CS 228 : Logic in Computer Science

S. Krishna

Satisfiability of FO over Words

- Recall : We work on FO over words
- Fix an alphabet Σ, and a signature $\tau = (S, <, Q_a, Q_b, \ldots, Q_z)$ (assume $\Sigma = \{a, \ldots, z\}$)

Satisfiability of FO over Words

- Recall : We work on FO over words
- Fix an alphabet Σ, and a signature $\tau = (S, <, Q_a, Q_b, \dots, Q_z)$ (assume $\Sigma = \{a, \dots, z\}$)
- Given an FO sentence φ over words over an alphabet Σ, $L(\varphi) = \{w \in \Sigma^* \mid w \models \varphi\}$.
- For example, for $\varphi = \forall x \exists y (x \leqslant y \wedge Q_a(y))$, $aba \models \varphi$. Hence, $aba \in L(\varphi)$.

Satisfiability of FO over Words

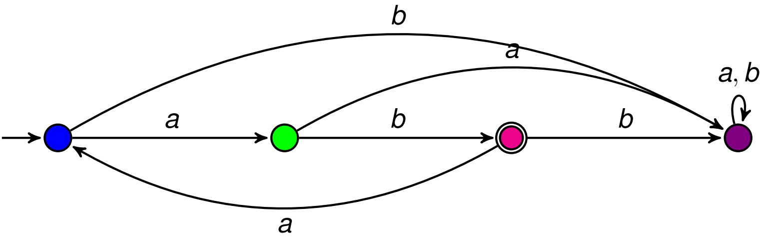
- Recall : We work on FO over words
- Fix an alphabet Σ, and a signature $\tau = (S, <, Q_a, Q_b, \ldots, Q_z)$ (assume $\Sigma = \{a, \ldots, z\}$)
- Given an FO sentence φ over words over an alphabet Σ, $L(\varphi) = \{w \in \Sigma^* \mid w \models \varphi\}$.
- For example, for $\varphi = \forall x \exists y (x \leqslant y \wedge Q_a(y))$, $aba \models \varphi$. Hence, $aba \in L(\varphi)$.
- Satisfiability Problem : Given an FO sentence φ, is φ satisfiable? That is, is $L(\varphi) \neq \emptyset$?

Satisfiability of FO over Words

- Recall : We work on FO over words
- Fix an alphabet Σ, and a signature $\tau = (S, <, Q_a, Q_b, \ldots, Q_z)$ (assume $\Sigma = \{a, \ldots, z\}$)
- Given an FO sentence φ over words over an alphabet Σ, $L(\varphi) = \{w \in \Sigma^* \mid w \models \varphi\}$.
- For example, for $\varphi = \forall x \exists y (x \leqslant y \wedge Q_a(y))$, $aba \models \varphi$. Hence, $aba \in L(\varphi)$.
- Satisfiability Problem : Given an FO sentence φ, is φ satisfiable? That is, is $L(\varphi) \neq \emptyset$?
- Validity Problem : Given an FO sentence φ, is φ valid? That is, does every word satisfy φ? That is, is $L(\varphi) = \Sigma^*$?

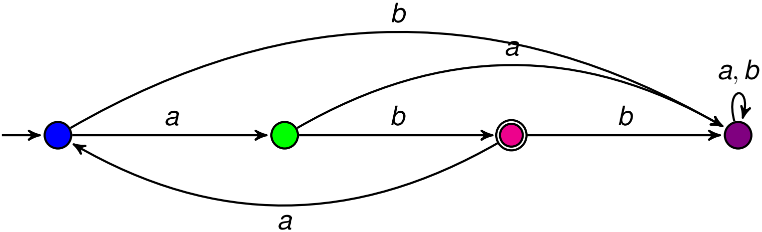
In Search of an Algorithm?

A First Machine A



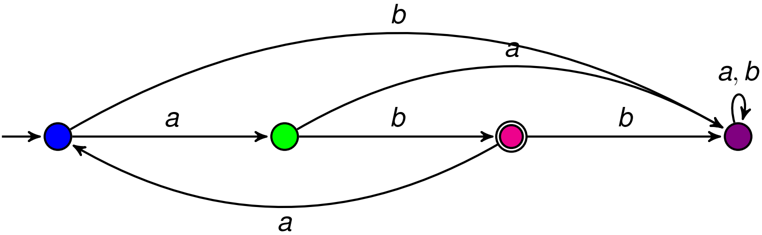
- Colored circles called states
- Arrows between circles called transitions
- Blue state called an initial state
- Doubly circled state called a final state

A First Machine A



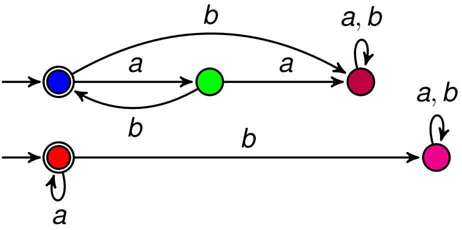
- A path from one state to another gives a word over $\Sigma = \{a, b, c\}$
- The machine accepts words along paths from an initial state to a final state
- The set of words accepted by the machine is called the language accepted by the machine

A First Machine A



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

A Second and a Third Machine B, C



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

Finite State Machines

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 $L = L(A)$.
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 $L \subseteq \Sigma^*$:

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