Automates et Langages

Formation Systèmes à Evénements Discrets

1^{re} édition
Janvier 2024

Société d'Automatique,
de Génie Industriel & de Productique

Plan

1. Langages
   - Définitions
   - Opérations
   - Expressions régulières
   - Propriétés

2. Automates à états
   - Définitions
   - Langages d'un automate
   - Propriétés
   - Opérations

Langages – Définitions

Lettres, alphabets et mots

Lettre

Suite indivisible de symboles.

Exemple : lettres possibles : a, dd, f3, α
Remarque : une lettre ne doit pas contenir d'autres lettres sinon il n'est plus possible de distinguer un mot d'une lettre.

Alphabet

Ensemble, non vide, a priori fini, de lettres, noté Σ.

Exemple : $\Sigma_1 = \{a, dd, f3, \alpha\}$

Mot

Suite (ou concaténation) de lettres d'un alphabet donné, notée s.

Exemple : Sur Σ_1 : $a, a.dd, f3.\alpha.\alpha.a$
Remarque : Un mot vide ne contenant aucune lettre est noté ε.

Lettres, alphabets et mots

Lettre

Suite indivisible de symboles.

Exemple : lettres possibles : a, dd, f3, α
Remarque : une lettre ne doit pas contenir d'autres lettres sinon il n'est plus possible de distinguer un mot d'une lettre.

Alphabet

Ensemble, non vide, a priori fini, de lettres, noté Σ.

Exemple : $\Sigma_1 = \{a, dd, f3, \alpha\}$

Mot

Suite (ou concaténation) de lettres d'un alphabet donné, notée s.

Exemple : Sur Σ_1 : $a, a.dd, f3.\alpha.\alpha.a$
Remarque : Un mot vide ne contenant aucune lettre est noté ε.

Lettres, alphabets et mots

Lettre

Suite indivisible de symboles.

Exemple : lettres possibles : a, dd, f3, α
Remarque : une lettre ne doit pas contenir d'autres lettres sinon il n'est plus possible de distinguer un mot d'une lettre.

Alphabet

Ensemble, non vide, a priori fini, de lettres, noté Σ.

Exemple : $\Sigma_1 = \{a, dd, f3, \alpha\}$

Mot

Suite (ou concaténation) de lettres d'un alphabet donné, notée s.

Exemple : Sur Σ_1 : $a, a.dd, f3.\alpha.\alpha.a$
Remarque : Un mot vide ne contenant aucune lettre est noté ε.

Langages et Σ^*

Langage

Ensemble de mots d'un alphabet donné, noté L.

Exemple : Sur $\Sigma_1, L = \{a, a.dd, f3.\alpha.\alpha.a\}$
Remarque : Langage vide : un langage ne contenant aucun mot est noté $\emptyset$ et le langage ne contenant que ϵ n'est pas vide.

Σ^*

Ensemble infini des mots construits sur l'alphabet Σ.

Exemple : Soit $\Sigma = \{a, b, c\}$:
$\Sigma^* = \{\epsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, ...\}$
Remarque :

- Tout mot s construit sur Σ est un élément de Σ^* : $s \subseteq \Sigma^*$
- Tout langage L construit sur Σ est un sous-ensemble de Σ^* : $L \subseteq \Sigma^*$

Langages et Σ^*

Langage

Ensemble de mots d'un alphabet donné, noté L.

Exemple : Sur $\Sigma_1, L = \{a, a.dd, f3.\alpha.\alpha.a\}$
Remarque : Langage vide : un langage ne contenant aucun mot est noté $\emptyset$ et le langage ne contenant que ϵ n'est pas vide.

Σ^*

Ensemble infini des mots construits sur l'alphabet Σ.

Exemple : Soit $\Sigma = \{a, b, c\}$:
$\Sigma^* = \{\epsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, ...\}$
Remarque :

- Tout mot s construit sur Σ est un élément de Σ^* : $s \subseteq \Sigma^*$
- Tout langage L construit sur Σ est un sous-ensemble de Σ^* : $L \subseteq \Sigma^*$

Langages – Opérations

Concaténation de mots

Concaténation de mots

Soient deux mots $s = s_1 s_2 ... s_p$ et $t = t_1 t_2 ... t_q$ définis sur sur l'alphabet Σ. On appelle produit (ou concaténation) le mot u tel que :

$$u = s.t = st = s_1 s_2 ... s_p t_1 t_2 ... t_q$$

Propriétés : Quels que soient les mots s, t et u définis sur l'alphabet Σ :

- $|s.t| = |s| + |t|$
- $(s.t).u = s.(t.u)$
- $s.\varepsilon = \varepsilon.s = s$

Soient $s, t, u \in \Sigma^*$:

- s est dit **préfixe** de t si et seulement si il existe un mot u tel que $s.u = t$
- s est dit **suffixe** de t si et seulement si il existe un mot u tel que $u.s = t$

Concaténation de langages et fermeture préfixielle

Concaténation de 2 langages

Soit $L_a L_b$ la concaténation de deux langages $L_a, L_b \subseteq \Sigma^*$:

$$L_a L_b = \{s \in \Sigma^* \mid s = s_a s_b \ \wedge \ s_a \in L_a \ \wedge \ s_b \in L_b\}$$

Exemple : Soient $L_a = \{abc, bac\}$ et $L_b = \{end, fin\}$:

$$L_a L_b = \{abcend, abcfin, bacend, bacfin\}$$

Fermeture préfixielle

Soit $Pref(L)$ (ou $\overline{L}$) la fermeture préfixielle de $L \subseteq \Sigma^*$:

$$Pref(L) = \{s \in \Sigma^* \mid \exists \ t \in \Sigma^* : st \in L\}$$

Exemple : Soient $\Sigma = \{a, c\}$ et $L = \{c, acc\}$:

$$Pref(L) = \{\varepsilon, a, c, ac, acc\}$$

Remarque : Un langage L est fermé par ses préfixes ssi $L = Pref(L)$.

Concaténation de langages et fermeture préfixielle

Concaténation de 2 langages

Soit L_aL_b la concaténation de deux langages $L_a, L_b \subseteq \Sigma^*$:

$$L_aL_b = \{s \in \Sigma^* \mid s = s_as_b \ \wedge \ s_a \in L_a \ \wedge \ s_b \in L_b\}$$

Exemple : Soient $L_a = \{abc, bac\}$ et $L_b = \{end, fin\}$:

$$L_aL_b = \{abcend, abcfin, bacend, bacfin\}$$

Fermeture préfixielle

Soit $Pref(L)$ (ou $\overline{L}$) la fermeture préfixielle de $L \subseteq \Sigma^*$:

$$Pref(L) = \{s \in \Sigma^* \mid \exists\ t \in \Sigma^* : st \in L\}$$

Exemple : Soient $\Sigma = \{a, c\}$ et $L = \{c, acc\}$:

$$Pref(L) = \{\varepsilon, a, c, ac, acc\}$$

Remarque : Un langage L est fermé par ses préfixes ssi $L = Pref(L)$.

Projection de mots

Projection d'un mot

Soient deux alphabets Σ_1 et Σ_2. La projection $P : \Sigma_1^* \to \Sigma_2^*$ est définie telle que :

$$\begin{array}{rcl} P(\varepsilon) & = & \varepsilon \\ P(e) & = & \begin{cases} e & \text{si } e \in \Sigma_2 \\ \varepsilon & \text{si } e \in \Sigma_1 \setminus \Sigma_2 \end{cases} \\ P(se) & = & P(s)P(e) \text{ pour } s \in \Sigma_1^* \text{ et } e \in \Sigma_1 \end{array}$$

Exemple : Soient $\Sigma_1 = \{a, b, c\}$ et $\Sigma_2 = \{a, b\}$:

$$P(abcab) = ab\varepsilon ab = abab$$

$$P(abcabc) = ab\varepsilon ab\varepsilon = abab$$

$$P(abcabcc) = ab\varepsilon ab\varepsilon\varepsilon = abab$$

Projection de mots

Projection d'un mot

Soient deux alphabets Σ_1 et Σ_2. La projection $P : \Sigma_1^* \to \Sigma_2^*$ est définie telle que :

$$\begin{array}{rcl} P(\varepsilon) & = & \varepsilon \\ P(e) & = & \begin{cases} e & \text{si } e \in \Sigma_2 \\ \varepsilon & \text{si } e \in \Sigma_1 \setminus \Sigma_2 \end{cases} \\ P(se) & = & P(s)P(e) \text{ pour } s \in \Sigma_1^* \text{ et } e \in \Sigma_1 \end{array}$$

Exemple : Soient $\Sigma_1 = \{a, b, c\}$ et $\Sigma_2 = \{a, b\}$:

$$P(abcab) = ab\varepsilon ab = abab$$

$$P(abcabc) = ab\varepsilon ab\varepsilon = abab$$

$$P(abcabcc) = ab\varepsilon ab\varepsilon\varepsilon = abab$$

Projection inverse de mots

Projection inverse d'un mot

La projection inverse $P^{-1} : \Sigma_2^* \rightarrow 2^{\Sigma_1^*}$ est définie telle que :

$$P^{-1}(t) = \{s \in \Sigma_1^* \mid P(s) = t\}$$

Note : $2^{\Sigma_1^*}$ représente l'ensemble de tous les sous-ensembles de Σ_1^*.

Exemple : Soient $\Sigma_1 = \{a, b, c\}$ et $\Sigma_2 = \{a, b\}$:

$$P^{-1}(\varepsilon) = c^*$$

$$P^{-1}(abab) = c^*ac^*bc^*ac^*bc^*$$

Projection inverse de mots

Projection inverse d'un mot

La projection inverse $P^{-1} : \Sigma_2^* \rightarrow 2^{\Sigma_1^*}$ est définie telle que :

$$P^{-1}(t) = \{s \in \Sigma_1^* \mid P(s) = t\}$$

Note : $2^{\Sigma_1^*}$ représente l'ensemble de tous les sous-ensembles de Σ_1^*.

Exemple : Soient $\Sigma_1 = \{a, b, c\}$ et $\Sigma_2 = \{a, b\}$:

$$P^{-1}(\varepsilon) = c^*$$

$$P^{-1}(abab) = c^* a c^* b c^* a c^* b c^*$$

Projection de langages

Projection d'un langage

Pour un langage L défini sur Σ_1^*, sa projection $P(L)$ sur Σ_2^* est définie par :

$$P(L) = \{t \in \Sigma_2^* \mid \exists\, s \in L : P(s) = t\}$$

Projection inverse d'un langage

Pour un langage L' défini sur Σ_2^*, sa projection $P^{-1}(L')$ sur Σ_1^* est définie par :

$$P^{-1}(L) = \{s \in \Sigma_1^* \mid \exists\, t \in L' : P(s) = t\}$$

Exemple : Soient $\Sigma_1 = \{a, b, c\}$ et $\Sigma_2 = \{a, b\}$:

$$P(\{c, abc, abcabc\}) = \{\varepsilon, ab\varepsilon, ab\varepsilon ab\varepsilon\} = \{\varepsilon, ab, abab\}$$

Remarque : $P[(P^{-1}(L)] = L$ mais $L \subseteq P^{-1}[P(L)]$

Projection de langages

Projection d'un langage

Pour un langage L défini sur Σ_1^*, sa projection $P(L)$ sur Σ_2^* est définie par :

$$P(L) = \{t \in \Sigma_2^* \mid \exists\ s \in L : P(s) = t\}$$

Projection inverse d'un langage

Pour un langage L' défini sur Σ_2^*, sa projection $P^{-1}(L')$ sur Σ_1^* est définie par :

$$P^{-1}(L) = \{s \in \Sigma_1^* \mid \exists\ t \in L' : P(s) = t\}$$

Exemple : Soient $\Sigma_1 = \{a, b, c\}$ et $\Sigma_2 = \{a, b\}$:

$$P(\{c, abc, abcabc\}) = \{\varepsilon, ab\varepsilon, ab\varepsilon ab\varepsilon\} = \{\varepsilon, ab, abab\}$$

Remarque : $P[(P^{-1}(L)] = L$ mais $L \subseteq P^{-1}[P(L)]$

Langages – Expressions régulières

Expressions régulières

Expressions régulières

Les expressions régulières sur Σ sont formées à partir des règles suivantes :

- $\emptyset$, ε et les lettres de Σ sont des expressions régulières,
- Si r_1 et r_2 sont des expressions régulières, alors $r_1.r_2$, $r_1 + r_2$, r_1^* et r_2^* sont des expressions régulières,
- Il n'existe pas d'autres expressions régulières que celles construites à partir des 2 règles précédentes appliquées un nombre fini de fois.

Remarque : Une expression régulière définit un langage quand les opérateurs « $+$ », « . » et « * » sont interprétés respectivement comme les opérateurs d'union, de concaténation et de fermeture itérative sur les ensembles.
Exemple : $b + a.b^*$ représente le langage $\{b\} \cup \{a\}.\{b\}^*$

Langages réguliers

Tout langage pouvant être défini par une expression régulière est appelé langage régulier.

Exemple : Soit le langage L construit sur $\Sigma = \{a, b\}$ comprenant les mots tels que a et b apparaissent alternativement et telles que a apparaît en premier.
Ce langage est régulier car il peut être décrit :

- par $L = (a.b)^*.(\varepsilon + a)$
- ou par $L = \varepsilon + a.(b.a)^*.(\varepsilon + b)$

Langages – Propriétés

Propriétés des langages

Langages réguliers, finis et infinis :

- Tout langage fini est un langage régulier,
- Un langage infini peut être un langage régulier (voir exemple précédent).

Langages et automates à états :

- Tout automate génère un langage régulier,
- Tout langage régulier est modélisable par un automate à états.

Tout langage généré par un automate à état est clos par ses préfixes (ou préfixe clos).

Plan

1. Langages
   - Définitions
   - Opérations
   - Expressions régulières
   - Propriétés

2. Automates à états
   - Définitions
   - Langages d'un automate
   - Propriétés
   - Opérations

Automates à états – Définitions

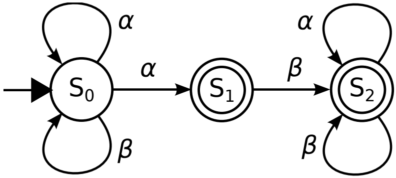
Automates à états

Un automate est défini par un 5-uplet $A = < X, \Sigma, \delta, X_0, X_m >$

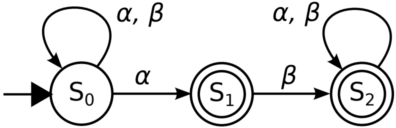
- X : ensemble appelé espace d'états de A
- Σ : ensemble appelé alphabet d'entrée de A
- δ : ensemble des transitions de A, $\delta \subseteq X \times \Sigma \times X$
- X_0 : ensemble des états initiaux, $X_0 \subseteq X$
- X_m : ensemble des états marqués, $X_m \subseteq X$

Représentation graphique d'un automate

Un automate peut être représenté par un multigraphe :



Le multigraphe peut être ramené à un graphe :



- États : $X = \{S_0, S_1, S_2\}$, $X_0 = \{S_0\}$, $X_m = \{S_1, S_2\}$
- Alphabet : $\Sigma = \{\alpha, \beta\}$
- Transitions : $\delta = \{ (S_0, \alpha, S_0), (S_0, \beta, S_0), (S_0, \alpha, S_1), (S_1, \beta, S_2), (S_2, \alpha, S_2), (S_2, \beta, S_2)\}$

Automate déterministe

Un automate déterministe est défini par un 5-uplet $A =< X, \Sigma, \delta, x_0, X_m >$

- X : ensemble appelé espace d'états de A
- Σ : ensemble appelé alphabet d'entrée de A
- δ : fonction partielle de transition de A, $\delta : X \times \Sigma \to X$
- x_0 : état initial, $x_0 \in X$
- X_m : ensemble des états marqués, $X_m \subseteq X$

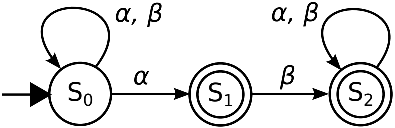
Automates à états – Langages d'un automate

Langage généré par un automate

- Le langage $L(A)$ représente l'ensemble des mots généré par A à partir de l'état initial :

$$L(A) = \{s \in \Sigma^* \mid \delta(x_0, s)!\}$$

Exemple : Soit l'automate A_1 :



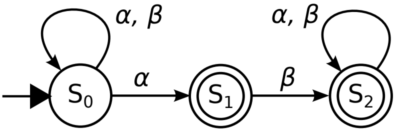
- $L(A_1) = (\alpha+\beta)^*(\varepsilon+\alpha+\alpha\beta)(\alpha+\beta)^*$

Langage marqué

- Le langage $L_m(A) \subseteq L(A)$ marqué par A contient tous les mots tels qu'à partir de l'état initial, la fonction de transition étendue contienne ce mot et mène à un état marqué :

$$L_m(A) = \{s \in \Sigma^* \mid \delta(x_0, s) \in X_m\}$$

Exemple : Soit l'automate A_1 :



- $L(A_1) = (\alpha+\beta)^*(\varepsilon+\alpha+\alpha\beta)(\alpha+\beta)^*$
- $L_m(A_1) = (\alpha+\beta)^*(\alpha+\alpha\beta)(\alpha+\beta)^*$

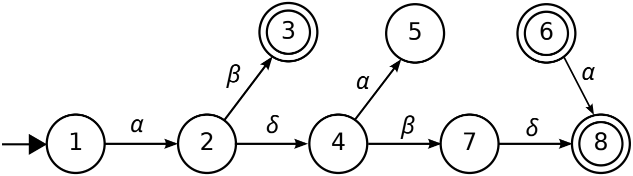
Automates à états – Propriétés

Etats utiles

Soit $A =< X, \Sigma, \delta, x_0, X_m >$:

- Un état q de A est accessible à partir d'un état p s'il existe une trajectoire dans A dont l'origine est p et dont q est l'extrémité. L'état q est dit accessible s'il est accessible à partir de l'état initial ;
- Un état p est co-accessible à un état q s'il existe une trajectoire dans A dont l'origine est p et dont q est l'extrémité. L'état p est dit co-accessible s'il est co-accessible à un état marqué ;
- Un état p est dit utile s'il est à la fois accessible et co-accessible.

Exemple :

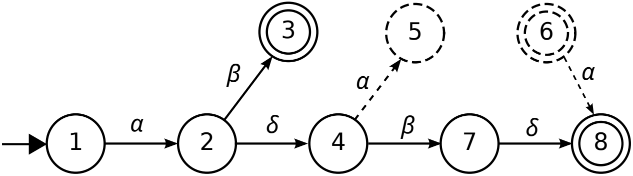


Automates émondés

Soit $A =< X, \Sigma, \delta, x_0, X_m >$:

- L'automate A est accessible (resp. co-accessible) si tous les états le sont ;
- L'automate A est émondé s'il est à la fois accessible et co-accessible, c'est-à-dire que tous les états sont utiles ;
- L'automate $\mathrm{ac}(A)$ (resp. $\mathrm{coac}(A)$) est l'automate obtenu à partir de A en supprimant tous les états non accessibles (resp. co-accessibles) ;
- L'automate $\mathrm{em}(A)$ est l'automate obtenu à partir de A en supprimant tous les états non accessibles ou non co-accessibles.

Exemple :



Automates à états – Opérations

La composition d'automates

Composition d'automates

Soient $A_1 = (X_1,\ \Sigma_1, \delta_1,\ x_{0,1},\ X_{m,1})$ et $A_2 = (X_2,\ \Sigma_2, \delta_2,\ x_{0,2},\ X_{m,2})$:

$$A_1 \parallel A_2 = ac(X_1 \times X_2,\ \Sigma_1 \cup \Sigma_2,\ \delta,\ (x_{0,1}, x_{0,2}),\ X_{m,1} \times X_{2,m})$$

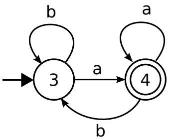
avec δ définie de la façon suivante :

$$\delta((x_1, x_2),\ \sigma) = \begin{cases} (\delta_1(x_1, \sigma),\ \delta_2(x_2,\ \sigma)) & \text{si } \delta(x_1,\ \sigma)! \text{ et } \delta(x_2,\ \sigma)! \\ (\delta_1(x_1, \sigma), x_2) & \text{si } \delta(x_1, \sigma)! \text{ et } \sigma \notin \Sigma_2 \\ (x_1,\ \delta_2(x_2,\ \sigma)) & \text{si } \sigma \notin \Sigma_1 \text{ et } \delta(x_2, \sigma)! \\ \text{non défini} & \text{sinon} \end{cases}$$

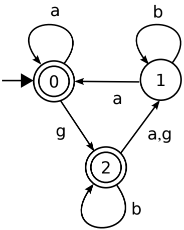
Note : « $\delta(q_1, \sigma)!$ » se lit « $\delta(q_1, \sigma)$ existe ».

Exemple de composition d'automates

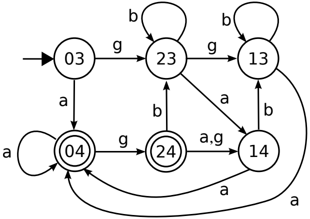
A_1



A_2



$A_1 \parallel A_2$



Le produit d'automates

Automates

Soient $A_1 = (X_1,\ \Sigma_1, \delta_1,\ x_{0,1},\ X_{m,1})$ et $A_2 = (X_2,\ \Sigma_2, \delta_2,\ x_{0,2},\ X_{m,2})$:

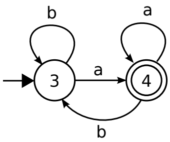
$$A_1 \times A_2 = ac(X_1 \times X_2,\, \Sigma_1 \cup \Sigma_2,\, \delta,\, (x_{0,1}, x_{0,2}),\, X_{m,1} \times X_{2,m})$$

avec δ définie de la façon suivante :

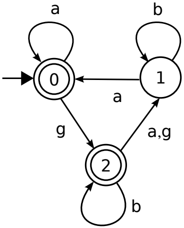
$$\delta((x_1, x_2), \sigma) = \begin{cases} (\delta_1(x_1, \sigma),\ \delta_2(x_2, \sigma)) & \text{si } \delta(x_1, \sigma)! \text{ et } \delta(x_2, \sigma)! \\ \text{non défini} & \text{sinon} \end{cases}$$

Exemple de produit d'automates

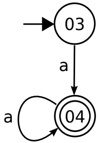
A_1



A_2



$A_1 \times A_2$



Références bibliographiques

Références bibliographiques

[1] Christos G Cassandras et Stéphane Lafortune.
Introduction to Discrete Event Systems. Springer, 2008.

[2] Yliès Falcone et Jean-Claude Fernandez.
Automates à états finis et langages réguliers. collection infoSup. Dunod, juill. 2020. isbn : 9782100808465.

Équipe pédagogique

Équipe pédagogique

Auteur.rice.s : Pascale Marangé, Alexandre Philippot, Laurent Piétrac, Ramla Saddem

Intervenant : Laurent Piétrac