Nondeterministic Finite Automata

COMPSCI 3331

Outline

- Nondeterminism
- NFAs.
- NFA acceptance
- NFAs vs DFAs.

Deterministic? Finite Automaton

What was "determinism"?

- Everything about our computation was unambiguous.
- We knew which state to start in.
- Given our current state and our current input symbol, we knew what state to proceed to.
- When computation completed, we checked whether or not it was a final state.

Nondeterminism

Nondeterminism is a central concept in theoretical computer science.

- Nondeterminism: viewed as a way for computers to make "guesses".
- Nondeterminism gives our automata several "choices" to explore at each step.
- Given a state and an input symbol, there are several states we could proceed to.
- Result: many different computations on one input.
- We accept if **any** of the possible computations succeed.

Example: Nondeterminism

$$L = \{xay \ : \ x \in \{a,b\}^*, y \in \{a,b\}\}.$$

Example: Nondeterminism

- **Not a DFA**: one state has two transitions labelled *a* leaving it.
- Idea: stay in state until we're sure that there are two more letters remaining. Then read *a* and the last letter.
- How does the automaton **know** there are only two letters left?
- It doesn't: several different possible computations on a given string.
- e.g., computation on string *aabaa*.
- NFA: accept if **ANY** of the computations accept.

Nondeterminism Finite Automata

An NFA is again a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

- Q is a finite set of states,
- Σ is a finite alphabet.
- $\delta : Q \times \Sigma \to 2^Q$ is the transition function.
- $q_0 \in Q$ is the start state.
- $F \subseteq Q$ is the set of final states.

Nondeterministic Transition Functions

What does $\delta : Q \times \Sigma \to 2^Q$ mean?

- DFA: the result of $\delta(q, a)$ is a single state which
  - the unique state we go to from state q when reading $a \in \Sigma$.
- NFA: the result of $\delta(q, a)$ is a **set of states**.
  - When in state q and reading $a \in \Sigma$, the NFA can choose to go to any state $q' \in \delta(q, a)$.

Accepting Strings Nondeterministically

- An NFA accepts a word if **any** of its **nondeterministic paths** accept.
- If none of the paths accept, then the word is rejected.
- "Guesses": an NFA accepts a word if it can guess a path through the NFA, and that path ends at a final state.
- If there is no possible guessed path from the start state to a final state, then the word is rejected.

Accepting Strings Nondeterministically

- To determine whether a word w is accepted by an NFA: **trace** all possible paths through the NFA.
- Alternate way: “show a path” through M from the start state to a final state labelled by w.
  - This can’t be used to show that word is **not** accepted by the NFA.

Accepting Strings Nondeterministically

We extend δ from acting on $Q \times \Sigma$ to $Q \times \Sigma^*$ with the following definition:

$$
\begin{aligned}
\delta(q,\varepsilon) &= \{q\} \quad \forall q \in Q \\
\delta(q,wa) &= \bigcup_{q' \in \delta(q,w)} \delta(q',a) \quad \forall q \in Q, w \in \Sigma^* \text{ and } a \in \Sigma
\end{aligned}
$$

What does $\delta : Q \times \Sigma^* \to 2^Q$ mean?

Accepting Strings Nondeterministically

$$\delta(q, wa) = \bigcup_{q' \in \delta(q,w)} \delta(q', a)$$

- $\delta(q, w)$ is a **set of states**.
- $q' \in \delta(q, w)$: q' is one of the states we can get to by reading the word w starting in state q.
- $\delta(q', a)$: the states we can get to from q' by reading a.
- Thus, $\bigcup_{q' \in \delta(q,w)} \delta(q', a)$ is the set of states we can get to by reading wa.

Accepting Strings Nondeterministically

Remember:

- a word $w \in \Sigma^*$ is accepted by an NFA $M = (Q, \Sigma, \delta, q_0, F)$ if there is any path labelled w from q_0 to a final state in M.
- for a word w, $\delta(q, w)$ represents the set of **all** possible paths from q labelled with w.

A word w is accepted by $M = (Q, \Sigma, \delta, q_0, F)$ if

$$\delta(q_0, w) \cap F \neq \emptyset.$$

In words: there is a final state in the set of all states we can get to by a path labelled with w starting at q_0.

The language accepted by an NFA

The language accepted by an NFA $M = (Q, \Sigma, \delta, q_0, F)$ is denoted $L(M)$:

$$L(M) = \{w \in \Sigma^* \ : \ \delta(q_0, w) \cap F \neq \emptyset\}.$$

Note:

- Every DFA is an NFA (i.e., NFAs accept all regular languages.)
- Are there languages which are not regular, but accepted by an NFA?

Computational Power of NFAs

Theorem: Let $M = (Q, \Sigma, \delta, q_0, F)$ be an NFA. Then there exists a DFA $M' = (Q', \Sigma, \delta', q_0', F')$ such that $L(M) = L(M')$.

Proof: By constructing the DFA M'. (in class)

More on Nondeterminism

- Nondeterminism is a central concept in the theory of computing: does being able to "guess" get us anything?
- Or: does being able to examine multiple possibilities simultaneously get us anything?
- DFA = NFA: nondeterminism is not more powerful for finite automata.
- For other classes of automata (which we see later), the answer is yes; nondeterminism gets us something in that case.
- For still other classes, we **don't know**.

More on Nondeterminism

- Nondeterminism is **an idea**: we don't actually implement nondeterminism.
- Nondeterminism is **important**: helps us understand algorithm design.
- Nondeterminism is **tempting**: sometimes hard problems become easy with nondeterminism.