CS480
Translators
FSAs (NDFAs to DFAs)
Chap. 3
```c
int main()
{
    int $;
    10.;
    - 10;
    #
}
```

OSU Oregon State University
Example Output from M2...

```
atom:    atom
left:    [
atom:    with
atom:    escape
right:   ]
op:      !!!
left:    [
op:      +
float:   3.0
atom:    b
right:   ]
end:
flip2   ~/cs480/private/m2 21%
```
Quiz #3

• **Tokenize** the following C statement:
  
  ```c
  float limitedSquare(x) float x; {
    /*returns x-squared, but never more than 100*/
    return (x<=-10.0||x>=10.0)?100:x*x;
  }
  ```

• Given Σ = {a,b}, provide regular expressions for languages below:
  
  – all strings beginning and ending in a
  
  – all strings of a’s and b’s of even length
  
  – all strings with an odd number of a’s
  
  – string of zero or more a’s followed by same number of b’s
NFA vs. DFA

• **Nondeterministic Finite Automata (NFA):**
  – $\varepsilon$ can label edges (these edges are called $\varepsilon$-transitions)
  – some character can label 2 or more edges out of the same state

• **Deterministic Finite Automata (DFA):**
  – no edges are labeled with $\varepsilon$
  – each character can label at most one edge out of the same state

• **NFA and DFA** accepts string $x$ if there exists a path from the start state to a final state labeled with characters in $x$
(a | b)*abb

Figure 3.24: A nondeterministic finite automaton

Figure 3.28: DFA accepting (a|b)*abb
### Transition Tables

- For NFA, each entry is a set of states:

<table>
<thead>
<tr>
<th>STATE</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>{0,1}</td>
<td>{0}</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>{2}</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>{3}</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- For DFA, each entry is a unique state:

<table>
<thead>
<tr>
<th>STATE</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
## NFA to DFA

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon$-closure($s$)</td>
<td>Set of NFA states reachable from NFA state $s$ on $\epsilon$-transitions alone.</td>
</tr>
<tr>
<td>$\epsilon$-closure($T$)</td>
<td>Set of NFA states reachable from some NFA state $s$ in set $T$ on $\epsilon$-transitions alone; $= \cup_{s \in T} \epsilon$-closure($s$).</td>
</tr>
<tr>
<td>$move(T, a)$</td>
<td>Set of NFA states to which there is a transition on input symbol $a$ from some state $s$ in $T$.</td>
</tr>
</tbody>
</table>

```c
while (there is an unmarked state $T$ in $Dstates$) {
    mark $T$;
    for (each input symbol $a$) {
        $U = \epsilon$-closure($move(T, a)$);
        if ($U$ is not in $Dstates$)
            add $U$ as an unmarked state to $Dstates$;
        $Dtran[T, a] = U$;
    }
}
```
Figure 3.34: NFA N for \((a|b)^*abb\)

<table>
<thead>
<tr>
<th>NFA STATE</th>
<th>DFA STATE</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>{0,1,2,4,7}</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>{1,2,3,4,6,7,8}</td>
<td>B</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>{1,2,4,5,6,7}</td>
<td>C</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>{1,2,4,5,6,7,9}</td>
<td>D</td>
<td>B</td>
<td>E</td>
</tr>
<tr>
<td>{1,2,4,5,6,7,10}</td>
<td>E</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>
DFA - \((a | b)^*abb\)

Figure 3.36: Result of applying the subset construction to Fig. 3.34
Crazy Semantics

https://www.destroyallsoftware.com/talks/wat

....Thank you Kevin Strasser