Lecture 7

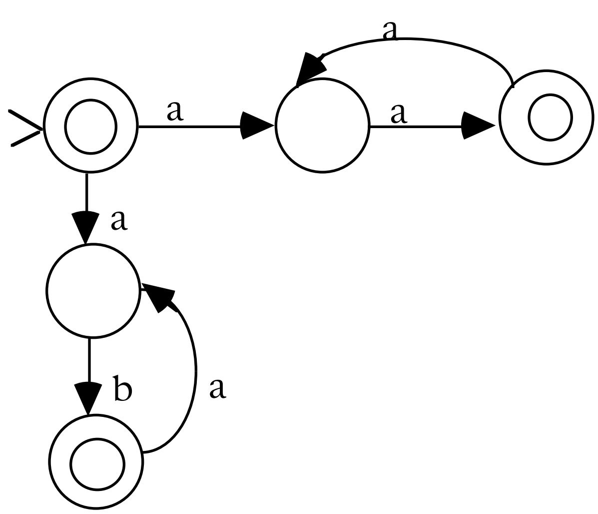
Homework #7: 2.2.1, 2.2.2, 2.2.3 (hand in c and d),
Misc: Given: M, a NFA
Prove: $(q,xy) \vdash^* (p,y)$ iff $(q,x) \vdash^* (p,e)$
(follow proof done in class today)

Last time: introduced our first computational model – the DFA.

Today we'll expand on the DFA model. This expansion will not increase the power of the model, but will increase the expressiveness of the model.

What are we adding? **Nondeterminism** - i.e., when in a given state, might be able to select from several transitions for a given input symbol.

For example. $(aa)^* \cup (ab)^*$



Note: if there is no transition specified for a particular symbol out of a state, the implication is that the transition would be to a "dead state".

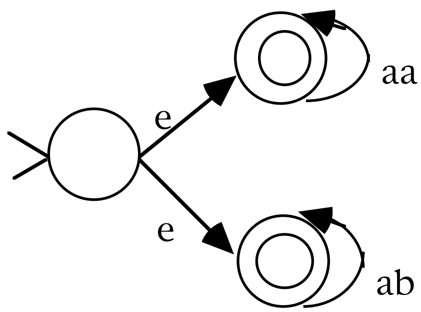
Also Note: acceptance is defined as follows - a string $w \in \Sigma^*$ is accepted by a NFA iff there is *some* way to get to a final state from the start state.

Why introduce nondeterminism?

1) equivalent to DFA
2) but provides additional representational flexibility

And we'll allow even more flexibility in NFAs:
1) e-transitions
2) transitions on strings rather than on symbols

An example.

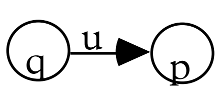


Exercise. Give an NFA to recognize reals (recall exercise for DFAs).

Def. A nondeterministic finite automaton (NFA or NDFA) is a quintuple $M = (K, \Sigma, \Delta, s, F)$, where

K is a finite set of states
Σ is an alphabet
$s \in K$ is the start state
$F \subseteq K$ is the set of final states
Δ is the transition **relation**, a **finite subset** of
$K \times \Sigma^* \times K$

$(q,u,p) \in \Delta$ iff $u \in \Sigma^*$ and



We'll largely be able to use the same notation and definitions as before, but we need to make a few modifications:

$\vdash$ is **"yields in one step"**

$(q,w) \vdash (q^1,w^1)$ iff
$w = uw^1$ and
$(q,u,q^1) \in \Delta$.

$\vdash^*$ is the reflexive, transitive closure of $\vdash$

$w \in \Sigma^*$ is accepted iff
$(s,w) \vdash^* (q,e)$ and $q \in F$.

Note: that the state at any point is determined by what's been read - you can't look ahead. And there is only limited memory of what's been read (just like a DFA).

Lemma. Let $M = (K, \Sigma, \Delta, s, F)$ be a NFA;
$q, r \in K$
$x, y \in \Sigma^*$
Then $(q, xy) \vdash^* (r,e)$ if, for some $p \in K$
$(q,x) \vdash^* (p,e)$ and $(p,y) \vdash^* (r,e)$.

We will prove this by induction on the length of $\vdash^*$ in
$(q,x) \vdash^* (p,e)$,
i.e., on the number of steps in the computation.

<u>Basis.</u> Let $(q,x) \vdash^* (p,e)$ in 0 steps.
Then $q = p$, and
$x = e$.
So $(q, xy) = (p, ey) = (p,y) \vdash^* (r,e)$

<u>Induction Step.</u> Assume $(q,x) \vdash^* (p,e)$ and $(p,y) \vdash^* (r,e)$
$\rightarrow (q, xy) \vdash^* (r,e)$, when $(q,x) \vdash^* (p,e)$ in $<$ n steps.

Now consider $(q,x) \vdash^* (p,e)$ and $(p,y) \vdash^* (r,e)$,
where $(q,x) \vdash^* (p,e)$ in n steps.

Then
$(q,x) = (q,ux^1) \vdash (q^1,x^1) \vdash^* (p,e)$, where
$x = ux^1$
$(q,u,q^1) \in \Delta$
$(q^1, x^1) \vdash^* (p,e)$ in (n-1) steps

But then $(q^1, x^1y) \vdash^* (r,e)$, by the induction hypothesis.

So $(q,xy) = (q,ux^1y) \vdash (q^1,x^1y) \vdash^* (r,e)$.

Note that the following is a stronger version that holds for DFAs:

Given $(p,y) \vdash^* (r,e)$
$(q,xy) \vdash^* (r,e)$ iff $(q,x) \vdash^* (p,e)$.

Now, let's show that **NFAs = DFAs.**

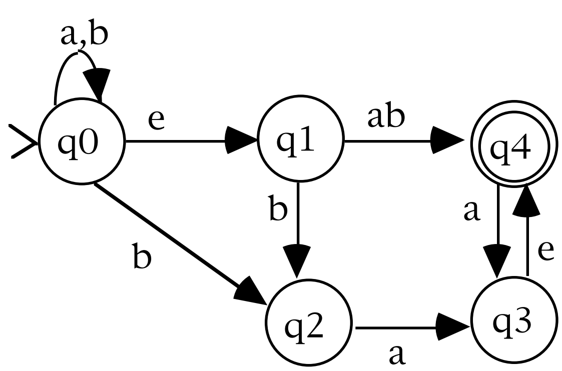
We'll say that two finite automata M_1 and M_2 are equivalent iff $L(M_1) = L(M_2)$.

Thm. For each NFA, there is an equivalent DFA.
Proof will proceed by construction. That is, we will show how to turn an NFA into a DFA.

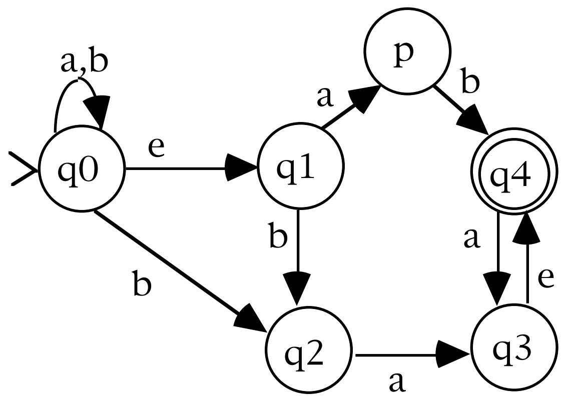
To do this, we need to:
(1) eliminate transitions on e.
(2) eliminate transitions on strings of length > 1.
(3) add transitions to have actions for all symbols in a given state.
(4) eliminate multiple transitions from 1 state.

(2) is easy - just "stretch" the NFA by adding states.

Example.



becomes



(of course, we'd need to prove that the "stretched" NFA is equiv to the original, but it's fairly obvious)

Basically, if $(q,u,q^1) \in \Delta$, and $|u| > 1$, then we can write $u = u_1 u_2 \ldots u_n$.
To "stretch" the NFA, we add transitions to Δ:

$(q,u_1,p_1), (p_1,u_2,p_2), \ldots, (p_{n-1},u_n,q^1)$.

Let's call the initial NFA $M = (K, \Sigma, \Delta, s, F)$, and

the stretched NFA $M^1 = (K^1, \Sigma, \Delta^1, s^1, F^1)$.

Now, for the remainder of the proof:

We will view the NFA as follows: **at any point in time, it can be in many states at once.**

Example cont'd. On input ab, it can be in any of {q0, q1, q2, q4}

We can view this as a single state in a DFA.
The idea is that we're building "multi-states".

Now, what are subsequent states?

Anything that can be reached from one of these on a given input symbol.

So, if the next symbol were "a": {q0, q1, p, q3}

Basically, **the DFA is simulating all moves of the NFA simultaneously.**

Now, before we formalize this, let's look at transitions on e - these need to be considered specially.

Define the states that are **reachable** from a state q **on no input**:

E(q) = $\{p \in K^1: (q,e) \mid\!\text{—}^* (p,e)\}$

or

E(q) = $\{p \in K^1: (q,w) \mid\!\text{—}^* (p,w)\}$

In our Example.

E(q0) = {q0, q1}
E(q1) = {q1}
E(q2) = {q2}
E(q3) = {q3, q4}
E(q4) = {q4}
E(p) = {p}

Before giving the formal construction, let's take these ideas and complete the construction of a DFA from the NFA above:

S" = {q0, q1}

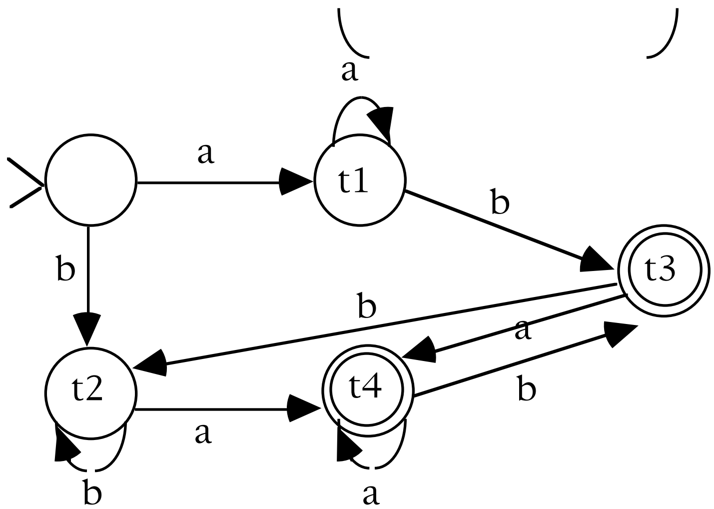
δ"(S", a) = E(q0) ∪ E(p) = {q0, q1, p} = t1
δ"(S", b) = E(q0) ∪ E(q2) = {q0, q1, q2} = t2

δ"(t1, a) = E(q0) ∪ E(p) = {q0, q1, p} = t1
δ"(t1, b) = E(q0) ∪ E(q2) ∪ E(q4)= {q0, q1, q2, q4} = t3 (FINAL STATE)

δ"(t2, a) = E(q0) ∪ E(p) ∪ E(q3) = {q0, q1, p, q3, q4} = t4 (FINAL)
δ"(t2, b) = E(q0) ∪ E(q2) = {q0, q1, q2} = t2

δ"(t3, a) = E(q0) ∪ E(p) ∪ E(q3) = {q0, q1, p, q3, q4} = t4
δ"(t3, b) = E(q0) ∪ E(q2) = {q0, q1, q2} = t2

δ"(t4, a) = E(q0) ∪ E(p) ∪ E(q3) = {q0, q1, p, q3, q4} = t4
δ"(t4, b) = E(q0) ∪ E(q2) ∪ E(q4) = {q0, q1, q2, q4} = t3


a
a
t1
b
b
t3
b
a
t2
t4
b
a
b
a