AAL

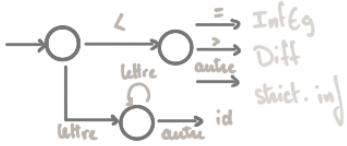
Introduc°

Automates : au départ pour l'analyse de langues
↳ découpage syntaxiq & traduc° automatiq
& traitem^t de txt

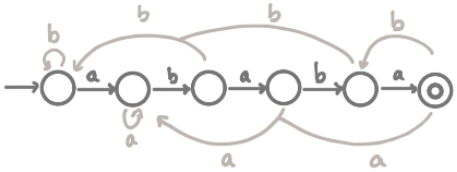
Analyse lexicale : 1ère étape vers la compila° de programme.
=> Découpe le programme en lexème
unité lexicale : mots-clefs, id, ==, <...

ex. Langage : <, ≤, <>, >, ≥, =

x1 ≤ ab2c



recherche de motif : abababab



hiérarchie des modèles de calculs
- automates finis (nb fini d'état)
- grammaire hors contexte => S -> AB => S -> AB -> aAB -> ...
- " générale A -> aA
- machine de Turing B -> bb

Alphabet

C'est un ensemble fini de symboles. Un mot de l'alphabet Σ.
C'est une sequence finie & ordonnée (ou vide ε) d'élém^t de Σ.

- Longueur n de w : $n = |w|$
- nb d'occurence d'une lettre x : $|w|_x$
- posi° i d'une lettre : w_i

Langage : ensemble de mots sur l'alphabet Σ

ex : $L = \{ab\}$ => 1 mots
$L = \{u \in \Sigma^* \mid |u|_a \equiv 0 \mod 2\}$
$L = \{a^n b^n \mid n \in \mathbb{N}\}$

concatena° : $u.v$; $|uv| = |u| + |v|$
Σ^* : ensemble des mots formés sur Σ.
↳ monoïde libre engendré par ∘
↳ · associative : $(u.v).w = u.(v.w)$
· élém^t neutre $\varepsilon \in \Sigma^*$

facteur : $v = w_g.u.w_d$, si $w_g = \varepsilon$ => u est prefixe (inv. w_d)
Sous-mot : u est un sous mot si en enlevant des lettres on a u

Opera° sur les langage

Rappel : $A \cup B$
$A \cap B$

si $L \subset \Sigma$, le complém^t de L noté $\overline{L}$

Si L & L' sont deux langages sur Σ, leur concatena° est $L.L' = \{ u.v \mid u \in L \text{ & } v \in L' \}$

ex. $\{a, aa\} \cdot \{b, bb\} = \{ab, abb, aab, aabb\}$

$\{a, aa\} \cdot \{a, aa\} = \{aa, aaa, aaaa\}$

$\{a^{2n} \mid n \in \mathbb{N}\} \cdot \{a^{2n+1} \mid n \in \mathbb{N}\} = \{a^{2n+1} \mid n \in \mathbb{N}\}$

$\emptyset . L = \emptyset \neq \{\varepsilon\} . L = L$

$L^n = \underbrace{L.L.L}_{n}$ $\quad L^0 = \{\varepsilon\}$

Etoile de K : $L^* = \underbrace{L^1 \cup L^2 \cup L^3 \ldots}_{*\text{ fois}}$
↳ $\{aa\}^* = \{\varepsilon, aa, a^4, a^6, \ldots\} = \{a^{2n} \mid n \in \mathbb{N}\}$

Def : L est ration^l, s'il s'obtient à partir de $\emptyset$, $\{x\}$ par un nb fini d'opera° $\cup$, . & *

ex : $Z = \{a,b\} . [\{a\} \cup \{b\} . \{a\} \cup \{b\}]^* \mid u$

On enlève les accolades & $u \Rightarrow +$
.

L est [...] le + petit ens. de langage contenant $\emptyset$ & $\{x\}$ pour tout $x \in \Sigma$

AAL

Automate fini déterministe

Un afd $\mathcal{A}$ est défini par : $\mathcal{A}(\Sigma, Q, q_0, F, \delta)$
- Un alphabet Σ
- L'ens. des états Q
- Un état initial $q_0 \in Q$
- L'ens. des états finaux $F \subset Q$
- La fonc° de transi° $\delta : Q \times \Sigma \to Q \cup \{\emptyset\}$

$\Rightarrow q' = \delta(q,a)$

on peut noter δ :

$\delta(0,a)=1 \quad \delta(0,b)=0$
$\delta(1,a)=2 \quad \delta(1,b)=1$
$\delta(2,a)=0 \quad \delta(2,b)=2$

$\Leftrightarrow$

	a	b
0	1	0
1	2	1
2	0	2

Si une transi° n'est pas définie : on dit q le calcul bloq

- fonc° de transi° étendue :

$\delta : Q \times \Sigma \to Q \cup \{\emptyset\} \Rightarrow \delta^* : Q \times \Sigma^* \to Q \cup \{\emptyset\}$ par

- $\delta^*(q, \varepsilon) = q$ pour tout etat
- $\delta^*(q, x_1 \dots x_n) = \begin{cases} \emptyset & \text{si } \delta(q,x_1)=\emptyset \\ \delta^*(\delta(q,x_1), x_2 \dots x_n) & \text{sinon} \end{cases}$

$\delta^*(q,w)=q'$: en partant de l'état q, on effectue le calcul du mot w sans bloquer et on arrive à l'etat q'

On dit qu'un mot $w \in \Sigma^*$ est accepté par si $\delta^*(q_0, w) \in F$. L'ens. $\mathcal{L}(\mathcal{A})$ des mots accep^t = langage reconnu par l'automate $\mathcal{A}$

Un langage L est reconnaissable si $\exists$ un automate $\mathcal{A}$ tel q $L = \mathcal{L}(\mathcal{A})$
↳ tout langage fini est reconnaissable

Un afd est complet, si $\forall$ etat q et $\forall$ lettre $a \in \Sigma$ $\delta(q,a) \neq \emptyset$

Automate fini non déterministe

Un afnd $\mathcal{A}$ est défini par : $\mathcal{A}(\Sigma, Q, Q_0, F, \delta)$
- Un alphabet Σ
- L'ens. des états Q
- L'ens. des états initiaux $Q_0 \subset Q$
- L'ens. des états finaux $F \subset Q$
- La fonc° de transi° $\delta : Q \times \Sigma \to \mathcal{P}(Q)$

ensemble des parties de Q

$\Rightarrow$ Pour un mot, on obtient une liste d'état possible
↳ un mot accep^t si $\delta^*(q_0, w)$ contient au-1 etat final.

Th. $\forall$ afnd $\mathcal{A}$, $\exists$ un afd $\mathcal{A}'$ t.q $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$

Propriété de clôture de Rec

- Rat : + petit ens. de langage contenant $\emptyset, \{x\}$, pour $x \in \Sigma$, et clos par $\cup, \cdot, *$

clos
↓
$L \cup L' \in Rat$
$L \cdot L' \in Rat$
$L^* \in Rat$

- But : $Rec \subseteq Rat$

complémentaire, si $L \in Rec$ alors ${}^cL \in Rec$
↳ Afd $\mathcal{A}$ pour L et $\mathcal{A}'$ pour ${}^cL$

1. Compléter $\mathcal{A}$ (états puits)
2. On inverse ET et CNT

---

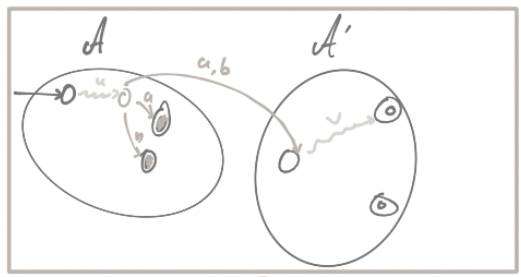
union : $L, L' \in Rec \Rightarrow L \cup L' \in Rec$
$\mathcal{A}, \mathcal{A}' \longrightarrow B$ pour $L \cup L'$

---

intersec° : $L \cap L' = {}^c({}^cL \cup {}^cL')$

---

concatena° : L, L'



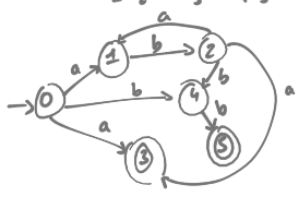
B

Etoile :

Algorithme de Glushkov (ER => AFND)

- linearisé l'ER
- transformer l'ERL en AFND
- passer de l'AFND pour l'ER linearisée à un AFND pour l'ER initial

ex : $e = (ab)^*(a+bb)$
1. $e' = (x_1x_2)^*(x_3 + x_4x_5)$



2.

	Succ
0	1, 3, 4
1	2
2	1,3,4
3	/
4	5
5	/

3.

	a	b
0	1,3	4
1	/	2
2	1,3	4
3	/	/
4	/	5
5	/	/

AAL

Lemme de l'etoile

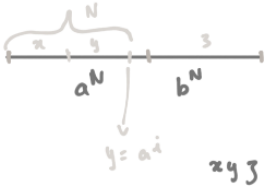
Limites des langages reconnaissable

$xyz \in L \Rightarrow xy^*z \in L$

Si $L \in Rec$ alors $\exists N \in \mathbb{N} \iff \forall u \in L, |u| \geq N$, il existe un découpage $u = xyz$ tq $\begin{cases} y \neq \varepsilon \\ |xy| \leq N \text{ tq } \forall k \in \mathbb{N} \quad xy^kz \in L \end{cases}$

ex. $L = \{a^n b^n \mid n \in \mathbb{N}\}$

si $L \in Rec$ alors le lemme donne N.
soit $u = a^N b^N \in L$, $|u| \geq N$

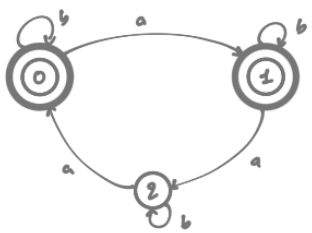


$xyz = a^{N+i} b^N \notin L$

Arden (A → er)

- $L = A.L + B$ a pour solu^{e} : $L = A^*B$ unic si $\varepsilon \notin A$

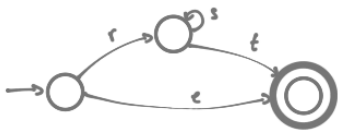
ex



$L_0 = bL_0 + aL_1 + \varepsilon$
$L_1 = bL_1 + aL_2 + \varepsilon$
$L_2 = bL_2 + aL_0$

états terminaux

Brzozowski - McCluskey (A → er)



er : $e + rs^*t$

- pour simplifier on ajoute un état i relié aux ε_i par ε-transi^{s} de $\hat{m}$ avec les ε_j

Classe d'equivalence

$u \sim_L v$ si $\forall w$, $uw \in L \iff vw$: - si $uw \in L \iff vw \in L$
- si $uw \notin L \iff vw \notin L$

un nb de classe d'eq infini $\Rightarrow$ pas Rec

Residu^{L} + Myhill-Nerode (er → AFD min)

- $u \sim_L v$ ssi $u^{-1}L = v^{-1}L$
- si $L \in Rec \Rightarrow$ AFD des residu^{L} est min, comp, uniq

On utilise l'arbre des residu^{L} pour trouver tous les états selon :

- $e = \varepsilon \Rightarrow x^{-1}L = \emptyset$ - $e = y, y \neq x \Rightarrow x^{-1}L = \emptyset$
- $e = x \Rightarrow x^{-1}L = \varepsilon$
- $e = e_1^* \Rightarrow x^{-1}L = (x^{-1}e_1).e_1^*$
- $e = e_1 + e_2 \Rightarrow x^{-1}L = (x^{-1}L_1) + (x^{-1}L_2)$
- $e = e_1.e_2 \Rightarrow x^{-1}L = \begin{cases} (x^{-1}e_1).e_2 & \text{si } \varepsilon \notin e_1 \\ (x^{-1}e_1)e_2 + x^{-1}e_2 \end{cases}$

Moore (minimisa^{s})

- On raffine une partit^{s} C des Etats
- On part de $C = \{Q \setminus F, F\}$
- On sépare les groupes tant q c'est possible
  - ↳ on regarde ds q^{l} groupe on arrive avec chaq lettre a regroupe ceux avec le $\hat{m}$ comportem^{t}

Brzozowski (Min)

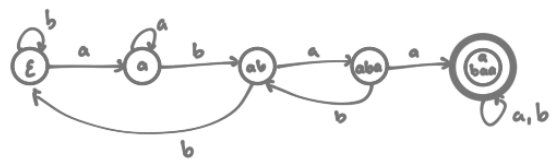
det(mir(det(mir(A)))) avec A => AFD comp

Algo de Knut - Morris Pratt (résidu^{L} motif)

$\tilde{u}$: + long suffixe de u q est prefixe du motif m

=> C.E : $[\varepsilon], [m_1], \ldots [m_1 \ldots m_n]$

ex



- $\widetilde{\varepsilon b}$: le + long suffixe de b q est prefixe de m = ε
- $\widetilde{\varepsilon a}$: ——— a ——— = a
- $\widetilde{aa}$: ——— aa ——— = a
- $\widetilde{ab}$: ——— ab ——— = ab