- Assume we can construct a formula

$$Visits(X_0, \dots, X_n)$$

which is true for $(w, \mathcal{I})$ iff

$$\mathcal{I}(X_0) = P_{q_0}, \dots, \mathcal{I}(X_n) = P_{q_n}$$

- Then $(w, \mathcal{I})$ satisfies the formula

$$\psi_A := \exists X_0 \dots \exists X_n \ \mathrm{Visits}(X_0, \dots X_n) \wedge \exists x \left(\mathrm{last}(x) \wedge \bigvee_{q_i \in F} x \in X_i \right)$$

iff w has a last letter and $w \in L$, and we easily get a formula expressing L .

- To construct $Visits(X_0, \ldots, X_n)$ we observe that the sets P_q are the unique sets satisfying
  a) $1 \in P_{\delta(q_0,a_1)}$ i.e., after reading the first letter the DFA is in state $\delta(q_0, a_1)$.
  b) The sets P_q build a partition of the set of positions, i.e., the DFA is always in exactly one state.
  c) If $i \in P_q$ and $\delta(q, a_{i+1}) = q'$ then $i + 1 \in P_{q'}$, i.e., the sets „match“ δ.
- We give formulas for a) , b), and c)

- Formula for a)

$$\mathrm{Init}(X_0,\ldots,X_n) = \exists x \left(\mathrm{first}(x) \wedge \left(\bigvee_{a\in\Sigma}(Q_a(x) \wedge x \in X_{i_a})\right)\right)$$

- Formula for b)

$$\mathrm{Partition}(X_0,\ldots,X_n) = \forall x \left(\bigvee_{i=0}^{n} x \in X_i \;\wedge \bigwedge_{\substack{i,j=0\\ i\neq j}}^{n} (x \in X_i \rightarrow x \notin X_j)\right)$$

- Formula for c)

$$\text{Respect}(X_0,\dots,X_n) =$$
$$\forall x \forall y \left(y = x+1 \rightarrow \bigvee_{\substack{a\in\Sigma \\ i,j\in\{0,\dots,n\} \\ \delta(q_i,a)=q_j}} (x \in X_i \wedge Q_a(x) \wedge y \in X_j) \right)$$

- Together:

$$\begin{aligned}\text{Visits}(X_0,\dots X_n) := {} & \text{Init}(X_0,\dots,X_n) \wedge \\ & \text{Partition}(X_0,\dots,X_n) \wedge \\ & \text{Respect}(X_0,\dots,X_n)\end{aligned}$$

Every language expressible in MSO logic is regular

- Recall: an interpretation of a formula is a pair $(w, \mathcal{I})$ consisting of a word w and assignments $\mathcal{I}$ to the free first and second order variables (and perhaps to others).

$$\left(aab\ ,\ \begin{array}{l} x \mapsto 1 \\ y \mapsto 3 \\ X \mapsto \{2,3\} \\ Y \mapsto \{1,2\} \end{array} \right) \qquad \left(ba\ ,\ \begin{array}{l} x \mapsto 2 \\ y \mapsto 1 \\ X \mapsto \emptyset \\ Y \mapsto \{1\} \end{array} \right)$$

- We encode interpretations as words.

$$\left(aab\ ,\ \begin{array}{l} x \mapsto 1 \\ y \mapsto 3 \\ X \mapsto \{2,3\} \\ Y \mapsto \{1,2\} \end{array} \right) \qquad \left(ba\ ,\ \begin{array}{l} x \mapsto 2 \\ y \mapsto 1 \\ X \mapsto \emptyset \\ Y \mapsto \{1\} \end{array} \right)$$

	a	a	b
x	1	0	0
y	0	0	1
X	0	1	1
Y	1	1	0

	b	a
x	0	1
y	1	0
X	0	0
Y	1	0

- Given a formula with n free variables, we encode an interpretation $(w, \mathcal{I})$ as a word $enc(w, \mathcal{I})$ over the alphabet $\Sigma \times \{0,1\}^n$.
- The language of the formula φ , denoted by $L(\varphi)$, is given by

$$L(\varphi)=\{enc(w, \mathcal{I}) \mid (w, \mathcal{I}) \models \varphi\}$$

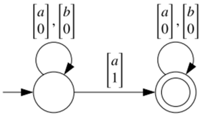
- We prove by induction on the structure of φ that $L(\varphi)$ is regular (and explicitely construct an automaton for it).

Case $\varphi = Q_a(x)$

- $\varphi = Q_a(x)$. Then $free(\varphi) = x$, and the interpretations of φ are encoded as words over $\Sigma \times \{0,1\}$. The language $L(\varphi)$ is given by

$$L(\varphi) = \left\{ \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \begin{matrix} \dots \\ \dots \end{matrix} \begin{bmatrix} a_k \\ b_k \end{bmatrix} \middle| \begin{array}{l} k \geq 0, \\ a_i \in \Sigma \text{ and } b_i \in \{0,1\} \text{ for every } i \in \{1,\dots,k\}, \text{ and} \\ b_i = 1 \text{ for exactly one index } i \in \{1,\dots,k\} \text{ such that } a_i = a \end{array} \right\}$$

and is recognized by

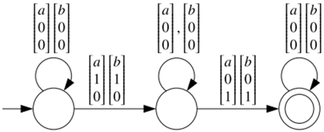


Case $\varphi = x < y$

- $\varphi = x < y$. Then $free(\varphi) = \{x, y\}$, and the interpretations of ϕ are encoded as words over $\Sigma \times \{0,1\}^2$. The language $L(\varphi)$ is given by

$$L(\varphi) = \left\{ \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} \cdots \begin{bmatrix} a_k \\ b_k \\ c_k \end{bmatrix} \;\middle|\; \begin{array}{l} k \geq 0, \\ a_i \in \Sigma \text{ and } b_i, c_i \in \{0,1\} \text{ for every } i \in \{1, \ldots, k\}, \\ b_i = 1 \text{ for exactly one index } i \in \{1, \ldots, k\}, \\ c_j = 1 \text{ for exactly one index } j \in \{1, \ldots, k\}, \text{ and} \\ i < j \end{array} \right\}$$

and is recognized by

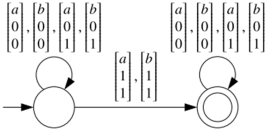


Case $\varphi = x \in X$

- $\varphi = x \in X$. Then $free(\varphi) = \{x, X\}$, and interpretations are encoded as words over $\Sigma \times \{0,1\}^2$. The language $L(\varphi)$ is given by

$$L(\varphi) = \left\{ \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} \begin{matrix} \dots \\ \dots \end{matrix} \begin{bmatrix} a_k \\ b_k \\ c_k \end{bmatrix} \;\middle|\; \begin{array}{l} k \geq 0, \\ a_i \in \Sigma \text{ and } b_i, c_i \in \{0,1\} \text{ for every } i \in \{1,\dots,k\}, \\ b_i = 1 \text{ for exactly one index } i \in \{1,\dots,k\}, \text{ and} \\ \text{for every } i \in \{1,\dots,k\}, \text{ if } b_i = 1 \text{ then } c_i = 1 \end{array} \right\}$$

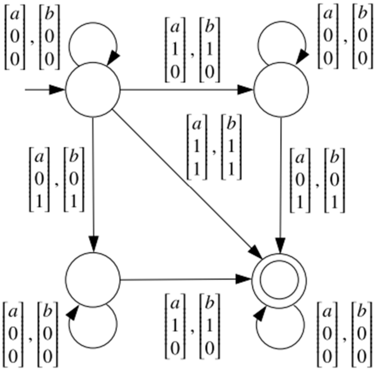
and is recognized by



Case $\varphi = \neg\psi$

- Then $free(\varphi) = free(\psi)$. By i.h. $L(\psi)$ is regular.
- $L(\varphi)$ is equal to $\overline{L(\psi)}$ minus the words that do not encode any implementation („the garbage“).
- Equivalently, $L(\varphi)$ is equal to the intersection of $\overline{L(\psi)}$ and the encodings of all interpretations of ψ.
- We show that the set of these encodings is regular.
  - Condition for encoding: Let x be a free first-oder variable of ψ . The projection of an encoding onto x must belong to 0^*10^* (because it represents one position).
  - So we just need an automaton for the words satisfying this condition for every free first-order variable.

Example: $free(\varphi) = \{x, y\}$

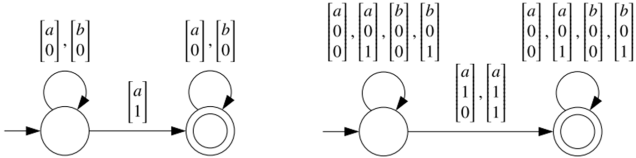


Case $\varphi = \varphi_1 \vee \varphi_2$

- Then $free(\varphi) = free(\varphi_1) \cup free(\varphi_2)$. By i.h. $L(\varphi_1)$ and $L(\varphi_2)$ are regular.
- If $free(\varphi_1) = free(\varphi_2)$ then $L(\varphi) = L(\varphi_1) \cup L(\varphi_2)$ and so $L(\varphi)$ is regular.
- If $free(\varphi_1) \neq free(\varphi_2)$ then we extend $L(\varphi_1)$ to a language L_1 encoding all interpretations of $free(\varphi_1) \cup free(\varphi_2)$ whose projection onto $free(\varphi_1)$ belongs to $L(\varphi_1)$. Similarly we extend $L(\varphi_2)$ to L_2. We have
  - L_1 and L_2 are regular.
  - $L(\varphi) = L_1 \cup L_2$.

Example: $\varphi = Q_a(x) \vee Q\_b(y)$

- L_1 contains the encodings of all interpretations $(w, \{x \mapsto n_1, y \mapsto n_2\})$ such that the encoding of $(w, \{x \mapsto n_1\})$ belongs to $L(Q_a(x))$.
- Automata for $L(Q_a(x))$ and L_1:

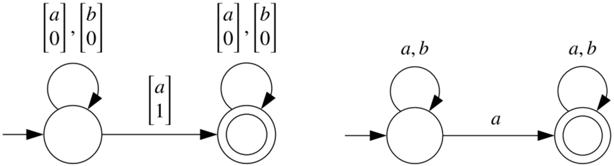


Cases $\varphi = \exists x\ \psi$ and $\varphi = \exists X\ \psi$

- Then $free(\varphi) = free(\psi) \setminus \{x\}$ or $free(\varphi) = free(\psi) \setminus \{X\}$
- By i.h. $L(\psi)$ is regular.
- $L(\varphi)$ is the result of projecting $L(\psi)$ onto the components for $free(\psi) \setminus \{x\}$ or $free(\psi) \setminus \{X\}$.

Example: $\varphi = Q_a(x)$

- Automata for $Q_a(x)$ and $\exists x\, Q_a(x)$



The mega-example

- We compute an automaton for

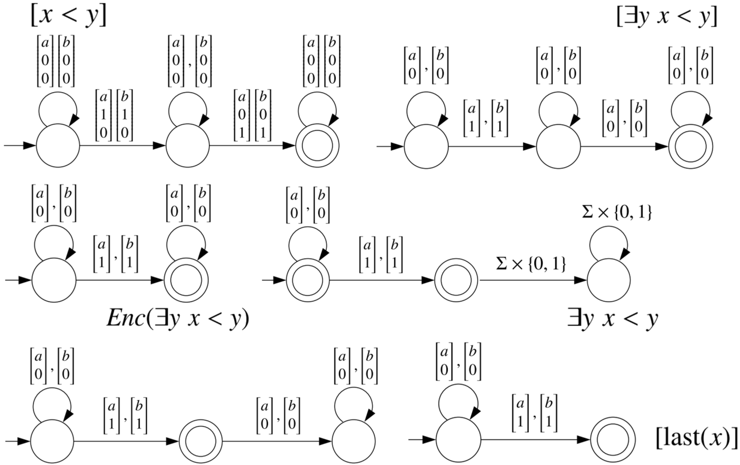
$$\exists x\,(\text{last}(x) \wedge Q_b(x)) \;\wedge\; \forall x\,(\neg\text{last}(x) \rightarrow Q_a(x))$$

- First we rewrite φ into

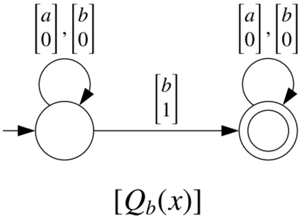
$$\exists x\,(\text{last}(x) \wedge Q_b(x)) \;\wedge\; \neg\exists x\,(\neg\text{last}(x) \wedge \neg Q_a(x))$$

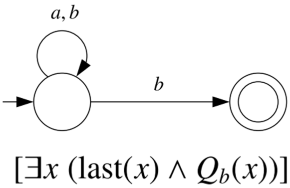
- In the next slides we
  1. compute a DFA for $last(x)$
  2. compute DFAs for $\exists x\;(last(x) \wedge Q_b(x))$ and $\neg\exists x\;(\neg last(x) \wedge \neg Q_a(x))$
  3. compute a DFA for the complete formula.
- We denote the DFA for a formula ψ by $[\psi]$.

$[last(x)]$

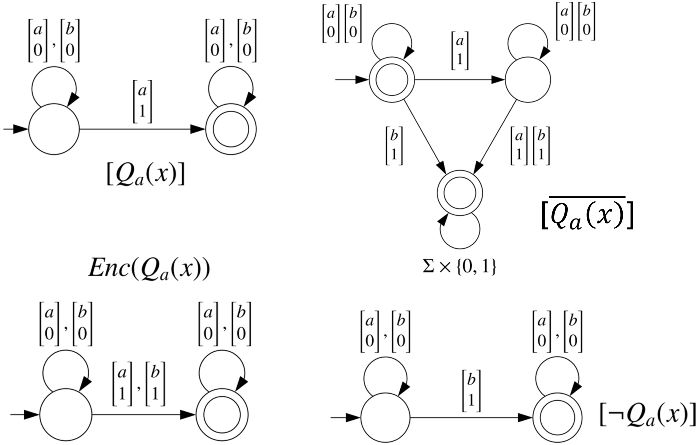


$[\exists x\,(last(x) \wedge Q_b(x))]$

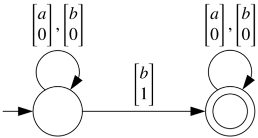




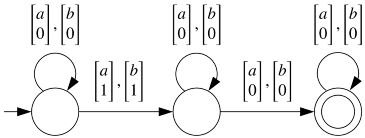
$[\neg Q_a(x)]$



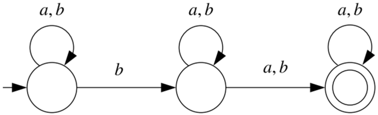
$[\neg\exists x\,(\neg last(x) \wedge \neg Q_a(x))]$



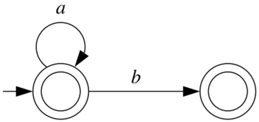
$[\neg Q_a(x)]$



$[\neg last(x)]$

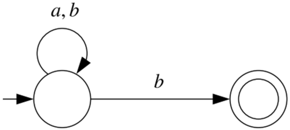


$[\exists x\,(\neg last(x) \wedge \neg Q_a(x))]$

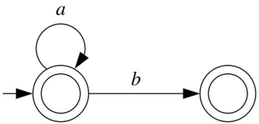


$[\neg\exists x\,(\neg last(x) \wedge \neg Q_a(x))]$

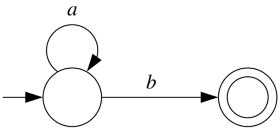
$$[\exists x\,(last(x) \land Q_b(x))$$
$$\land \neg\exists x\,(\neg last(x) \land \neg Q_a(x))]$$



$[\exists x\,(\mathrm{last}(x) \land Q_b(x))]$



$[\neg\exists x\,(\neg last(x) \land \neg Q_a(x))]$



$[\exists x\,(\mathrm{last}(x) \land Q_b(x)) \;\land\; \neg\exists x\,(\neg\mathrm{last}(x) \land \neg Q_a(x))]$