CS 228 : Logic in Computer Science

S. Krishna

Satisfiability to Model Checking

- Satisfiability of FO over words
- Model checking
  - System abstracted as a model DFA/NFA A
  - Specification written in FO as formula φ
  - Does system model $\models \varphi$
  - $L(A) \subseteq L(\varphi)$?
  - $L(A) \cap \overline{L(\varphi)} = \emptyset$?
- FO-definable $\subseteq$ *REG*
- Is there a logic equivalent to regular languages?

Monadic Second Order Logic (MSO)

Symbols in MSO

Formulae of MSO, over signature τ, are sequences of symbols, where each symbol is one of the following:

- The symbol $\perp$ called false
- An element of the infinite set $\mathcal{V}_1 = \{x_1, x_2, \ldots\}$ of first order variables
- An element of the infinite set $\mathcal{V}_2 = \{X_1, X_2, \ldots\}$ of second order variables where each variable has arity 1 (new!)
- Constants and relations from τ
- The symbol $\rightarrow$ called implication
- The symbol $\forall$ called the universal quantifier
- The symbols (and) called paranthesis

Well formed Formulae

A well-formed formula (wff) over a signature τ is inductively defined as follows:

- $\bot$ is a wff
- If t_1, t_2 are either variables or constants in τ, then $t_1 = t_2$ is a wff
- If t_i is either a first order variable or a constant, for $1 \leqslant i \leqslant k$ and R is a k-ary relation symbol in τ, then $R(t_1, \dots, t_k)$ is a wff
- If t is either a first order variable or a constant, X is a second order variable, then $X(t)$ is a wff
- If φ and ψ are wff, then $\varphi \to \psi$ is a wff
- If φ is a wff and x is a first order variable, then $(\forall x)\varphi$ is a wff
- If φ is a wff and X is a second order variable, then $(\forall X)\varphi$ is a wff

Free and Bound Variables

- Free, Bound Variables and Scope same as in FO
- In a wff $\varphi = \forall X \psi$, every occurrence of X in ψ is bound
- A sentence is a formula with no free first order and second order variables

Assignments on τ-structures

Assignments

For a τ-structure $\mathcal{A}$, an assignment over $\mathcal{A}$ is a pair of functions (α_1, α_2), where

- $\alpha_1 : \mathcal{V}_1 \to u(\mathcal{A})$ assigns every first order variable $x \in \mathcal{V}_1$ a value $\alpha_1(x) \in u(\mathcal{A})$. If t is a constant symbol c, then $\alpha_1(t)$ is $c^{\mathcal{A}}$.
- $\alpha_2 : \mathcal{V}_2 \to 2^{u(\mathcal{A})}$ assigns to every second order variable $X \in \mathcal{V}_2$, $\alpha_2(X) \subseteq u(\mathcal{A})$.

Binding on a Variable

For an assignment $\alpha = (\alpha_1, \alpha_2)$ over $\mathcal{A}$, and $x \in \mathcal{V}_i$, $i = 1, 2$,
$\alpha_i[x \mapsto a]$ is the assignment $\alpha_i[x \mapsto a](y) = \begin{cases} \alpha_i(y), y \neq x, \\ a, y = x \end{cases}$

Satisfaction

We define the relation $\mathcal{A} \models_\alpha \varphi$ (read as φ is true in $\mathcal{A}$ under the assignment α) inductively:

- $\mathcal{A} \not\models_\alpha \perp$
- $\mathcal{A} \models_\alpha t_1 = t_2$ iff $\alpha_1(t_1) = \alpha_1(t_2)$
- $\mathcal{A} \models_\alpha R(t_1, \dots, t_k)$ iff $(\alpha_1(t_1), \dots, \alpha_1(t_k)) \in R^{\mathcal{A}}$
- $\mathcal{A} \models_\alpha X(t)$ iff $\alpha_1(t) \in \alpha_2(X)$ (new)
- $\mathcal{A} \models_\alpha (\varphi \to \psi)$ iff $\mathcal{A} \not\models_\alpha \varphi$ or $\mathcal{A} \models_\alpha \psi$
- $\mathcal{A} \models_\alpha (\forall x)\varphi$ iff for every $a \in u(\mathcal{A})$, $\mathcal{A} \models_{\alpha[x \mapsto a]} \varphi$
- $\mathcal{A} \models_\alpha (\forall X)\varphi$ iff for every $S \subseteq u(\mathcal{A})$, $\mathcal{A} \models_{\alpha[X \mapsto S]} \varphi$ (new)

Examples

Recall the signature for the graph structure, $\tau = \{E\}$

- The graph is 3-colorable

Examples

Recall the signature for the graph structure, $\tau = \{E\}$

- The graph is 3-colorable

$$\exists X \exists Y \exists Z (\forall x [X(x) \lor Y(x) \lor Z(x)] \land$$

$$\forall x \forall y [E(x,y) \to \{\neg(X(x) \land X(y)) \land \neg(Y(x) \land Y(y)) \land \neg(Z(x) \land Z(y))\}])$$

Examples

Recall the signature for the graph structure, $\tau = \{E\}$

- The graph has an independent set of size $\geqslant k$

Examples

Recall the signature for the graph structure, $\tau = \{E\}$

- The graph has an independent set of size $\geqslant k$

$$\exists I\{\forall x \forall y[(\neg(x = y) \wedge I(x) \wedge I(y)) \rightarrow \neg E(x, y)] \wedge$$

$$\exists x_1 \dots x_k[\bigwedge_{i \neq j} \neg(x_i = x_j) \wedge \bigwedge_i I(x_i)]\}$$

Examples

Recall the signature τ for the word structure, $\tau = \{Q_a, Q_b, <, S\}$ for $\Sigma = \{a, b\}$

- Words of even length

Examples

Recall the signature τ for the word structure, $\tau = \{Q_a, Q_b, <, S\}$ for $\Sigma = \{a, b\}$

- Words of even length

$$\exists E \exists O \{\forall x[(first(x) \rightarrow E(x)) \wedge (last(x) \rightarrow O(x))]$$

Examples

Recall the signature τ for the word structure, $\tau = \{Q_a, Q_b, <, S\}$ for $\Sigma = \{a, b\}$

- Words of even length

$$\exists E \exists O \{\forall x[(first(x) \rightarrow E(x)) \wedge (last(x) \rightarrow O(x))]$$

$$\wedge \forall x[(E(x) \vee O(x)) \wedge \neg(E(x) \wedge O(x))]$$

Examples

Recall the signature τ for the word structure, $\tau = \{Q_a, Q_b, <, S\}$ for $\Sigma = \{a, b\}$

- Words of even length

$$\exists E \exists O \{\forall x [(first(x) \rightarrow E(x)) \wedge (last(x) \rightarrow O(x))]$$

$$\wedge \forall x [(E(x) \vee O(x)) \wedge \neg (E(x) \wedge O(x))]$$

$$\wedge \forall x \forall y [S(x, y) \wedge O(x) \rightarrow E(y)]$$

$$\wedge \forall x \forall y [S(x, y) \wedge E(x) \rightarrow O(y)]\}$$

MSO on Words : Satisfiability

MSO on Words

- Signature $\tau = (Q_\Sigma, <, S)$, domain or universe = set of positions of a word
- MSO over words: Atomic formulae

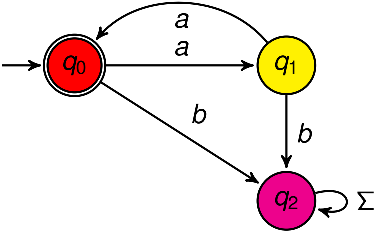
$$X(x) | Q_\Sigma(x) | x = y | x < y | S(x, y)$$

- Given a MSO sentence φ, $L(\varphi)$ defined as usual
- A language $L \subseteq \Sigma^*$ is MSO definable iff there is an MSO formula φ such that $L = L(\varphi)$
- Given an MSO sentence φ, is it satisfiable/valid?

MSO Expressiveness

- Clearly, $FO \subseteq MSO$
- $FO \subset$ Regular
- MSO=Regular

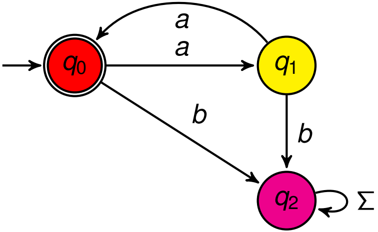
Regular Languages to MSO



Runs of words:

$a \quad a \quad a \quad a$

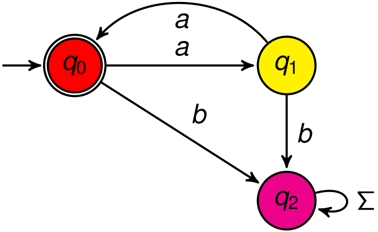
Regular Languages to MSO



Runs of words:

a	a	a	a	
q_0	q_1	q_0	q_1	q_0

Regular Languages to MSO



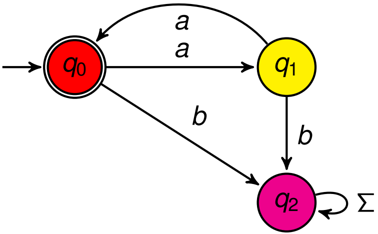
Runs of words:

$a \quad a \quad a \quad a$

$q_0 \quad q_1 \quad q_0 \quad q_1 \quad q_0$

$a \quad a \quad b \quad a$

Regular Languages to MSO



Runs of words:

a	a	a	a	
q_0	q_1	q_0	q_1	q_0

a	a	b	a	
q_0	q_1	q_0	q_2	q_2

Regular Languages to MSO

Given a regular language L, and a DFA such that $L = L(A)$,

- Run of a word : at every position of the word, we are in some unique state

Regular Languages to MSO

Given a regular language L, and a DFA such that $L = L(A)$,

- Run of a word : at every position of the word, we are in some unique state

Position x	:	0	1	2	3	
		a	a	b	a	
		q_0	q_1	q_0	q_2	q_2

Regular Languages to MSO

Given a regular language L, and a DFA such that $L = L(A)$,

- Run of a word : at every position of the word, we are in some unique state

Position x	:	0	1	2	3	
		a	a	b	a	
		q_0	q_1	q_0	q_2	q_2

- For a state $q \in Q$, let X_q=the set of positions of the word where the state is q in the run

Regular Languages to MSO

Given a regular language L, and a DFA such that $L = L(A)$,

- Run of a word : at every position of the word, we are in some unique state

Position x	:	0	1	2	3	
		a	a	b	a	
		q_0	q_1	q_0	q_2	q_2

- For a state $q \in Q$, let X_q=the set of positions of the word where the state is q in the run
- $X_{q_0} = \{0, 2\}, X_{q_1} = \{1\}, X_{q_2} = \{3\}$

Regular Languages to MSO

Given a regular language L, and a DFA such that $L = L(A)$,

- Run of a word : at every position of the word, we are in some unique state

Position x	:	0	1	2	3	
		a	a	b	a	
		q_0	q_1	q_0	q_2	q_2

- For a state $q \in Q$, let X_q=the set of positions of the word where the state is q in the run
- $X_{q_0} = \{0, 2\}, X_{q_1} = \{1\}, X_{q_2} = \{3\}$
- The initial position of any word must belong to X_{q_0} : $0 \in X_{q_0}$

Regular Languages to MSO

- If a word *wa* is accepted, then
  - The last position x of the word satisfies $Q_a(x)$
  - For some state q, we have $X_q(x)$ and there is a transition $\delta(q, a) = q_f \in F$

Regular Languages to MSO

- If a word *wa* is accepted, then
  - The last position x of the word satisfies $Q_a(x)$
  - For some state q, we have $X_q(x)$ and there is a transition $\delta(q, a) = q_f \in F$

$$\begin{array}{llccccc} \text{Position } x & : & 0 & 1 & 2 & 3 & \\ & & a & a & b & a & \\ & & q_0 & q_1 & q_0 & q_2 & q_2 \end{array}$$

Regular Languages to MSO

- If a word wa is accepted, then
  - The last position x of the word satisfies $Q_a(x)$
  - For some state q, we have $X_q(x)$ and there is a transition $\delta(q, a) = q_f \in F$

Position x	:	0	1	2	3	
		a	a	b	a	
		q_0	q_1	q_0	q_2	q_2

- $Q_a(3)$ and $3 \in X_{q_2}$. $\delta(q_2, a) = q_2 \notin F$
- If x, y are consecutive positions in the word, and if $X_q(x) \wedge Q_a(x)$, then it must be that $X_t(y)$ such that $\delta(q, a) = t$

Regular Languages to MSO

- If a word wa is accepted, then
  - The last position x of the word satisfies $Q_a(x)$
  - For some state q, we have $X_q(x)$ and there is a transition $\delta(q, a) = q_f \in F$

Position x	:	0	1	2	3	
		a	a	b	a	
		q_0	q_1	q_0	q_2	q_2

- $Q_a(3)$ and $3 \in X_{q_2}$. $\delta(q_2, a) = q_2 \notin F$
- If x, y are consecutive positions in the word, and if $X_q(x) \wedge Q_a(x)$, then it must be that $X_t(y)$ such that $\delta(q, a) = t$
- $X_{q_0}(0)$, $X_{q_1}(1)$ and $Q_a(0)$. $\delta(q_0, a) = q_1$.
- $X_{q_1}(1)$, $X_{q_0}(2)$ and $Q_a(1)$. $\delta(q_1, a) = q_0$.

Regular Languages to MSO

Given a DFA $A = (Q, \Sigma, \delta, q_0, F)$, a word w is accepted iff it satisfies

$$\exists X_0 \exists X_1 \ldots X_n \{[\forall x (X_0(x) \vee \cdots \vee X_n(x)) \wedge \forall x \bigwedge_{i \neq j} \neg (X_i(x) \wedge X_j(x))] \wedge$$

Regular Languages to MSO

Given a DFA $A = (Q, \Sigma, \delta, q_0, F)$, a word w is accepted iff it satisfies

$$\exists X_0 \exists X_1 \dots X_n \{[\forall x (X_0(x) \vee \cdots \vee X_n(x)) \wedge \forall x \bigwedge_{i \neq j} \neg(X_i(x) \wedge X_j(x))] \wedge$$

$$[\exists x (first(x) \wedge X_0(x))] \wedge$$

Regular Languages to MSO

Given a DFA $A = (Q, \Sigma, \delta, q_0, F)$, a word w is accepted iff it satisfies

$$\exists X_0 \exists X_1 \ldots X_n \{[\forall x (X_0(x) \vee \cdots \vee X_n(x)) \wedge \forall x \bigwedge_{i \neq j} \neg (X_i(x) \wedge X_j(x))] \wedge$$

$$[\exists x (\mathit{first}(x) \wedge X_0(x))] \wedge$$

$$\forall x \forall y [S(x, y) \rightarrow \bigvee_{\delta(i,a)=j} [X_i(x) \wedge Q_a(x) \wedge X_j(y)]] \wedge$$

Regular Languages to MSO

Given a DFA $A = (Q, \Sigma, \delta, q_0, F)$, a word w is accepted iff it satisfies

$$\exists X_0 \exists X_1 \dots X_n \{[\forall x (X_0(x) \vee \cdots \vee X_n(x)) \wedge \forall x \bigwedge_{i \neq j} \neg (X_i(x) \wedge X_j(x))] \wedge$$

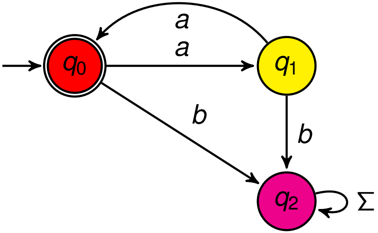
$$[\exists x (\mathit{first}(x) \wedge X_0(x))] \wedge$$

$$\forall x \forall y [S(x,y) \rightarrow \bigvee_{\delta(i,a)=j} [X_i(x) \wedge Q_a(x) \wedge X_j(y)]] \wedge$$

$$\exists x [\mathit{last}(x) \wedge \bigvee_{\delta(i,a)=j \in F} [X_i(x) \wedge Q_a(x)]]\}$$

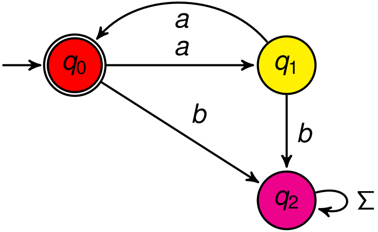
- $w \in L(A)$ iff $w \models \varphi$

Example : Regular to MSO



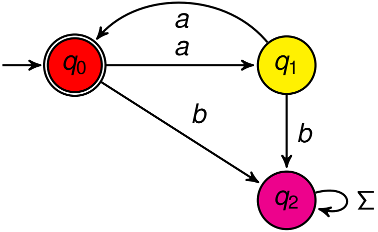
$\exists X_0 \exists X_1 \exists X_2 \{[\forall x(X_0(x) \vee X_1(x) \vee X_2(x)) \wedge \forall x[\neg(X_0(x) \wedge X_1(x)) \wedge$
$\neg(X_0(x) \wedge X_2(x)) \wedge \neg(X_1(x) \wedge X_2(x))] \wedge$

Example : Regular to MSO



$$\exists X_0 \exists X_1 \exists X_2 \{[\forall x(X_0(x) \vee X_1(x) \vee X_2(x)) \wedge \forall x[\neg(X_0(x) \wedge X_1(x)) \wedge \neg(X_0(x) \wedge X_2(x)) \wedge \neg(X_1(x) \wedge X_2(x))] \wedge [\exists x(first(x) \wedge X_0(x))] \wedge$$

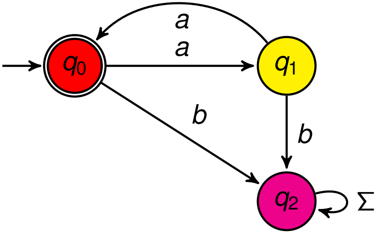
Example : Regular to MSO



$$\exists X_0 \exists X_1 \exists X_2 \{[\forall x(X_0(x) \vee X_1(x) \vee X_2(x)) \wedge \forall x[\neg(X_0(x) \wedge X_1(x)) \wedge \neg(X_0(x) \wedge X_2(x)) \wedge \neg(X_1(x) \wedge X_2(x))] \wedge [\exists x(first(x) \wedge X_0(x))] \wedge$$

$$\forall x \forall y[S(x,y) \rightarrow [(X_0(x) \wedge Q_a(x) \wedge X_1(y)) \vee (X_0(x) \wedge Q_b(x) \wedge X_2(y)) \vee (X_1(x) \wedge Q_a(x) \wedge X_0(y)) \vee (X_1(x) \wedge Q_b(x) \wedge X_2(y)) \vee (X_2(x) \wedge Q_a(x) \wedge X_2(y)) \vee (X_2(x) \wedge Q_b(x) \wedge X_2(y))]]$$

Example : Regular to MSO



$$\exists X_0 \exists X_1 \exists X_2 \{ [\forall x (X_0(x) \vee X_1(x) \vee X_2(x)) \wedge \forall x [\neg (X_0(x) \wedge X_1(x)) \wedge$$
$$\neg (X_0(x) \wedge X_2(x)) \wedge \neg (X_1(x) \wedge X_2(x))] \wedge [\exists x (first(x) \wedge X_0(x))] \wedge$$

$$\forall x \forall y [S(x,y) \rightarrow [(X_0(x) \wedge Q_a(x) \wedge X_1(y)) \vee$$
$$(X_0(x) \wedge Q_b(x) \wedge X_2(y)) \vee (X_1(x) \wedge Q_a(x) \wedge X_0(y)) \vee$$
$$(X_1(x) \wedge Q_b(x) \wedge X_2(y)) \vee (X_2(x) \wedge Q_a(x) \wedge X_2(y)) \vee (X_2(x) \wedge Q_b(x) \wedge X_2(y))]]$$

$$\wedge \exists x [last(x) \wedge (X_1(x) \wedge Q_a(x))] \}$$

Quiz

$$\psi = \exists X_0 \exists X_1 \exists X_2 \{[\forall x (X_0(x) \vee X_1(x) \vee X_2(x)) \wedge \forall x [\neg (X_0(x) \wedge X_1(x)) \wedge$$

$$\neg (X_0(x) \wedge X_2(x)) \wedge \neg (X_1(x) \wedge X_2(x))] \wedge [\exists x (first(x) \wedge X_0(x))] \wedge$$

$$\forall x \forall y [S(x,y) \rightarrow [(X_0(x) \wedge Q_a(x) \wedge X_1(y)) \vee$$

$$(X_0(x) \wedge Q_b(x) \wedge X_2(y)) \vee (X_1(x) \wedge Q_b(x) \wedge X_0(y)) \vee$$

$$(X_1(x) \wedge Q_a(x) \wedge X_2(y)) \vee (X_2(x) \wedge Q_a(x) \wedge X_2(y)) \vee (X_2(x) \wedge Q_b(x) \wedge X_2(y))]]$$

$$\wedge \exists x [last(x) \wedge (X_1(x) \wedge Q_b(x))]\}$$

Let $L = L(\psi)$. Define the morphism $h : \{a, b\} \rightarrow \{b\}$ as $h(a) = h(b) = b$. Then $h : \{a, b\}^* \rightarrow b^*$ maps words $w \in \{a, b\}^*$ to words in b^* : for example, $h(abab) = bbbb$, $h(ab) = bb$ and so on.

MSO to Regular Languages

- Every MSO sentence φ over words can be converted into a DFA A_φ such that $L(\varphi) = L(A_\varphi)$.
- Start with atomic formulae, construct DFA for each of them.
- Conjunctions, Disjunctions, Negation easily handled via union, intersection and complementation of respective DFA
- Handling quantifiers?

Atomic Formulae to DFA

- $Q_a(x)$: All words which have an a. Need to fix a position for x, where a holds.

Atomic Formulae to DFA

- $Q_a(x)$: All words which have an a. Need to fix a position for x, where a holds.
- Think of a word *baab* which satisfies $Q_a(x)$ as $\begin{matrix} baab \\ 0010 \end{matrix}$ or $\begin{matrix} baab \\ 0100 \end{matrix}$

Atomic Formulae to DFA

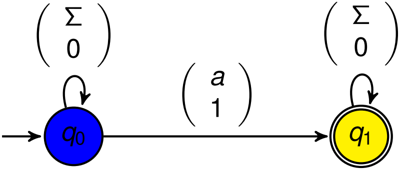
- $Q_a(x)$: All words which have an a. Need to fix a position for x, where a holds.
- Think of a word *baab* which satisfies $Q_a(x)$ as $\begin{matrix} baab \\ 0010 \end{matrix}$ or $\begin{matrix} baab \\ 0100 \end{matrix}$
- The first row is over Σ, and the second row captures a possible assignment to x

Atomic Formulae to DFA

- $Q_a(x)$: All words which have an *a*. Need to fix a position for x, where *a* holds.
- Think of a word *baab* which satisfies $Q_a(x)$ as $\begin{matrix} baab \\ 0010 \end{matrix}$ or $\begin{matrix} baab \\ 0100 \end{matrix}$
- The first row is over Σ, and the second row captures a possible assignment to x
- Think of an extended alphabet $\Sigma' = \Sigma \times \{0, 1\}$, and construct an automaton over Σ'.

Atomic Formulae to DFA

- $Q_a(x)$: All words which have an a. Need to fix a position for x, where a holds.
- Think of a word *baab* which satisfies $Q_a(x)$ as $\begin{matrix} baab \\ 0010 \end{matrix}$ or $\begin{matrix} baab \\ 0100 \end{matrix}$
- The first row is over Σ, and the second row captures a possible assignment to x
- Think of an extended alphabet $\Sigma' = \Sigma \times \{0, 1\}$, and construct an automaton over Σ'.
- Deterministic, not complete.



Simple Formulae to DFA

- $Q_a(x) \wedge X(x)$ means that the position x is in the set X, and letter a is true when $x = 1$.

Simple Formulae to DFA

- $Q_a(x) \wedge X(x)$ means that the position x is in the set X, and letter a is true when $x = 1$.
- Think of a word *baab* which satisfies $Q_a(x) \wedge X(x)$ as

baab		*baab*
0010	or	0100
$DD1D$		$D1DD$

  where D stands for *dont care*. X can have value 0 or 1 at D.

Simple Formulae to DFA

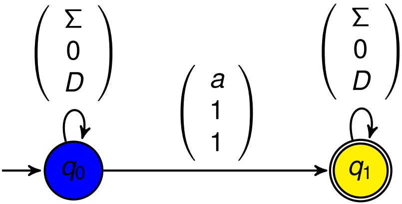
- $Q_a(x) \wedge X(x)$ means that the position x is in the set X, and letter a is true when $x = 1$.
- Think of a word *baab* which satisfies $Q_a(x) \wedge X(x)$ as

baab		*baab*
0010	or	0100
$DD1D$		$D1DD$

  where D stands for *dont care*. X can have value 0 or 1 at D.
- However, the position where $x = 1$ must belong to X.

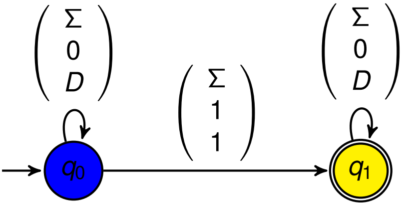
Simple Formulae to DFA

- The first row is over Σ, and the second row captures a possible assignment to x, and the third row captures a possible assignment to X.
- Think of an extended alphabet $\Sigma' = \Sigma \times \{0,1\} \times \{0,1\}$, and construct an automaton over Σ'.
- $Q_a(x) \wedge X(x)$: deterministic, not complete



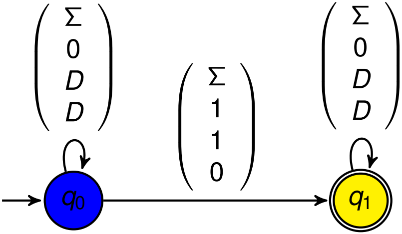
Simple Formulae to DFA

- $X(x)$



Simple Formulae to DFA

- $X(x) \wedge \neg Y(x)$
- $\Sigma' = \Sigma \times \{0,1\} \times \{0,1\} \times \{0,1\}$



Formulae to DFA

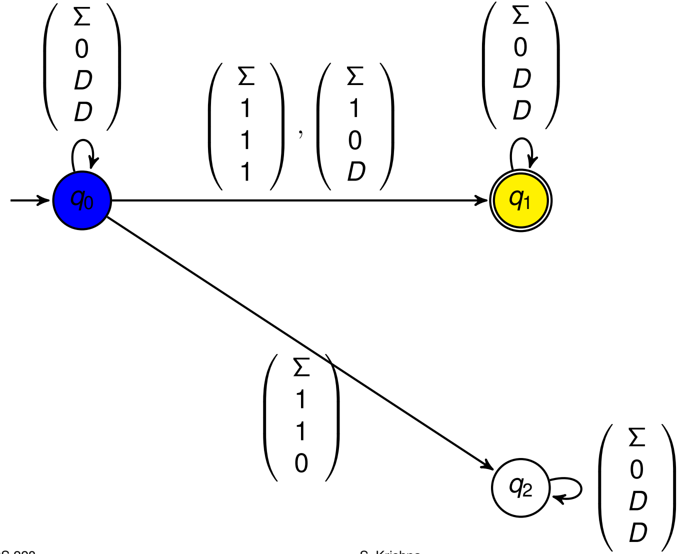
- Given $\varphi(x_1, \dots, x_n, X_1, \dots, X_m)$, an MSO formula over Σ, consider the extended alphabet

$$\Sigma' = \Sigma \times \{0, 1\}^{m+n}$$

- Assign values to x_i, X_j at every position as seen in the cases of atomic formulae
- Keep in mind that every x_i can be assigned 1 at a unique position

Handling Quantifiers

- $\exists X \exists Y \forall x [X(x) \to Y(x)]$



Points to Remember

- Given $\varphi(x_1, \ldots, x_n, X_1, \ldots, X_m)$, construct automaton for atomic MSO formulae over the extended alphabet $\Sigma \times \{0, 1\}^{m+n}$
- Intersect with the regular language where every x_i is assigned 1 exactly at one position
- Given a sentence $Q_{x_1} \ldots Q_{x_n} Q_{X_1} \ldots Q_{X_m} \varphi$, first construct the automaton for the formula $\varphi(x_1, \ldots, x_n, X_1, \ldots, X_m)$
- Replace $\forall$ in terms of $\exists$

Points to Remember

- Given the automaton for $\varphi(x_1, \ldots, x_n, X_1, \ldots, X_n)$, the automaton for $\exists X_i \varphi(x_1, \ldots, x_n, X_1, \ldots, X_n)$ is obtained by projecting out the row of X_i
- This may result in an NFA
- Determinize it and complement it to get a DFA for $\neg\exists x_i \varphi(x_1, \ldots, x_n, X_1, \ldots, X_n)$
- Intersect with the regular language where each of $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ are assigned 1 exactly at one position

The Automaton-Logic Connection

Given any MSO sentence φ, one can construct a DFA A_φ such that $L(\varphi) = L(A_\varphi)$. If a language L is regular, one can construct an MSO sentence φ such that $L = L(\varphi)$.