CS 228 : Logic in Computer Science

Krishna. S

Recap

- *DFA* $=$ *NFA* $=$ ϵ-NFA=regular languages
- Closure properties of regular languages

Agenda

- FO-definable $\Rightarrow$ regular
- Given an FO formula φ, construct a DFA A_φ such that $L(\varphi) = L(A_\varphi)$
- If $L(A_\varphi) = \emptyset$, then φ is unsatisfiable
- If $L(A_\varphi) \neq \emptyset$, then φ is satisfiable

FO to Regular Languages

- Every FO 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 of formulae easily handled via union, intersection and complementation of 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.
- *baab* satisfies $Q_a(x)$ with assignment $x = 1$ or $x = 2$.
- Think of this as $\begin{matrix} baab \\ 0010 \end{matrix}$ or $\begin{matrix} baab \\ 0100 \end{matrix}$

Atomic Formulae to DFA

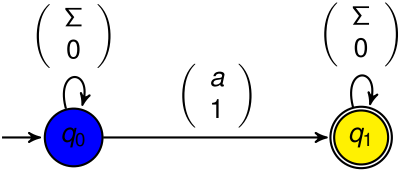
- $Q_a(x)$: All words which have an a. Need to fix a position for x, where a holds.
- $baab$ satisfies $Q_a(x)$ with assignment $x = 1$ or $x = 2$.
- Think of this as $\begin{matrix} baab \\ 0010 \end{matrix}$ or $\begin{matrix} baab \\ 0100 \end{matrix}$
- The first row is over Σ, and the second row 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.
- *baab* satisfies $Q_a(x)$ with assignment $x = 1$ or $x = 2$.
- Think of this 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.
- *baab* satisfies $Q_a(x)$ with assignment $x = 1$ or $x = 2$.
- Think of this 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.



Atomic Formulae to DFA : $S(x, y)$

- *bab* satisfies $S(x, y)$ with assignment $x = 0$ or $y = 1$ or $x = 1, y = 2$.
- Think of this as $\begin{matrix} bab \\ 100 \\ 010 \end{matrix}$ or $\begin{matrix} bab \\ 010 \\ 001 \end{matrix}$

Atomic Formulae to DFA : $S(x, y)$

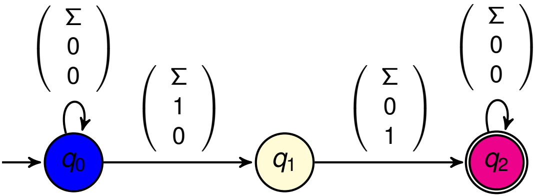
- *bab* satisfies $S(x, y)$ with assignment $x = 0$ or $y = 1$ or $x = 1, y = 2$.
- Think of this as $\begin{matrix} bab \\ 100 \\ 010 \end{matrix}$ or $\begin{matrix} bab \\ 010 \\ 001 \end{matrix}$
- The first row is over Σ, and the second, third rows capture a possible assignment to x, y

Atomic Formulae to DFA : $S(x, y)$

- *bab* satisfies $S(x, y)$ with assignment $x = 0$ or $y = 1$ or $x = 1, y = 2$.
- Think of this as $\begin{array}{c} bab \\ 100 \\ 010 \end{array}$ or $\begin{array}{c} bab \\ 010 \\ 001 \end{array}$
- The first row is over Σ, and the second, third rows capture a possible assignment to x, y
- Think of an extended alphabet $\Sigma' = \Sigma \times \{0, 1\}^2$, and construct an automaton over Σ'.

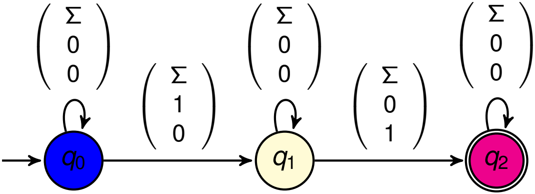
Atomic Formulae to DFA : $S(x, y)$

- *bab* satisfies $S(x, y)$ with assignment $x = 0$ or $y = 1$ or $x = 1, y = 2$.
- Think of this as $\begin{matrix} bab \\ 100 \\ 010 \end{matrix}$ or $\begin{matrix} bab \\ 010 \\ 001 \end{matrix}$
- The first row is over Σ, and the second, third rows capture a possible assignment to x, y
- Think of an extended alphabet $\Sigma' = \Sigma \times \{0, 1\}^2$, and construct an automaton over Σ'.
- Deterministic, not complete.



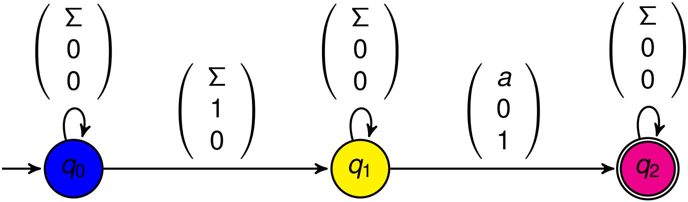
Atomic Formulae to DFA : $x < y$

- *bab* satisfies $x < y$ with assignment $x = 0$ or $y = 1$ or $x = 1, y = 2$ or $x = 0, y = 2$.



Simple Formulae to DFA

- $x < y \wedge Q_a(y)$
- $\Sigma' = \Sigma \times \{0, 1\} \times \{0, 1\}$
- Obtain intersection of DFA for $x < y$ and $Q_a(y)$

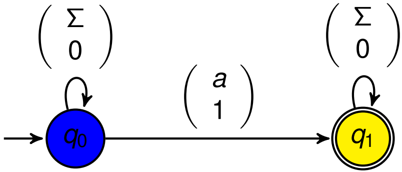


Formulae to DFA

- Given $\varphi(x_1, \ldots, x_n)$, a FO formula over Σ, consider the extended alphabet
$$\Sigma' = \Sigma \times \{0, 1\}^n$$
- Assign values to x_i 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

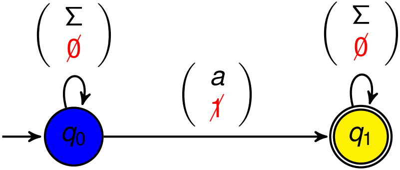
Quantifiers

- $Q_a(x)$



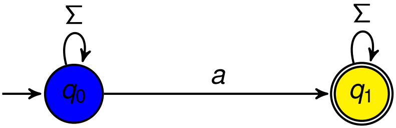
Handling Quantifiers

- $\exists x Q_a(x)$



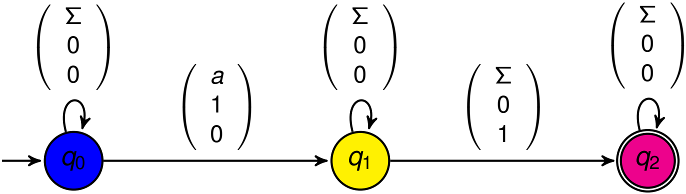
Handling Quantifiers

- $\exists x Q_a(x)$



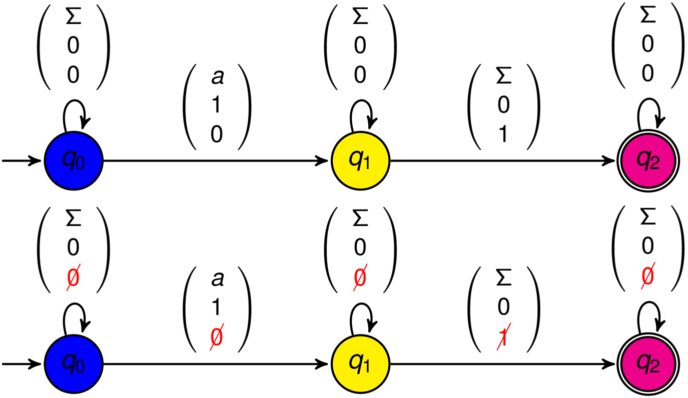
Handling Quantifiers

- $Q_a(x) \wedge \exists y(x < y)$



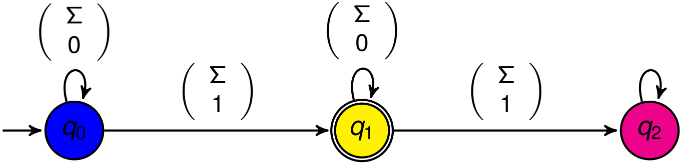
Handling Quantifiers

- $Q_a(x) \land \exists y(x < y)$



Handling Quantifiers: $\forall x(x \neq x)$

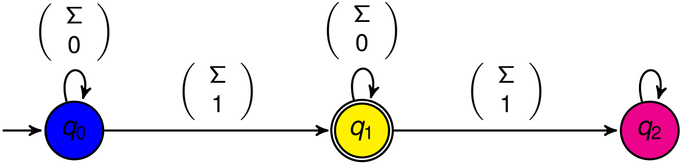
- $(x = x)$



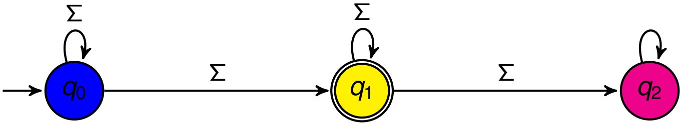
- $\exists x(x = x)$

Handling Quantifiers: $\forall x(x \neq x)$

- $(x = x)$



- $\exists x(x = x)$



- $\neg\exists x(x = x)$

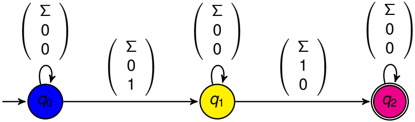
Handling Quantifiers : Summary

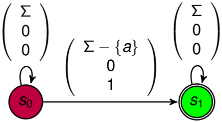
Quantifier Lemma

Let $L \subseteq (\Sigma \times \{0,1\}^n)^*$ be defined by $\varphi(x_1, \ldots, x_n)$. Let $f : (\Sigma \times \{0,1\}^n)^* \to (\Sigma \times \{0,1\}^{n-1})^*$ be the projection $f(w, c_1, \ldots, c_n) = (w, c_1, \ldots, c_{n-1})$. Then $\exists x_n \varphi(x_1, \ldots, x_{n-1})$ defines $f(L)$.

Handling Quantifiers

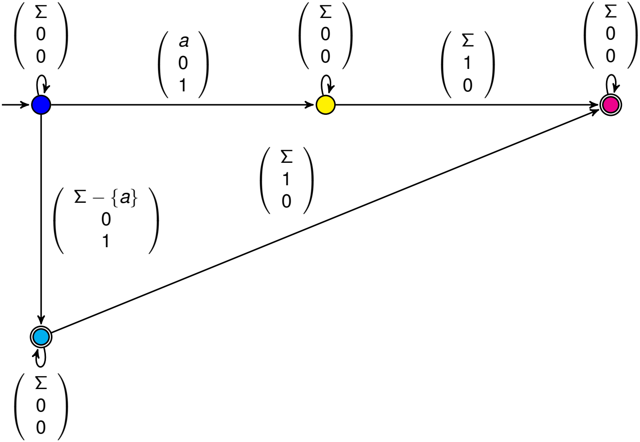
- $\exists y \forall x [x \leqslant y \wedge Q_a(y)] = \exists y [\neg \exists x [x > y \vee \neg Q_a(y)]]$
- $[x > y \vee \neg Q_a(y)]$
- Check the automaton and correct/complete it!





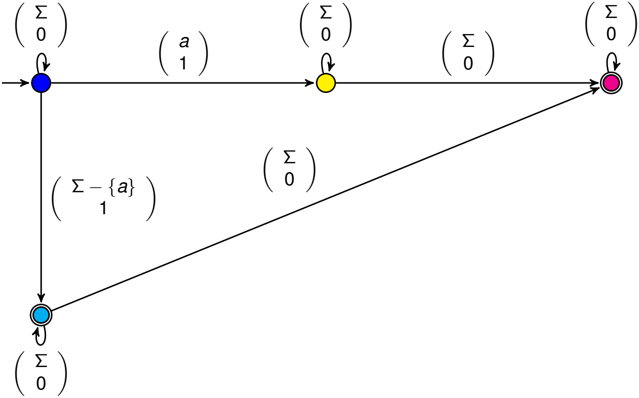
Handling Quantifiers

- $[x > y \vee \neg Q_a(y)]$



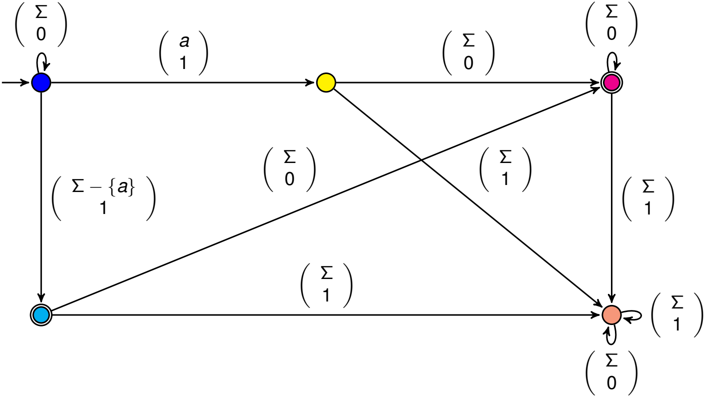
Handling Quantifiers

- $\exists x[x > y \vee \neg Q_a(y)]$



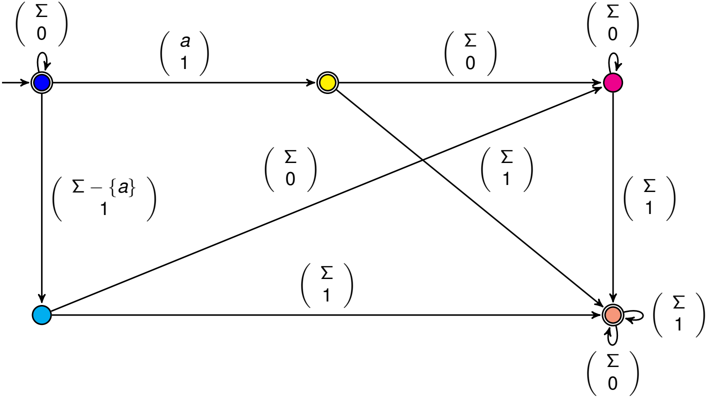
Handling Quantifiers

- $\neg \exists x[x > y \vee \neg Q_a(y)]$
- Deterministic Automaton corresponding to $\exists x[x > y \vee \neg Q_a(y)]$



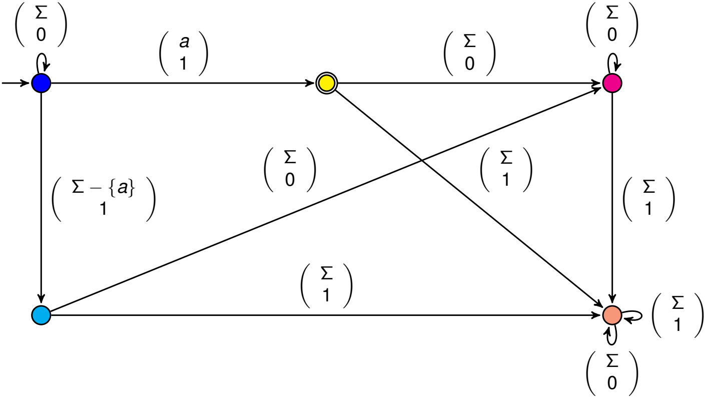
Handling Quantifiers

- Incorrect deterministic Automaton corresponding to $\neg\exists x[x > y \vee \neg Q_a(y)]$



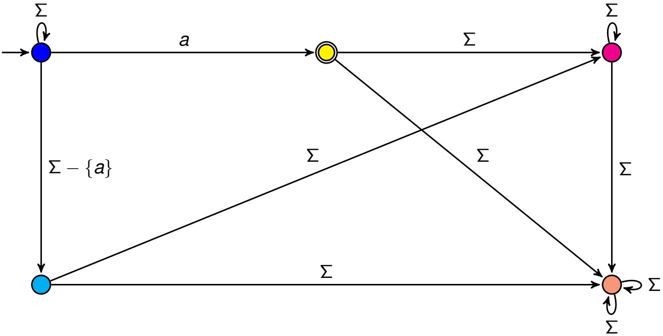
Handling Quantifiers

- $\neg\exists x[x > y \vee \neg Q_a(y)]$
- Deterministic Automaton corresponding to $\neg\exists x[x > y \vee \neg Q_a(y)]$
- Intersect with $\begin{pmatrix} \Sigma \\ 0 \end{pmatrix}^* \begin{pmatrix} \Sigma \\ 1 \end{pmatrix} \begin{pmatrix} \Sigma \\ 0 \end{pmatrix}^*$



Handling Quantifiers

- $\exists y \neg \exists x [x > y \vee \neg Q_a(y)] = \exists y \forall x [x \leqslant y \wedge Q_a(y)]$



Points to Remember

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

Points to Remember

- Given the automaton for $\varphi(x_1, \ldots, x_n)$, the automaton for $\exists x_i \varphi(x_1, \ldots, x_n)$ 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 $\neg\exists x_i \varphi(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$
- 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 Computational Effort

Given NFAs A_1, A_2 each with atmost n states,

- The union has atmost $2n + 1$ states
- Intersection has atmost n^2 states
- The complement has atmost 2^n states
- The projection has atmost n states

The Computational Effort

- $\psi = Q_1 \dots Q_n \varphi$. If $Q_i = \exists$ for all i, then size of A_ψ is same the size of A_φ.
- When $Q_1 = \exists, Q_2 = \forall, \dots$: each $\forall$ quantifier can create a 2^n blowup in automaton size
- Size of automaton is

$$2^{2^{2^{2^{2^{2^{2^n}}}}}}$$

  where the tower height k is the quantifier alternation size.
- This number is indeed a lower bound!

The Automaton-Logic Connection

Given any FO sentence φ, one can construct a DFA A_φ such that $L(\varphi) = L(A_\varphi)$.