CS241 Lecture 10

Graham Cooper

June 8th 2015

Recall:
A DFA is a 5-tuble $(\Sigma, Q, q_0, A.\delta)$

- Σ, Q finite, non-empty sets (alphabet, states)
- $q_0 \in Q$ (start)
- $A \subseteq Q$ (accepting)
- $\delta : Q \times \Sigma \implies Q$ (transition)

It accepts a word w if $\delta^*(q_0, w) \in A$

Implementing a DFA

```
int state = q_0
char c
while not EOF do
-- read c
-- case state of
-- -- q_0: case c of
-- -- -- a: state = ...
-- -- -- b: state= ...
-- -- -- ...
-- -- q_1: case c of
-- -- -- a: state = ...
-- -- -- b: state = ...
-- -- -- ...
...
-- -- q_n: case c of
-- -- -- a: state = ...
-- -- -- b: state = ...
-- -- -- ...
-- end case
end while
return true if state is in A
```

Or use a lookup table

	states		
characters		next state	

DFA's with actions

We can attach computations to the arcs of a DFA.

eg: L = {binary numbers with no loading 0's}
- Compute the value of the number
- 1(0|1)*|0
SET PICTURE

What do we gain by making DFA's more complex?
eg:

L = $\{w \in \{a, b\} * | wendswithabb\}$
SEE PICTURE

What if we allowed more than one arc with the same char from the same state?

What would this mean?
Machine chooses one direction or the other (i.e. the machine is non-deterministic).
Accept if some set of choices leads to an accepting state.
With non-determinism, the machine becomes: SEE PICTURE

The machine gueses to stay in the first state until it reaches the final abb, then transitions to accepting.
- NFA's often simpler than DFA's

Formally: - An NFA is a 5-tuble $(\Sigma, Q, q_0, A, \delta)$ where:

- Σ is a finite, non-empty set (Alphabet)
- Q is a finite, non-empty set (states)
- $q_0 \in Q$ (start state)

- $A \subseteq Q$ (accepting states)
- $\delta : Q \times \Sigma \implies$ subsets of Q (2^Q) - non-determinism

Want to acept if some path through the NFA leads to acceptance for the given word (reject if none do)

δ^* for NFA's: set of states X $\Sigma \implies$ set of states

$\delta^*(qs, \epsilon) =$ qs
qs (spoken as multiple q states)
$\delta^*(qs, cw) = \bigcup_{q \in qs}(\delta(q, c), w)$

Then accept w if $\delta^*(\{q_0\}, w) \bigcap A \neq \emptyset$

NFA simulation procedure:
States $\leftarrow \{q_0\}$
while not eof do
– read ch
– states $\leftarrow \bigcup_{q \in states}(\delta(q, ch))$
end do return states $\cap A \neq \emptyset$

Using the NFA:
$\rightarrow$ 1(loops on a,b)$\xrightarrow{a}$ 2 $\xrightarrow{b}$ 3 $\xrightarrow{b}$ 4 accept
simulate baabb

Already read input	Unread Input	states	rename
ϵ	baabb	{1}	A
b	aabb	{1}	A
ba	abb	{1,2}	B
baa	bb	{1,2}	B
baab	b	{1,3}	C
baabb	ϵ	{1,4}	D

{1,4} $\cap$ {4} = {4} $\neq \emptyset$, accept!