NFA-to-DFA conversion

Slides from

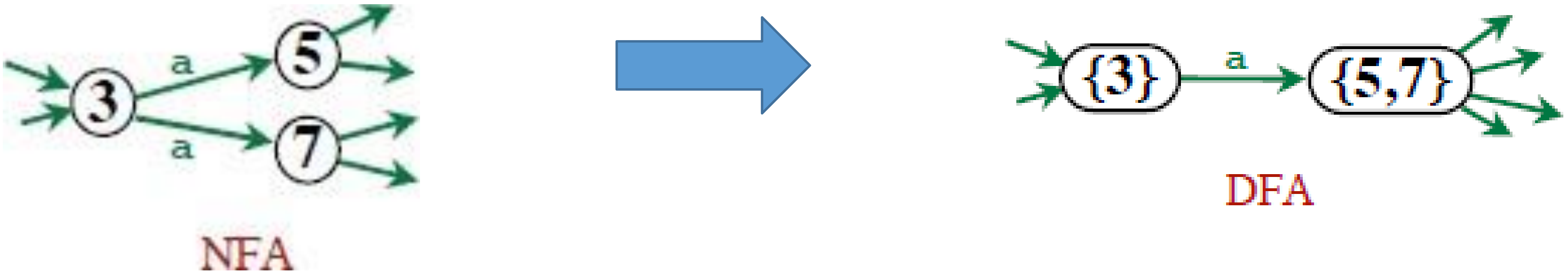
http://web.cecs.pdx.edu/~harry/compilers/slides/LexicalPart3.pdf

Two ways to deal with an NFA

- Convert the NFA to an equivalent DFA first
- Use the NFA directly

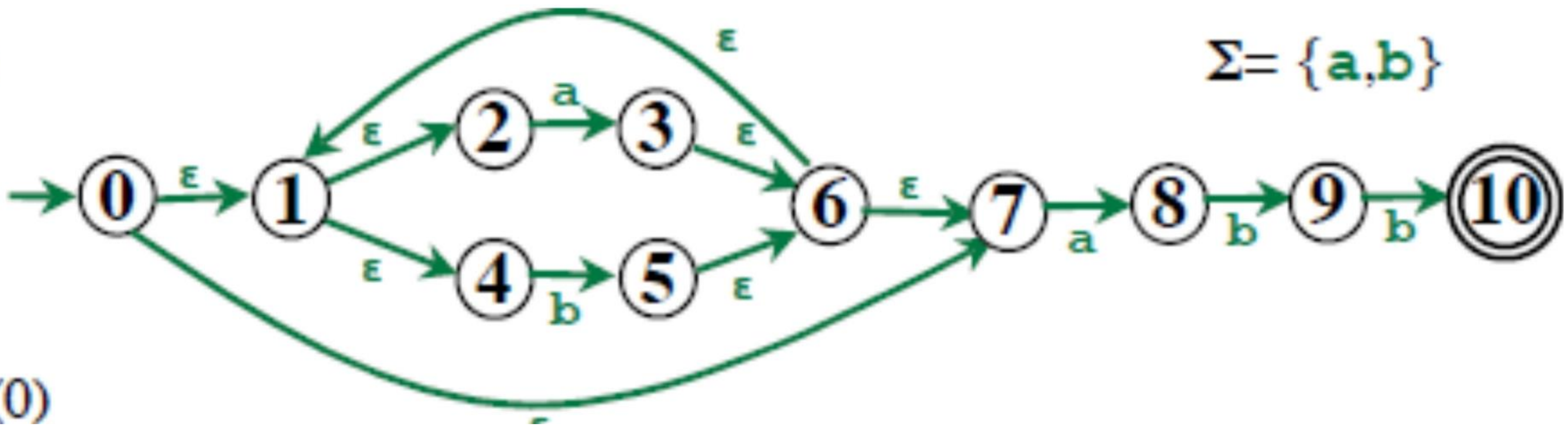
Converting an NFA to a DFA

- Input: an NFA
- Output: a DFA, which is equivalent
- Idea: Each state in the DFA corresponds to a set of NFA states



Example

$\Sigma = \{a,b\}$



tart state

ε-Clost e (0)

Move$_{DF.}$ .a

$= \varepsilon$-Closure (Move$_{NFA}$(A,a))

$= \varepsilon$-Closure ({3,8})

$= \{1,2,3,4,6,7,8\}$

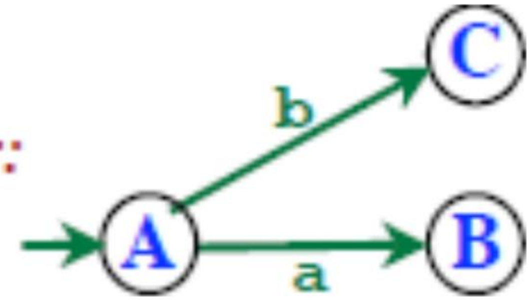
Move$_{DFA}$(A,b)

$= \varepsilon$-Closure (Move$_{NFA}$(A,b))

$= \varepsilon$- Closure ({5})

$\{1,2,4,5,6,7\} =$

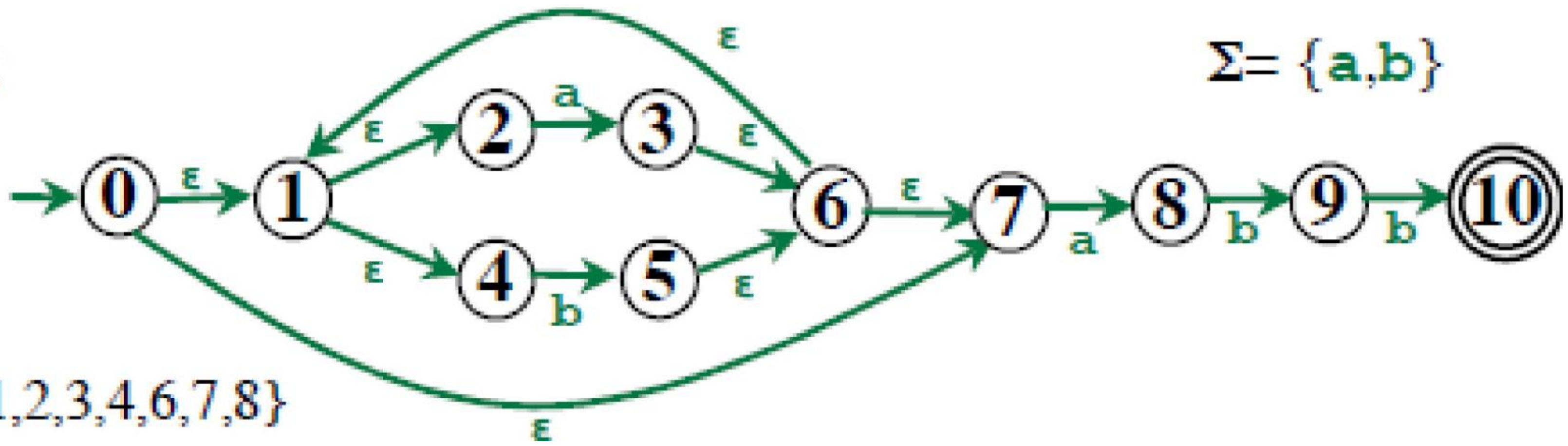
So far:



A is now done; mark it!
B and C are unmarked.
Let's do B next...

Example

$\Sigma = \{a,b\}$



Process B = {1,2,3,4,6,7,8}

$\text{Move}_{\text{DFA}}(B,a)$

= ε-Closure (Move_{NF} (B,a))

= ε-Closure ({3,8})

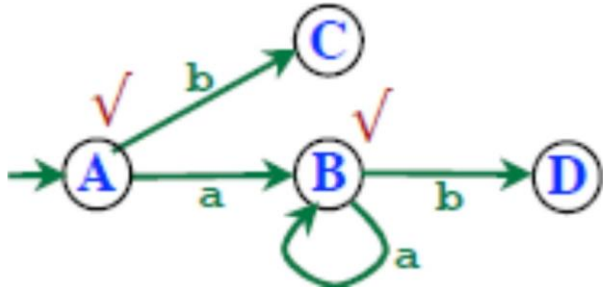
1,2,3,4,6, = B

$\text{Move}_{\text{DFA}}(B,b)$

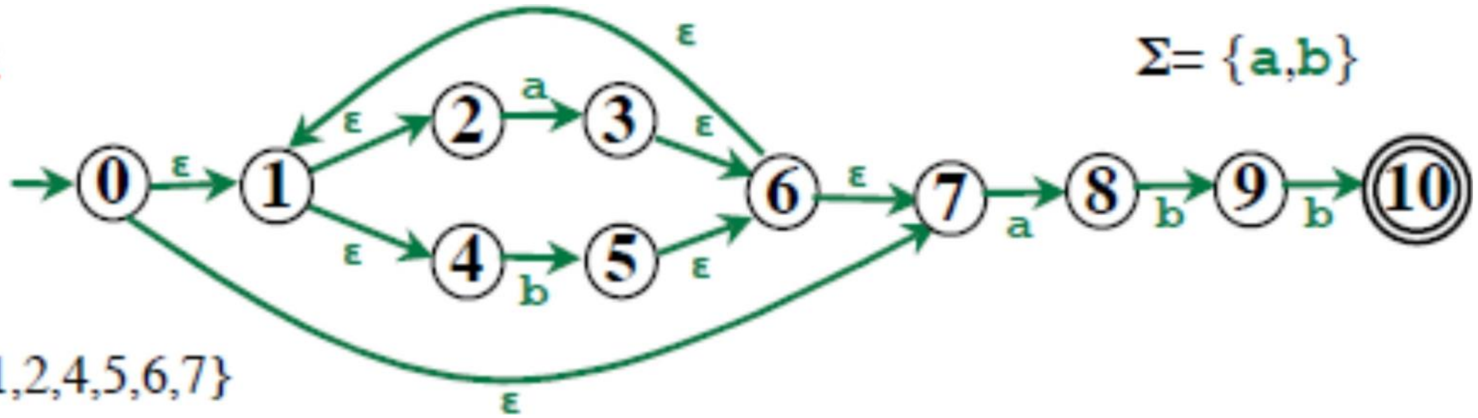
= ε - Closure (Mo e_ (B,b))

= ε - Closure ({5.9}

= {1,2,4,5,6,7,9} = D



Example



$\Sigma = \{a,b\}$

Process C = {1,2,4,5,6,7}

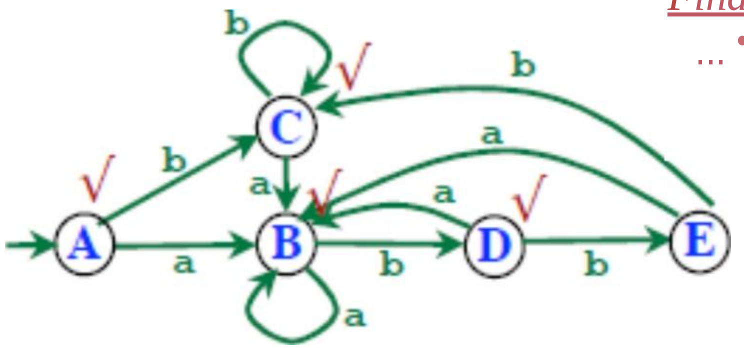
$Move_{DFA}(C,a) = \{1,2,3,4,6,7,8\}$
$Move_{FA}(C,b) = \{1,2,4,5,6,7\} =$

Process E ,2,4,5,6,7,10}

$Move_{DFA}(E,a) = \{1,2,3,4,6,7,8\} =$
$Move_{DFA}(E,b)$ 1,2,4,5,6,7} =

Process D {1,2,4,5,6,7,9}

$Move_{DFA}(D,a)$ {1,2,3,4,6,7,8}
$Move_{DFA}(D,b)$ {1,2,4,5,6,7,10} =



Final . tares ;11DF .
... *•hicl1 statcf s co11rain 10?*

Algorithm: Convert NFA to DFA

5nFA = {}
Add £-Clos e (s) to SOFA as th start state
Se the nlys ate in FA o n:marke n
hile FA c n 1.ns unmar state .....Q.
Let T bet at -u .-rke d sta ar

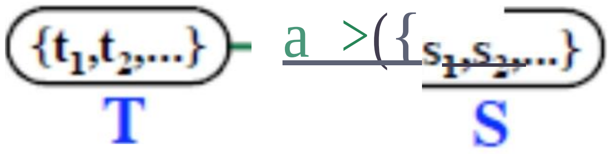
A set of.· F stat

for ea a .l.. o
S = E-C os re(e (T , a))
if S is not 'n SDFA alrea y then

Everywhere ou could possibly t to011 an

Add S to 5nFA (as an nmar e " sate)
end.If
Set Move$_{DFA}$(T,a) to S

i. dda11 edgcio t/J . DF.. ...



endFor
endWhile
for each S 'n S_0
_:_fanys ES is a fi al sta e i e en
Mark S an a finals ate i e DFA
e dlf
end.For

Use NFA directly

- Your code will keep track of "current search states". Once you reach the end of the input symbol sequence, check one of the current search states is a final state.

- Note that you do not need to enumerate all possible paths explicitly. You just need to keep track of multiple current search states.

- The method is explained in Section 2.2.5 in Jurasfsky & Martin (2^{nd} edition).