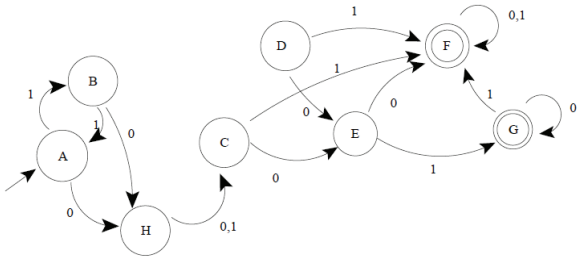
CMSC 330: Organization of Programming Languages

DFA minimization

Complex DFA

- This DFA is more complex than it needs to be:



Source: http://www.cs.uiuc.edu/class/fa07/cs273/Handouts/minimization/minimization.pdf

DFA minimization algorithm

The DFA minimization algorithm marks which states are distinct. States not marked as distinct can then be merged, to create a simpler DFA.

Suppose M = (Q, $\sum$, δ, $q0$,F) is a DFA. We define distinctness inductively.

Specifically, two states p and q in M are distinct if
a) $p \in F$ and $q \notin F$, or vice versa, or
b) for some a $\in \sum$, δ(p, a) and δ(q, a) are distinct

DFA minimization algorithm

The algorithm for marking distinct states follows this inductive definition.

Create a table DISTINCT with an entry for each pair of states. Table cells are initially blank.

(1) For every pair of states (p,q)
If p is final and q is not, or vice versa,
Set DISTINCT(p,q) to be x

(2) Loop until there is no change in the table contents
For each pair of states (p,q) and each character a in the alphabet
if DISTINCT(p,q) is empty and DISTINCT($\delta(p,a)$, $\delta(q,a)$) is not empty
Set DISTINCT(p,q) to be x.

(3) Two states p and q are distinct iff DISTINCT(p,q) is not empty.

DFA minimization algorithm

After step(1):

b							
c							
d							
e							
f	x	x	x	x	x		
g	x	x	x	x	x		
h						x	x
	a	b	c	d	e	f	g

DFA minimization algorithm

After one iteration of step(2):

b							
c	x	x					
d	x	x					
e	x	x	x	x			
f	x	x	x	x	x		
g	x	x	x	x	x		
h			x	x	x	x	x
	a	b	c	d	e	f	g

DFA minimization algorithm

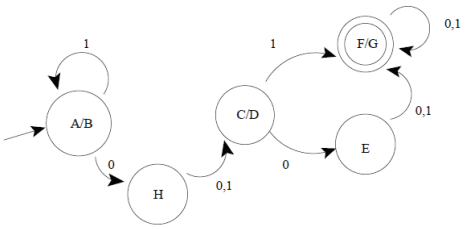
After the second iteration of step(2):

b							
c	*x*	*x*					
d	*x*	*x*					
e	*x*	*x*	*x*	*x*			
f	*x*	*x*	*x*	*x*	*x*		
g	*x*	*x*	*x*	*x*	*x*		
h	*x*	*x*	*x*	*x*	*x*	*x*	*x*
	a	b	c	d	e	f	g

Minimized DFA

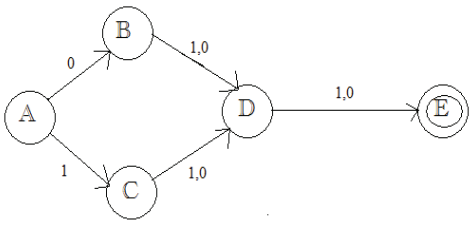
Third iteration of step (2) makes no changes to the table, so we halt.

The cells *(a,b), (c,d)* and *(f,g)* are still empty, so these pairs of states are not distinct. Merging them produces the following simpler DFA recognizing the same language.



More Examples - 1

- Lets now minimize the following DFA



More Examples -1

After step(1) and one iteration of step(2):

B				
C				
D	×	×	×	
E	×	×	×	×
	A	**B**	**C**	**D**

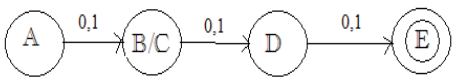
More Examples -1

After another iteration of step(2):

B	×			
C	×			
D	×	×	×	
E	×	×	×	×
	A	**B**	**C**	**D**

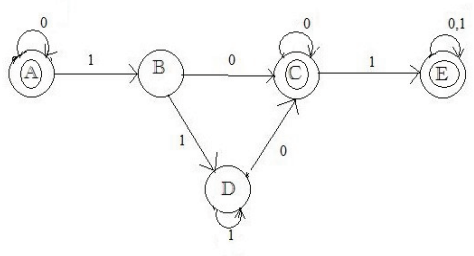
More Examples -1

- Minimized DFA:



More Examples - 2

- Lets now minimize the following DFA



More Examples -2

After step(1):

B	x			
C		x		
D	x		x	
E		x		x
	A	**B**	**C**	**D**

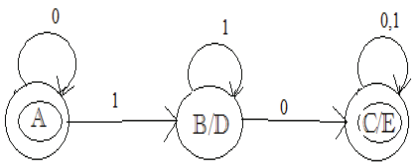
More Examples -2

After one iteration of step(2):

B	x			
C	x	x		
D	x		x	
E	x	x		x
	A	**B**	**C**	**D**

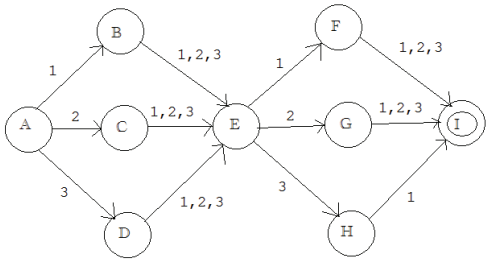
More Examples -2

- So the final minimized DFA is:



More Examples - 3

- Lets now minimize the following DFA



More Examples -3

After step(1):

B								
C								
D								
E								
F								
G								
H								
I	x	x	x	x	x	x	x	x
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**

More Examples -3

After one iteration of step(2):

B								
C								
D								
E								
F	x	x	x	x	x			
G	x	x	x	x	x			
H	x	x	x	x	x	x	x	
I	x	x	x	x	x	x	x	x
	A	B	C	D	E	F	G	H

More Examples -3

After three iterations of step(2):

B	x							
C	x							
D	x							
E	x	x	x	x				
F	x	x	x	x	x			
G	x	x	x	x	x			
H	x	x	x	x	x	x	x	
I	x	x	x	x	x	x	x	x
	A	B	C	D	E	F	G	H

More Examples -3

- So the final minimized DFA is: