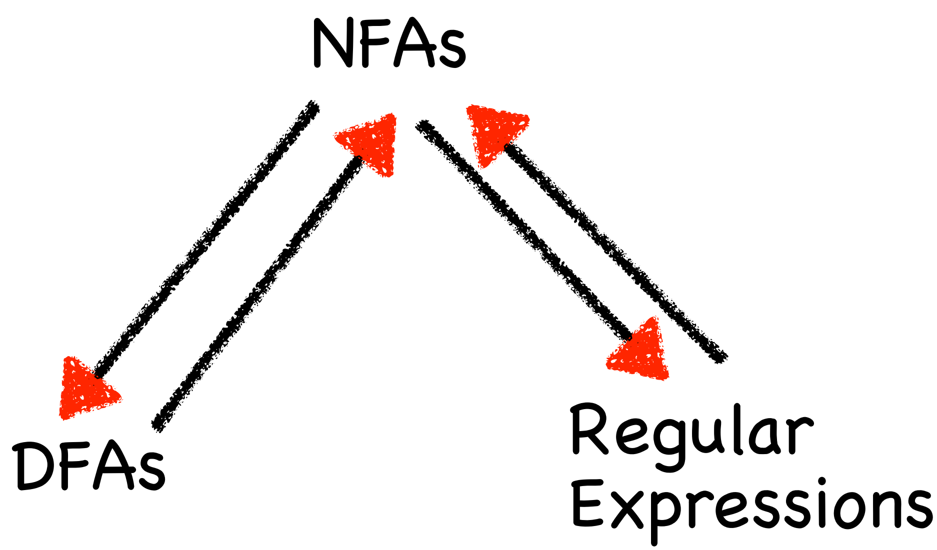
Still more on Regular Languages

CS 81: Computability and Logic

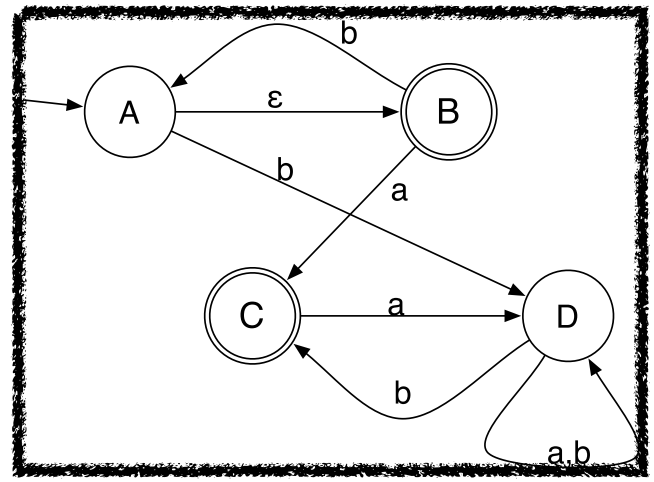
October 21, 2010

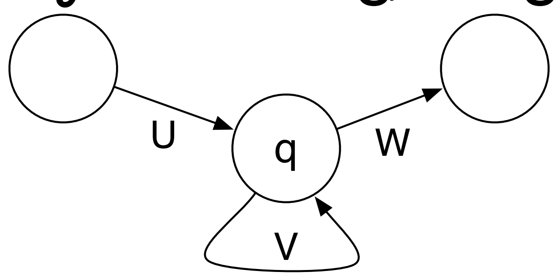
Representing Regular Languages: Conclusion



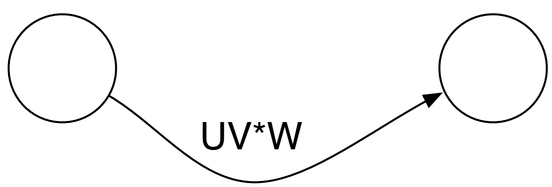
Removing States

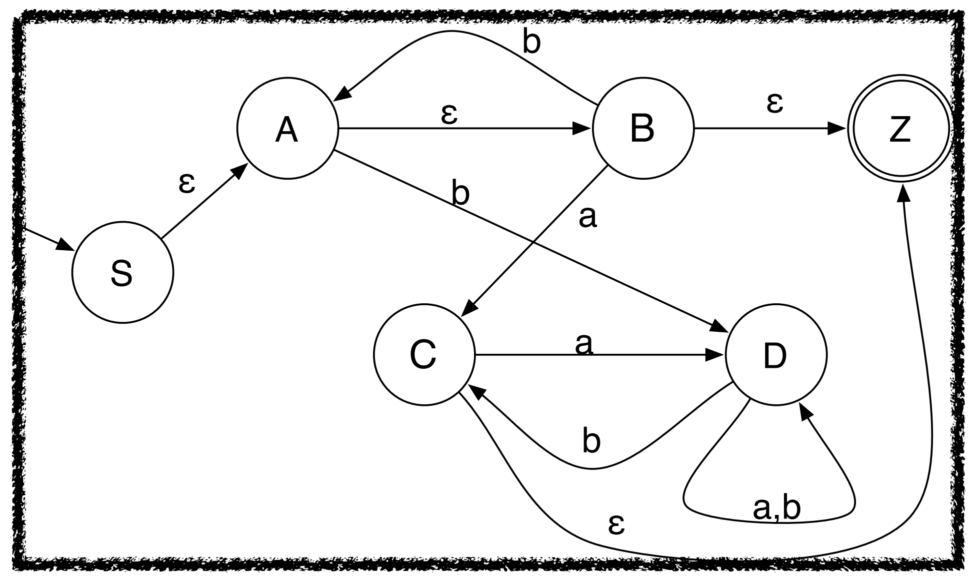
- Pick a state q to remove.
- For every incoming/outgoing pair





- replace by the direct edge





Closure Properties

- A family of languages is a set of languages.
  - The family of all finite languages
  - The family of all languages
  - The family of all regular languages
- A family F is closed under an operation if applying the operation to languages in F always produces a result in F.

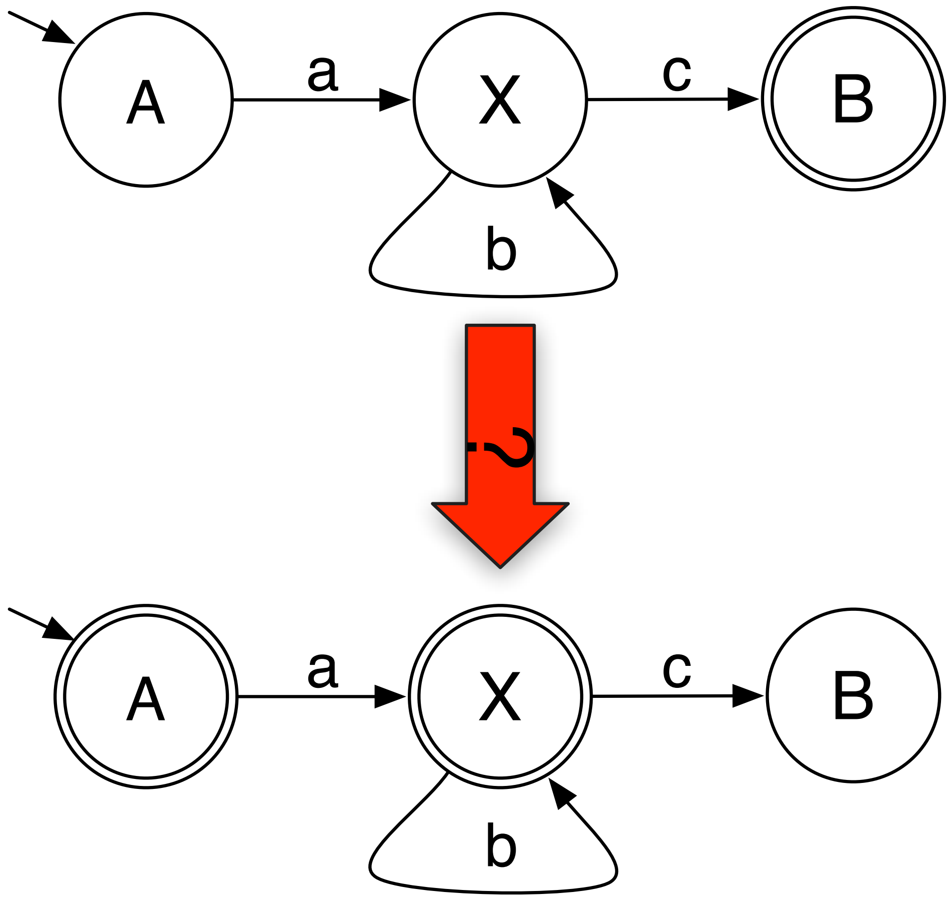
Example: Finite Languages

- Are the finite languages closed under:
  - Union? ($A \cup B$)
  - Intersection? ($A \cap B$)
  - Concatenation (AB)
  - Star? (A^*)
  - Complement? (A^c)

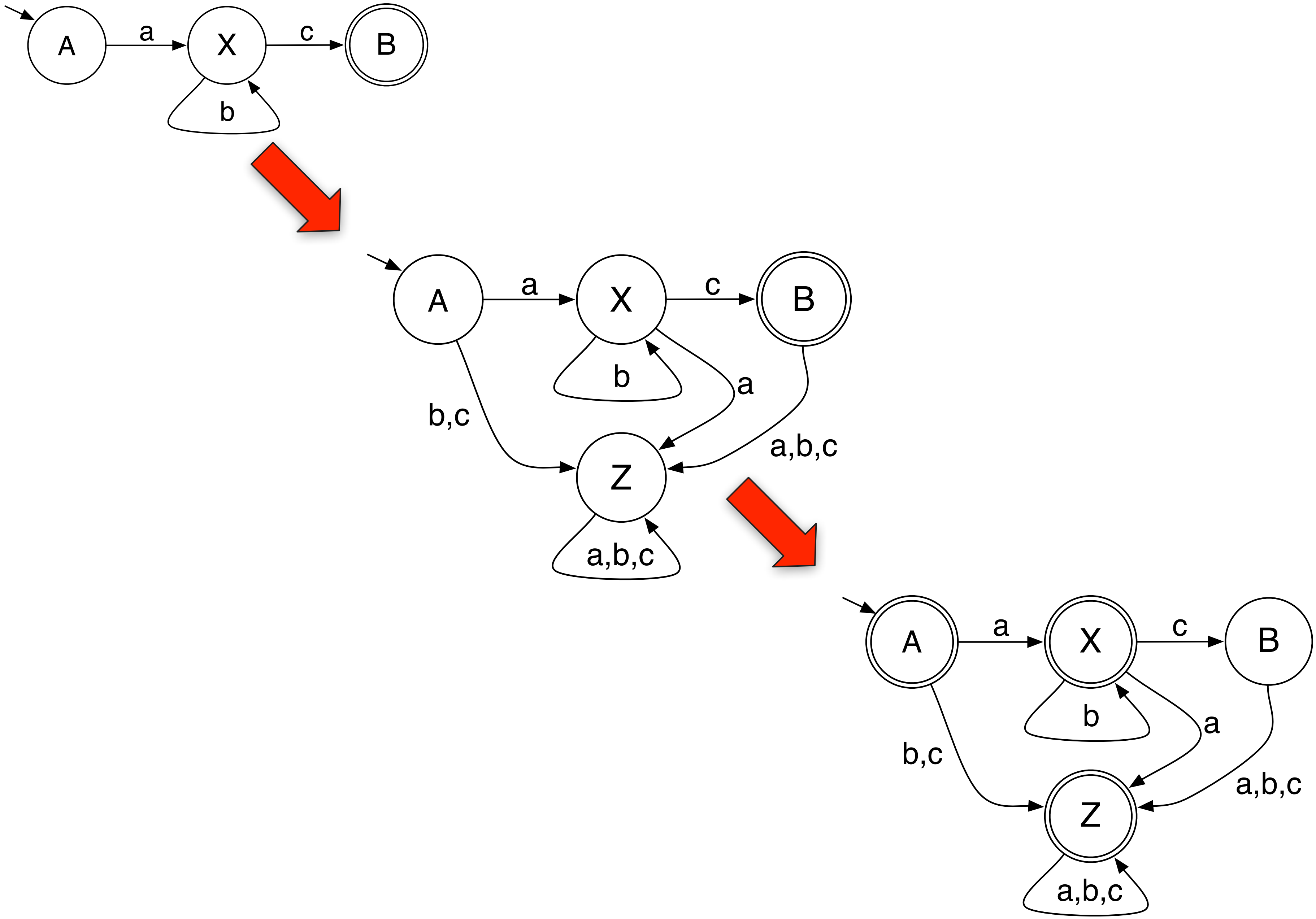
Closure Properties for Regular Languages

- Regular languages are closed under
  - Concatenation
  - Union
  - Star
  - Complement
  - Intersection
- Proofs: Consider the corresponding automata...

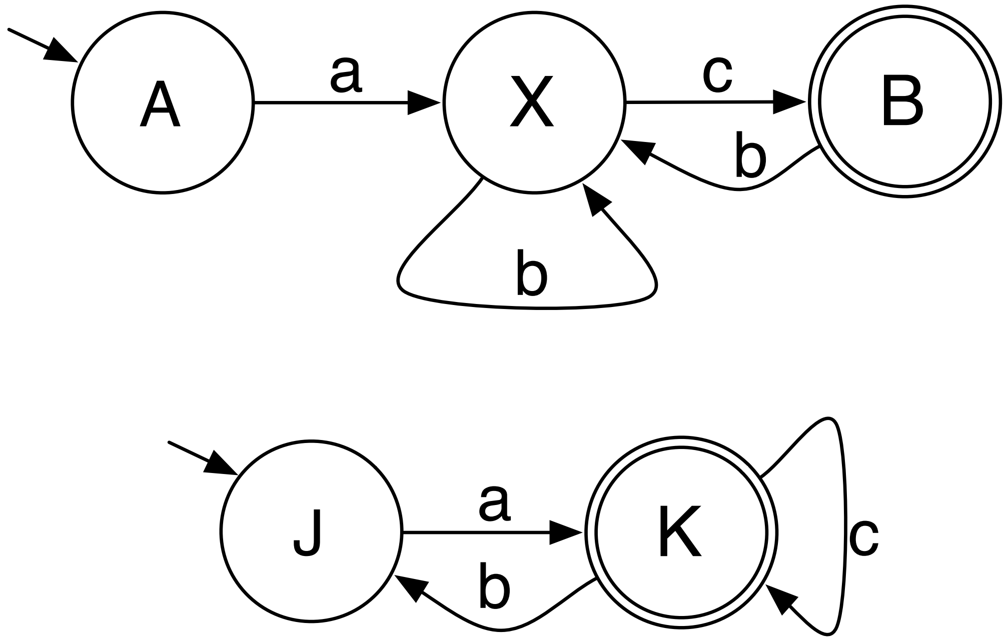
Complement?



Complement



Intersection



Abstract Machines

- Abstraction of computation.
- Simplest form: Recognizers

abbaabb...

Black Box

(Light goes on or off after each symbol)

Recall

$$L_w := \{ y \in \Sigma^* \mid wy \in L \}$$

$$\forall y \in \Sigma^*. \quad y \in L_w \leftrightarrow wy \in L$$

$$L_{uv} = (L_u)_v$$

Intrinsic States

$$\Sigma = \{a\}$$

$$L = \{ a^{3n} \mid n \geq 0 \}$$

$L_\varepsilon = \{ \varepsilon, aaa, aaaaaa, \ldots \}$

$L_a = \{ aa, aaaaa, aaaaaaaa, \ldots \}$

$L_{aa} = \{ a, aaaa, aaaaaaa, \ldots \}$

$L_{aaa} = \{ \varepsilon, aaa, aaaaaa, \ldots \}$

$L_{aaaa} = \{ aa, aaaaa, aaaaaaaa, \ldots \}$

...

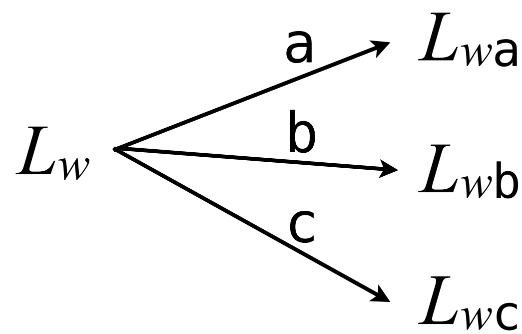
$$L_w := \{ y \in \Sigma^* \mid wy \in L \}$$

$$\forall y \in \Sigma^*. \quad y \in L_w \leftrightarrow wy \in L$$

$$L_{uv} = (L_u)_v$$

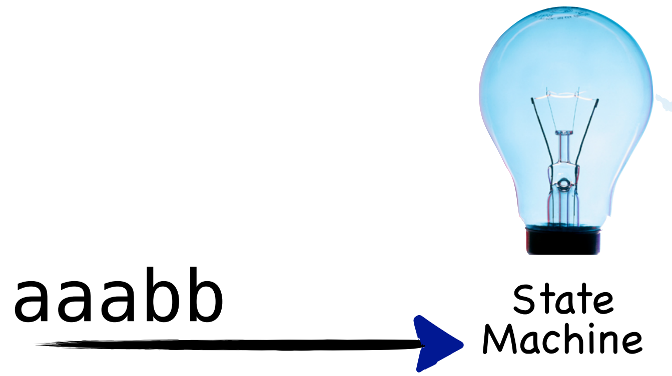
Language to Machine

- States
- Transitions
- Start State
- Accept State



The resulting machine is
deterministic and minimal

A Different Language



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

$L = \{\ a^n b^n \mid n \geq 0\ \}$

$L_b = L_{bb} = L_{bbb} = \cdots = L_{ba} = L_{baa} = L_{bbaab} = \varnothing$

$L_{ab} = L_{aabb} = L_{aaabbb} = \cdots = \{\ \varepsilon\ \}$

$L_\varepsilon = \{\ \varepsilon, ab, aabb, aaabbb, \ \ldots\}$
$L_a = \{b, abb, aabbb, \ldots\}$
$L_{aa} = \{bb, abbb, aabbbb, \ldots\}$
$L_{aaa} = \cdots$

$L_{aab} = L_{aaabb} = L_{aaaabbb} = \cdots = \{\ b\ \}$
$L_{aaab} = L_{aaaabb} = \cdots = \{\ bb\ \}$
$L_{aaaab} = L_{aaaaabb} = \cdots = \{\ bbb\ \}$

Proving Myhill-Nerode

Theorem [Myhill-Nerode]:

A language is regular iff $\{ L_w \mid w \in \Sigma^* \}$ is finite.

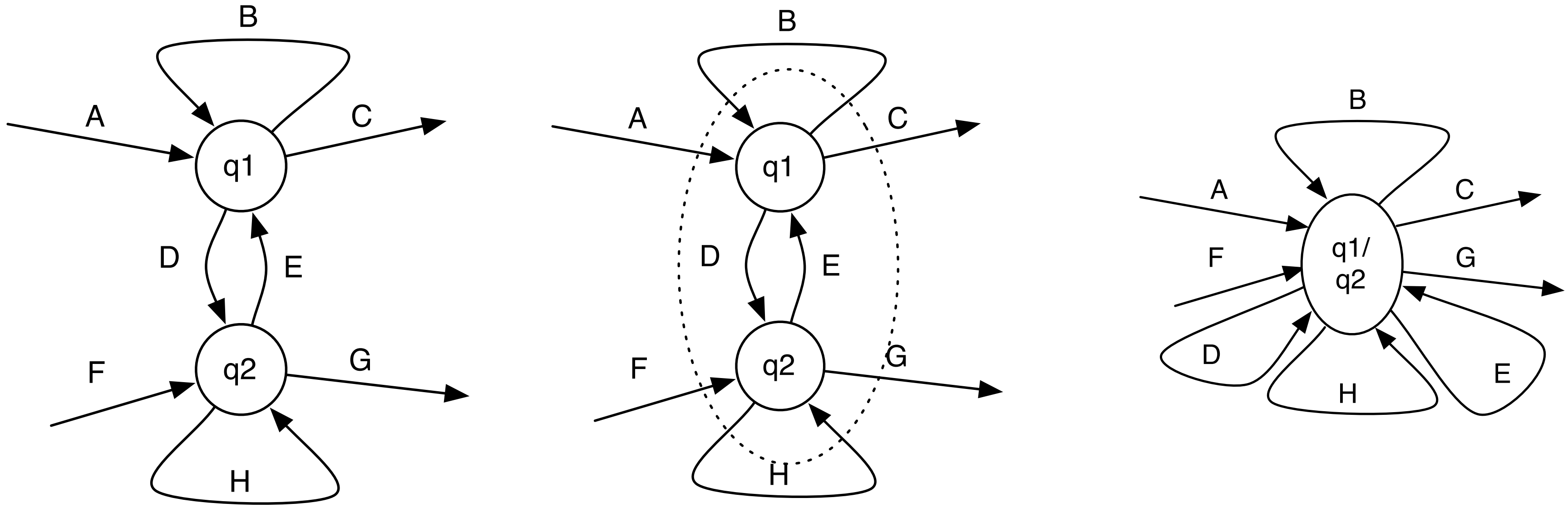
Proof:

If: For any language, we can construct a state machine from $\{ L_w \mid w \in \Sigma^* \}$. If this set is finite, we get a DFA.

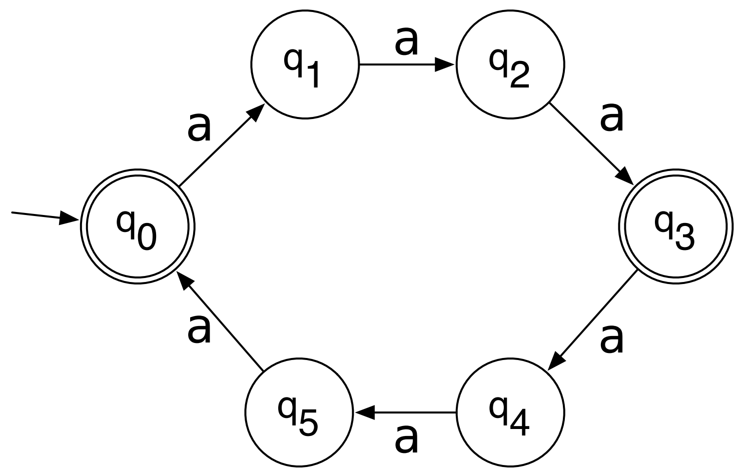
Only if: Assume we have a DFA for L. Run w through the DFA, reaching some state q. Then $L_w = L_q$. Since there are only finitely many L_q's, there can only be finitely many different L_w's.

State Minimization

- If two states have the same language, they can be merged without changing the language of the state machine.

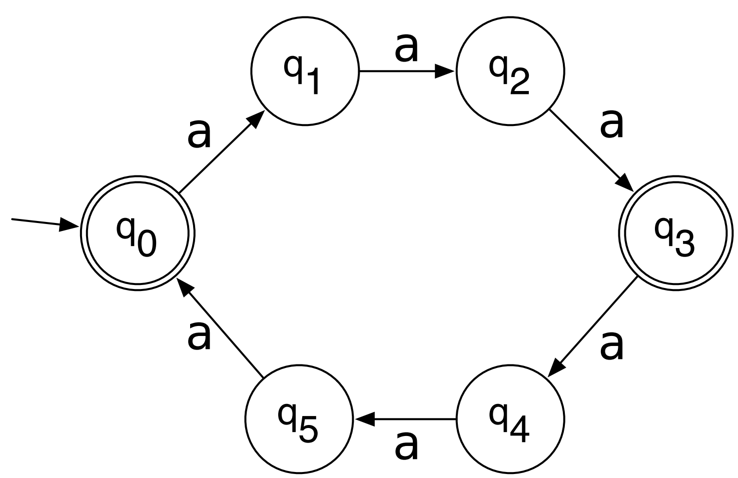


Example

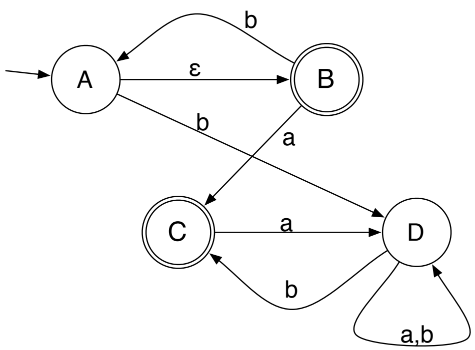


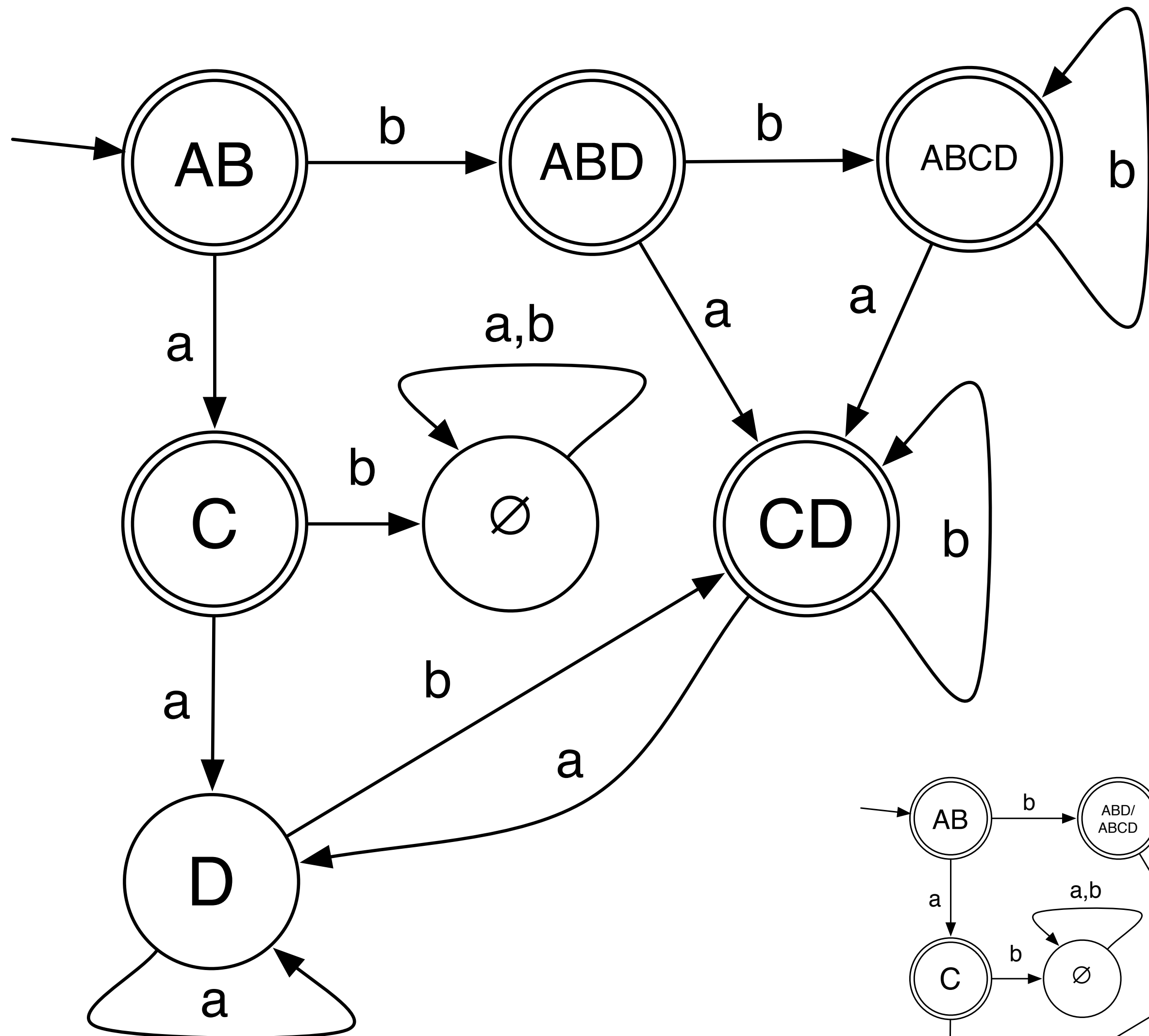
One Minimization Algorithm

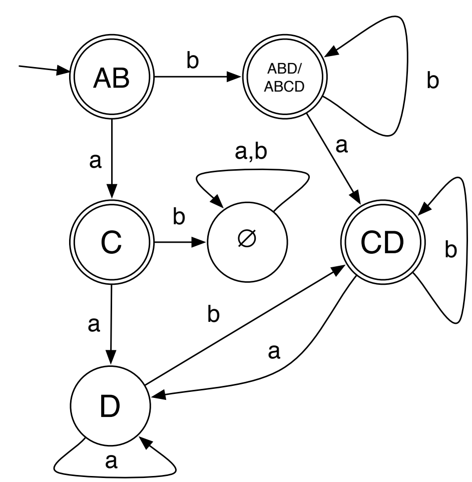
- Assume all states are mergable unless there's evidence they're not otherwise.
  - Accepting vs. nonaccepting
  - One symbol takes us to two states known to be different.



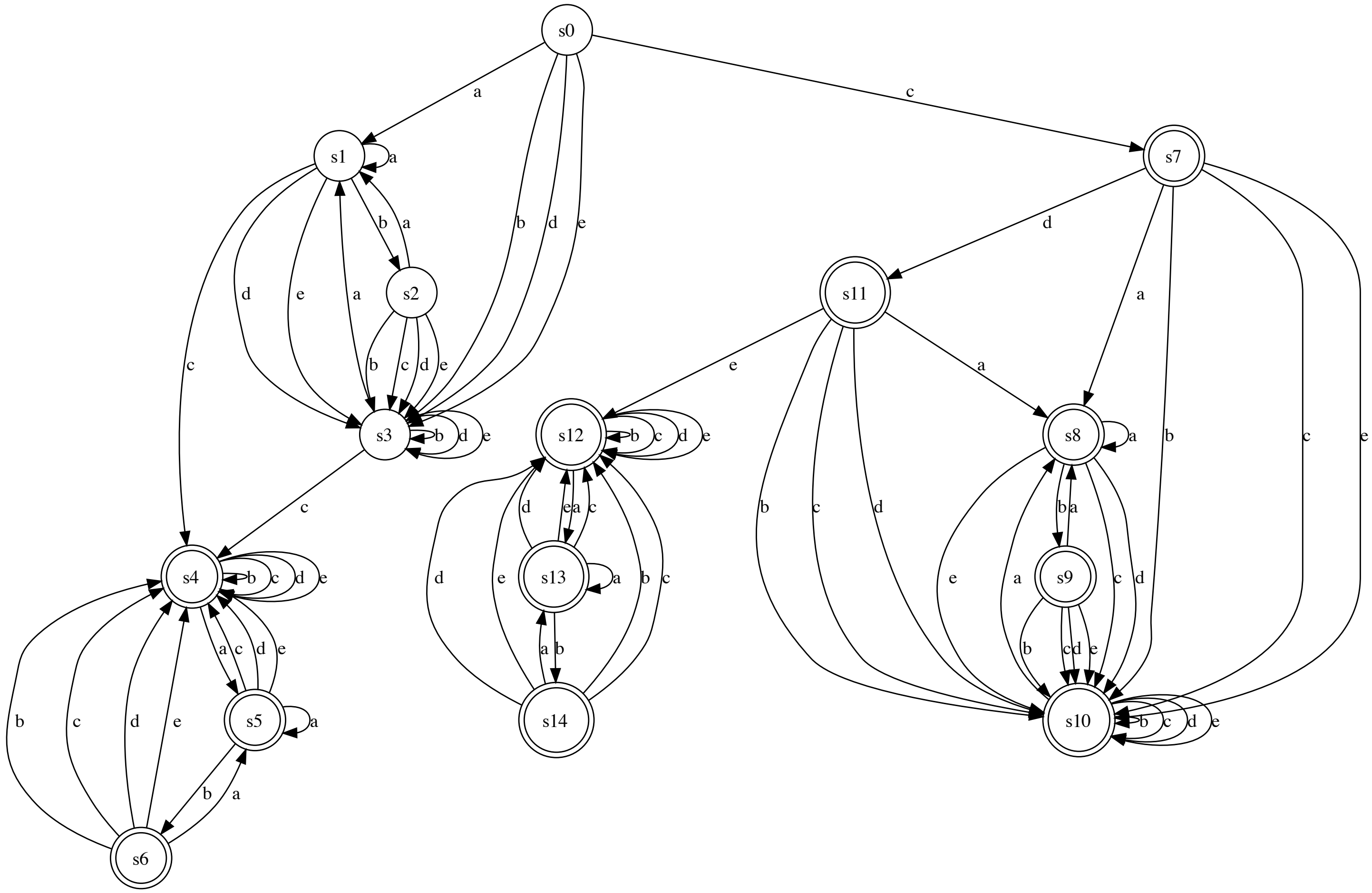
More Complex Example

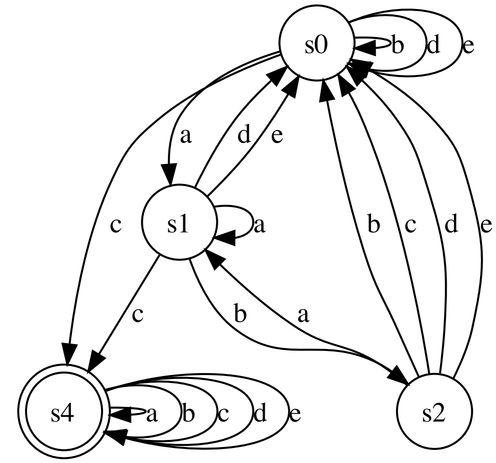






More Minimization





Mazes

- Suppose you enter a maze of twisty passages, all alike.
  (but with doors that open from only one side)
- You happen to know that this maze has exactly 19 rooms.
  You start wandering and pass through 27 rooms.
  What can you conclude?
- This wandering has brought you to an exit.
  What can you conclude about other solutions to the maze?
- Was there anything special about the numbers 19 and 27?

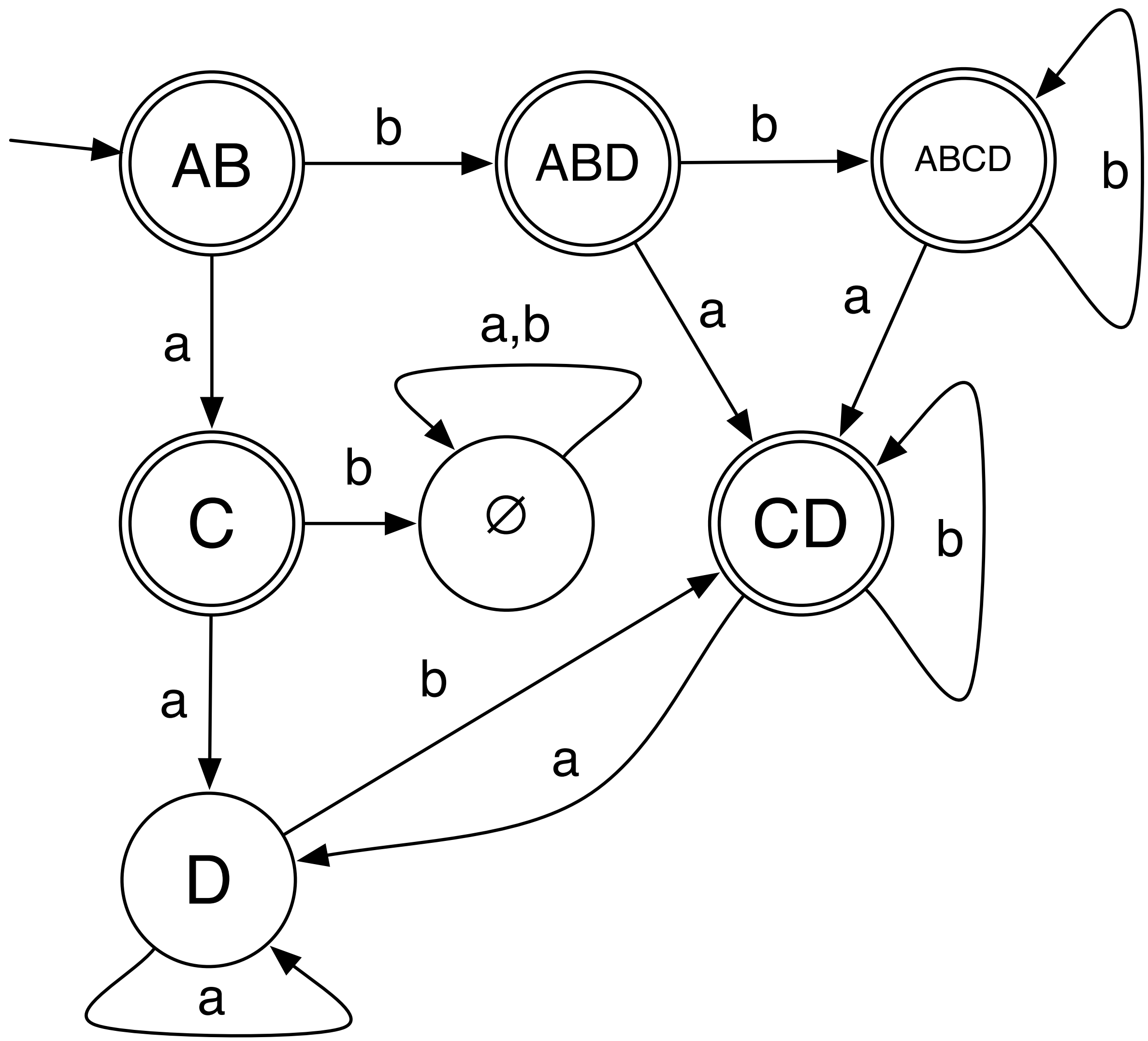
A "Theorem" About Finite Mazes

For every finite maze, there is a number p such that

For every path through the maze s with $|s| \geq p$:

The path s contains at least one loop starting and ending within the first p steps.

Finite Automata as Mazes



Pumping Lemma

If L is a regular language, then there exists a number p such that:

For every $s \in L$ with $|s| \geq p$:

There's at least one way to decompose s into xyz where

$|y| > 0$ and

$|xy| \leq p$ and

$xy^iz \in L$ for every $i \geq 0$.