



Langages et automates

Principales définitions et «recettes» pour rédiger des démonstrations correctes

Rappels sur les langages

Définition Un *alphabet* est un ensemble fini de symboles. Un *mot* sur un alphabet A est une séquence finie de symboles de A . La séquence vide, ne contenant aucun symbole de l'alphabet, est un mot appelé le mot vide, et noté ε . L'ensemble de tous les mots sur l'alphabet A est un ensemble infini noté A^* (qui contient *toujours* ε). Un *langage* L sur A , aussi appelé un langage de A^* , est un sous-ensemble de $A^* : L \subseteq A^*$.

Opérations sur les langages

Les langages étant des ensembles, on peut utiliser les définitions ensemblistes connues : $L_1 \cup L_2$, $L_1 \cap L_2$, $\overline{L_1}$

Définition Si L_1 et L_2 sont deux langages sur un alphabet A , $L_1 \cdot L_2$ est le *langage produit* (ou *concaténation*) défini par : $L_1 \cdot L_2 = \{w \in A^* \mid w = u \cdot v, u \in L_1, v \in L_2\}$.

Montrons que $u \in L_1 \cdot L_2$.

Montrons que $u = v_1 \cdot v_2$ avec $v_1 \in L_1$ et $v_2 \in L_2$

...

À partir de l'hypothèse $u \in L_1 \cdot L_2$

on peut en déduire qu'il existe v_1, v_2

tels que $u = v_1 \cdot v_2$ et $v_1 \in L_1$ et $v_2 \in L_2$.

Définition Soit $L \subseteq A^*$ un langage. On définit inductivement $L^0 = \{\varepsilon\}$, et, pour tout $n \geq 0$, $L^{n+1} = L \cdot L^n$. On définit ensuite $L^* = \bigcup_{n \geq 0} L^n$ et $L^+ = \bigcup_{n \geq 1} L^n$.

Montrons que $u \in L^*$.

Montrons qu'il existe un entier $n \geq 0$

tel que $u \in L^n$.

...

A partir de l'hypothèse $u \in L^*$, on peut déduire

l'existence d'un entier $n \geq 0$ tel que $u \in L^n$

Montrons que $u \in L^+$.

Montrons qu'il existe un entier $n \geq 1$

tel que $u \in L^n$.

...

A partir de l'hypothèse $u \in L^+$, on peut déduire

l'existence d'un entier $n \geq 1$ tel que $u \in L^n$.

Automates

Définition Sur un alphabet A , un automate fini est donné par $\mathcal{A} = (S, T, I, F)$ où

- S est un ensemble fini d'*états*,
- $T \subseteq S \times A \times S$ est une *relation de transition*,
- $I \subseteq S$ est l'ensemble des *états initiaux*,
- $F \subseteq S$ est l'ensemble des *états finaux*

Définition Une exécution de \mathcal{A} est une séquence finie $s_0 a_1 s_1 a_2 \dots a_n s_n$, souvent notée $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \dots s_{n-1} \xrightarrow{a_n} s_n$, telle que :

- $s_0 \in I$, est un état initial,
- pour tout $0 \leq i < n$, $(s_i, a_{i+1}, s_{i+1}) \in T$ est une transition autorisée par la relation de transition.

La séquence $a_1 a_2 \dots a_n$ est un mot de A , qui *étiquette* l'exécution. Une exécution est *acceptante* si $s_n \in F$. Un mot $u \in A^*$ est *accepté par* \mathcal{A} s'il est étiquette d'une exécution acceptante.

Montrons que $u = u_1 \dots u_n$ est accepté par $\mathcal{A} = (S, T, I, F)$.

Montrons qu'il existe $s_0, \dots, s_n \in S$ tels que

$s_0 \xrightarrow{u_1} s_1 \xrightarrow{u_2} s_2 \dots s_{n-1} \xrightarrow{u_n} s_n$

est une exécution acceptante.

A partir de l'hypothèse u est accepté par \mathcal{A} ,

on peut déduire qu'il existe $s_0, \dots, s_n \in S$ tels que

$s_0 \xrightarrow{u_1} s_1 \xrightarrow{u_2} s_2 \dots s_{n-1} \xrightarrow{u_n} s_n$

est une exécution acceptante.

Définition Le langage de \mathcal{A} est donné par $L(\mathcal{A}) = \{u \in A^* \mid u \text{ est accepté par } \mathcal{A}\}$.

Montrons que $L(\mathcal{A}) = L$.

1– Montrons que $L(\mathcal{A}) \subseteq L$

Soit $u \in L(\mathcal{A})$. Montrons que $u \in L$

...

2– Montrons que $L \subseteq L(\mathcal{A})$

Soit $u \in L$. Montrons que $u \in L(\mathcal{A})$

...

Définition Soit A un alphabet. Un langage $L \subseteq A^*$ est *reconnaissable* s'il existe un automate fini \mathcal{A} tel que $L = L(\mathcal{A})$.

Montrons que L est reconnaissable.

Soit \mathcal{A} un automate fini défini par

$\mathcal{A} = \boxed{\text{à trouver}}$.

Montrons que $L(\mathcal{A}) = L$

...

À partir de l'hypothèse L est reconnaissable, on peut déduire l'existence d'un automate \mathcal{A} tel que $L(\mathcal{A}) = L$.

Définition Deux automates finis sont *équivalents* s'ils acceptent le même langage.

Montrons que \mathcal{A} et \mathcal{B} sont équivalents.

1– Montrons $L(\mathcal{A}) \subseteq L(\mathcal{B})$

...

2– Montrons $L(\mathcal{B}) \subseteq L(\mathcal{A})$

...

Automates complets

Définition Un automate $\mathcal{A} = (S, T, I, F)$ sur un alphabet A est *complet* si, pour tout $s \in S$, pour tout $a \in A$, il existe $s' \in S$ et $(s, a, s') \in T$.

Montrons que \mathcal{A} automate sur $A = \{a_1, \dots, a_n\}$ est complet.

Soient s_1, \dots, s_k les états de \mathcal{A} .

1–1 Montrons qu'il existe $s' \in S$ tel que $(s_1, a_1, s') \in T$.

...

...

1–n Montrons qu'il existe $s' \in S$ tel que $(s_1, a_n, s') \in T$.

...

...

k–1 Montrons qu'il existe $s' \in S$ tel que $(s_k, a_1, s') \in T$.

...

...

k–n Montrons qu'il existe $s' \in S$ tel que $(s_k, a_n, s') \in T$.

...

À partir de l'hypothèse \mathcal{A} est complet, on peut déduire $(s, a, s') \in T$ pour tous $s, s' \in S$ et pour tout $a \in A$

Théorème Tout automate fini est équivalent à un automate complet.

Soit $\mathcal{A} = (S, T, I, F)$ un automate fini sur un alphabet A . On construit $\text{comp}(\mathcal{A}) = (S \uplus \{\perp\}, T', I, F)$ avec $T' = T \uplus \{(s, a, \perp) \mid s \in S \uplus \{\perp\}, a \in A, \text{ et pour tout } s' \in S, (s, a, s') \notin T\}$.

— $\text{comp}(\mathcal{A})$ est complet.

— $L(\text{comp}(\mathcal{A})) = L(\mathcal{A})$.

Automates déterministes

Définition Un automate fini déterministe sur un alphabet A est un automate donné par $\mathcal{A} = (S, T, i, F)$ avec i unique état initial, et T tel que pour tout $s \in S$, pour tout $a \in A$, si $(s, a, s') \in T$ et $(s, a, s'') \in T$, alors $s' = s''$.

Montrons que \mathcal{A} est

un automate déterministe.

Soient s_1, \dots, s_k les états de \mathcal{A} .

1–1 Montrons qu'il existe au plus un état $s' \in S$ tel que $(s_1, a_1, s') \in T$.

...

1–n Montrons qu'il existe au plus un état $s' \in S$ tel que $(s_1, a_n, s') \in T$.

...

k–1 Montrons qu'il existe au plus un état $s' \in S$ tel que $(s_k, a_1, s') \in T$.

...

k–n Montrons qu'il existe au plus un état $s' \in S$ tel que $(s_k, a_n, s') \in T$.

...

A partir d'un mot $u \in A^*$

et de l'hypothèse

« \mathcal{A} est déterministe»,

on peut déduire

\mathcal{A} a au plus une exécution sur u .

Théorème Pour tout automate fini \mathcal{A} , on peut construire un automate fini déterministe équivalent.

Soit $\mathcal{A} = (S, T, I, F)$ un automate fini non déterministe sur un alphabet A . On définit $\det(\mathcal{A}) = (\mathcal{P}(S), T', \{I\}, F')$ avec

— Pour $X \in \mathcal{P}(S)$, pour $a \in A$, $T'(X, a) = \{s' \in S \mid \text{il existe } s \in X \text{ et } (s, a, s') \in T\}$.

— $F' = \{X \subseteq S \mid X \cap F \neq \emptyset\}$.

Alors,

— $\det(\mathcal{A})$ est déterministe.

— $L(\mathcal{A}) = L(\det(\mathcal{A}))$.

Propriétés de clôture des langages reconnaissables

Théorème Si $L \subseteq A^*$ est un langage reconnaissable, alors \bar{L} est reconnaissable.

Soit \mathcal{A} tel que $L(\mathcal{A}) = L$. On suppose que \mathcal{A} est *déterministe* et *complet* (sinon on le détermine et/ou complète). Si $\mathcal{A} = (S, T, i, F)$ alors on construit $\mathcal{B} = (S, T, i, F')$ avec $F' = S \setminus F$, et on a $L(\mathcal{B}) = \bar{L}(\mathcal{A})$

Théorème Si $L_1, L_2 \subseteq A^*$ sont deux langages reconnaissables, alors les langages $L_1 \cap L_2$ et $L_1 \cup L_2$ sont reconnaissables.

Soient $\mathcal{A}_1 = (S_1, T_1, I_1, F_1)$ et $\mathcal{A}_2 = (S_2, T_2, I_2, F_2)$ deux automates reconnaissant respectivement L_1 et L_2 . On définit l'automate produit $\mathcal{A} = (S_1 \times S_2, T, I_1 \times I_2, F)$ avec $T = \{((s_1, s_2), a, (s'_1, s'_2)) \mid (s_1, a, s'_1) \in T_1 \text{ et } (s_2, a, s'_2) \in T_2\}$

— **Intersection** Si $F = F_1 \times F_2$, $L(\mathcal{A}) = L_1 \cap L_2$.

— **Union** Si \mathcal{A}_1 et \mathcal{A}_2 sont complets, et $F = (F_1 \times S_2) \cup (S_1 \times F_2)$, alors $L(\mathcal{A}) = L_1 \cup L_2$.

La construction préserve le déterminisme : si \mathcal{A}_1 et \mathcal{A}_2 sont déterministes, \mathcal{A} est déterministe aussi.

Autre construction pour l'union : Soient $\mathcal{A}_1 = (S_1, T_1, I_1, F_1)$ et $\mathcal{A}_2 = (S_2, T_2, I_2, F_2)$ deux automates reconnaissant respectivement L_1 et L_2 . Si $S_1 \cap S_2 = \emptyset$, ce qui est toujours possible par renommage, on construit $\mathcal{A} = (S_1 \cup S_2, T_1 \cup T_2, I_1 \cup I_2, F_1 \cup F_2)$.

$L(\mathcal{A}) = L_1 \cup L_2$.

Théorème Si $L_1, L_2 \subseteq A^*$ sont deux langages reconnaissables, alors $L_1 \cdot L_2$ est reconnaissable.

Soient $\mathcal{A}_1 = (S_1, T_1, I_1, F_1)$ et $\mathcal{A}_2 = (S_2, T_2, I_2, F_2)$ deux automates reconnaissant respectivement L_1 et L_2 . On construit $\mathcal{A} = (S, T, I, F_2)$ avec $S = S_1 \cup S_2$, $T = T_1 \cup T_2 \cup \{(s, a, i) \mid i \in I_2 \text{ et il existe } f \in F_1 \text{ tel que } (s, a, f) \in T_1\}$

et $I = \begin{cases} I_1 & \text{si } I_1 \cap F_1 = \emptyset \\ I_1 \cup I_2 & \text{sinon.} \end{cases}$

Alors $L(\mathcal{A}) = L_1 \cdot L_2$

Théorème Soit $L \subseteq A^*$ un langage reconnaissable, alors L^+ et L^* sont reconnaissables.

Soit $\mathcal{A} = (S, T, I, F)$ un automate reconnaissant L . On construit d'abord $\mathcal{A}_+ = (S, T_+, I, F)$ un automate reconnaissant L^+ : $T_+ = T \uplus \{(s, a, i) \mid i \in I \text{ et il existe } f \in F, (s, a, f) \in T\}$. On construit alors $\mathcal{A}_* = (S \uplus \{j\}, T_+, I \uplus \{j\}, F \uplus \{j\})$.

$L(\mathcal{A}_+) = L^+$ et $L(\mathcal{A}_*) = L^*$.