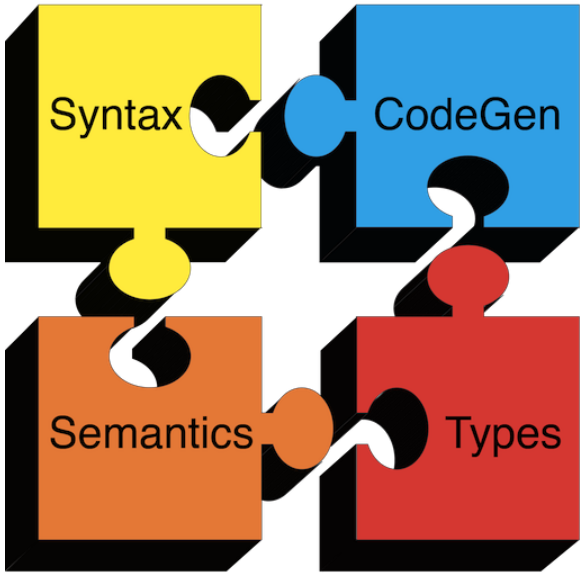
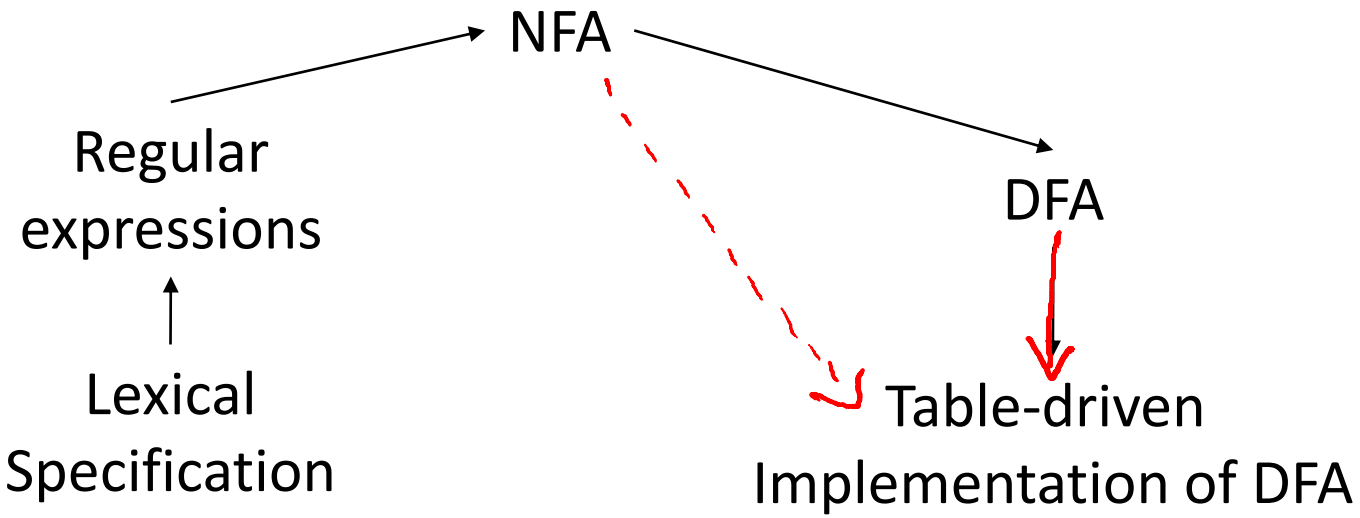


Compilers

Implementing Finite Automata

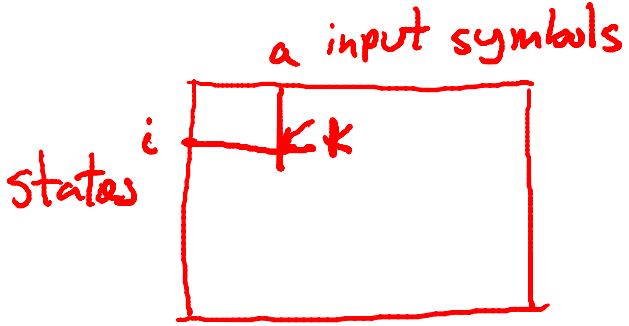
Alex Aiken

Implementing FA

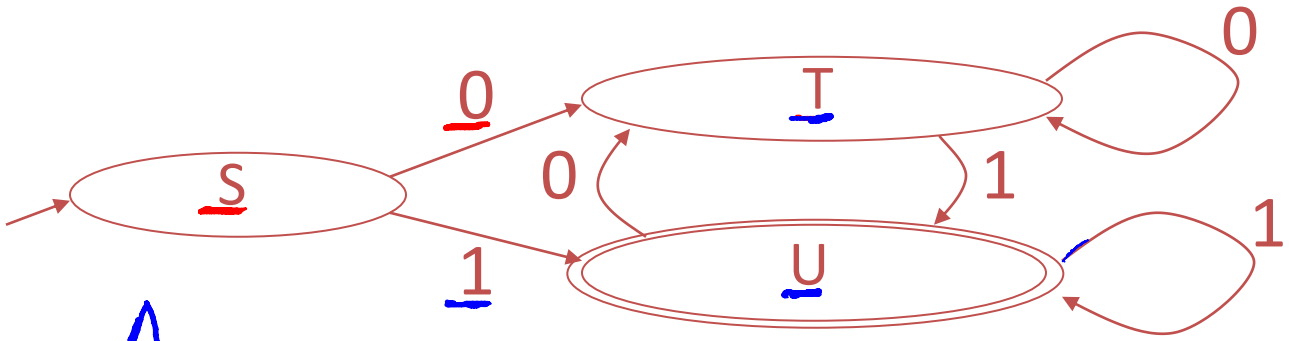


Implementing FA

- A DFA can be implemented by a 2D table T
  - One dimension is states
  - Other dimension is input symbol
  - For every transition $S_i \rightarrow^a S_k$ define $T[i,a] = k$



Implementing FA

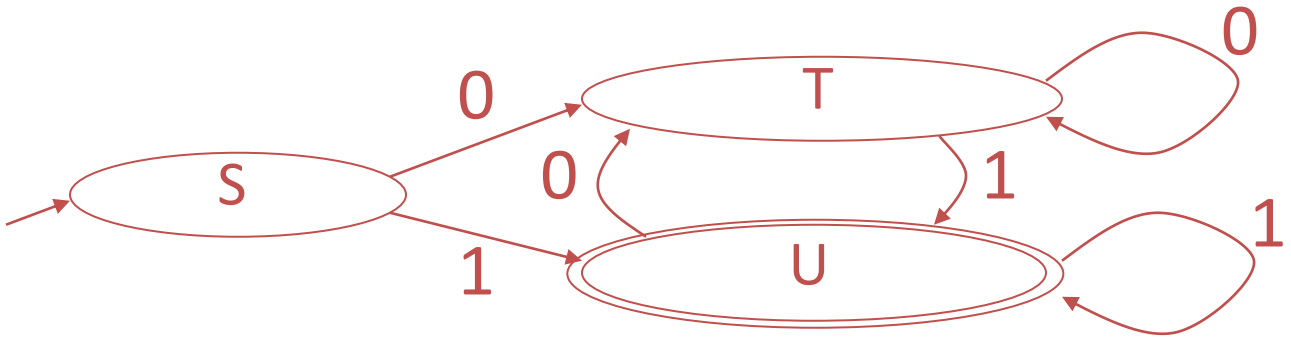


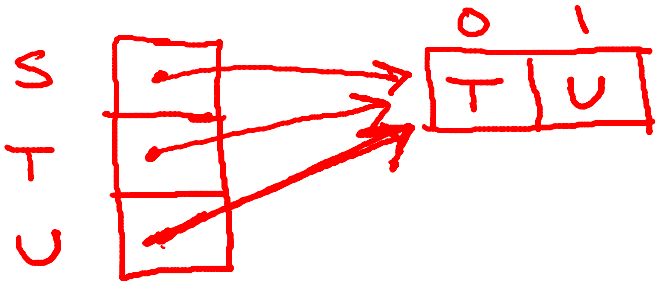
A

	0	1
→ S	T	U
T	T	U
U	T	U

```
i = 0;
state = 0;
while (input[i]) {
  state = A[state, input[i++]];
}
```

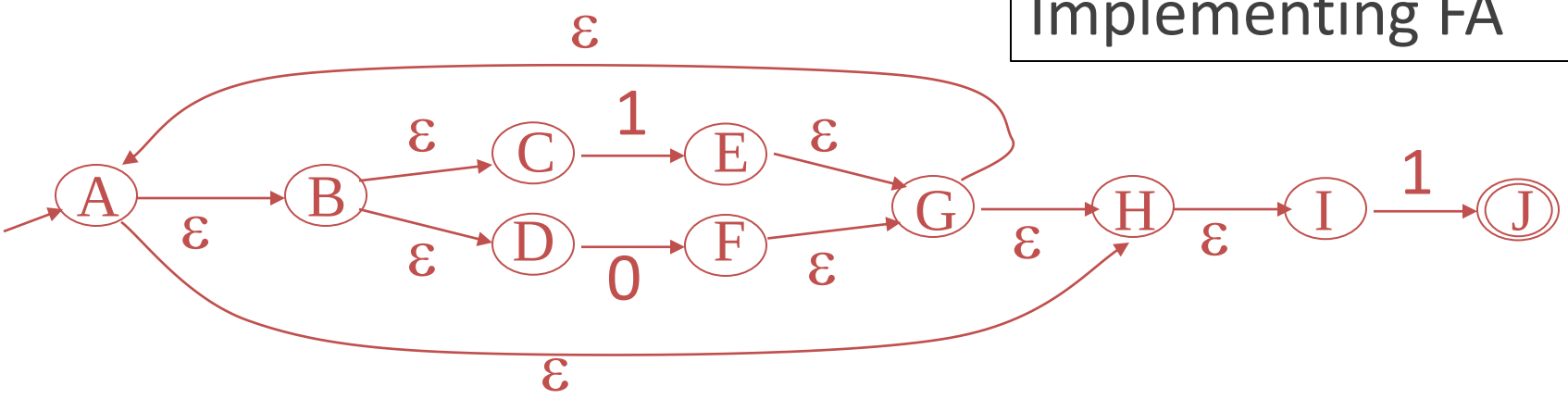
Implementing FA





$2^N - 1$ DFA for an NFA w/ n states

Implementing FA



	0	1	ε
A			{B}
B			{C, D}
C		{E}	
D	{F}		
⋮			

sets of states

Implementing FA

- NFA -> DFA conversion is key

- Tools trade between speed and space

DFAs faster, less compact

NFAs slower, concise