même lettre aboutissent au même état:

$$(q_1, a, q_2) \in E \wedge (q_3, a, q_4) \in E \implies q_2 = q_4$$

Automate local

Définition

Un automate déterministe (Σ, Q, q_0, F, E) est **local** si toutes les transitions étiquetées par la même lettre aboutissent au même état:

$$(q_1, a, q_2) \in E \wedge (q_3, a, q_4) \in E \implies q_2 = q_4$$

Théorème

Tout langage local L est reconnu par un automate local.

Automate local

Définition

Un automate déterministe (Σ, Q, q_0, F, E) est **local** si toutes les transitions étiquetées par la même lettre aboutissent au même état:

$$(q_1, a, q_2) \in E \wedge (q_3, a, q_4) \in E \implies q_2 = q_4$$

Théorème

Tout langage local L est reconnu par un automate local.

<u>Preuve</u>:

Si L ne contient pas ε, il est reconnu par (Σ, Q, q_0, F, E) où:

- $Q = \Sigma \cup \{q_0\}$: un état correspond à la dernière lettre lue
- $F = S(L)$
- $E = \{(q_0, a, a) \mid a \in P(L)\} \cup \{(a, b, b) \mid ab \in F(L)\}$

Automate local

Définition

Un automate déterministe (Σ, Q, q_0, F, E) est **local** si toutes les transitions étiquetées par la même lettre aboutissent au même état:

$$(q_1, a, q_2) \in E \wedge (q_3, a, q_4) \in E \implies q_2 = q_4$$

Théorème

Tout langage local L est reconnu par un automate local.

Preuve:
Si L ne contient pas ε, il est reconnu par (Σ, Q, q_0, F, E) où:

- $Q = \Sigma \cup \{q_0\}$: un état correspond à la dernière lettre lue
- $F = S(L)$
- $E = \{(q_0, a, a) \mid a \in P(L)\} \cup \{(a, b, b) \mid ab \in F(L)\}$

Exemple: construire un automate local reconnaissant $a(b^* + c)$.

Algorithme de Berry-Sethi

Soit e une expression rationnelle.

1. On linéarise e en e'. On note φ la fonction qui à chaque lettre de e' associe la lettre correspondante de e.
2. On construit un automate local A reconnaissant $L(e')$. Pour cela il faut calculer $P(L(e'))$, $S(L(e'))$, $F(L(e'))$.
3. On remplace chaque étiquette a de A par $\varphi(a)$. On obtient alors un automate (de Glushkov) reconnaissant $L(e)$.

On en déduit:

Théorème

L est un langage rationnel $\Longrightarrow$ L est reconnaissable.

Algorithme de Berry-Sethi

Soit e une expression rationnelle.

1. On linéarise e en e'. On note φ la fonction qui à chaque lettre de e' associe la lettre correspondante de e.
2. On construit un automate local A reconnaissant $L(e')$. Pour cela il faut calculer $P(L(e'))$, $S(L(e'))$, $F(L(e'))$.
3. On remplace chaque étiquette a de A par $\varphi(a)$. On obtient alors un automate (de Glushkov) reconnaissant $L(e)$.

On en déduit:

Théorème

L est un langage rationnel $\implies$ L est reconnaissable.

Exemple: construire l'automate de Glushkov reconnaissant $L(a(a+b)^*)$.