La famille Rat(X*)

- **Définition**: La famille Rat(X*) est la plus petite famille de langages contenant les parties finies de X* et fermée par union, produit et étoile.

  - il en découle que Rat(X*) est l'intersection de toutes les familles de langages possédant ces propriétés.

- **Propriété**: on peut donner une construction de cette famille :

  - prenons pour R_0 l'ensemble des parties finies de X*
  - soit pour $i>0$,
    $R_i = R_{i-1} \cup \{L^*, L \cup M, L.M \,/\, L \in R_{i-1} \text{ et } M \in R_{i-1}\}$

  La proposition $\mathbf{Rat(X^*) = \bigcup_{i\geq 0} R_i}$ a sa preuve dans le poly p47-48

Langages rationnels

- On dira qu'un langage est rationnel s'il appartient à la famille Rat(X*)

- Tout langage rationnel possède une expression sous la forme d'un nombre **fini** d'unions, produits et étoiles de parties finies.

$$L = ((\{ab , aa\} ^* \cup \{a , b\})^* . \{aaa , ba\} ^*) \in R_4$$

Constructions similaires

- Soit F_0 un ensemble de variables propositionnelles, on peut construire l'ensemble F des formules défini comme le plus petit ensemble contenant F_0 et contenant les formules $(A \wedge B)$, $(A \vee B)$, $(A \Rightarrow B)$ et $\neg A$ pour toutes formules A et B de F.

  - soit pour $i>0$,
    $F_i = F_{i-1} \cup \{\neg A, (A \wedge B), (A \vee B), (A \Rightarrow B) \,/\, A \in F_{i-1} \text{ et } B \in F_{i-1}\}$

  On a $F = \bigcup_{i \geq 0} F_i$

- Soit R une relation binaire, on désigne par R* la fermeture réflexive et transitive de R. Cette nouvelle relation est constructible sur le même modèle. De plus, si R est une relation finie (graphe fini), l'union à calculer est une union finie (algorithme de Warshall)

Théorème de Kleene

- **Rec(X*) = Rat(X*)**

  - sens Rat(X*) $\subseteq$ Rec(X*) : on a déjà vu que Rec(X*) contient les parties finies et est fermée par union, produit et étoile,

    donc elle contient la plus petite famille contenant les parties finies et fermée par union, produit et étoile : Rat(X*).

  - sens Rec(X*) $\subseteq$ Rat(X*)

    algorithme de M^{c} Naughton & Yamada ou résolution de systèmes d'équations (cf TD7).

Construction des expressions rationnelles à partir de l'automate

- L_{pq}^{P} = {f $\in$ X* / f est trace d'un chemin menant de p à q n'utilisant que P comme états intermédiaires}

  - si card(P)=0 : $L_{pq}^{P} \subseteq X \cup \{\varepsilon\}$; il est donc rationnel

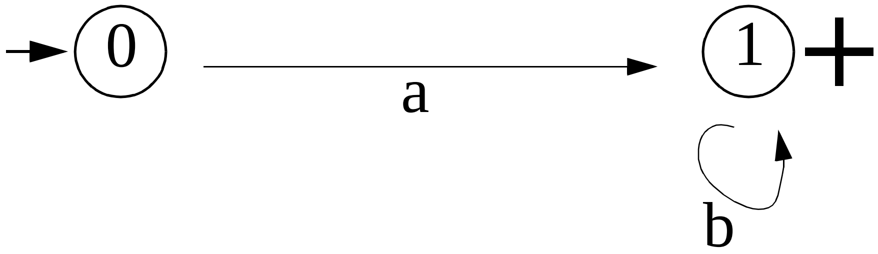
  - si card(P)>0 : L_{pq}^{P} se définit en utilisant un nombre fini d'opérations d'union, produit et étoile et de $L_{p'q'}^{P'}$ avec $P' \subset P$

  - par induction sur la taille de P, tous ces langages sont rationnels

- $L(\mathcal{A}) = \bigcup_{q\in F} L_{q_0q}^{Q}$ est donc rationnel

Application de l'algorithme de Mc Naughton & Yamada

- on peut calculer le langage reconnu par cet automate



$L_{00}^{\varnothing} = \{\varepsilon\}$

$L_{01}^{\varnothing} = \{a\}$

$L_{10}^{\varnothing} = \varnothing$

$L_{11}^{\varnothing} = \{b, \varepsilon\}$

se voit en consultant δ

- $L(A) = L_{01}^{\{0,1\}}$

$= (L_{00}^{\{1\}})^* \,.\, L_{01}^{\varnothing} \,.\, (L_{11}^{\varnothing})^*$

$= (L_{00}^{\varnothing} \cup L_{01}^{\varnothing} \,.(L_{11}^{\varnothing})^* \,.\, L_{10}^{\varnothing})^* \,.\, L_{01}^{\varnothing} \,.\, (L_{11}^{\varnothing})^*$

$= (\{\varepsilon\} \cup \{a\} \,.\, \{b, \varepsilon\}^* \,.\, \varnothing)^* \,.\, \{a\} \,.\, \{b, \varepsilon\}^*$

$= (\varnothing)^* \,.\, \{a\} \,.\, \{b\}^*$

$= \{a\}.\{b\}^*$

Expressions rationnelles

- X étant un alphabet, **ER(X)** est le plus petit ensemble de mots sur $X \cup \{ (,) , + , . , * , 0 , 1 \}$ vérifiant :

  - $0 \in ER(X)$

  - pour tout $x \in X$, on a $x \in ER(X)$

  - pour tous $e_1, e_2 \in ER(X)$, on a $(e_1 + e_2) \in ER(X)$

    $(e_1 . e_2) \in ER(X)$

    et $e_1{}^* \in ER(X)$

- Comme pour Rat(X*), on peut donner une construction de ER(X)

- On appelle ER(X) l'ensemble des expressions rationnelles définies sur X.
- Par raccourci de notation, 1 désigne 0*
- Langage dénoté par une expression rationnelle :

  Le langage $|e|$ dénoté par l'expression e est défini inductivement sur la structure de l'expression e :

  - 0 dénote le langage vide,
  - pour tout $x \in X$, l'expression x dénote le langage $\{ x \}$
  - pour tous $e_1, e_2 \in ER(X)$,

    l'expression $(e_1 + e_2)$ dénote le langage $|e_1| \cup |e_2|$

    l'expression $(e_1 . e_2)$ dénote $|e_1|.|e_2|$

    l'expression e_1* dénote $|e_1|*$

Exemples

- Que dénote l'expression ((a+b).0*)* ?

- Tout langage rationnel possède une expression rationnelle le dénotant

$$L = (\{ab , aa\} ^* \cup \{a , b\})^* . \{aaa , ba\}^*$$

est dénoté par l'expression

$$(\ (((a.b) + (a.a))^* + (a + b))^* \ . \ ((a.(a.a)) + (b.a))^* \ )$$

soit, après suppression des parenthèses inutiles,

$$((ab + aa)^* + a + b)^* \ . \ (aaa + ba)^*$$

Équivalence d'expressions rationnelles

- e_1 et $e_2 \in$ ER(X) sont dites équivalentes si $|e_1| = |e_2|$
- on a :

  - $(e_1 + e_2) \equiv (e_2 + e_1)$
  - $((e_1 + e_2) + e_3) \equiv (e_1 + (e_2 + e_3))$
  - $((e_1 . e_2) . e_3) \equiv (e_1 . (e_2 . e_3))$
  - $(e_1 . (e_2 + e_3)) \equiv ((e_1 . e_2) + (e_1 . e_3))$ etc...

Résiduels

- soit L un langage sur X* et u un mot de X*, on appelle résiduel de L par rapport à u , et on le note $u^{-1}.L$, le langage

$$u^{-1}. L = \{ v \in X^* / u . v \in L \}$$

(parfois noté dL/du dans la biblio)

- on a $(u . v)^{-1} . L = v^{-1} . (u^{-1} . L)$

- on étudie deux exemples :

$L_{pi} = \{f \in \{a, b\}^*, |f|_a$ est pair et $|f|_b$ est impair$\}$

puis $L = \{a^n b^n, n \geq 0\}$

Exemple 1

- soit $L = L_{pi} = \{f \in \{a, b\}^*, |f|_a$ est pair et $|f|_b$ est impair$\}$, on a :

$\varepsilon^{-1}.L = \{f \in \{a, b\}^*, |f|_a$ est pair et $|f|_b$ est impair$\} = L_{pi}$

$a^{-1}.L = \{f \in \{a, b\}^*, |f|_a$ est impair et $|f|_b$ est impair$\} = L_{ii}$

$b^{-1}.L = \{f \in \{a, b\}^*, |f|_a$ est pair et $|f|_b$ est pair$\} = L_{pp}$

$(ab)^{-1}.L = \{f \in \{a, b\}^*, |f|_a$ est impair et $|f|_b$ est pair$\} = L_{ip}$

$(abbb)^{-1}.L = (bbab)^{-1}.L = (babbbb)^{-1}.L = \ldots = (ab)^{-1}.L$

L_{pi} possède exactement 4 résiduels distincts deux à deux.

Exemple 2

- soit $L = \{a^n b^n, n \geq 0\}$

$(ab)^{-1}.L = (a^2b^2)^{-1}.L = (a^5b^5)^{-1}.L = \ldots = \{\varepsilon\}$: L possède une infinité de résiduels identiques

$a^{-1}.L = \{a^m b^{m+1}, m \geq 0\}$ son mot le plus court est b^1
$(aa)^{-1}.L = \{a^m b^{m+2}, m \geq 0\}$ son mot le plus court est b^2
$(a^k)^{-1}.L = \{a^m b^{m+k}, m \geq 0\}$ son mot le plus court est b^k
...

L possède une infinité de résiduels distincts deux à deux

Critère de reconnaissabilité

- notation : l'ensemble des résiduels d'un langage L est noté *Res*(L)

$$Res(L) = \{u^{-1}.L\ ,\ u \in X^*\}$$

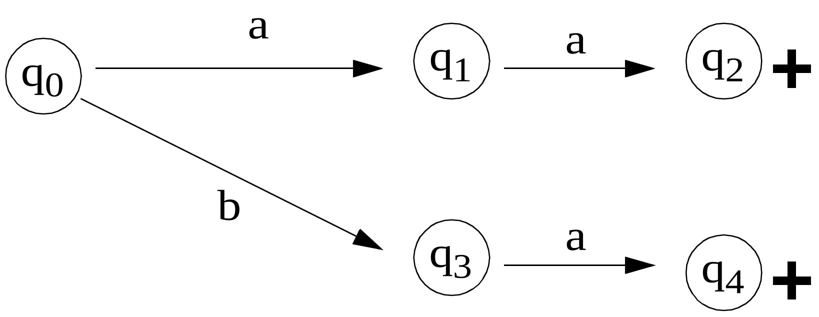
par exemple, $Res(L_{pi}) = \{L_{pi}, L_{pp}, L_{ii}, L_{ip}\}$

- Proposition : un langage est reconnaissable si et seulement si il possède un nombre fini de résiduels distincts.

Preuve du sens "seulement si"

- Soit L un langage reconnaissable
- Soit $\mathcal{A}$=< X, Q, q_o, F, δ > un aut. fini déterministe complet le reconnaissant,

pour q ∈ Q, notons $L_{\mathbf{q}} = \{ v \in X^* \,/\, \hat{\delta}(q,v) \in F \}$ l'ensemble des mots reconnus par $\mathcal{A}$ " à partir de l'état q"

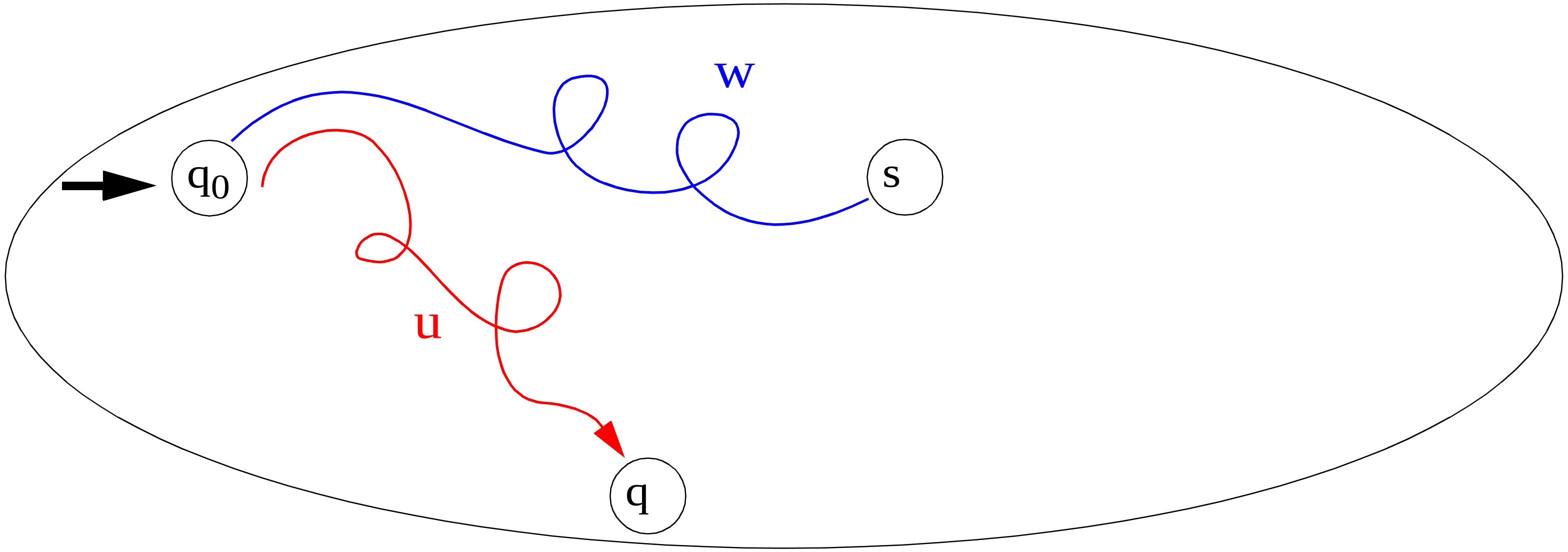


$$\mathbf{L_{q_1} = L_{q_3} = \{a\}}$$

$$\mathbf{L_{q_2} = L_{q_4} = \{\varepsilon\}}$$

Lemme : $\textit{Res}(L) = \{L_q \,/\, q \in Q,\ q \text{ accessible}\}$

- idée :



- $u \in X^*$ $\rightarrow$ $q = \hat{\delta}(q_o , u)$

  alors $u^{-1} . L = L_q$

- $s \in Q$, accessible $\rightarrow$ $w \in X^*$ tel que $\hat{\delta}(q_o , w) = s$

  alors $L_s = w^{-1} . L$

- on a ainsi prouvé : si un langage L est reconnaissable alors il possède un nombre fini de résiduels distincts

- de plus, pour tout langage reconnaissable L et pour tout a.f.d.c. $\mathcal{A}$=< X, Q, q_o, F, δ > reconnaissant L, on a

$$Res(\mathrm{L}) = \{\mathbf{L_q} \;/\; q \in Q,\ q \text{ accessible}\}$$

donc, pour tout a.f.d.c.accessible reconnaissant L on a

$$|Q| \geq |\{\ \mathbf{L_q} \;/\; q \in Q\ \}| = |Res(\mathrm{L})|$$

Preuve du sens " si" : automate des résiduels

Soit L un langage possédant un nombre fini de résiduels

distincts, posons $\mathcal{A}^r = < X, Q, q_o, F, \delta >$ avec

$Q = \{ [u^{-1} . L] \,/\, u \in X^* \} =$ *Res*(L)

$q_o = [\varepsilon^{-1} . L] = L$

$F = \{[u^{-1} . L] \,/\, \varepsilon \in u^{-1} . L \}$

$\delta([u^{-1} . L], x) = [(u x)^{-1} . L]$ $= [x^{-1} . (u^{-1} . L)]$ pour le calcul

On vérifie ultérieurement la pertinence de cette définition et on montre que cet automate reconnaît L

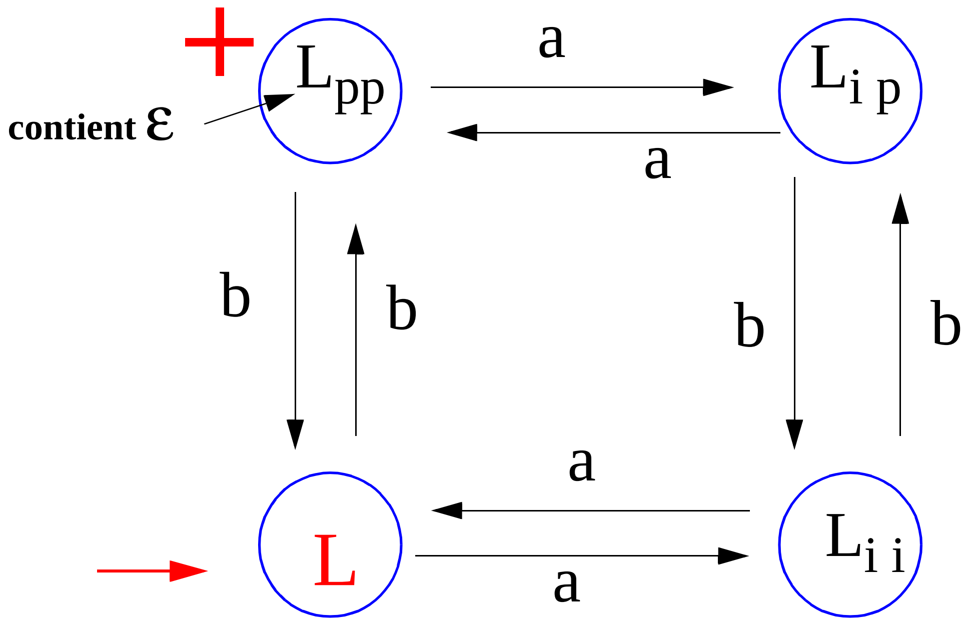
Application à l'exemple 1

$L = \{f \in \{a, b\}^*, |f|_a \text{ est pair et } |f|_b \text{ est impair}\}$

$a^{-1}.L = \{f \in \{a, b\}^*, |f|_a \text{ est impair et } |f|_b \text{ est impair}\} = L_{i\,i}$

$b^{-1}.L = \{f \in \{a, b\}^*, |f|_a \text{ est pair et } |f|_b \text{ est pair}\} = L_{pp}$

$a^{-1}.L_{pp} = \{f \in \{a, b\}^*, |f|_a \text{ est impair et } |f|_b \text{ est pair}\} = L_{i\,p}$



Pertinence de cette définition

- (1) on a bien défini un afdc :

  $Q = \{ [u^{-1} . L] \ / \ u \in X^* \} = Res(L)$ est fini par hypothèse !!!

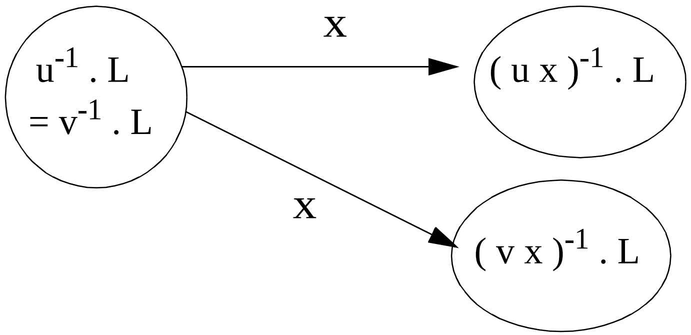
  $$\delta([u^{-1} . L], x) = [(u\,x)^{-1} . L]$$
  $$= [\ x^{-1} . (u^{-1} . L)\ ]$$ lors du calcul

  ne dépend pas du choix de u ! (donc A^r est déterministe)

- (2) pour tout mot u de X*, $\hat{\delta}(q_o , u) = [u^{-1} . L]$

- (3) cet automate reconnaît L

(1)



cette situation ne peut pas se produire

- Si $u^{-1} . L = v^{-1} . L$, alors pour tout $x \in X$, $(u\,x)^{-1} . L = (v\,x)^{-1} . L$,

  - donc $\delta ([u^{-1} . L] , x) = [(u\,x)^{-1} . L]$ ne dépend pas du choix de u

  - donc δ est une fonction totale : $Q \times X \rightarrow Q$,

  - ainsi $\mathcal{A}^r$ est un a.f.déterministe complet

(2)

pour tout mot u de X*, $\hat{\delta}(q_o , u) = [u^{-1} . L]$

se montre par récurrence sur |u| en prenant u = v.x

- ainsi A^r est un a.f.déterministe complet accessible

(3)

Cet automate reconnaît L

pour tout mot u de X*,

$\hat{\delta}(q_o , u) \in F$

ssi $[u^{-1} . L] \in F$ (prop 2)

ssi $\varepsilon \in u^{-1} . L$ (définition de F)

ssi $u \in L$ (définition de $u^{-1} . L$)

- on a ainsi prouvé : si un langage L possède un nombre fini de résiduels distincts, alors il est reconnaissable

Le problème ...

- ce n'est pas forcément facile de calculer TOUS les résiduels d'un langage
- ce n'est pas forcément facile de s'assurer qu'ils sont distincts deux à deux
- exemples

  - L= $\{f \in \{a, b\}^*$, f contient le facteur abaa$\}$

  - L= $\{f \in \{0, 1\}^*$, f est l'écriture en base 2 d'un multiple de 3$\}$