# Logique sur les mots

## Définition : Syntaxe de $\mathrm{MSO}(\Sigma, <)$

$$\varphi ::= \bot \mid P_a(x) \mid x < y \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \exists X\, \varphi$$

avec $a \in \Sigma$, $\{x, y, \ldots\}$ variables du premier ordre, {X,Y,...} variables monadiques du second ordre.

## Définition : Sémantique de $\mathrm{MSO}(\Sigma, <)$

Soit $w = w_1 w_2 \cdots w_n \in \Sigma^+$ un mot et $\mathrm{pos}(w) = \{1, 2, \ldots, n\}$ les positions du mot.
Soit $\sigma$ une valuation :
$\sigma(x) \in \mathrm{pos}(w)$ si $x$ est une variable du premier ordre et
$\sigma(X) \subseteq \mathrm{pos}(w)$ si $X$ est une variable monadique du second ordre.

$$\begin{array}{lcl}
w, \sigma \models P_a(x) & \text{si} & w_{\sigma(x)} = a \\
w, \sigma \models x < y & \text{si} & \sigma(x) < \sigma(y) \\
w, \sigma \models x \in X & \text{si} & \sigma(x) \in \sigma(X) \\
w, \sigma \models \exists x\, \varphi & \text{si} & \exists\, i \in \mathrm{pos}(w) \text{ tel que } w, \sigma[x \mapsto i] \models \varphi \\
w, \sigma \models \exists X\, \varphi & \text{si} & \exists\, I \subseteq \mathrm{pos}(w) \text{ tel que } w, \sigma[X \mapsto I] \models \varphi
\end{array}$$

# MSO: Raccourcis

- On peut exprimer $\forall x$, $\forall X$, $\vee$, etc comme d'habitude.
- $X \subseteq Y$:
$$\forall x.(x \in X \rightarrow x \in Y)$$
- $Z = X \cup Y$:
$$\forall x.(x \in Z \leftrightarrow x \in X \vee x \in Y)$$
- $Partition(X_1, \ldots, X_m)$:
$$\left(\forall x. \bigvee_{i=1}^{m} x \in X_i\right) \wedge \left(\bigwedge_{i=1}^{m} \bigwedge_{j \neq i} \forall x.(x \notin X_i \vee x \notin X_j)\right)$$
- $First(x)$:
$$\forall y : x \leq y$$
- $Succ(x, y)$:
$$x < y \wedge \forall z : \neg(x < z \wedge z < y)$$
- En plus, $X = \emptyset$, $X = \{x\}$, $X = Y$,...

# Logique sur les mots

## Définition : Codage d'une valuation dans l'alphabet

Soit $V$ un ensemble de variables, on note $\Sigma_V = \Sigma \times \{0,1\}^V$.
Un couple $(w, \sigma)$ où $w = w_1 w_2 \cdots w_n \in \Sigma^+$ est un mot sur l'alphabet $\Sigma$ et $\sigma$ est une valuation des variables de $V$ est codé par un mot $W = (w_1, \tau_1) \cdots (w_n, \tau_n) \in \Sigma_V^+$ sur l'alphabet $\Sigma_V$ avec:

- $\forall i \in \mathrm{pos}(w)$, $\tau_i(x) = 1$ ssi $\sigma(x) = i$
  si $x \in V$ est une variable du premier ordre,
- $\forall i \in \mathrm{pos}(w)$, $\tau_i(X) = 1$ ssi $i \in \sigma(X)$
  si $X \in V$ est une variable monadique du second ordre.

Un mot $W \in \Sigma_V^+$ est valide si il code un couple $(w, \sigma)$. On identifie $W$ et $(w, \sigma)$.

Remarque: $\{W \in \Sigma_V^+ \mid W \text{ est valide}\}$ est reconnaissable.

## Définition : Sémantique de $\mathrm{MSO}(\Sigma, <)$

Soit $\varphi \in \mathrm{MSO}(\Sigma, <)$ et soit $V$ un ensemble de variables contenant les variables libres de $\varphi$,

$$\mathcal{L}_V(\varphi) = \{W \in \Sigma_V^+ \mid W = (w, \sigma) \text{ est valide et } (w, \sigma) \models \varphi\}$$

# Exemples

- Soit $w = abc$ et $\sigma(x) = 2$, $\sigma(y) = 3$, and $\sigma(Z) = \{1, 3\}$.
  Alors $\langle w, \sigma \rangle$ est codé par $\langle a, 001\rangle\langle b, 100\rangle\langle c, 011\rangle$.

- Si on n'a que $\sigma'(x) = 2$, alors $\langle w, \sigma' \rangle$ est codé par $\langle a, 0\rangle\langle b, 1\rangle\langle c, 0\rangle$.

- $w, \sigma' \models P_b(x)$, du coup $\langle w, \sigma' \rangle \in \mathcal{L}(P_b(x))$

- $w \in \mathcal{L}(\exists x.P_b(x))$

# Logique sur les mots

**Théorème : Büchi 1960, Elgot 1961, Trakhtenbrot 1961**

Un langage $L \subseteq \Sigma^+$ est reconnaissable si et seulement si il est définissable par une formule close $\varphi \in \mathrm{MSO}(\Sigma, <)$.

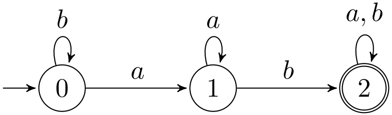
**Preuve**

$\Longrightarrow$ : Si $L$ est reconnu par un automate $\mathcal{A}$ ayant $n$ états $Q = \{q_1, \ldots, q_n\}$, on écrit une formule de la forme $\varphi = \exists X_1 \cdots \exists X_n \, \psi(X_1, \ldots, X_n)$ qui caractérise l'existence d'un calcul acceptant de $\mathcal{A}$ sur un mot $w \in \Sigma^+$.
$X_i$ est l'ensemble des positions de $w$ pour lesquelles le calcul est dans l'état $q_i$.
La formule $\psi$ assure que les transitions de l'automate sont respectées.

$\Longleftarrow$ : On donne des expressions rationnelles pour les formules atomiques.
On note $\Sigma_V^{x=1} = \{(a, \tau) \in \Sigma_V \mid \tau(x) = 1\}$, de même $\Sigma_V^{x=0}$ ou $\Sigma_V^{X=1}$ou $\Sigma_V^{X=0}$.
$\mathcal{L}_V(P_a(x)) = (\Sigma_V^{x=0})^*(\Sigma_V^{x=1} \cap (\{a\} \times \{0,1\}^V))(\Sigma_V^{x=0})^*$.
$\mathcal{L}_V(x \in X) = (\Sigma_V^{x=0})^*(\Sigma_V^{x=1} \cap \Sigma_V^{X=1})(\Sigma_V^{x=0})^*$.
$\mathcal{L}_V(x < y) = (\Sigma_V^{x=y=0})^*(\Sigma_V^{x=1} \cap \Sigma_V^{y=0})(\Sigma_V^{x=y=0})^*(\Sigma_V^{x=0} \cap \Sigma_V^{y=1})(\Sigma_V^{x=y=0})^*$.
On utilise les propriétés de clôture des langages reconnaissables :
union ($\vee$), complémentaire ($\neg$) et projection ($\exists x$ et $\exists X$).

# Exemple: Automate $\rightarrow$ MSO

- Automate :



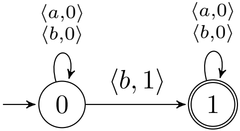

- Formule MSO :

$$
\begin{aligned}
\phi \quad = \quad & \exists Q_0, Q_1, Q_2. Partition(Q_0, Q_1, Q_2) \\
& \wedge \ \forall x. (First(x) \rightarrow x \in Q_0) \\
& \wedge \ \forall x, y. (Succ(x, y) \wedge x \in Q_0 \wedge P_a(x) \rightarrow y \in Q_1) \\
& \wedge \ \forall x, y. (Succ(x, y) \wedge x \in Q_0 \wedge P_b(x) \rightarrow y \in Q_0) \\
& \wedge \ \cdots \\
& \wedge \ \forall x. Last(x) \rightarrow (x \in Q_2 \vee (x \in Q_1 \wedge P_b(x)))
\end{aligned}
$$

# Exemple: MSO $\rightarrow$ Automate

Soit $\Sigma = \{a, b\}$.

- $\phi = P_b(x)$ donne l'automate $\mathcal{A}_\phi$ suivant :




- $\phi' = \exists x.P_b(x)$; on l'obtient $\mathcal{A}_{\phi'}$ en supprimant $x$ dans $\mathcal{A}_\phi$ (autrement dit, on projète toute lettre $\langle a, x\rangle$ vers $a$).