CSCI 742 - Compiler Construction

Lecture 7

Regular Expression - Automata Conversion

Instructor: Hossein Hojjat

February 6, 2017

Lexer Automatic Construction: Big Picture

Input: Token Spec

- List of regular expressions (RE) in priority order

Output: Lexer

- Reads an input stream and breaks it up into tokens according to REs

Algorithm

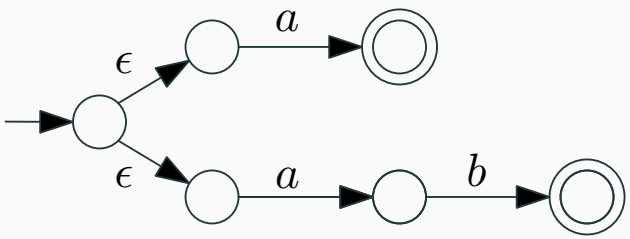
- Convert REs into non-deterministic finite automata (NFA)
- Convert NFA to DFA
- Convert DFA into transition table

Lexer Automatic Construction: Example

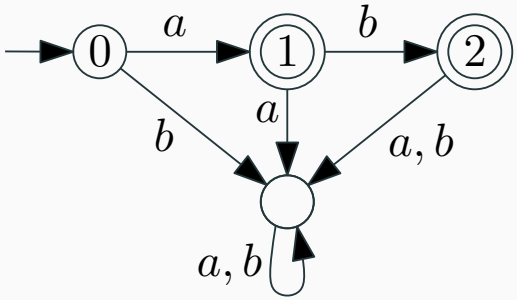
- RE for tokens:

$$(a|ab)$$

- NFA:



- DFA:



- Transition Table:

	a	b
0	1	Error
1	Error	2
2	Error	Error

Kleene's Theorem

Theorem

A language L can be described by regular expression if and only if L is the language accepted by a finite automaton.

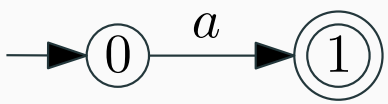
Algorithms:

- regular expression $\Rightarrow$ automaton — important for lexer construction
- automaton $\Rightarrow$ regular expression — interesting method in formal languages theory

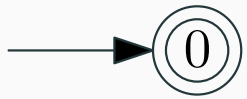
RE $\Rightarrow$ Finite Automaton

- Build the finite automaton inductively, based on the definition of regular expressions

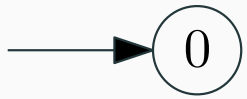
a



ϵ

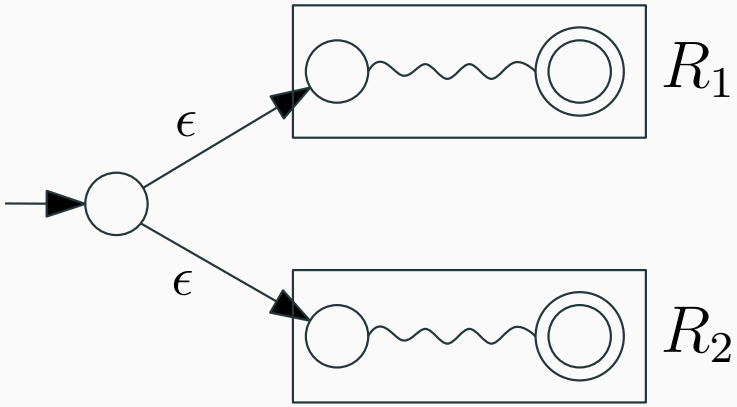


$\emptyset$



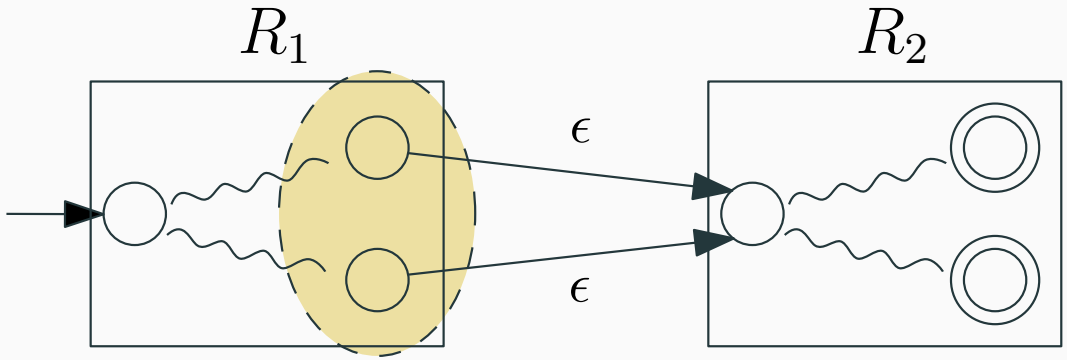
RE $\Rightarrow$ Finite Automaton

Alternation $R_1 \mid R_2$



Contcatenation

$R_1 \ . \ R_2$

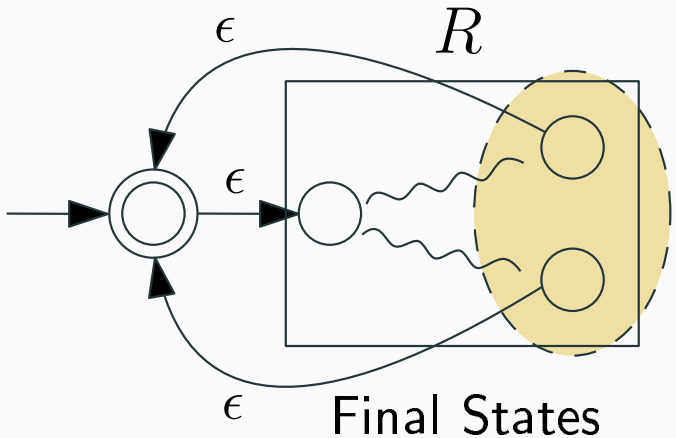


Final States

no final anymore

RE $\Rightarrow$ Finite Automaton

Alternation $R*$



Exercise

Question

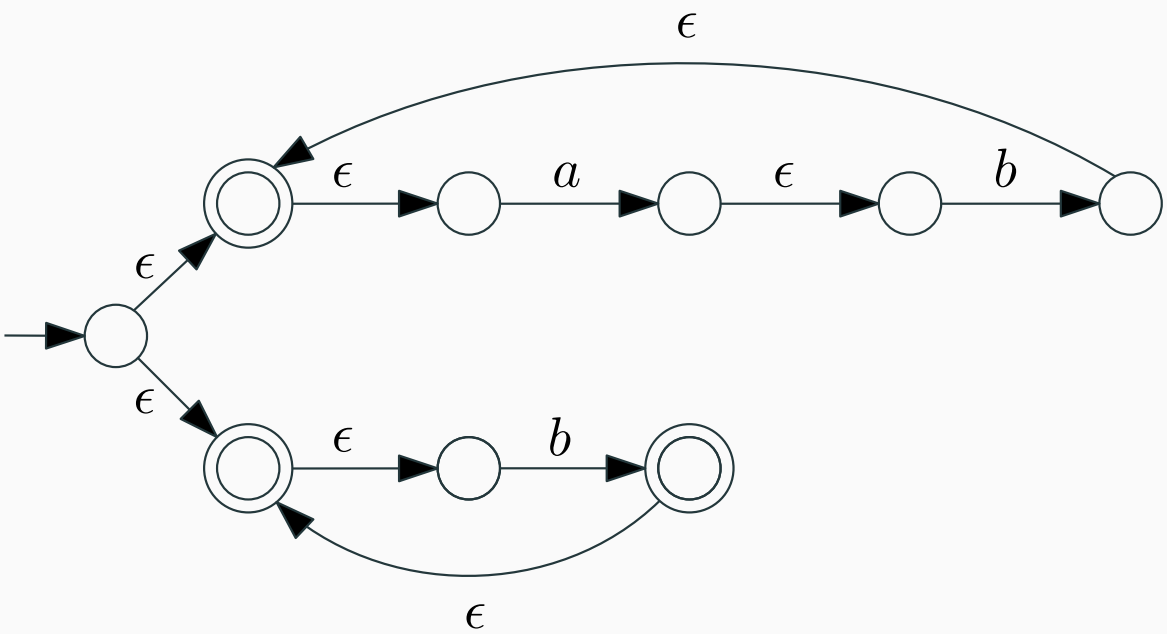
- Construct an NFA for the regular expression $(ab)* \mid b*$

Exercise

Question

- Construct an NFA for the regular expression $(ab)* \mid b*$

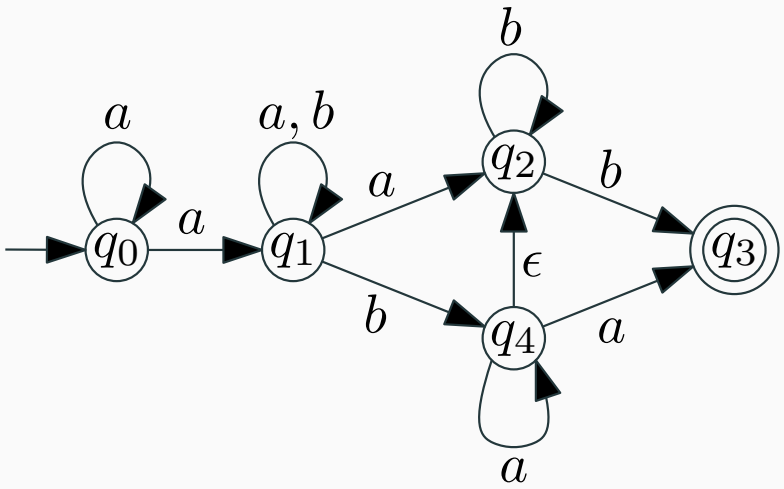
Answer



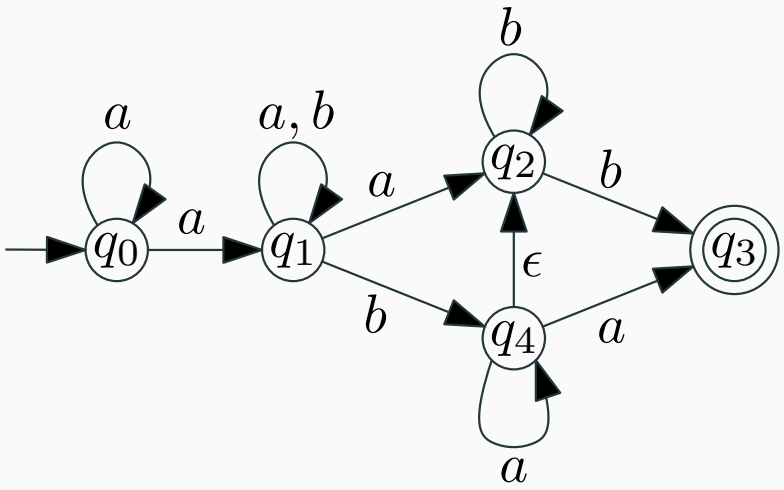
From NFA to DFA

- For every NFA there exists an equivalent DFA that accepts the same set of strings
- NFAs could be exponentially smaller (succinct)
- Idea: keep track of a set of all possible states in which the automaton could be
- View this finite set as one state of new automaton

From NFA to DFA: Example

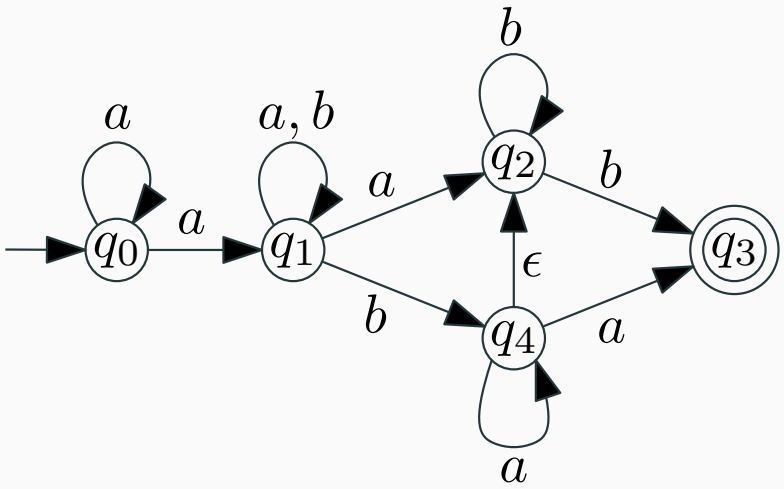


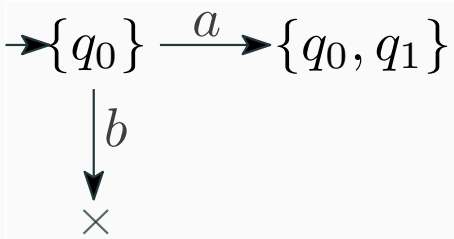
From NFA to DFA: Example



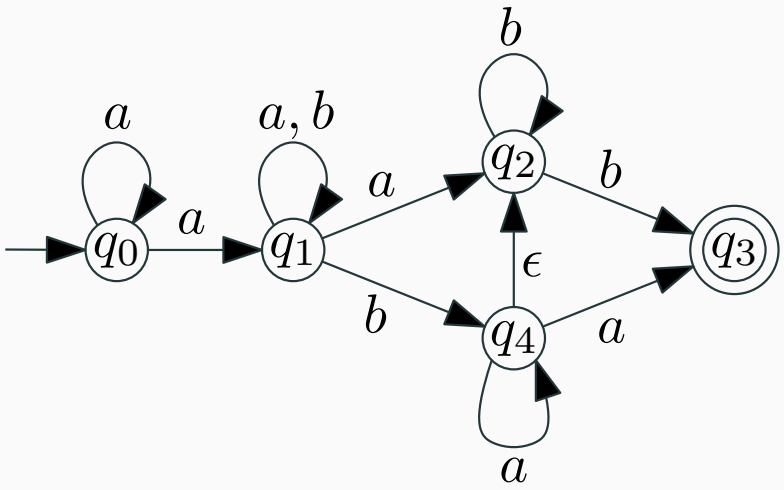
$\rightarrow\{q_0\}$

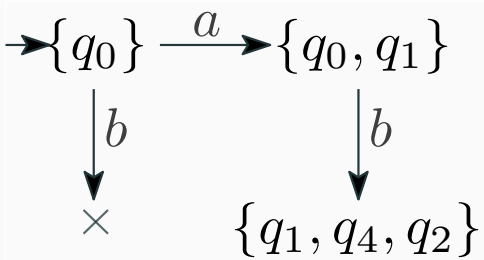
From NFA to DFA: Example



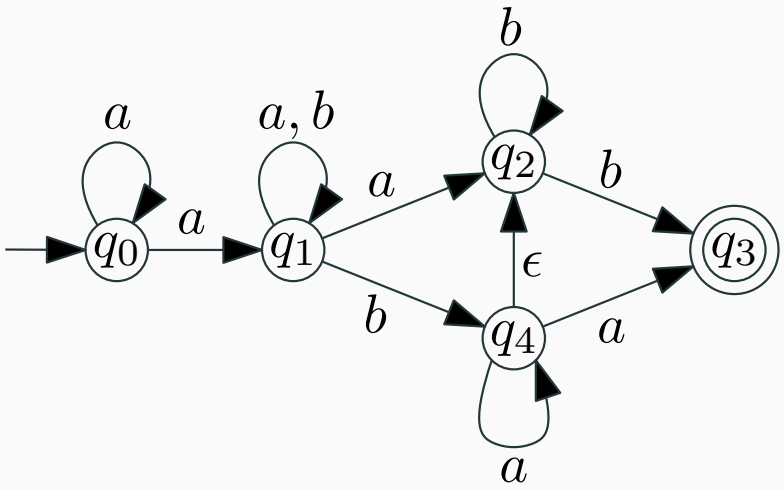


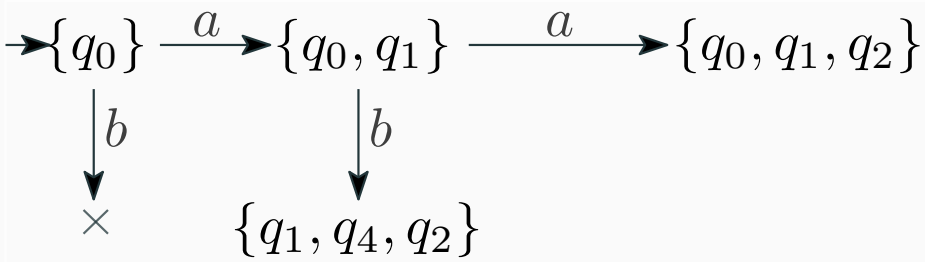
From NFA to DFA: Example



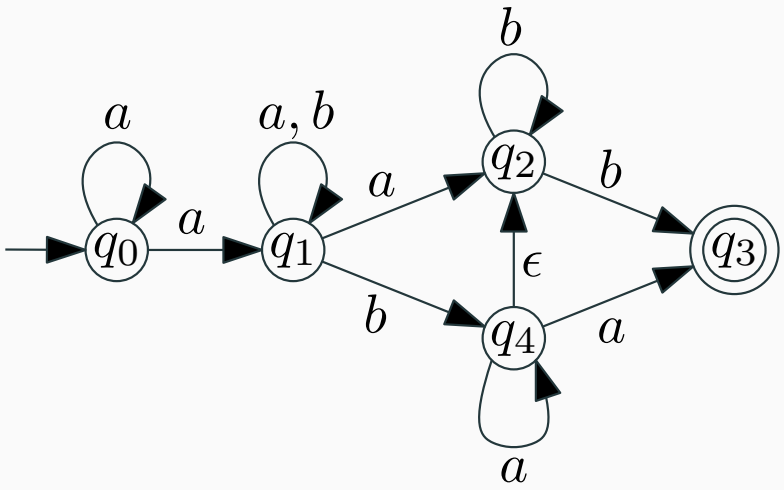


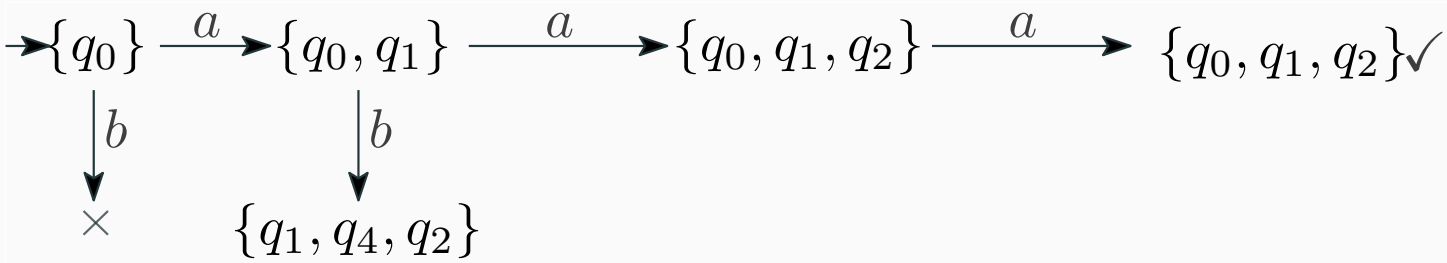
From NFA to DFA: Example





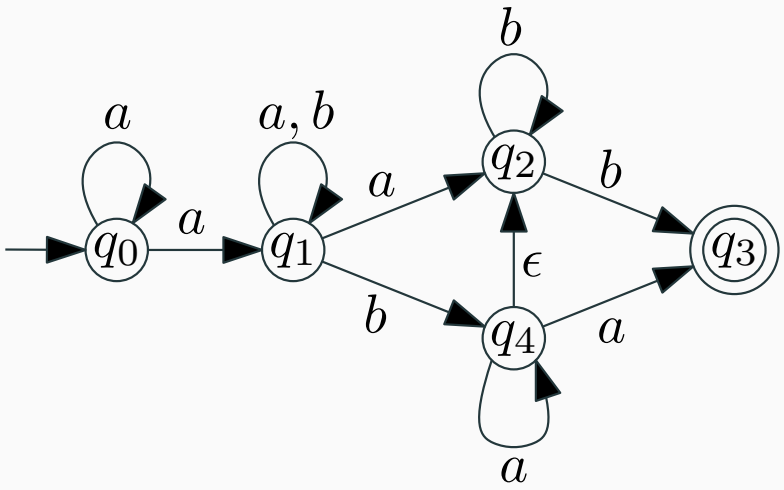
From NFA to DFA: Example

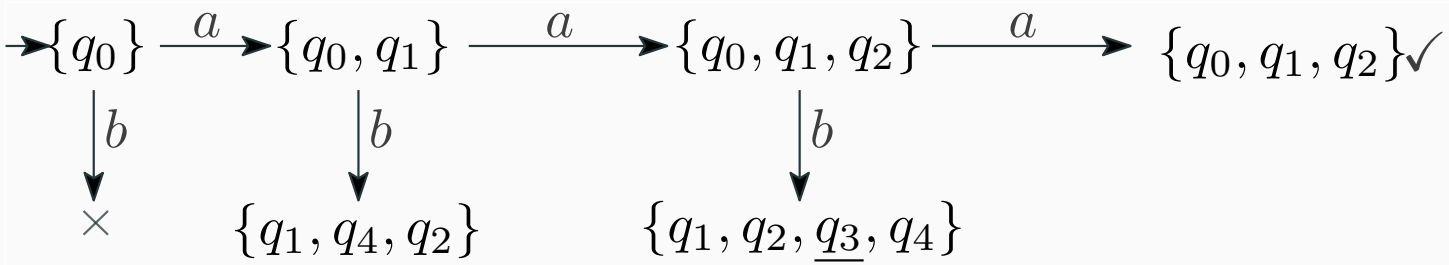




- When processing if we see a set exactly the same as a set constructed earlier we mark it

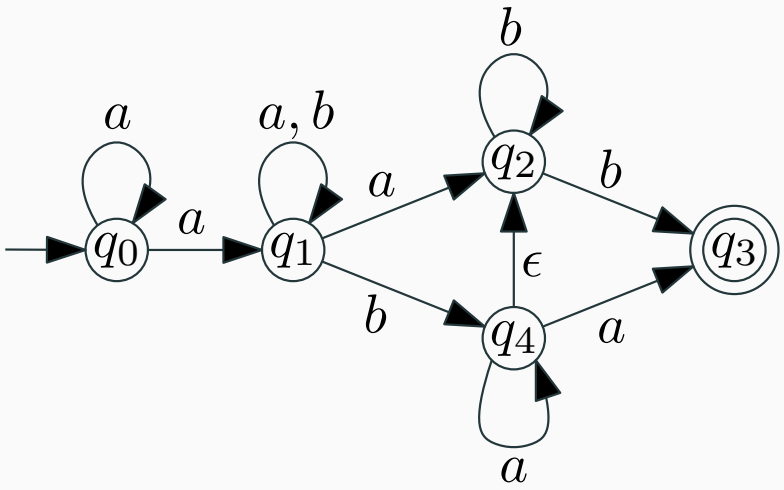
From NFA to DFA: Example

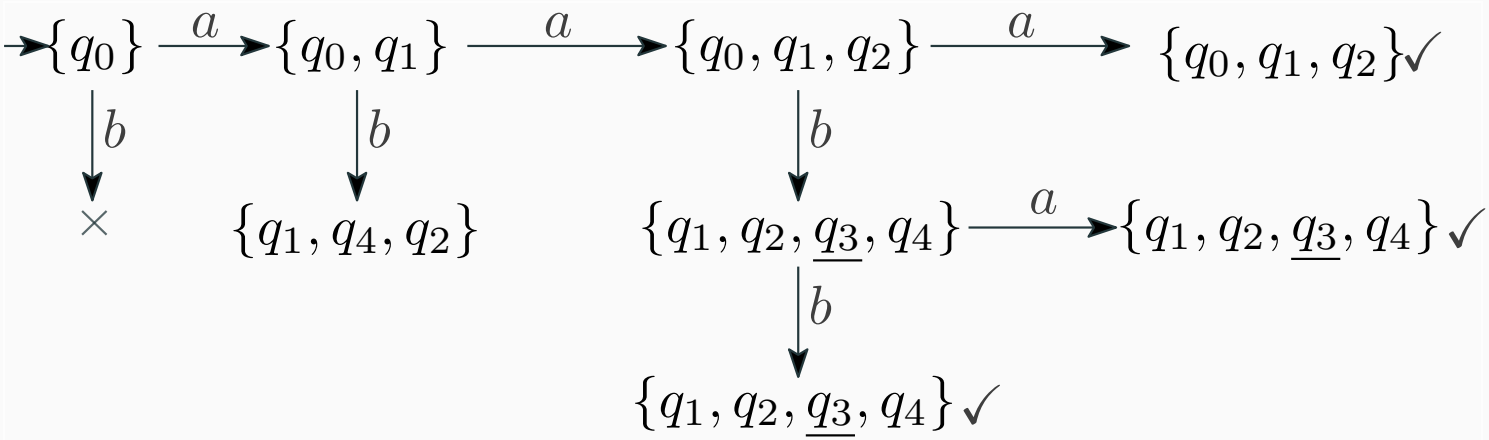




- When processing if we see a set exactly the same as a set constructed earlier we mark it

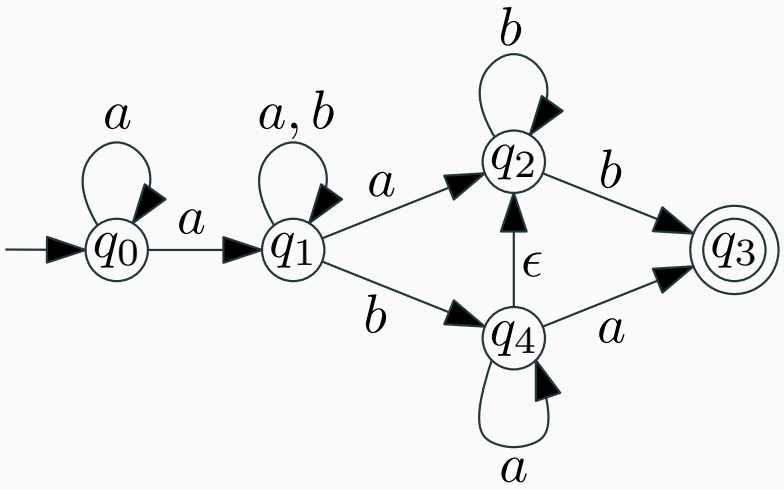
From NFA to DFA: Example

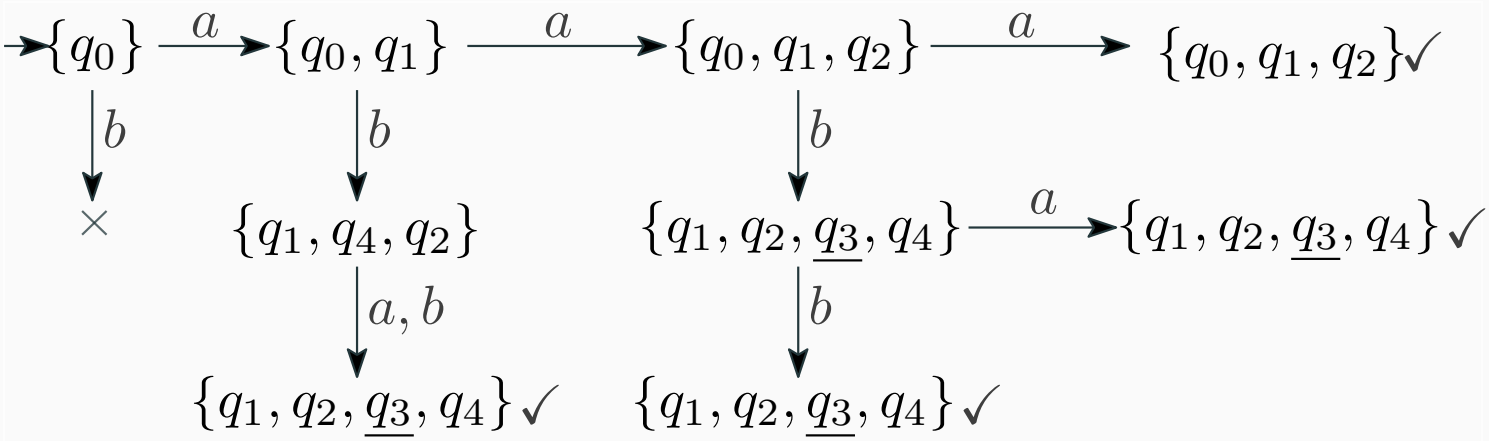




- When processing if we see a set exactly the same as a set constructed earlier we mark it

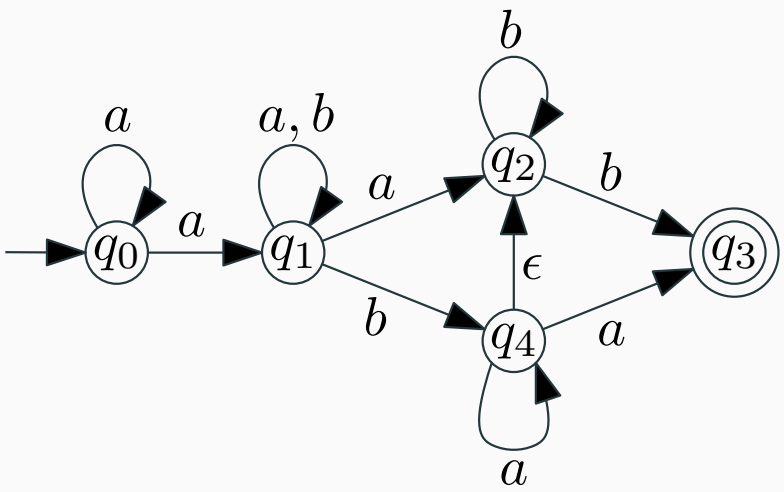
From NFA to DFA: Example

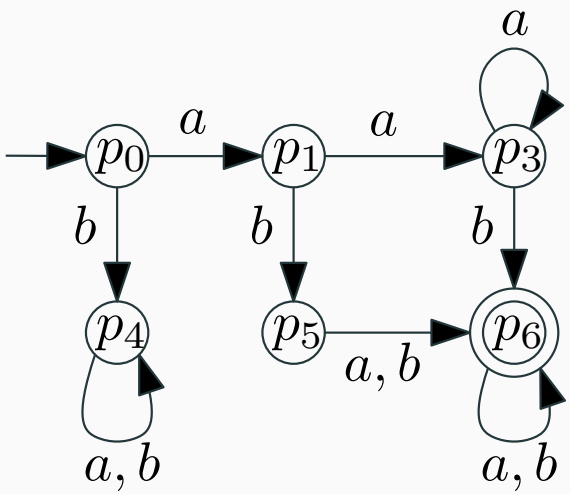


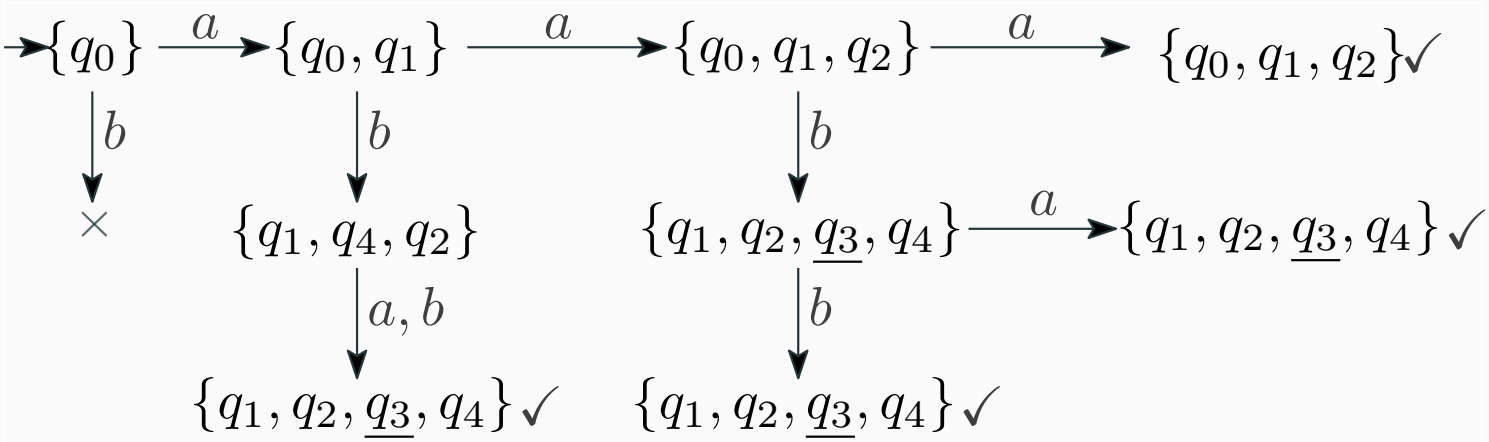


- When processing if we see a set exactly the same as a set constructed earlier we mark it

From NFA to DFA: Example







- When processing if we see a set exactly the same as a set constructed earlier we mark it

Exercise

Question

- Construct an NFA for the regular expression $(bb*)|a*$ over alphabet $\{a, b\}$ and determinize it.