CS 228 : Logic in Computer Science

Krishna. S

Recap

- We focus on FO over words : the signature has $<, S, Q_a, Q_b, \ldots$. Remember you always have $=$ with you. Recall the terms structure, universe, and assignment.

Recap

- We focus on FO over words : the signature has $<, S, Q_a, Q_b, \dots$. Remember you always have $=$ with you. Recall the terms structure, universe, and assignment.
- Consider the formula $\varphi(y) = Q_b(y) \wedge \forall x(x < y \rightarrow Q_a(x))$, and the word $W = aabacabacaa$. Does $W \models_\alpha \varphi(y)$ for some assignment α?
- Let $\psi(y, w)$ be the formula $Q_a(w) \wedge Q_b(y) \wedge \forall x(Q_a(x) \rightarrow x > y) \wedge \exists z[Q_b(z) \wedge \forall t(z \geqslant t)]$. What is $L(\psi)$?

Recap

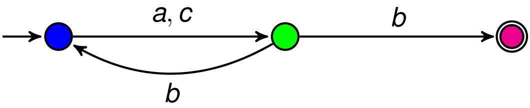
- We focus on FO over words : the signature has $<, S, Q_a, Q_b, \ldots$. Remember you always have $=$ with you. Recall the terms structure, universe, and assignment.
- Consider the formula $\varphi(y) = Q_b(y) \wedge \forall x(x < y \rightarrow Q_a(x))$, and the word $W = aabacabacaa$. Does $W \models_\alpha \varphi(y)$ for some assignment α?
- Let $\psi(y, w)$ be the formula $Q_a(w) \wedge Q_b(y) \wedge \forall x(Q_a(x) \rightarrow x > y) \wedge \exists z[Q_b(z) \wedge \forall t(z \geqslant t)]$. What is $L(\psi)$?
- Formula φ is satisfiable iff $L(\varphi) \neq \emptyset$.
- Formula φ is valid iff $L(\varphi) = \Sigma^*$.

Recap

- We focus on FO over words : the signature has $<, S, Q_a, Q_b, \ldots$. Remember you always have $=$ with you. Recall the terms structure, universe, and assignment.
- Consider the formula $\varphi(y) = Q_b(y) \wedge \forall x(x < y \to Q_a(x))$, and the word $W = aabacabacaa$. Does $W \models_\alpha \varphi(y)$ for some assignment α?
- Let $\psi(y, w)$ be the formula $Q_a(w) \wedge Q_b(y) \wedge \forall x(Q_a(x) \to x > y) \wedge \exists z[Q_b(z) \wedge \forall t(z \geqslant t)]$. What is $L(\psi)$?
- Formula φ is satisfiable iff $L(\varphi) \neq \emptyset$.
- Formula φ is valid iff $L(\varphi) = \Sigma^*$.
- Question : How to check satisfiability of FO over words?

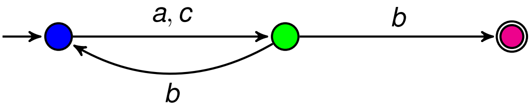
Idea for SAT checking

- Given FO formula φ over an alphabet Σ, construct an edge labeled graph G_φ : a graph whose edges are labeled by Σ.



Idea for SAT checking

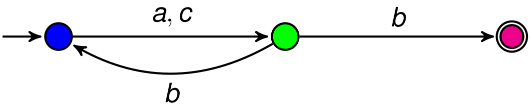
- Given FO formula φ over an alphabet Σ, construct an edge labeled graph G_φ : a graph whose edges are labeled by Σ.



- Each path in the graph gives rise to a word over Σ, obtained by reading off the labels on the edges

Idea for SAT checking

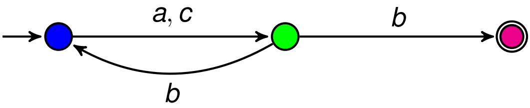
- Given FO formula φ over an alphabet Σ, construct an edge labeled graph G_φ : a graph whose edges are labeled by Σ.



- Each path in the graph gives rise to a word over Σ, obtained by reading off the labels on the edges
- G_φ has some special kinds of vertices
  - There is a unique vertex called the start vertex (blue vertex)
  - There are some vertices called good vertices (magenta vertex)

Idea for SAT checking

- Given FO formula φ over an alphabet Σ, construct an edge labeled graph G_φ : a graph whose edges are labeled by Σ.



- Each path in the graph gives rise to a word over Σ, obtained by reading off the labels on the edges
- G_φ has some special kinds of vertices
  - There is a unique vertex called the start vertex (blue vertex)
  - There are some vertices called good vertices (magenta vertex)
- Read off words on paths from the start vertex to any final vertex and call this set of words $L(G_\varphi)$
- Ensure that G_φ is constructed such that $L(\varphi) = L(G_\varphi)$.

Idea for SAT checking

- Why does this help?

Idea for SAT checking

- Why does this help?
- We know how to check the existence of a path between 2 vertices in a graph easily (how?)

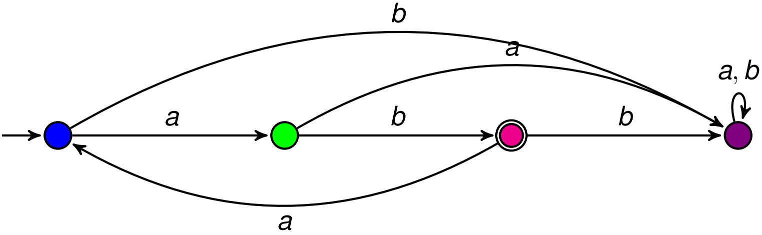
Idea for SAT checking

- Why does this help?
- We know how to check the existence of a path between 2 vertices in a graph easily (how?)
- If somehow we manage to construct G_φ correctly, then checking satisfiability of φ is same as checking the reachability of some good vertex from the start vertex of G_φ.

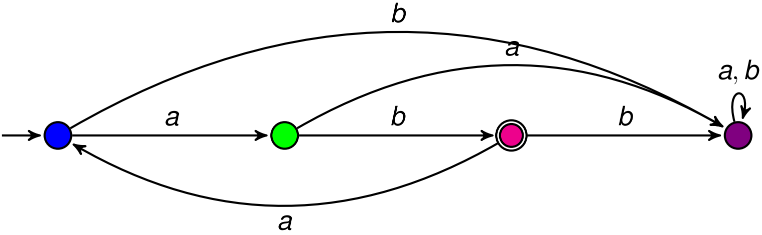
Idea for SAT checking

- Why does this help?
- We know how to check the existence of a path between 2 vertices in a graph easily (how?)
- If somehow we manage to construct G_φ correctly, then checking satisfiability of φ is same as checking the reachability of some good vertex from the start vertex of G_φ.
- How to construct G_φ?

A First Labeled Graph A

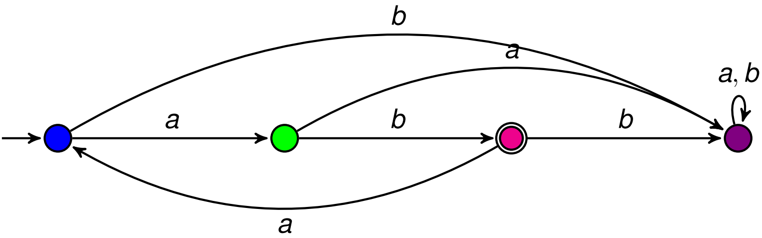


A First Labeled Graph A



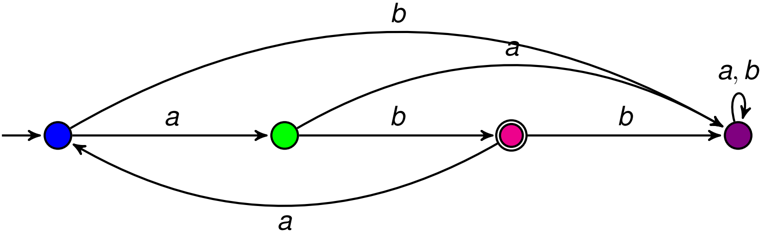
- Let us call the vertices of the graph states

A First Labeled Graph A



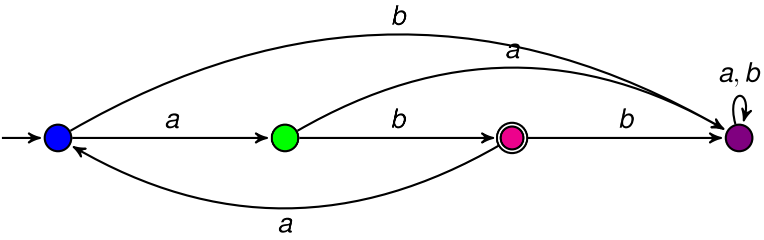
- Let us call the vertices of the graph states
- A path from one state to another gives a word over $\Sigma = \{a, b\}$

A First Labeled Graph A



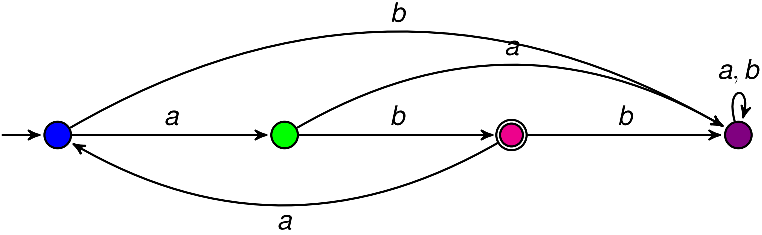
- Let us call the vertices of the graph states
- A path from one state to another gives a word over $\Sigma = \{a, b\}$
- The graph accepts words along paths from an initial state to a good state

A First Labeled Graph *A*



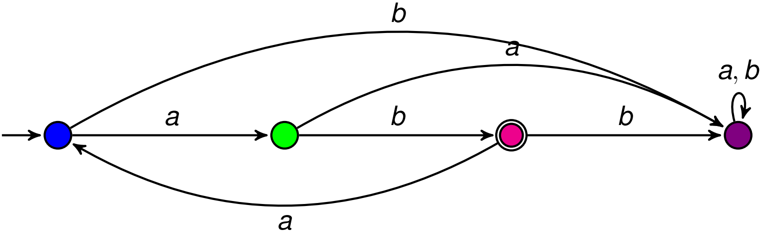
- Let us call the vertices of the graph states
- A path from one state to another gives a word over $\Sigma = \{a, b\}$
- The graph accepts words along paths from an initial state to a good state
- The set of words accepted by the graph is called the language of the graph

A First Labeled Graph A



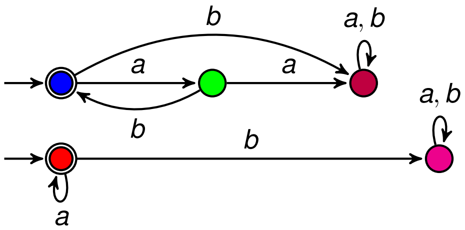
- What is the language L accepted by this graph, $L(A)$?

A First Labeled Graph A

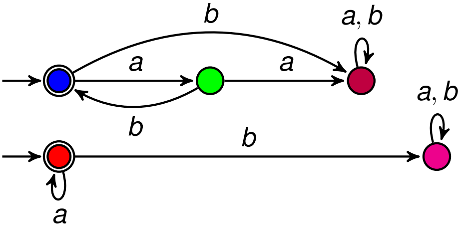


- What is the language L accepted by this graph, $L(A)$?
- Write an FO formula φ such that $L(\varphi) = L(A)$

A Second and a Third Graph B, C

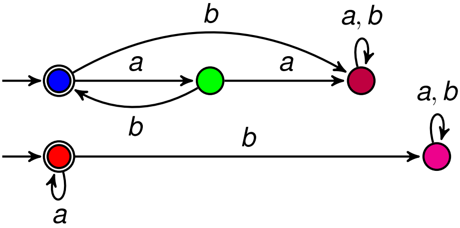


A Second and a Third Graph B, C



- What are $L(B), L(C)$?

A Second and a Third Graph B, C



- What are $L(B), L(C)$?
- Give an FO formula φ such that $L(\varphi) = L(B) \cup L(C)$

These graphs are called........

A deterministic finite state automaton (DFA) $A = (Q, \Sigma, \delta, q_0, F)$

These graphs are called........

A deterministic finite state automaton (DFA) $A = (Q, \Sigma, \delta, q_0, F)$

- Q is a finite set of states

These graphs are called........

A deterministic finite state automaton (DFA) $A = (Q, \Sigma, \delta, q_0, F)$

- Q is a finite set of states
- Σ is a finite alphabet

These graphs are called........

A deterministic finite state automaton (DFA) $A = (Q, \Sigma, \delta, q_0, F)$

- Q is a finite set of states
- Σ is a finite alphabet
- $\delta : Q \times \Sigma \to Q$ is the transition function

These graphs are called........

A deterministic finite state automaton (DFA) $A = (Q, \Sigma, \delta, q_0, F)$

- Q is a finite set of states
- Σ is a finite alphabet
- $\delta : Q \times \Sigma \to Q$ is the transition function
- $q_0 \in Q$ is the initial state

These graphs are called........

A deterministic finite state automaton (DFA) $A = (Q, \Sigma, \delta, q_0, F)$

- Q is a finite set of states
- Σ is a finite alphabet
- $\delta : Q \times \Sigma \to Q$ is the transition function
- $q_0 \in Q$ is the initial state
- $F \subseteq Q$ is the set of final states

These graphs are called........

A deterministic finite state automaton (DFA) $A = (Q, \Sigma, \delta, q_0, F)$

- Q is a finite set of states
- Σ is a finite alphabet
- $\delta : Q \times \Sigma \to Q$ is the transition function
- $q_0 \in Q$ is the initial state
- $F \subseteq Q$ is the set of final states
- $L(A)$=all words leading from q_0 to some $f \in F$

Languages, Machines and Logic

A language $L \subseteq \Sigma^*$ is called regular iff there exists some DFA A such that $L = L(A)$.

Languages, Machines and Logic

A language $L \subseteq \Sigma^*$ is called regular iff there exists some DFA A such that $L = L(A)$.
A language $L \subseteq \Sigma^*$ is called FO-definable iff there exists an FO formula φ such that $L = L(\varphi)$.

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Contains *abb*

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Contains *abb*
- Even length words

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Contains *abb*
- Even length words
- Contains a *b* and ends with *aa*

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Contains *abb*
- Even length words
- Contains a *b* and ends with *aa*
- Begins with *a*, ends with *b*, and has a pair of consecutive *a*'s

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Contains *abb*
- Even length words
- Contains a *b* and ends with *aa*
- Begins with *a*, ends with *b*, and has a pair of consecutive *a*'s
- There are two occurrences of *b* between which only *a*'s occur

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Contains *abb*
- Even length words
- Contains a *b* and ends with *aa*
- Begins with *a*, ends with *b*, and has a pair of consecutive *a*'s
- There are two occurrences of *b* between which only *a*'s occur
- Right before the last position is an *a*

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Contains *abb*
- Even length words
- Contains a *b* and ends with *aa*
- Begins with *a*, ends with *b*, and has a pair of consecutive *a*'s
- There are two occurrences of *b* between which only *a*'s occur
- Right before the last position is an *a*