CSC301

Lecture 4 summary

1910456

Nondeterministic Finite Automata (NFA)

What makes an automata deterministic?

For each state, there must be only one transition, a, for each symbol.
In other words, it must be stated what ONE action to take for each symbols in it's alphabet.

But, for NFAs, there can be multiple actions taken by the same symbol or there can be no transition specified for a symbol in a given state (In that case, if we do get a symbol which is not specified, we automatically die).

Other than the differences mentioned above, NFAs are the same as DFAs.

With DFAs, we check if a string is accepted or not.

With NFAs we do the same.

---

https://imgur.com/a/g2yAFxl

Example 1

If q_0 reads 1, we stay at q_0 but also move to q_1. We are at both states simultaneously.

Then if we get 0, we remain at q_0 because of the loop but also transition from q_1 to q_2.

Then if we read 1, we stay at q_0, move to q_1 from q_0 and move to q_3 from q_2.

Now we are at q_0, q_1, q_3 simultaneously.

Upon reading 0 or 1, in q_3, we die.

Example 2 (Contains)

This is an NFA over {0, 1} that accepts any string that contains 001.

Upon reaching q_3 (final state), we have a self-loop which means the NFA can accept anything if it has passed the test of 001.

Formal Definition of NFA

And NFA (like DFA) is a 5–tuple (Q, Σ, δ, q_0. F)

The only difference is in the transition function where,

δ : Q x ($\Sigma \cup \{\varepsilon\}$) $\rightarrow$ subsets of Q

e.g., for example 1, ($\{q_0\}$, 1) $\rightarrow$ $\{q_0, q_1\}$

In other cases, something like this may happen $(\{q_0\}, \varepsilon) \rightarrow \{q_3\}$

Which means even if nothing happens (without reading anything), we can change state. [Epsilon/Empty transition]

Definition of Language of an NFA same as DFA.

Except, we can remain at multiple states once all the actions are taken.

If there is at least one state which is a subset of F then the string is part of the language of the NFA.

Example 3

This NFA accepts a, b, aab, bab, aabab, ...

For the string "bab",

$(q_0, bab) \rightarrow (\{q_0, q_1\}, bab)$ //due to the ε we move to q_1 for free.

$(\{q_0, q_1\}, bab) \rightarrow (\{q_2\}, ab) \rightarrow (\{q_1\}, b) \rightarrow (\{q_2\}, \varepsilon)$ //$q_2 \subset F$, therefore accepted.

Example 4

It will accept 10010.

It will not accept a string with substring "11" as then all states will die.

Therefore, even if all 3 states are active,

$(\{q_0, q_1, q_2\}, 1100) \rightarrow (\{q_1, q_2\}, 100) \rightarrow (\varnothing, 00)$ //all states died, no need to continue.

Example 5 (EITHER OR)

An NFA which accepts all strings with an even number of 0s or an odd number of 1s.

We construct two DFAs. Top is for even 0's requirement and Bottom one is for odd 1's requirement.

Since we are directed to construct NFA which says either one is acceptable, we join the two with a q_0 state with ε going to both the start states of the DFAs. This means we start from both Top and Bottom DFAs simultaneously.