2.16 Caractérisation avec MSO

Definition 34 *Une formule MSO φ close définit le langage* $\{u \in \Sigma^+ \mid u, \lambda_1, \lambda_2 \models \varphi\}$.

Le but de cette section est de démontré le théorème suivant :

Theorem 13 *Un langage L qui ne contient pas ϵ est rationnel ssi il est défini par une formule de MSO.*

On démontre les deux sens.

Proposition 34 *Si L est reconnu par un automate non-déterministe $\mathcal{A}$, alors on construit une formule MSO φ en temps polynomial en $\mathcal{A}$ qui définit L.*

Proof.
$\boxed{\text{enlever les } \epsilon\text{-transitions}}$
Si jamais, l'automate contient des ϵ-transitions, on le convertit au préalable en un automate sans ϵ-transitions. Ca se fait en temps polynomial. Soit $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ l'automate non-déterministe sans ϵ-transitions qui reconnaît L. On note l'ensemble des états par $Q = \{q_0, q_1, \dots, q_{n-1}\}$.

$\boxed{\text{idée générale}}$ On utilise le fait que MSO est proche de la langue naturelle où on peut exprimer 'je veux tous les mots acceptés par l'automate $\mathcal{A}$'. On construit une formule φ de la forme

$$\exists X_0 \dots \exists X_{n-1} \psi(X_0, \dots, X_{n-1}),$$

qui dit 'le mot est tel qu'il existe un calcul acceptant de $\mathcal{A}$'. Intuitivement, X_i correspondant à l'ensemble des positions dans le mot pour lesquelles le calcul est dans l'état q_i. Plus précisément, $X_i(x)$ se lit 'avant de lire la lettre en position x, on se trouve dans l'état i'. La formule ψ assure que les transitions de l'automate sont respectées.

$\boxed{\text{définition formelle}}$ La formule ψ est la conjonction[2] de

- $\bigwedge_{(i,j) \in \{0,\dots,n-1\} \mid i \neq j} \forall x, \neg(X_i(x) \wedge X_j(x))$: on ne peut être dans deux états en même temps ;
- $\forall x, (premiere(x) \rightarrow X_0(x))$: au début, on est dans l'état initial 0 ;

  où $premiere(x)$ signifie 'x est la position initial du mot' ;
- $\forall x, y, S(x, y) \rightarrow \bigvee_{(i,a,j) \in \delta} (X_i(x) \wedge a(x) \wedge X_j(y))$: entre deux positions successives, on respecte une des transitions de l'automate ;
- $\forall x, derniere(x) \rightarrow \bigvee_{(i,a,j) \in \delta, j \in F} (X_i(x) \wedge a(x)$: à la fin, on arrive dans un état final.

  où $derniere(x)$ signifie 'x est la position final du mot'.

$\boxed{\text{idée de la démonstration}}$ Soit $u \in \Sigma^+$. Montrons que

[2] Le 'et' des formules.

(il existe une exécution acceptante depuis q_0 de $\mathcal{A}$ pour le mot u)

ssi

($u, \lambda_1, \lambda_2 \models \varphi$, pour λ_1, λ_2 arbitraires[3]).

$\boxed{\Downarrow}$ Soit $u = u_1 \dots u_n$. L'exécution acceptante est de la forme

$$q_{i_1} \to^{u_1} q_{i_2} \to^{u_2} \dots \to^{u_n} q_{i_{n+1}}$$

avec $i_1 = 0$ et $q_{i_{n+1}} \in F$.

On montre que $u, \lambda_1, \lambda_2' \models \psi$ où $\lambda_2'(X_i) = \{j \mid i_j = i\}$ pour tout $i \in \{0, \dots, n-1\}$. En image :

$$\begin{array}{cccc} u_1 & u_2 & \dots & u_n \\ X_{i_1} & X_{i_2} & \dots & X_{i_n} \end{array}$$

Par exemple, $u, \lambda_1, \lambda_2' \models \forall x, (premiere(x) \to X_0(x))$ est vraie car $1 \in \lambda_2(X_0)$ (informellement, en la position 1, il y a X_0).

$\boxed{\Uparrow}$

Réciproquement, si $u, \lambda_1, \lambda_2 \models \varphi$ alors il existe λ_2' tel que $u, \lambda_1, \lambda_2' \models \psi$. Comme $u, \lambda_1, \lambda_2' \models \bigwedge_{(i,j)\in\{0,\dots,n-1\} \mid i \neq j} \forall x, \neg(X_i(x) \wedge X_j(x))$, les ensembles $(\lambda_2'(X_i))_{i\in\{0,\dots,n-1\}}$ sont disjoints. En image,

$$\begin{array}{cccc} u_1 & u_2 & \dots & u_n \\ X_{i_1} & X_{i_2} & \dots & X_{i_n} \end{array}$$

Comme $u, \lambda_1, \lambda_2' \models \forall x, (premiere(x) \to X_0(x))$, on a $i_1 = 0$.

De la même façon, on montre que

$$q_{i_1} \to^{u_1} q_{i_2} \to^{u_2} \dots \to^{u_{n-1}} q_{i_n}$$

est une exécution et qu'il existe un état $q_{i_{n+1}} \in F$ avec $q_{i_n} \to^{u_n} q_{i_{n+1}}$.

■

Proposition 35 *Tout langage décrit par une formule MSO est rationnel. On peut construire un automate $\mathcal{A}$ de taille non-élémentaire en la taille de la formule.*

Proof.

Réciproquement, on montre par induction sur les formules de MSO que l'on peut associer un langage régulier $L(\varphi)$ à toute formule de MSO φ. D'abord nous allons enrichir l'alphabet pour représenter toute l'information dans u, λ_1, λ_2.

[3] On peut prendre λ_1, λ_2 arbitraires car la formule φ dans le sens où toutes les variables apparaissant dans φ sont sous la portée d'un quantificateur.

L'alphabet est $\Sigma \times \{0,1\}^{V_1 \cup V_2}$.

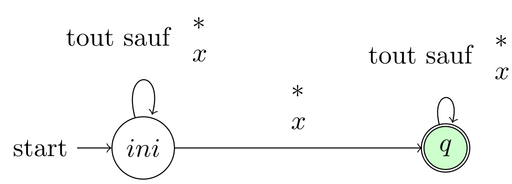
Par exemple le mot *abcde* avec $\lambda_1(x) = 2, \lambda_1(y) = 3$ et $\lambda_2(X) = \{2,3,5\}$ s'écrit maintenant :

$$\begin{array}{ccccc} a & b & c & d & e \\ & x & & & \\ & & y & & \\ & X & X & & X \end{array}$$

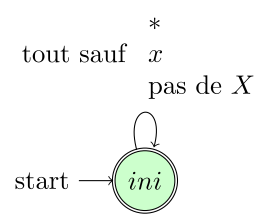
Dans $x = y$, les variables x et y sont libres : elles ne sont pas sous la portée d'un quantificateur. Donc, si on veut construire des automates pour vérifier, par exemple, la propriété $x = y$, il faut avoir à disposition l'information des positions dénotés par x et y. C'est à cela que sert l'alphabet enrichi.

Soit $P(\varphi)$ la propriété 'il existe un automate non-déterministe avec ϵ-transitions $\mathcal{A}_\varphi$ qui reconnaît $L(\varphi)$. On construit des automates :

- pour la contrainte x est une variable du premier ordre (c'est à dire x pointe vers une et une seule position) que l'on note $Sing(x)$:

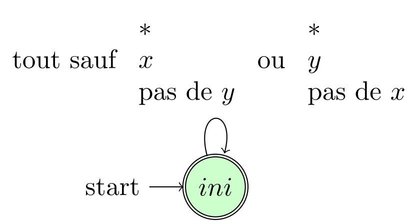


- pour $x \in X$:

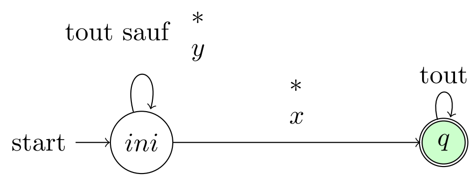


  En faisant le produit avec l'automate $Sing(x)$, on obtient un automate qui reconnaît les mots écrits dans le sur-alphabet avec $x \in X$.

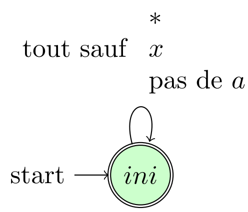
- $S(x, y)$: on construit l'automate de $(x < y \wedge \neg\exists z, (x < z \wedge z < y))$.
- $x = y$:



- $x < y$:



- $a(x)$:

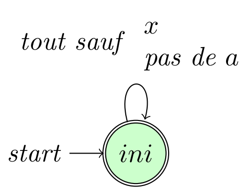


Montrons maintenant les cas inductifs.

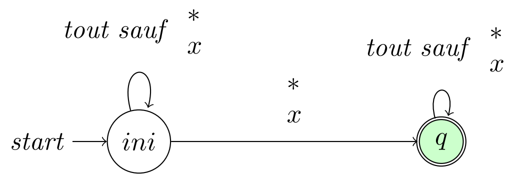
- Supposons que $P(\varphi)$. On construit $\mathcal{A}_{\neg\varphi}$ en passant au complémentaire à partir de $\mathcal{A}_\varphi$. Attention, la taille de $\mathcal{A}_{\neg\varphi}$ peut être exponentiel en la taille de $\mathcal{A}_\varphi$.
- Supposons que $P(\varphi)$ et $P(\psi)$. On construit $\mathcal{A}_{(\varphi\vee\psi)}$ en faisant l'union des automate $\mathcal{A}_\varphi$ et $\mathcal{A}_\psi$.
- Supposons que $P(\varphi)$. Montrons $P(\exists X\varphi)$. On construit $\mathcal{A}_{\exists X\varphi}$ en prenant l'automate $\mathcal{A}_\varphi$ et en oubliant les X.
- Supposons que $P(\varphi)$. Montrons $P(\exists x\varphi)$. On construit $\mathcal{A}_{\exists x\varphi}$ en faisant le produit de l'automate $\mathcal{A}_\varphi$et l'automate pour $Sing(x)$. Ensuite, on oublie les x.

■

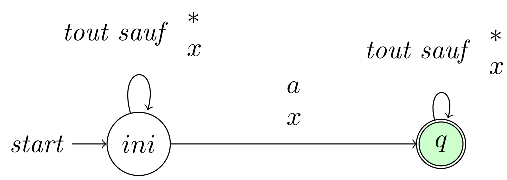
Example 39 *Construisons l'automate pour $\exists x, a(x)$.*
On prend l'automate de $a(x)$:



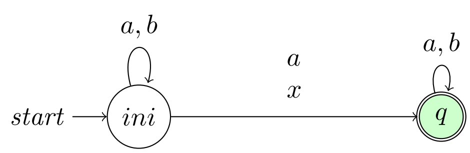
On fait le produit avec l'automate pour $Sing(x)$ qui est :



Ce produit est :



Explicitons le :



Maintenant, on obtient l'automate pour $\exists x, a(x)$ *en oubliant* x *:*