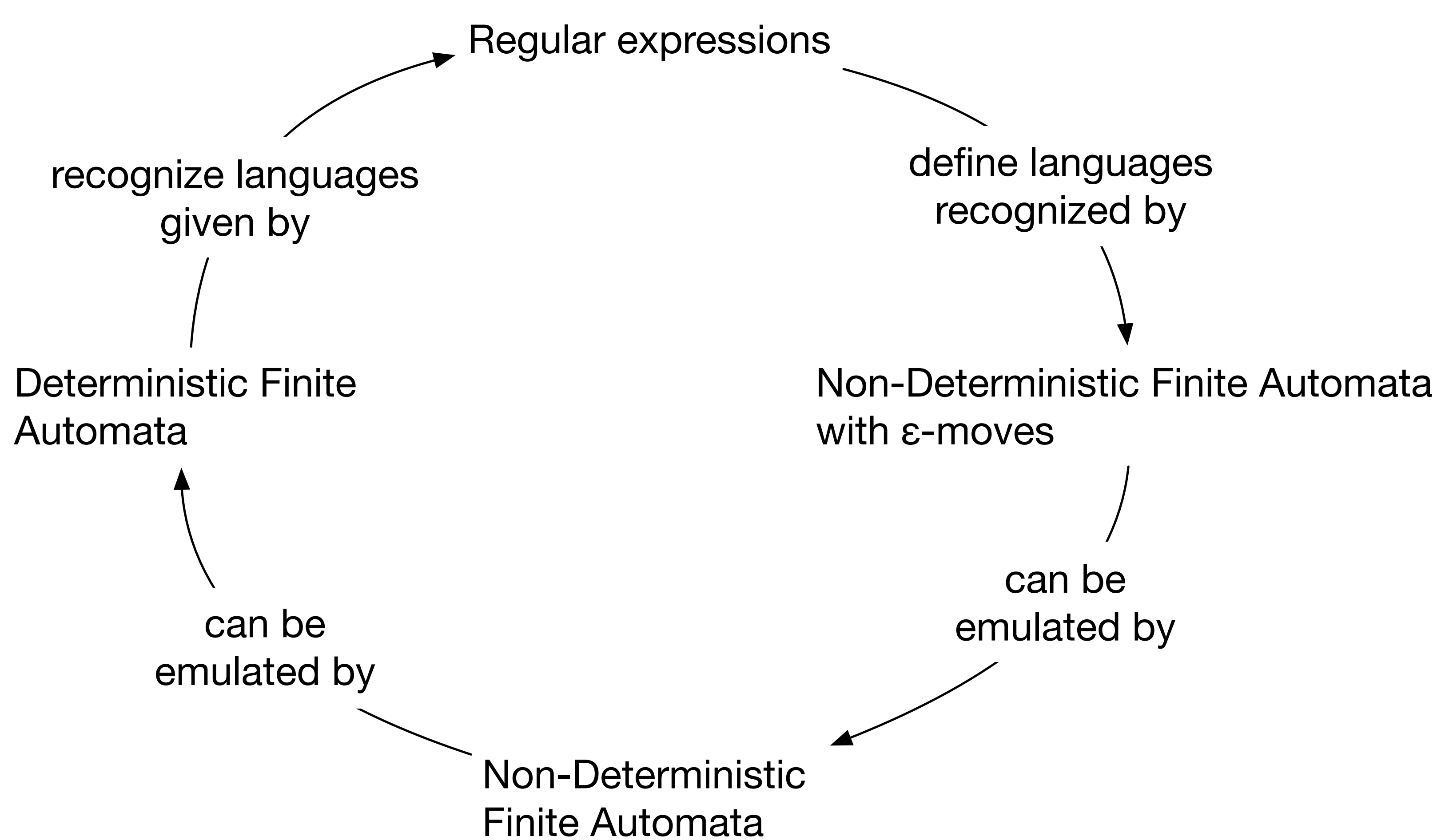
Regular Expressions and Deterministic Finite Automata

Thomas Schwarz, SJ

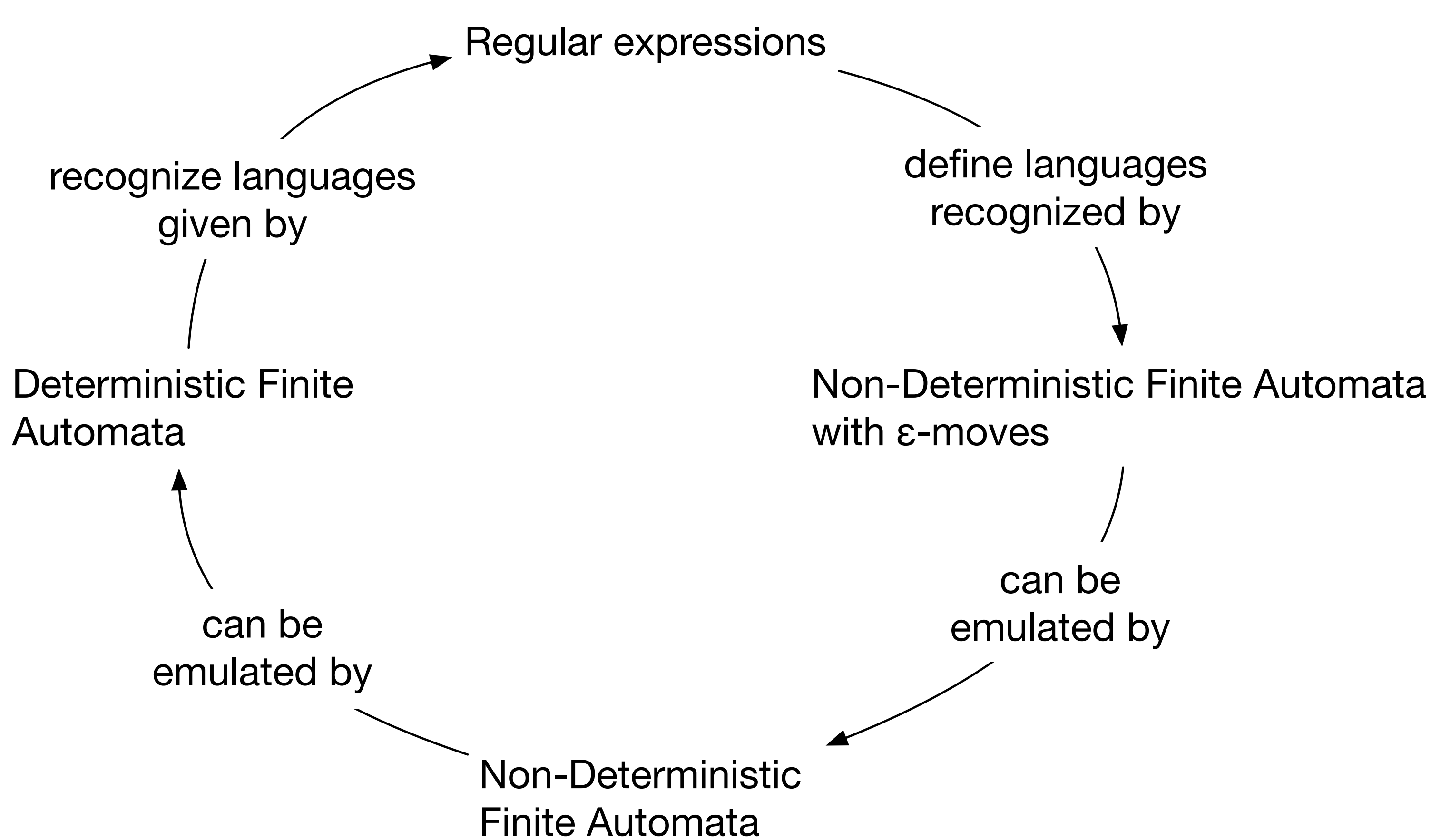
Regular Expressions and Deterministic Finite Automata

- We want to show that regular expressions are exactly those recognized by a finite automaton.
    - The proof follows a simple scheme



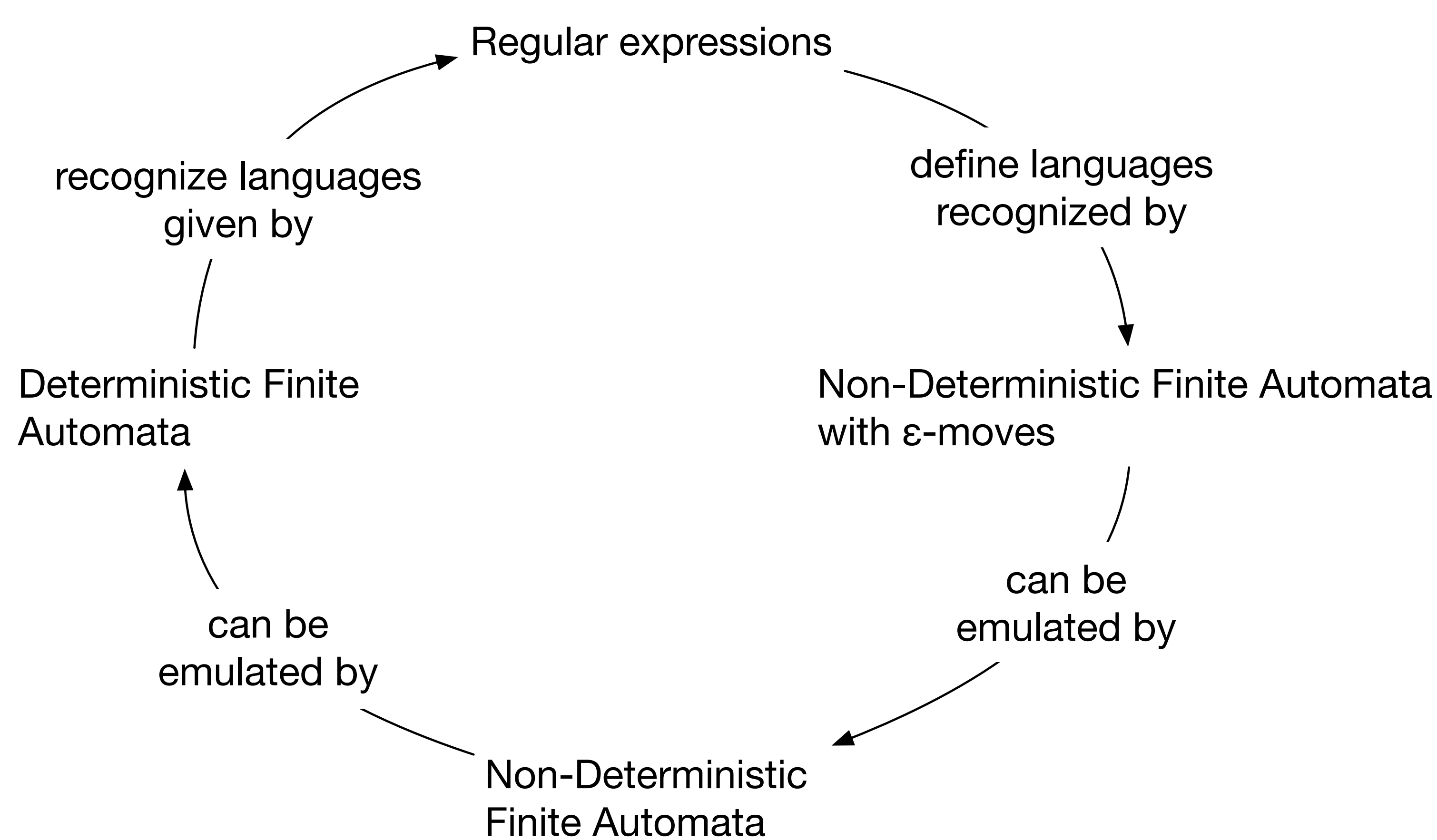
Regular Expressions and Deterministic Finite Automata

- We already have shown that:
  - NFAs with ϵ-moves can be emulated by NFAs
  - NFAs can be emulated by DFAs



Regular Expressions and Deterministic Finite Automata

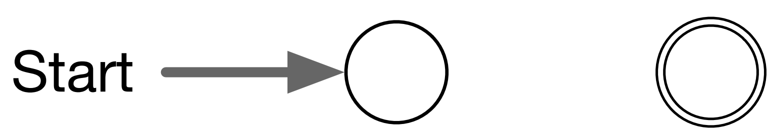
- Left to do:
  - Regular expressions define languages recognized by NFA with ϵ moves
  - DFA recognize languages given by regular expressions

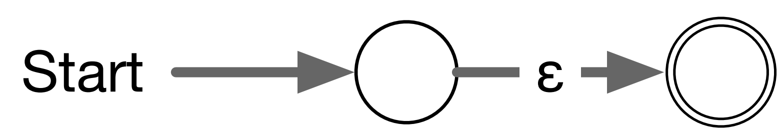


Regular expressions and NFAs with ϵ-moves

- Regular expressions are defined recursively
  - We need to give a construction for each step

- Base:





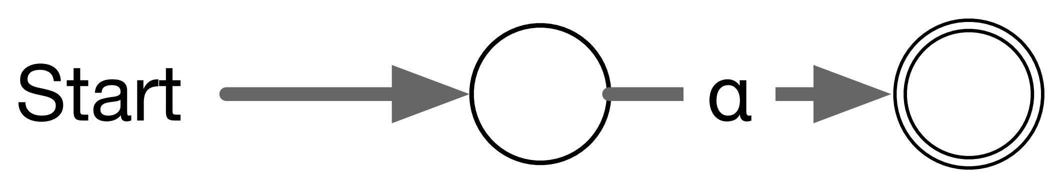
$\varnothing = \varnothing$

$\epsilon = \{\epsilon\}$

Regular expressions and NFAs with ϵ-moves

- Regular expressions are defined recursively
  - We need to give a construction for each step

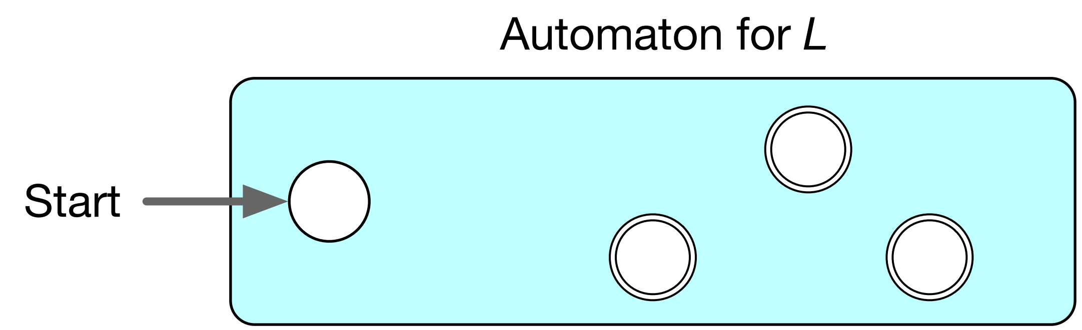
- Base: For a letter $\alpha \in \Sigma$

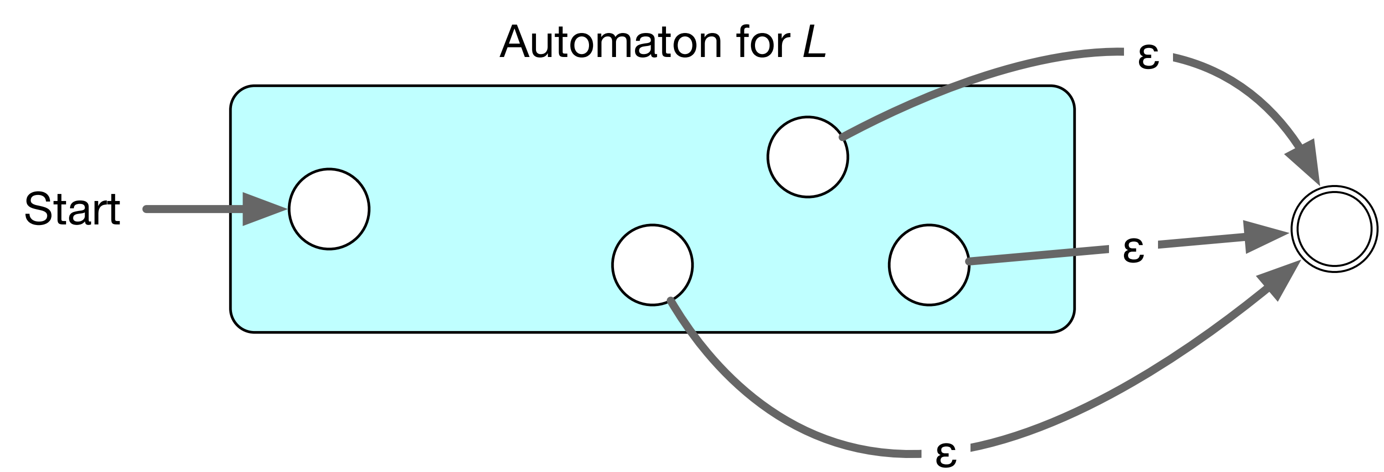


$$\alpha = \{\alpha\}$$

Regular expressions and NFAs with ϵ-moves

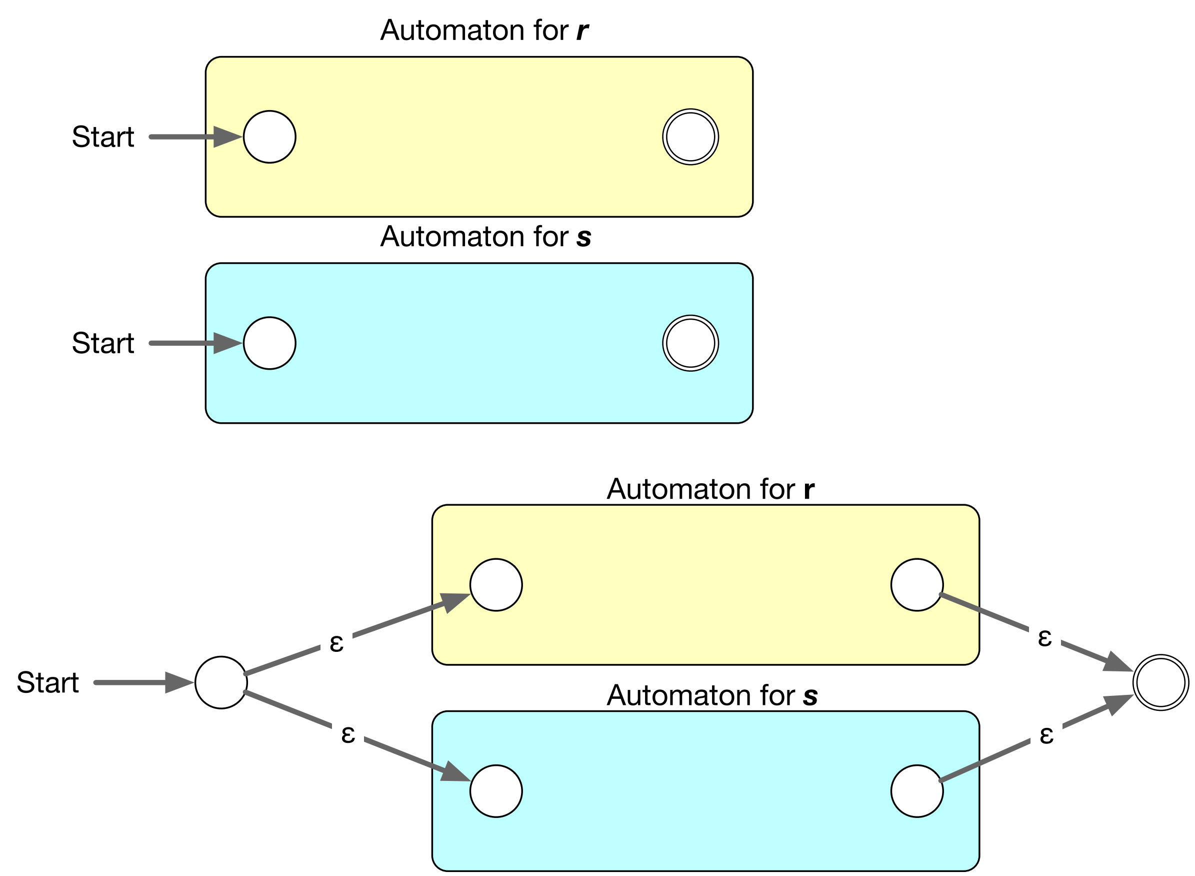
- We can always assume that an automaton has a singular final state
  - By replacing the automaton by one where:
    - The final states are no longer final
    - There is a new final state
    - There are ϵ - transition from the former final states to the new, single final state





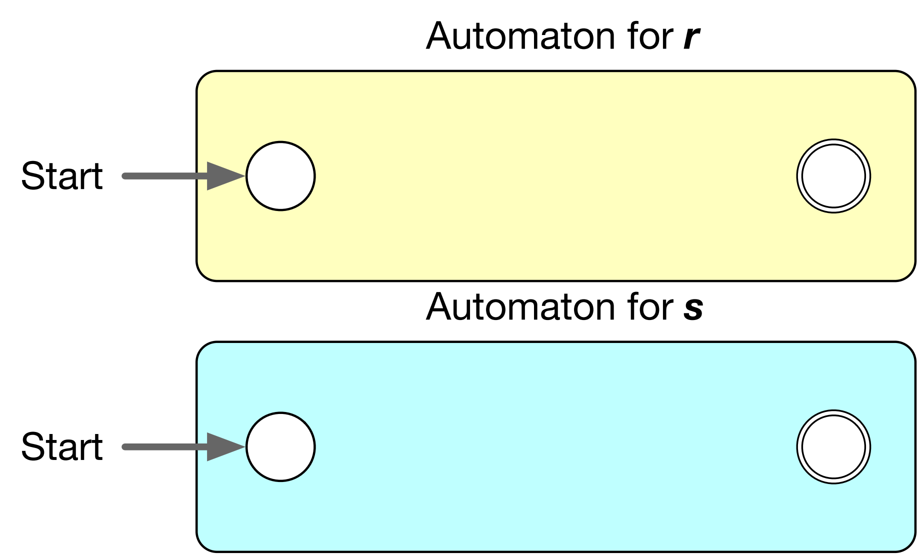
Regular expressions and NFAs with ϵ-moves

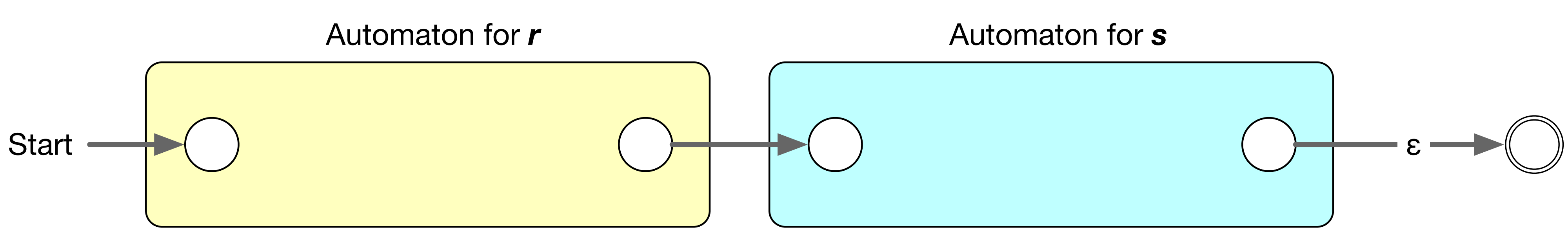
- Union $\mathbf{r} + \mathbf{s}$: Get two machines that recognize r and s
  - Connect a new start state to the start states of the two machines with an ϵ transition
  - Connect all final states with a new, single final state with an ϵ transition



Regular expressions and NFAs with ϵ-moves

- Concatenation $\mathbf{r} \cdot \mathbf{s}$
  - Connect the final state of the automaton that recognizes $\mathbf{r}$ with the start state of the automaton that recognizes $\mathbf{s}$
  - Add an ϵ transition to a new final state (not really necessary)

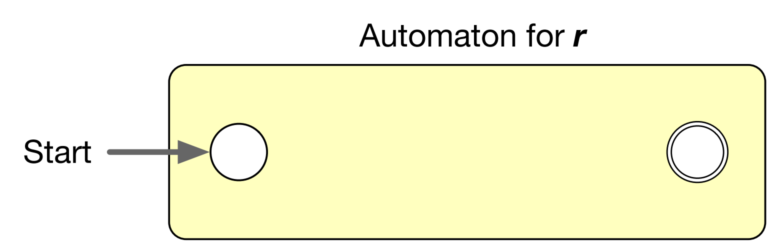


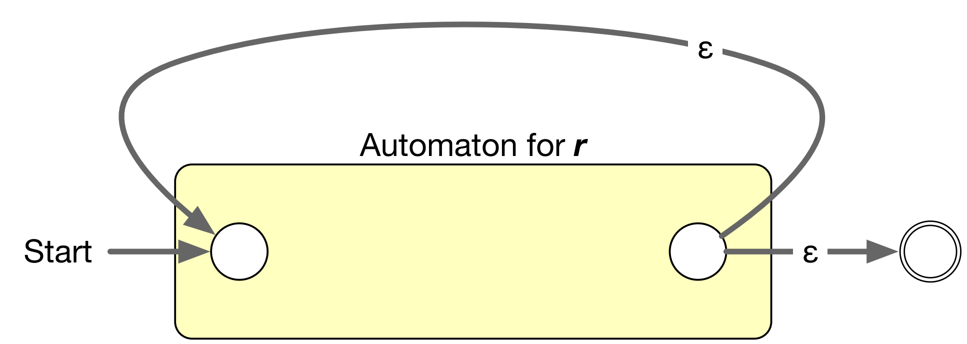


Automata for ***rs***

Regular expressions and NFAs with ϵ-moves

- Not strictly necessary: $\mathbf{r}^+$
  - Add an ϵ-transition from the accepting state of $\mathbf{r}$ to the start state
  - We can now transit the automaton several times, but at least one

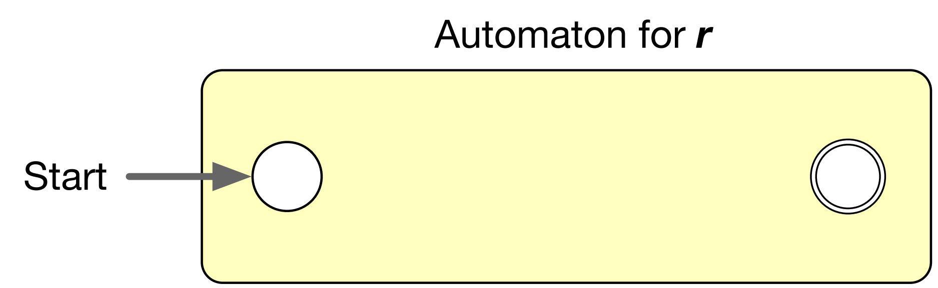


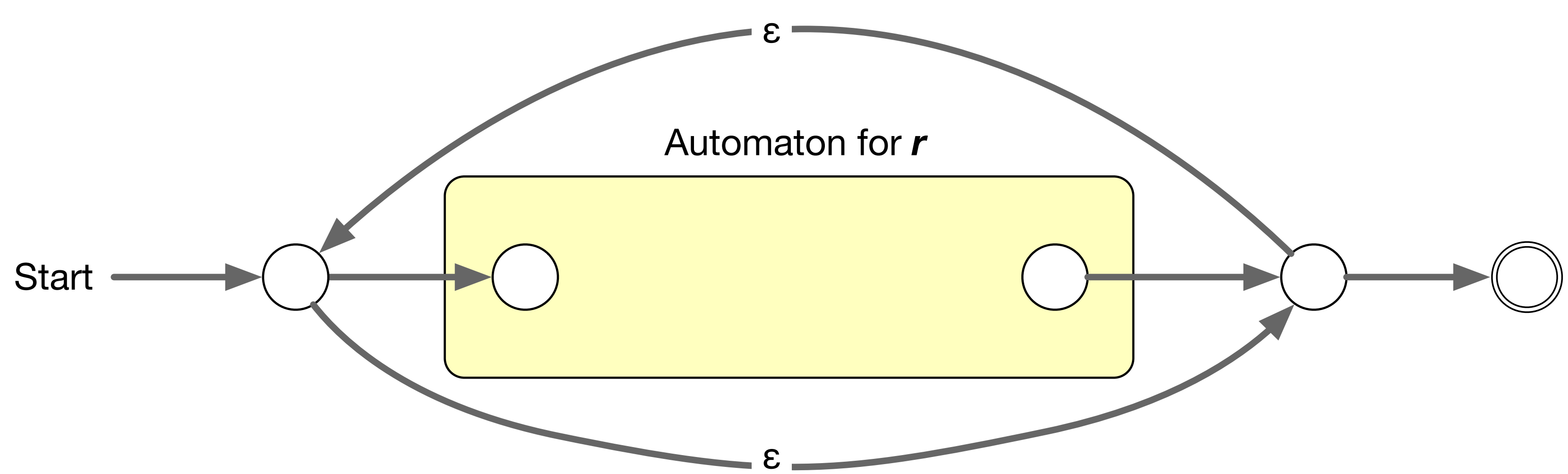


Automata for r^+

Regular expressions and NFAs with ϵ-moves

- $\mathbf{r^*}$: Use additional states and ϵ transitions that allow you to bypass the automaton.





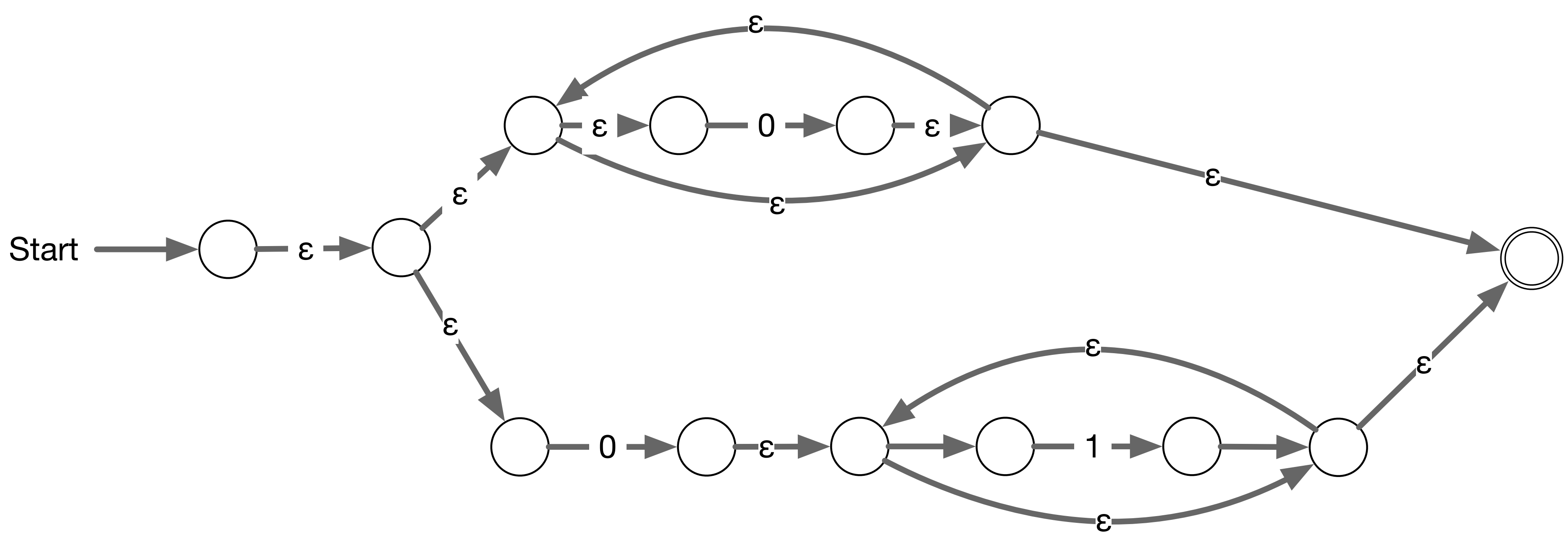
Automaton for ***r****

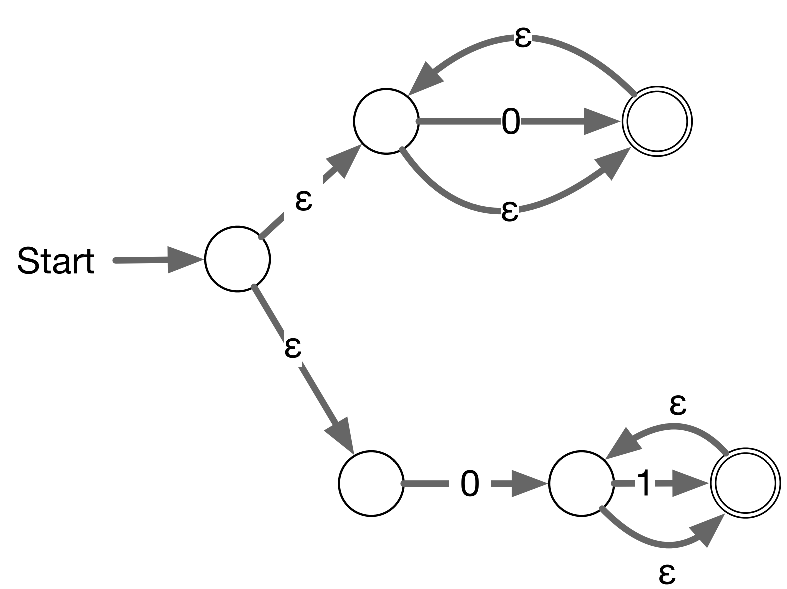
Examples

- Our recipes add many states and ϵ-transitions
  - Not necessary to keep them
  - There are actually optimization procedures to reduce the number of states and transitions

Examples

- **0* + 01***





Examples

- $(\mathbf{01} + \mathbf{10})^+ = \{01, 10, 0101, 0110, 1001, 1010, 010101, \ldots\}$