RICE

Early due date?
Friday or Saturday?

COMP 412
FALL 2013

Lexical Analysis — Part III
From NFA to DFA:
the Subset Construction

Comp 412

With a quick look at Brzozowski's Minimization Algorithm

Where are we? Why are we doing this?

RE → NFA *(Thompson's construction)* ✓

- Build an NFA for each term
- Combine them with ε-moves

NFA → DFA *(subset construction)*

- Build the simulation

DFA → Minimal DFA

- Hopcroft's algorithm

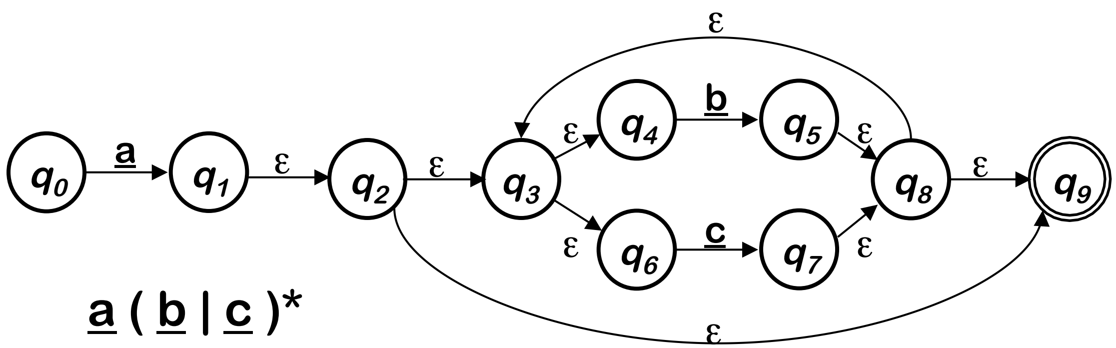
DFA → RE

- All pairs, all paths problem
- Union together paths from s_0 to a final state

The Cycle of Constructions

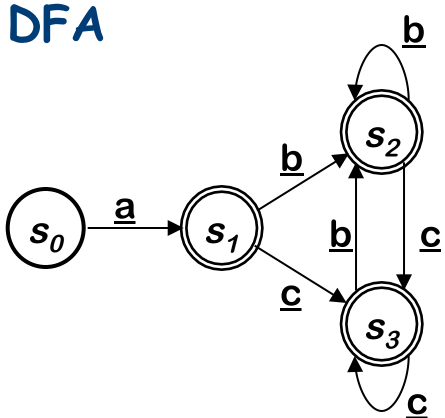
Simulating an NFA with a DFA

NFA



a (b | c)*

DFA



Where the mapping between NFA states and DFA states is:

DFA	NFA
s_0	q_0
s_1	$q_1, q_2, q_3, q_4, q_6, q_9$
s_2	$q_5, q_8, q_9, q_3, q_4, q_6$
s_3	$q_7, q_8, q_9, q_3, q_4, q_6$

NFA →DFA with Subset Construction

Need to build a simulation of the NFA

Two key functions

- *Move*(s_i , $\underline{a}$) is the set of states reachable from s_i by $\underline{a}$
- ε-*closure*(s_i) is the set of states reachable from s_i by ε

The algorithm:

- Start state derived from s_0 of the NFA
- Take its ε-closure $S_0 = \varepsilon\text{-closure}(\{s_0\})$
- Take the image of S_0, Move(S_0, α) for each $\alpha \in \Sigma$, and take its ε-closure
- Iterate until no more states are added

Sounds more complex than it is…

Rabin & Scott, 1959

Any DFA state containing a final state of the NFA becomes a final state of the DFA.

NFA →DFA with Subset Construction

The algorithm:

```
s0 ← ε-closure({n0})
S ← { s0 }
W ← { s0 }
while ( W ≠ Ø )
  select and remove s from W
  for each α ∈ Σ
    t ← ε-closure(Move(s,α))
    T[s,α] ← t
    if ( t ∉ S ) then
      add t to S
      add t to W
```

Let's think about why this works

s_0 is a set of states
S & W are sets of sets of states

The algorithm halts:

1. *S* contains no duplicates (test before adding)
2. $2^{\{\text{NFA states}\}}$ is finite
3. while loop adds to *S*, but does not remove from *S* *(monotone)*

⇒ the loop halts

S contains all the reachable NFA states

It tries each character in each s_i.

It builds every possible NFA configuration.

⇒ *S and T form the DFA*

This test is a little tricky

NFA →DFA with Subset Construction

Example of a *fixed-point* computation

- Monotone construction of some finite set
- Halts when it stops adding to the set
- Proofs of halting & correctness are similar
- These computations arise in many contexts

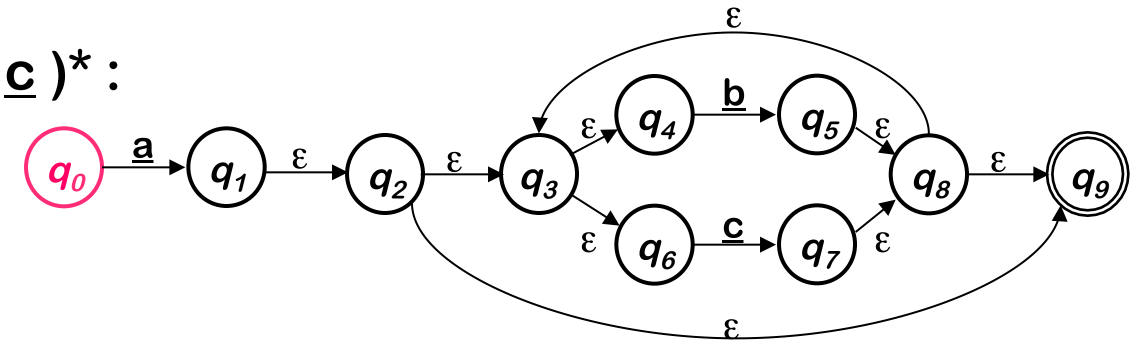
Other fixed-point computations

- Canonical construction of sets of LR(1) items
  - Quite similar to the subset construction
- Classic data-flow analysis & Gaussian Elimination
  - Solving sets of simultaneous set equations

We will see many more fixed-point computations

NFA →DFA with Subset Construction

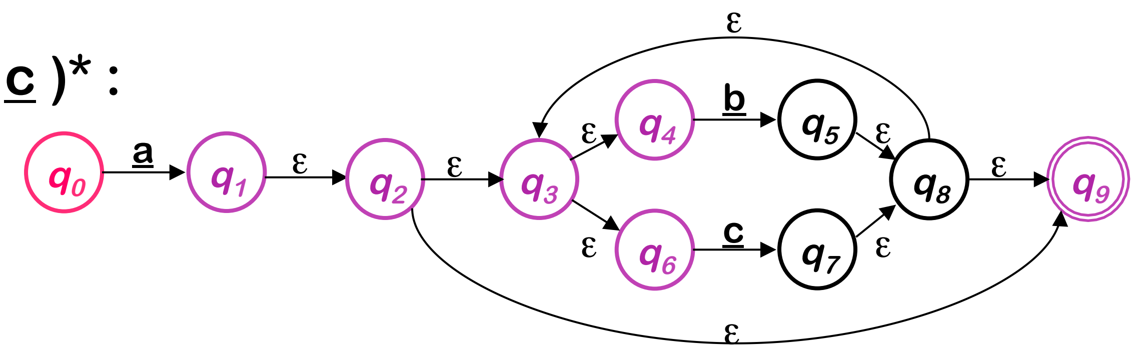
a (b | c)* :



States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0			

NFA →DFA with Subset Construction

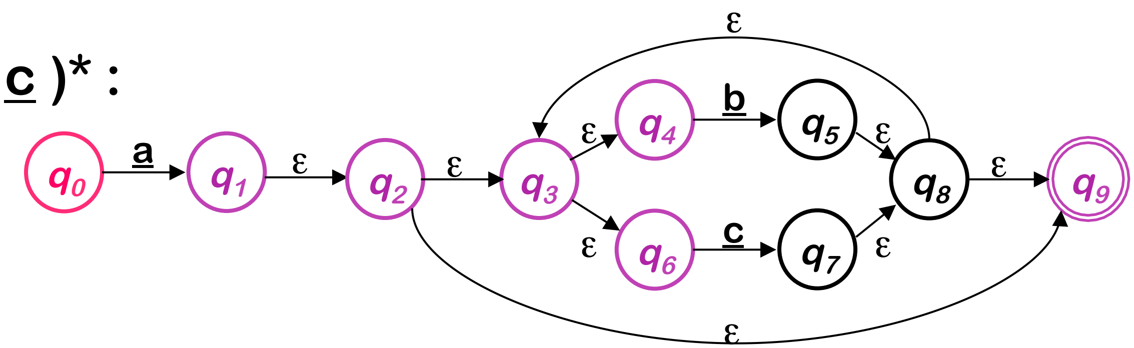
a (b | c)* :



States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0	$q_1, q_2, q_3, q_4, q_6, q_9$		

NFA →DFA with Subset Construction

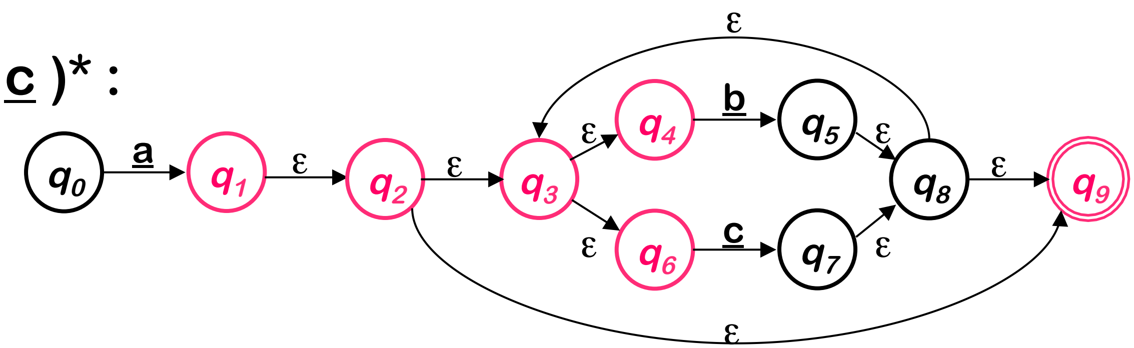
$\underline{a}\ (\ \underline{b}\ |\ \underline{c}\ )^*$:



States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	*none*

NFA →DFA with Subset Construction

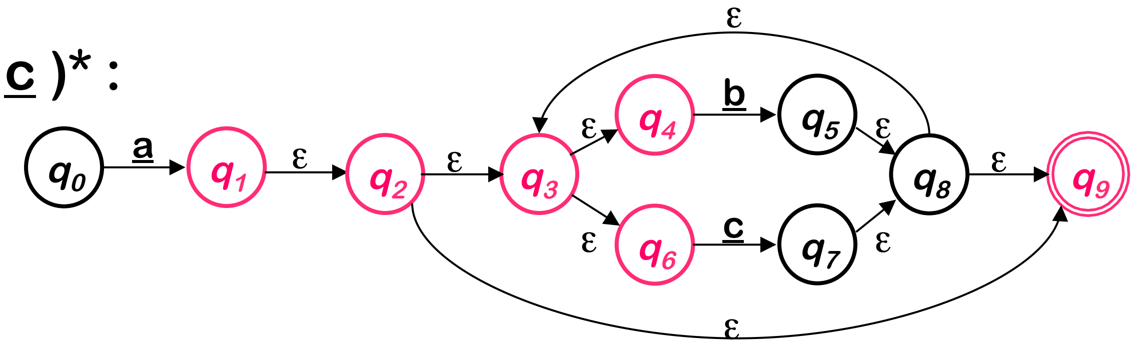
a (b | c)* :



States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	*none*
s_1	$q_1, q_2, q_3, q_4, q_6, q_9$			

NFA →DFA with Subset Construction

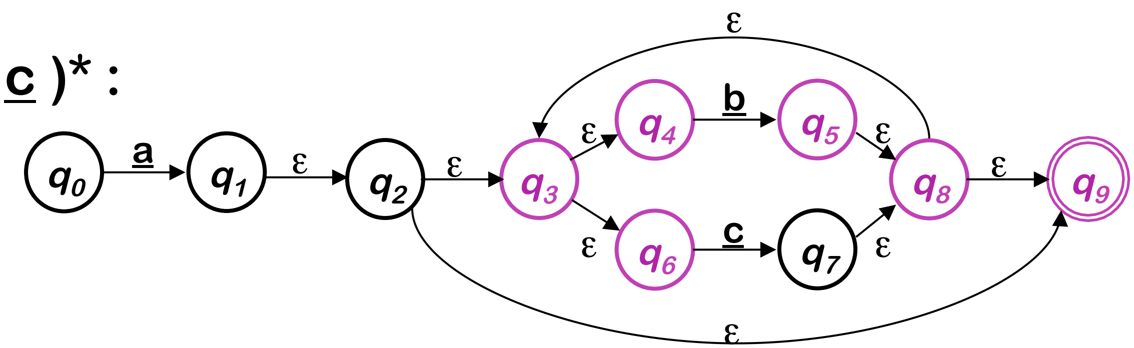
a (b | c)* :



States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	*none*
s_1	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*		

NFA →DFA with Subset Construction

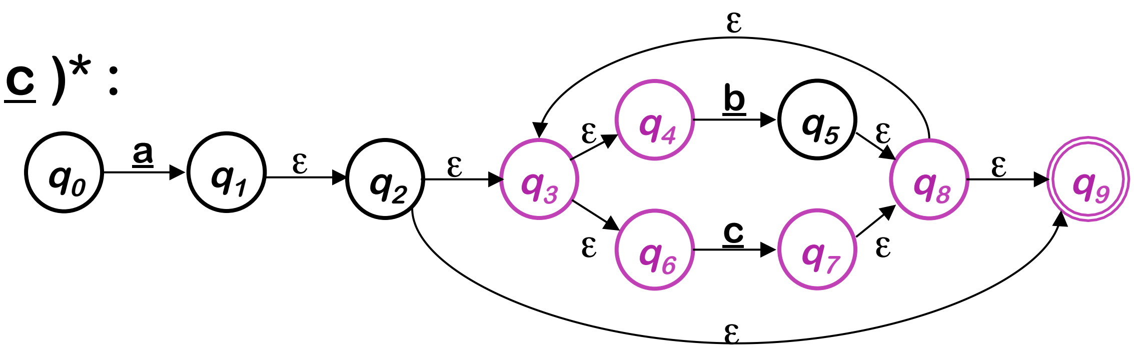
a (b | c)* :



States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0	$q_1, q_2, q_3, q_4, q_6, q_9$	none	none
s_1	$q_1, q_2, q_3, q_4, q_6, q_9$	none	$q_5, q_8, q_9, q_3, q_4, q_6$	

NFA →DFA with Subset Construction

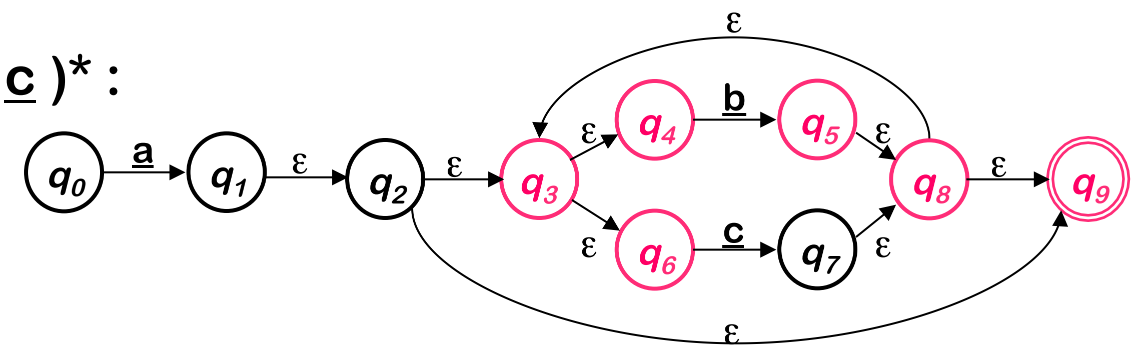
a (b | c)* :



States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	*none*
s_1	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	$q_5, q_8, q_9, q_3, q_4, q_6$	$q_7, q_8, q_9, q_3, q_4, q_6$

NFA →DFA with Subset Construction

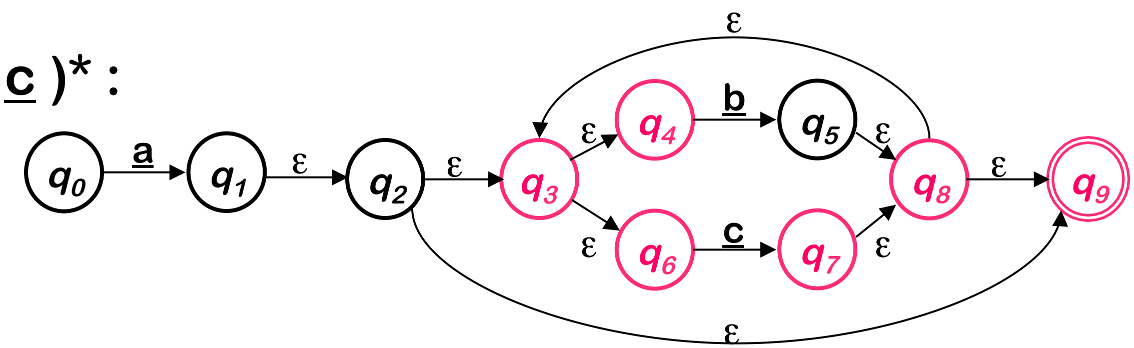
a (b | c)* :



States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	*none*
s_1	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	$q_5, q_8, q_9, q_3, q_4, q_6$	$q_7, q_8, q_9, q_3, q_4, q_6$
s_2	$q_5, q_8, q_9, q_3, q_4, q_6$			

NFA →DFA with Subset Construction

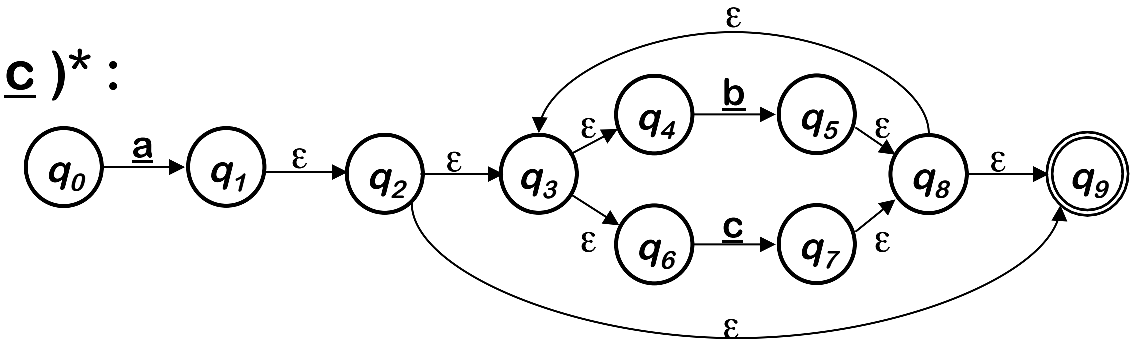
a (b | c)* :



States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	*none*
s_1	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	$q_5, q_8, q_9, q_3, q_4, q_6$	$q_7, q_8, q_9, q_3, q_4, q_6$
s_2	$q_5, q_8, q_9, q_3, q_4, q_6$			
s_3	$q_7, q_8, q_9, q_3, q_4, q_6$			

NFA →DFA with Subset Construction

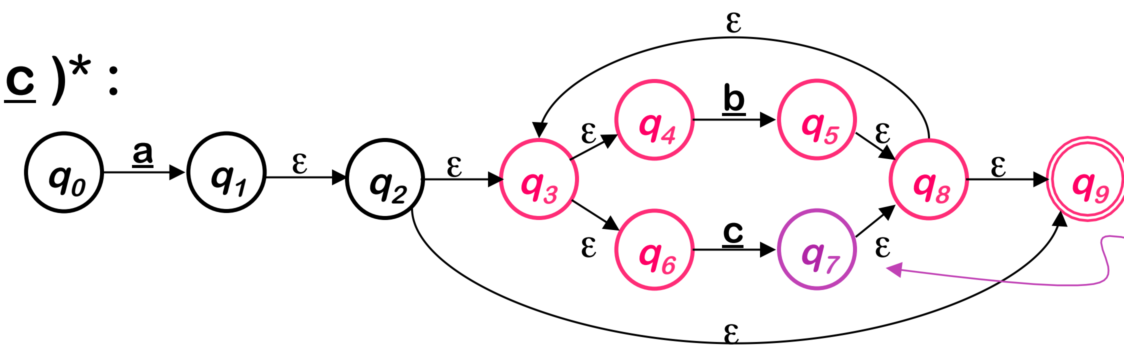
a (b | c)* :



States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	*none*
s_1	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	$q_5, q_8, q_9, q_3, q_4, q_6$	$q_7, q_8, q_9, q_3, q_4, q_6$
s_2	$q_5, q_8, q_9, q_3, q_4, q_6$	*none*		
s_3	$q_7, q_8, q_9, q_3, q_4, q_6$	*none*		

NFA →DFA with Subset Construction

a (b | c)* :

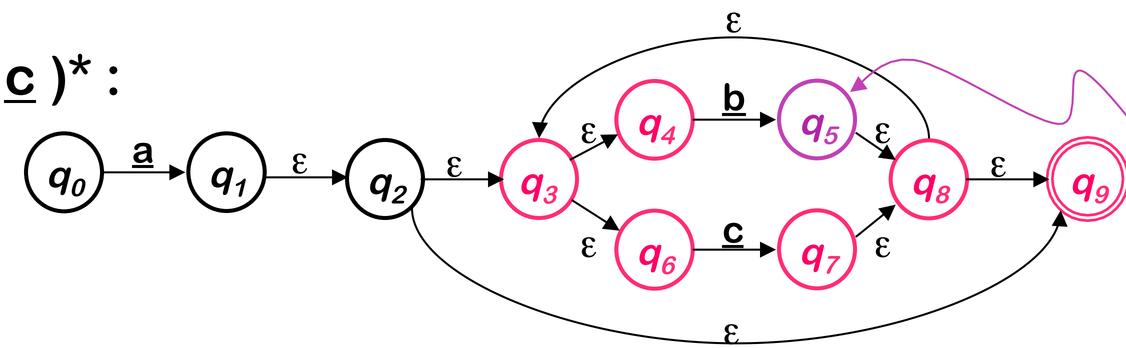


q_7 is the core state of s_3

States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0	$q_1, q_2, q_3, q_4, q_6, q_9$	none	none
s_1	$q_1, q_2, q_3, q_4, q_6, q_9$	none	$q_5, q_8, q_9, q_3, q_4, q_6$	$q_7, q_8, q_9, q_3, q_4, q_6$
s_2	$q_5, q_8, q_9, q_3, q_4, q_6$	none	s_2	s_3
s_3	$q_7, q_8, q_9, q_3, q_4, q_6$	none		

NFA →DFA with Subset Construction

a (b | c)* :

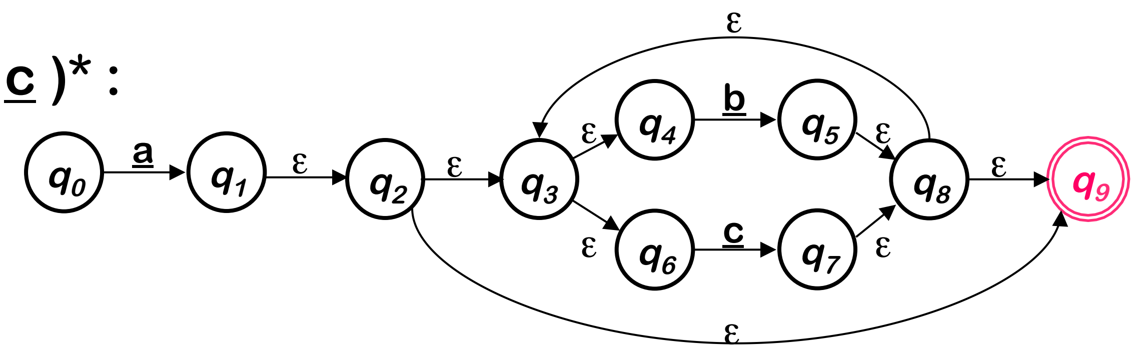


q_5 is the core state of s_2

States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	*none*
s_1	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	$q_5, q_8, q_9, q_3, q_4, q_6$	$q_7, q_8, q_9, q_3, q_4, q_6$
s_2	$q_5, q_8, q_9, q_3, q_4, q_6$	*none*	s_2	s_3
s_3	$q_7, q_8, q_9, q_3, q_4, q_6$	*none*	s_2	s_3

NFA →DFA with Subset Construction

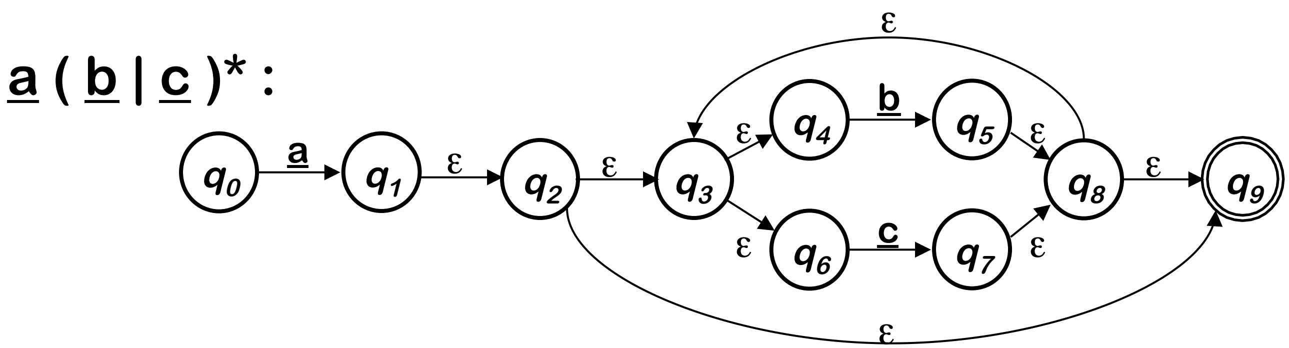
a (b | c)* :



States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	*none*
s_1	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	$q_5, q_8, q_9, q_3, q_4, q_6$	$q_7, q_8, q_9, q_3, q_4, q_6$
s_2	$q_5, q_8, q_9, q_3, q_4, q_6$	*none*	s_2	s_3
s_3	$q_7, q_8, q_9, q_3, q_4, q_6$	*none*	s_2	s_3

Final states because of q_9

NFA →DFA with Subset Construction

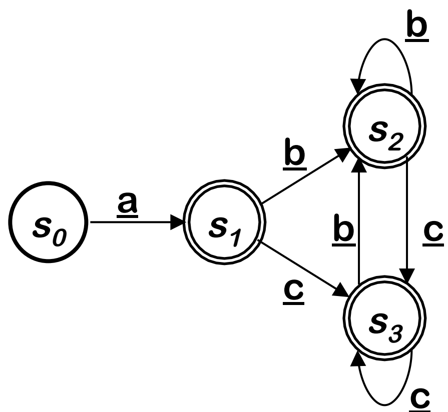


States		ε-closure(Move(s,*))		
DFA	NFA	a	b	c
s_0	q_0	s_1	*none*	*none*
s_1	$q_1, q_2, q_3, q_4, q_6, q_9$	*none*	s_2	s_3
s_2	$q_5, q_8, q_9, q_3, q_4, q_6$	*none*	s_2	s_3
s_3	$q_7, q_8, q_9, q_3, q_4, q_6$	*none*	s_2	s_3

Transition table for the DFA

NFA →DFA with Subset Construction

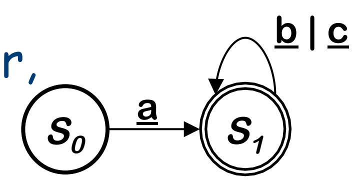
The DFA for a (b | c)*



	a	b	c
s_0	s_1	*none*	*none*
s_1	*none*	s_2	s_3
s_2	*none*	s_2	s_3
s_3	*none*	s_2	s_3

- Much smaller than the NFA (no ε-transitions)
- All transitions are deterministic
- Use same code skeleton as before

But, remember,
our goal was:



Where are we? Why are we doing this?

RE → NFA *(Thompson's construction)* ✓

- Build an NFA for each term
- Combine them with ε-moves

NFA → DFA *(subset construction)* ✓

- Build the simulation

DFA → Minimal DFA ⇐

- Hopcroft's algorithm

DFA → RE

- All pairs, all paths problem
- Union together paths from s_0 to a final state

The Cycle of Constructions

Not enough time to teach Hopcroft's algorithm today

Rabin and Scott, 1959 (page 8)

chines are more general than the ordinary ones, but this is not the case. We shall give a direct construction of an ordinary automaton, defining exactly the same set of tapes as a given nondeterministic machine.

Definition 11. *Let $\mathfrak{A}=(S,M,S_0,F)$ be a nondeterministic automaton. $\mathfrak{D}(\mathfrak{A})$ is the system (T,N,t_0,G) where T is the set of all subsets of S, N is a function on $T\times\Sigma$ such that $N(t,\sigma)$ is the union of the sets $M(s,\sigma)$ for s in t, $t_0=S_0$, and G is the set of all subsets of S containing at least one member of F.*

Clearly $\mathfrak{D}(\mathfrak{A})$ is an ordinary automaton, but it is actually equivalent to $\mathfrak{A}$.

Theorem 11. *If $\mathfrak{A}$ is a nondeterministic automaton, then $T(\mathfrak{A})=T(\mathfrak{D}(\mathfrak{A}))$.*

Proof: Assume first tha

You must love the "*clearly*"

$T(\mathfrak{A})$ and let $s_0,s_1,\ldots$ states satisfying the conditions of Definition 10. We show by induction that for $k\leq n$, s_k is in $N(t_0,{}_0x_k)$. For $k=0$, $N(t_0,{}_0x_k)=N(t_0,\Lambda)=t_0=S_0$ and we were given that s_0 is in S_0. Assume the result for $k-1$. By definition, $N(t_0,{}_0x_k)=N(N(t_0,{}_0x_{k-1}),\sigma_{k-1})$. But we have assumed s_{k-1} is in $N(t_0,{}_0x_{k-1})$ so that from the definition of N we have $M(s_{k-1},\sigma_{k-1})\subset N(t_0,{}_0x_k)$. However, s_k is in $M(s_{k-1},\sigma_{k-1})$, and so the result is established. In particular s_n is in $N(t_0,{}_0x_n)=N(t_0,x)$, and since s_n is in F, we have $N(t_0,x)$ in G, which proves that x is in $T(\mathfrak{D}(\mathfrak{A}))$. Hence, we have shown that

$$T(\mathfrak{A})\subset T(\mathfrak{D}(\mathfrak{A})).$$

Assume next that a tape $x=\sigma_0\sigma_1\ldots\sigma_{n-1}$ is in $T(\mathfrak{D}(\mathfrak{A}))$. Let for each $k\leq n$, $t_k=N(t_0,{}_0x_k)$. We shall work backwards. First, we know that t_n is in G. Let then s_n be any internal state of $\mathfrak{A}$ such that s_n is in t_n and s_n is in F. Since s_n is in

$$t_n=N(t_0,{}_0x_n)=N(t_{n-1},\sigma_{n-1}),$$

we have from the definition of N that s_n is in $M(s_{n-1},\sigma_{n-1})$ for some s_{n-1} in t_{n-1}. But

Definition 12. *Let $\mathfrak{A}=(S,M,S_0,F)$ be a nondeterministic automaton. The dual of $\mathfrak{A}$ is the machine $\mathfrak{A}^*=(S,M^*,F,S_0)$ where the function M^* is defined by the condition*

s' is in $M^*(s,\sigma)$ if and only if s is in $M(s',\sigma)$.

Notice that we have at once the equation $\mathfrak{A}^{**}=\mathfrak{A}$. The relation between the sets defined by an automaton and its dual is as follows.

Theorem 12. *If $\mathfrak{A}$ is a nondeterministic automaton, then $T(\mathfrak{A}^*)=T(\mathfrak{A})^*$.*

Proof: In view of the equality $\mathfrak{A}^{**}=\mathfrak{A}$, we need only show $T(\mathfrak{A}^*)\subset T(\mathfrak{A})^*$. Let $x=\sigma_0\sigma_1\ldots\sigma_{n-1}$ be a tape show that x^* is in $T(\mathfrak{A})$. Let $s_0,s_1,$ ence of internal states of $\mathfrak{A}^*$ such s in S_0 and s_k is in $M^*(s_{k-1},\sigma_{k-1})$ for $k=1,2,\ldots,n$. Define a new sequence $s'_0,s'_1,\ldots,s'_n$ by the equation $s'_k=s_{n-k}$ for $k\leq n$. Obviously, s'_0 is in S_0 and s'_n is in F. Further, for $k>0$ and $k\leq n$, $s'_{k-1}=s_{n-k+1}$ is in $M^*(s_{n-k},\sigma_{n-k})$, or in other words, $s_{n-k}=s'_k$ is in $M(s'_{k-1},\sigma_{n-k})$. Now defining a new sequence of symbols $\sigma'_0\sigma'_1\ldots\sigma'_{n-1}$ by the formula $\sigma'_k=\sigma_{n-k-1}$, we see that $\sigma'_{k-1}=\sigma_{n-k}$ and $\sigma'_0\sigma'_1\ldots\sigma'_{n-1}=x^*$. Thus, x^* is in $T(\mathfrak{A})$ as was to be proved.

It should be noted that Theorem 12 together with Theorem 11 yields a direct construction and proof for Theorem 4 of Section 3 which was first proved by the indirect method of Theorem 1. In the next section we make heavy use of the direct constructions supplied by the nondeterministic machines to obtain results not easily apparent from the mathematical characterizations of Theorems 1 and 2.

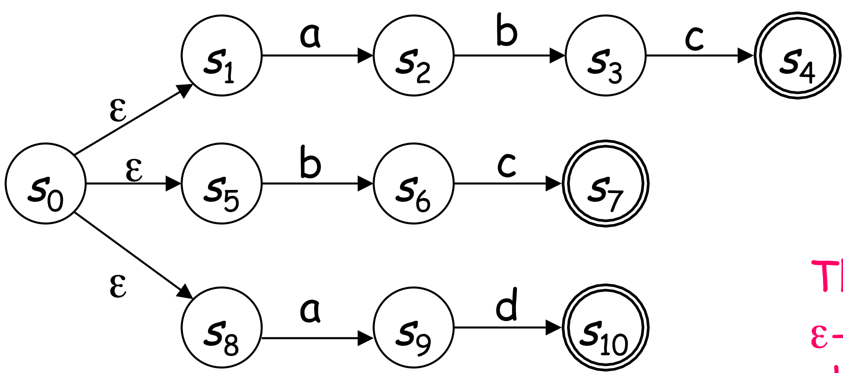
6. *Further closure properties*

Simplifying a result due originally to Kleene, Myhill in unpublished work has shown that the class $\mathcal{T}$ can be characterized as the least class of sets of tapes containing the finite sets and closed under some simple operations on sets of tapes. We indicate here a different proof using

Alternative Approach to DFA Minimization

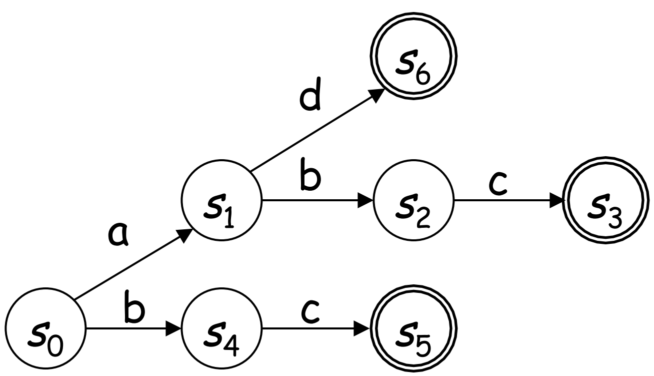
The Intuition

- The subset construction merges prefixes in the NFA



abc | bc | ad

Thompson's construction would leave ε-transitions between each single-character automaton



Subset construction eliminates ε-transitions and merges the paths for <u>a</u>. It leaves duplicate tails, such as <u>bc</u>.

Alternative Approach to DFA Minimization

Idea: use the subset construction twice

- For an NFA *N*
  - Let *reverse(N)* be the NFA constructed by making initial states final (& vice-versa) and reversing the edges
  - Let *subset(N)* be the DFA that results from applying the subset construction to *N*
  - Let *reachable(N)* be *N* after removing all states that are not reachable from the initial state
- Then,

reachable(subset(reverse[reachable(subset(reverse(N))]))

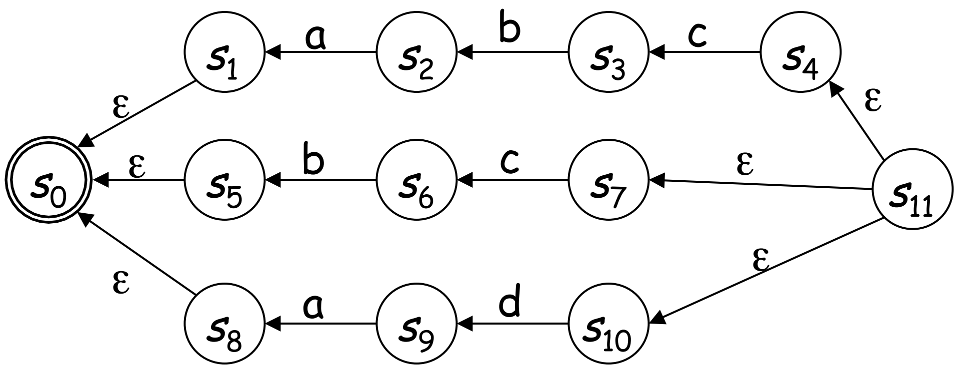
is the minimal DFA that implements *N*   [Brzozowski, 1962]

This result is not intuitive, but it is true.
Neither algorithm dominates the other.

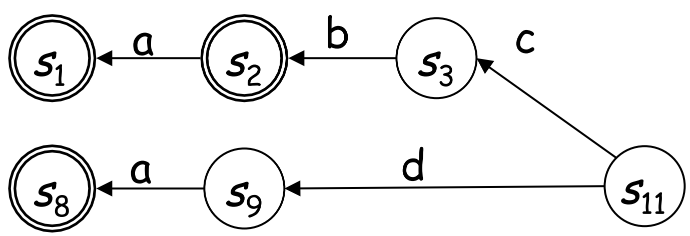
Alternative Approach to DFA Minimization

Step 1

- The subset construction on *reverse(NFA)* merges suffixes in original NFA



Reversed NFA

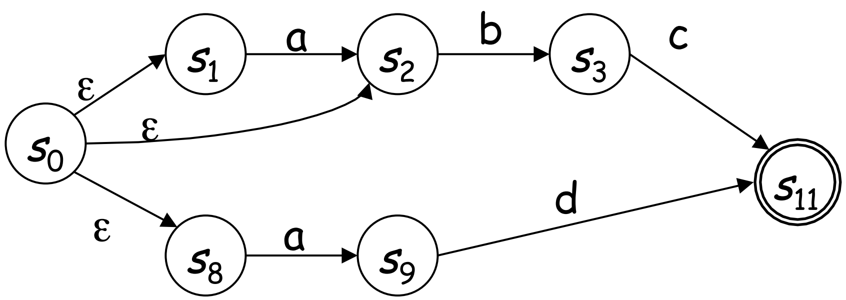


subset(reverse(NFA))

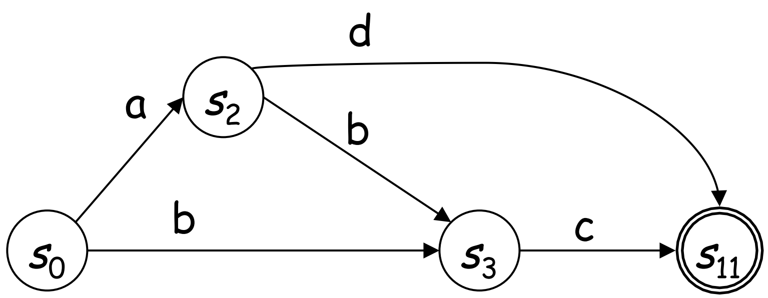
Alternative Approach to DFA Minimization

Step 2

- Reverse it again & use subset to merge prefixes …



Reverse it, again



Minimal DFA

And subset it, again

The Cycle of Constructions