Section 2:

NFA2DFA, Efficient Lexing, CFGs

CS 164 @ UC Berkeley, Spring 2024

http://tinyurl.com/164section1

Reminders

WA 1 is released, and is due on Feb 7 at 11:59PM PST.

(No slip days for written assignments)

PA1 is released, and is due on Feb 19 at 5PM PST.

Reminder to take care of yourselves, and to prioritize your health! Feel free to submit extension requests if you need them!

Converting NFAs to DFAs

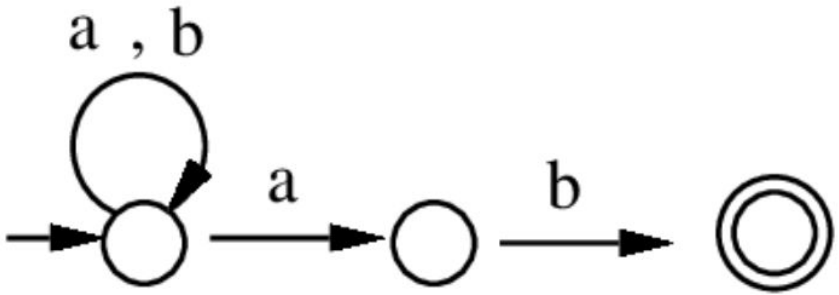
Basic intuition:

1. Combine the NFA's start states, and those reachable via an empty string to a single start state for the DFA.
2. At each step:
    a. For each possible transition:
        i. Combine the states reachable via the transition (including empty strings) into a single state, and add a transition
3. Any state with a goal state as one of its components is a goal state.

Converting NFAs to DFAs: Formal requirements

- Each state *S* of the DFA is a subset of the NFA's states *Q*. ($\forall S: S \subseteq Q$)
- The DFA's start state is the set of NFA states reachable via ε-moves from the NFA's start states
- The DFA has a transition *a* from S to S' iff S' is the set of states reachable from any state in S via *a* and ε-moves

Example



CFG's

Consist of replacement rules:

$S \rightarrow SS$,

$S \rightarrow (S)$,

$S \rightarrow ()$

CFG derivations and parse trees

S → SS,

S → (S),

S → ()

Example: (())()

Ambiguous CFG's can produce multiple parse trees

S → SS,

S → (S),

S → ()

S → S+S

Example: ()+()+()

Anonymous feedback form:
http://tinyurl.com/SoraDisFeedback