CS420

Introduction to the Theory of Computation

Lecture : Module 1 recap

Tiago Cogumbreiro

Mini-test 1

Location: University Hall

2nd floor, Classroom 2330

5:30pm~7:30pm

What you will need to know for mini-test 1

- Operators for DFAs (union, char, empty, nil)
- Convert an NFA into a DFA
- Convert a REGEX into an NFA
- Convert an NFA into a REGEX
- Design an DFA/NFA/REGEX that recognizes a language
- Prove that a language is not regular (Pumping Lemma)
- Design a CFG that recognizes a language
- The algorithm that returns the Chomsky Normal Form

Today we will recap...

- Drawing a state diagram systematically
- The union operator
- Converting an NFA into a DFA
- Converting an NFA into a REGEX
- Removing unit-rules

Tip 1

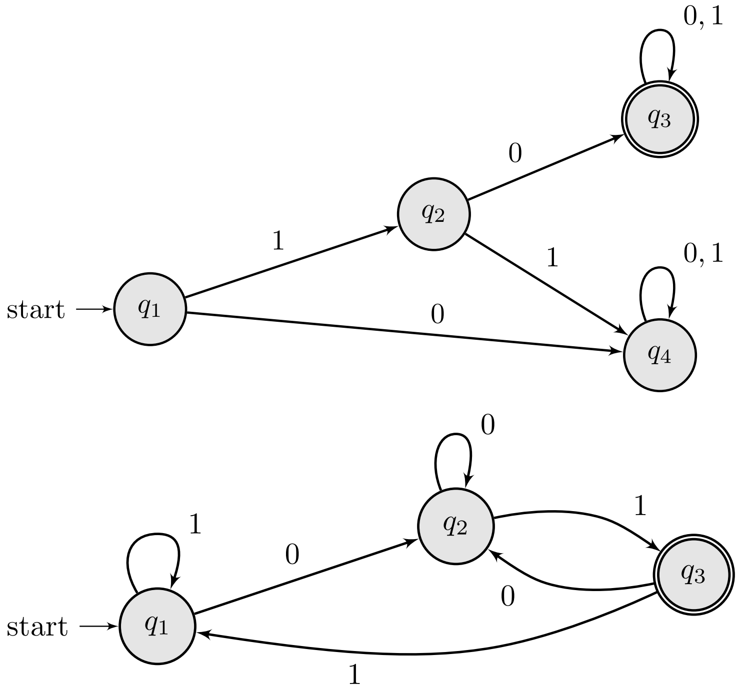
1. Derive the transitions
2. Draw the state diagram

(If you do both at once, you might forget transitions)

The union operator

The union operator

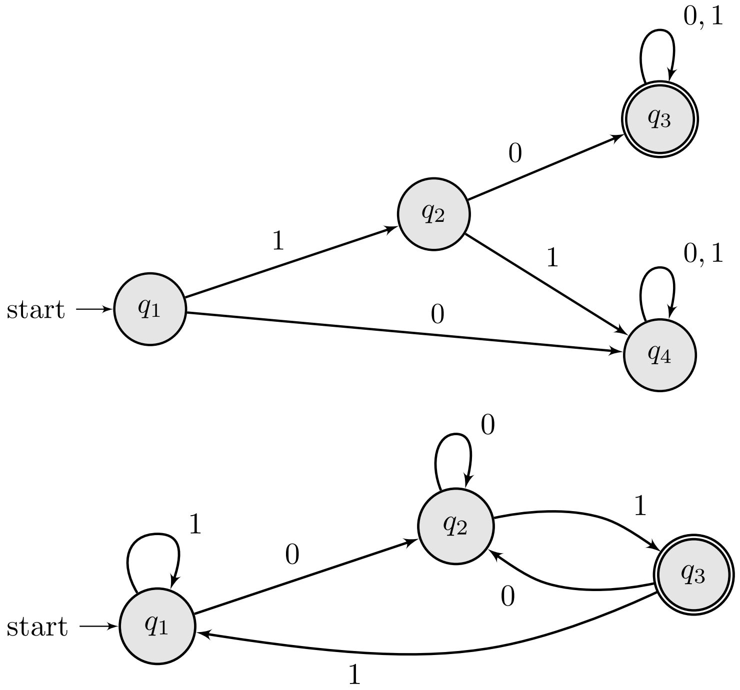
First, list the transitions of the new state diagram



Source	Edge	Target	Done
(q_1, q_1)	0		
(q_1, q_1)	1		

The union operator

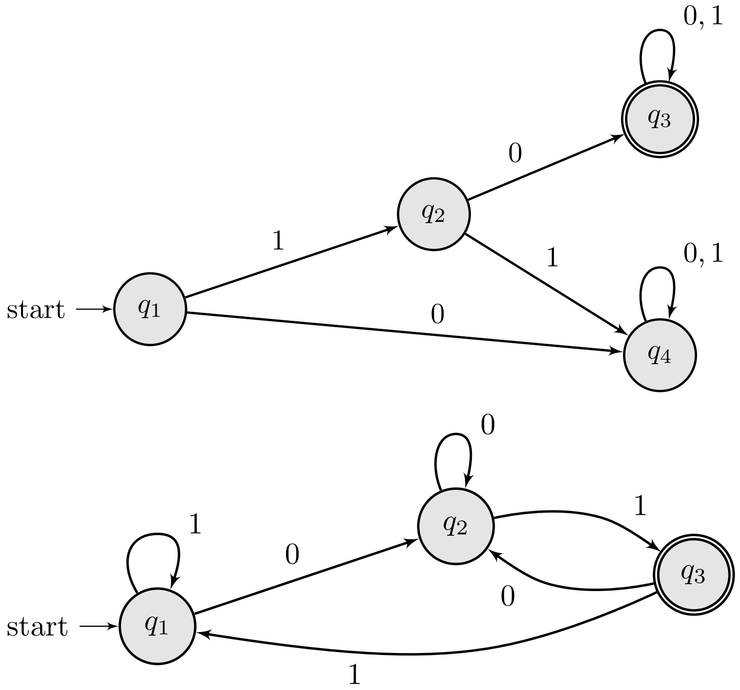
- First, list the transitions of the new state diagram



Source	Edge	Target	Done
(q_1, q_1)	0	(q_4, q_2)	
(q_1, q_1)	1	(q_2, q_1)	
(q_4, q_2)	0		
(q_4, q_2)	1		
(q_2, q_1)	0		
(q_2, q_1)	1		

The union operator

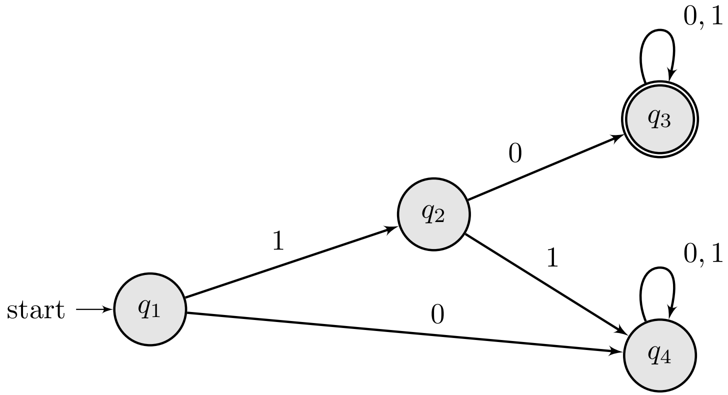
First, list the transitions of the new state diagram

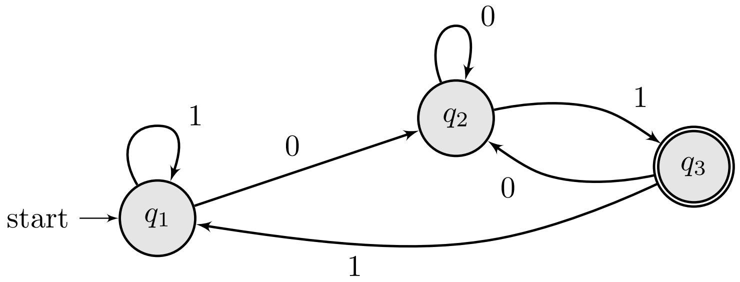


Source	**Edge**	**Target**	**Done**
(q_1, q_1)	0	(q_4, q_2)	
(q_1, q_1)	1	(q_2, q_1)	
(q_4, q_2)	0		
(q_4, q_2)	1		
(q_2, q_1)	0		
(q_2, q_1)	1		

The union operator

> First, list the transitions of the new state diagram

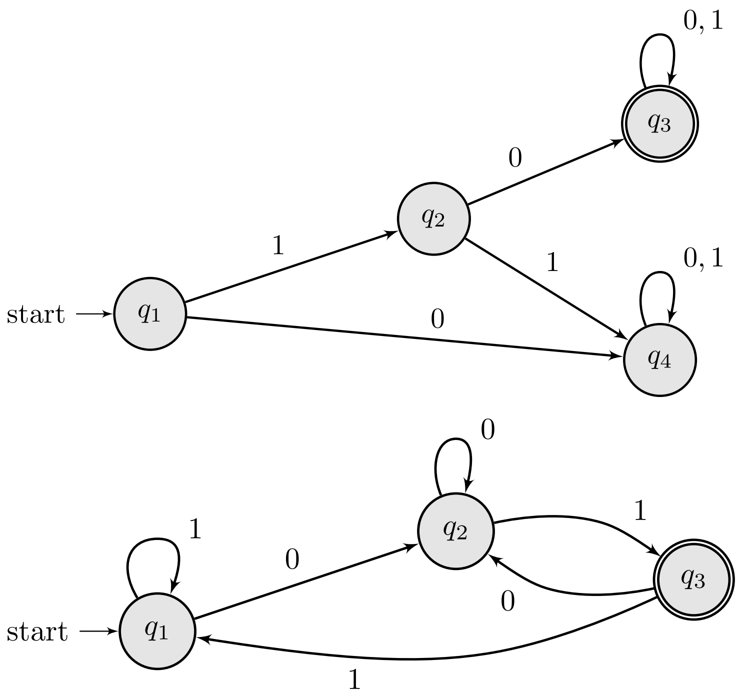




Source	Edge	Target	Done
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	
(q_2, q_1)	0		
(q_2, q_1)	1		
(q_4, q_3)	0		
(q_4, q_3)	1		

The union operator

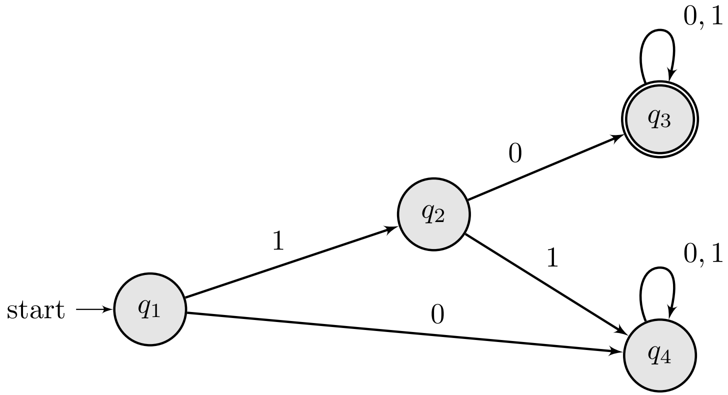
First, list the transitions of the new state diagram

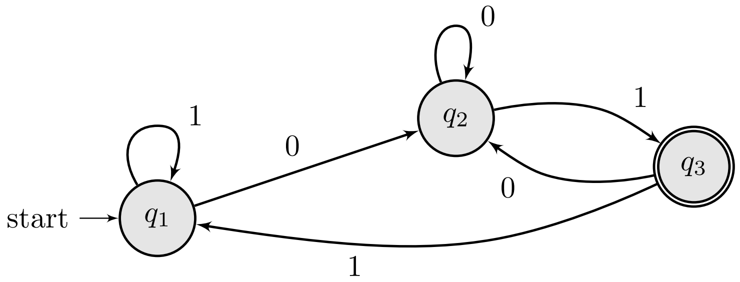


Source	***Edge***	***Target***	***Done***
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	
(q_2, q_1)	0	(q_3, q_2)	
(q_2, q_1)	1	(q_4, q_1)	
(q_4, q_3)	0		
(q_4, q_3)	1		
(q_3, q_2)	0		
(q_3, q_2)	1		
(q_4, q_1)	0		
(q_4, q_1)	1		

The union operator

First, list the transitions of the new state diagram

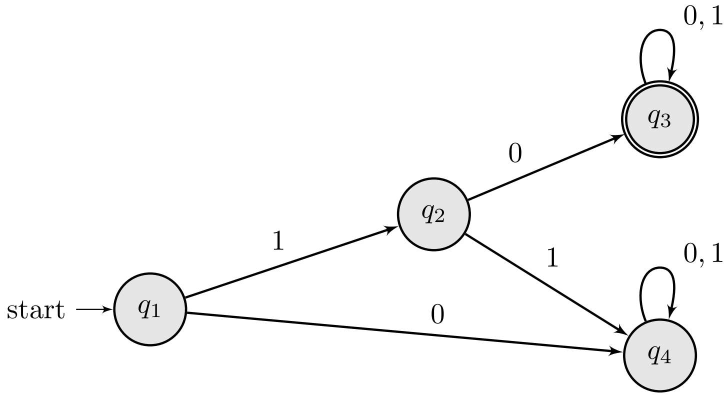


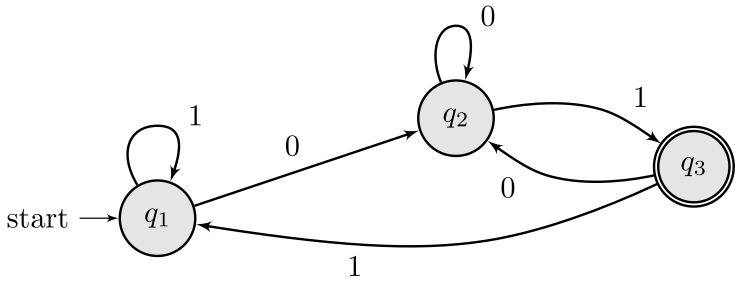


Source	**Edge**	**Target**	**Done**
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	
(q_2, q_1)	1	(q_4, q_1)	
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	
(q_3, q_2)	0		
(q_3, q_2)	1		
(q_4, q_1)	0		
(q_4, q_1)	1		

The union operator

First, list the transitions of the new state diagram

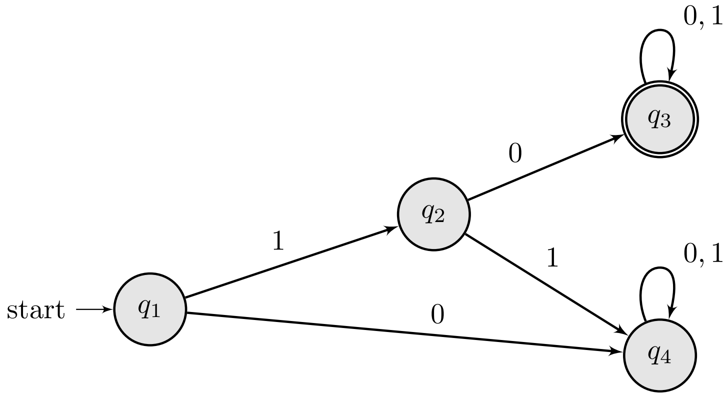


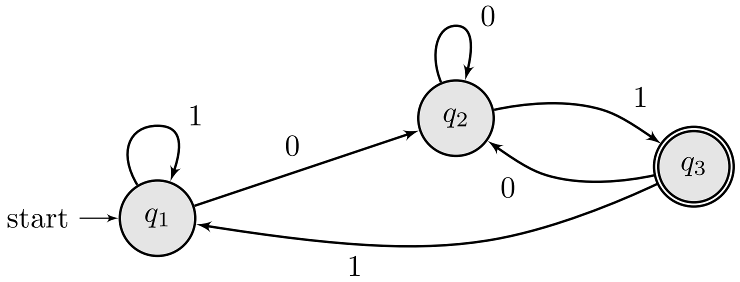


Source	Edge	Target	Done
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	x
(q_2, q_1)	1	(q_4, q_1)	
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	
(q_3, q_2)	0	(q_3, q_2)	x
(q_3, q_2)	1	(q_3, q_3)	
(q_4, q_1)	0		
(q_4, q_1)	1		
(q_3, q_3)	0		
(q_3, q_3)	1		

The union operator

First, list the transitions of the new state diagram

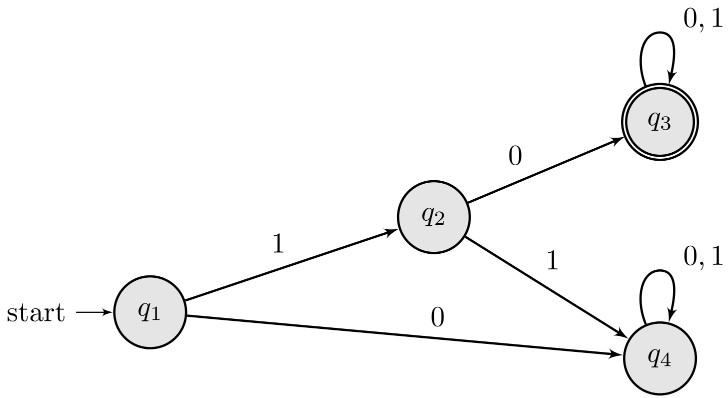


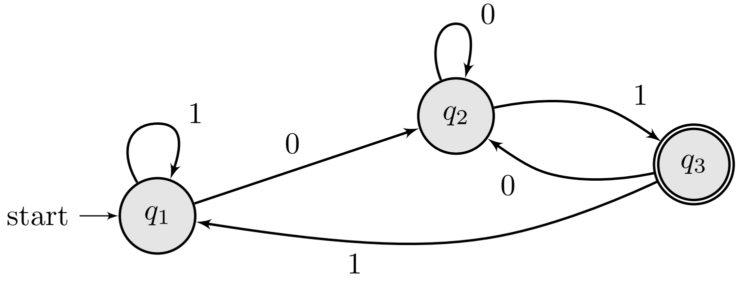


Source	**Edge**	**Target**	**Done**
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	x
(q_2, q_1)	1	(q_4, q_1)	x
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	x
(q_3, q_2)	0	(q_3, q_2)	x
(q_3, q_2)	1	(q_3, q_3)	
(q_4, q_1)	0	(q_4, q_2)	x
(q_4, q_1)	1	(q_4, q_1)	x
(q_3, q_3)	0		
(q_3, q_3)	1		

The union operator

First, list the transitions of the new state diagram

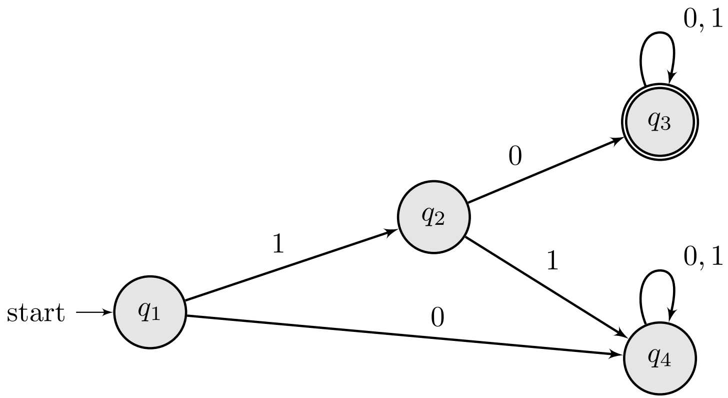


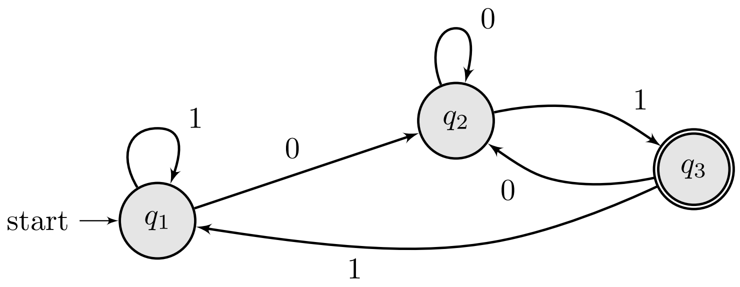


Source	Edge	Target	Done
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	x
(q_2, q_1)	1	(q_4, q_1)	x
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	x
(q_3, q_2)	0	(q_3, q_2)	x
(q_3, q_2)	1	(q_3, q_3)	x
(q_4, q_1)	0	(q_4, q_2)	x
(q_4, q_1)	1	(q_4, q_1)	x
(q_3, q_3)	0	(q_3, q_2)	x
(q_3, q_3)	1	(q_3, q_1)	
(q_3, q_1)	0		
(q_3, q_1)	1		

The union operator

First, list the transitions of the new state diagram



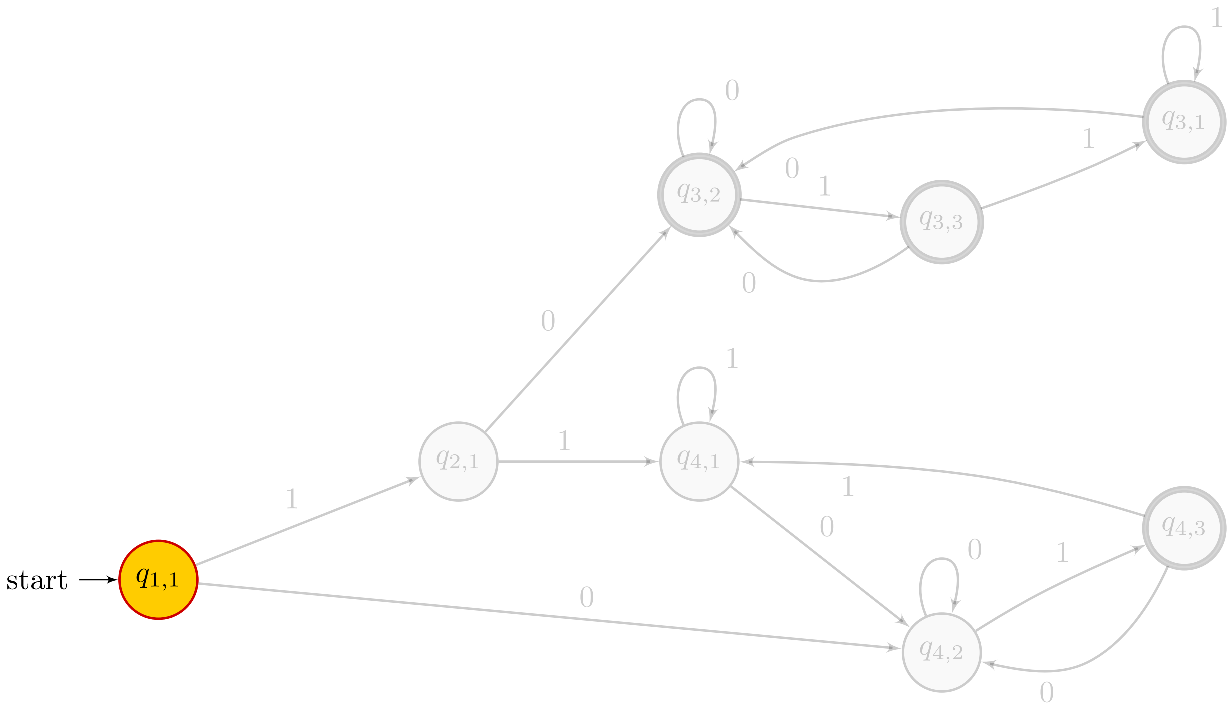


Source	Edge	Target	Done
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	x
(q_2, q_1)	1	(q_4, q_1)	x
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	x
(q_3, q_2)	0	(q_3, q_2)	x
(q_3, q_2)	1	(q_3, q_3)	x
(q_4, q_1)	0	(q_4, q_2)	x
(q_4, q_1)	1	(q_4, q_1)	x
(q_3, q_3)	0	(q_3, q_2)	x
(q_3, q_3)	1	(q_3, q_1)	x
(q_3, q_1)	0	(q_3, q_2)	x
(q_3, q_1)	1	(q_3, q_1)	x

Drawing state diagram

Drawing state diagram

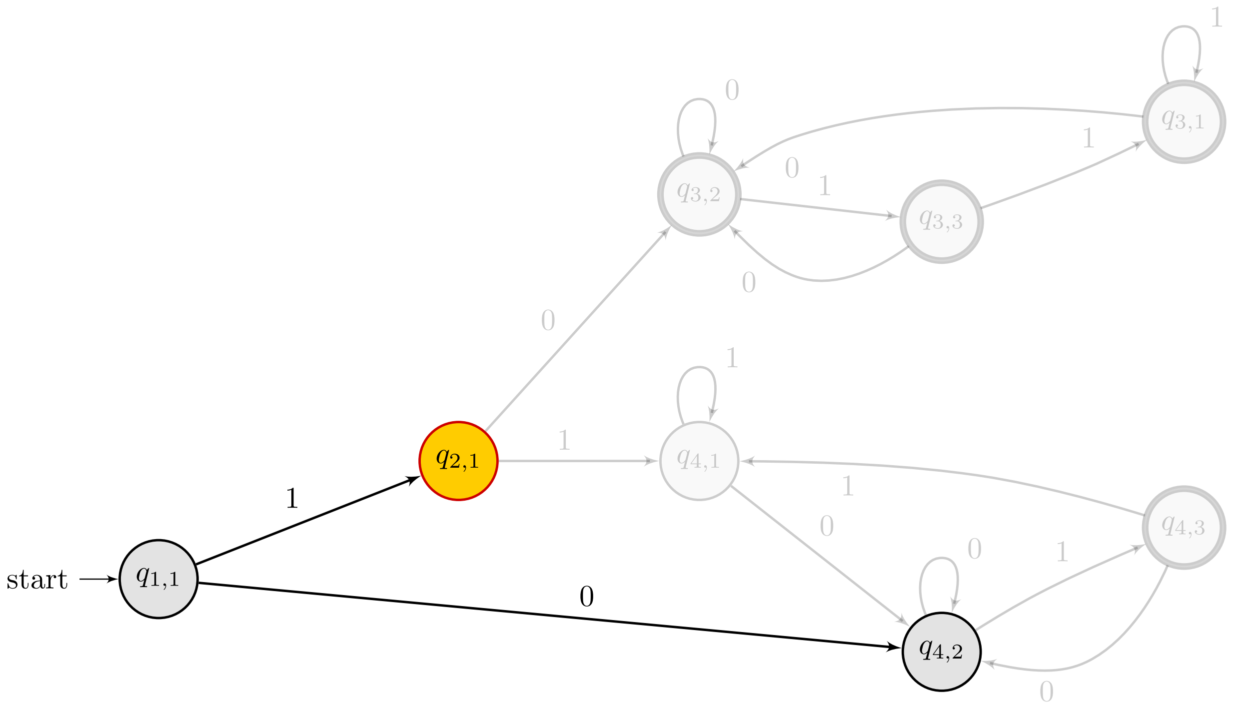
Source	Edge	Target	Done
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	x
(q_2, q_1)	1	(q_4, q_1)	x
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	x
(q_3, q_2)	0	(q_3, q_2)	x
(q_3, q_2)	1	(q_3, q_3)	x
(q_4, q_1)	0	(q_4, q_2)	x
(q_4, q_1)	1	(q_4, q_1)	x
(q_3, q_3)	0	(q_3, q_2)	x
(q_3, q_3)	1	(q_3, q_1)	x
(q_3, q_1)	0	(q_3, q_2)	x
(q_3, q_1)	1	(q_3, q_1)	x



Start with the initial state $q_{1,1}$ and draw its outgoing edges. Lookup the edges in the transition table.

Source	Edge	Target	Done
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	x
(q_2, q_1)	1	(q_4, q_1)	x
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	x
(q_3, q_2)	0	(q_3, q_2)	x
(q_3, q_2)	1	(q_3, q_3)	x
(q_4, q_1)	0	(q_4, q_2)	x
(q_4, q_1)	1	(q_4, q_1)	x
(q_3, q_3)	0	(q_3, q_2)	x
(q_3, q_3)	1	(q_3, q_1)	x
(q_3, q_1)	0	(q_3, q_2)	x
(q_3, q_1)	1	(q_3, q_1)	x

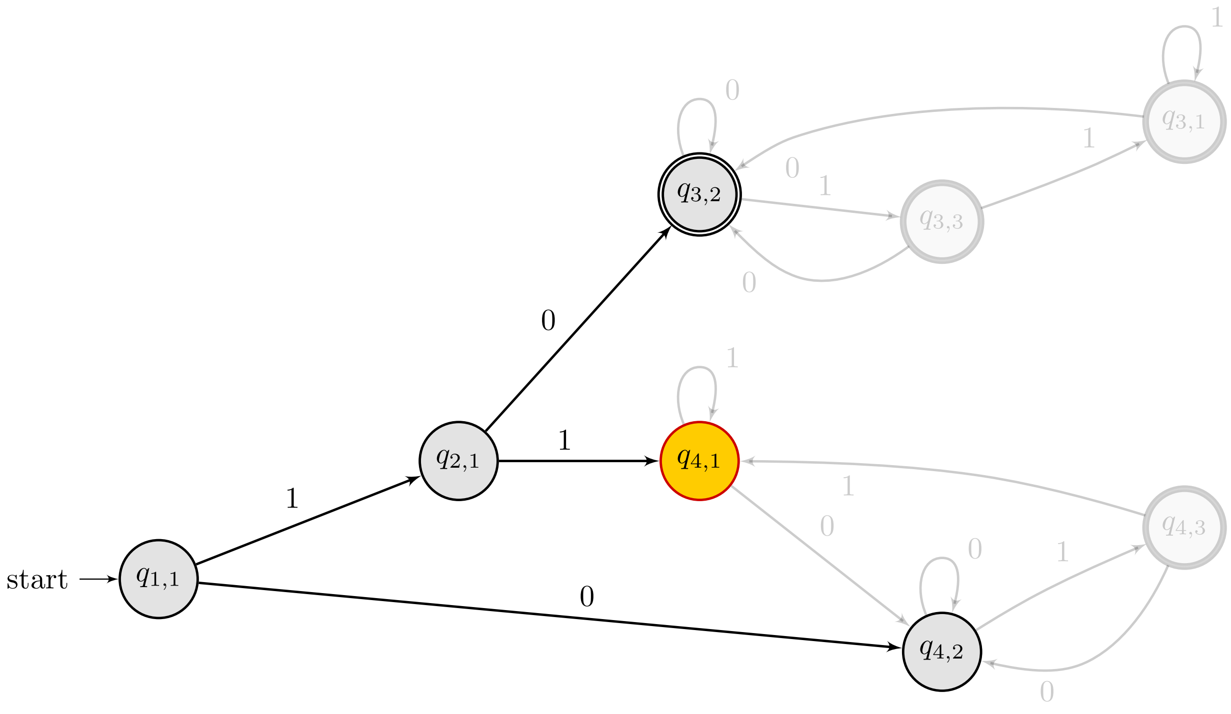
Drawing state diagram



Pick one state without outgoing edges, say $q_{2,1}$, and draw its outgoing edges. Lookup the edges in the transition table.

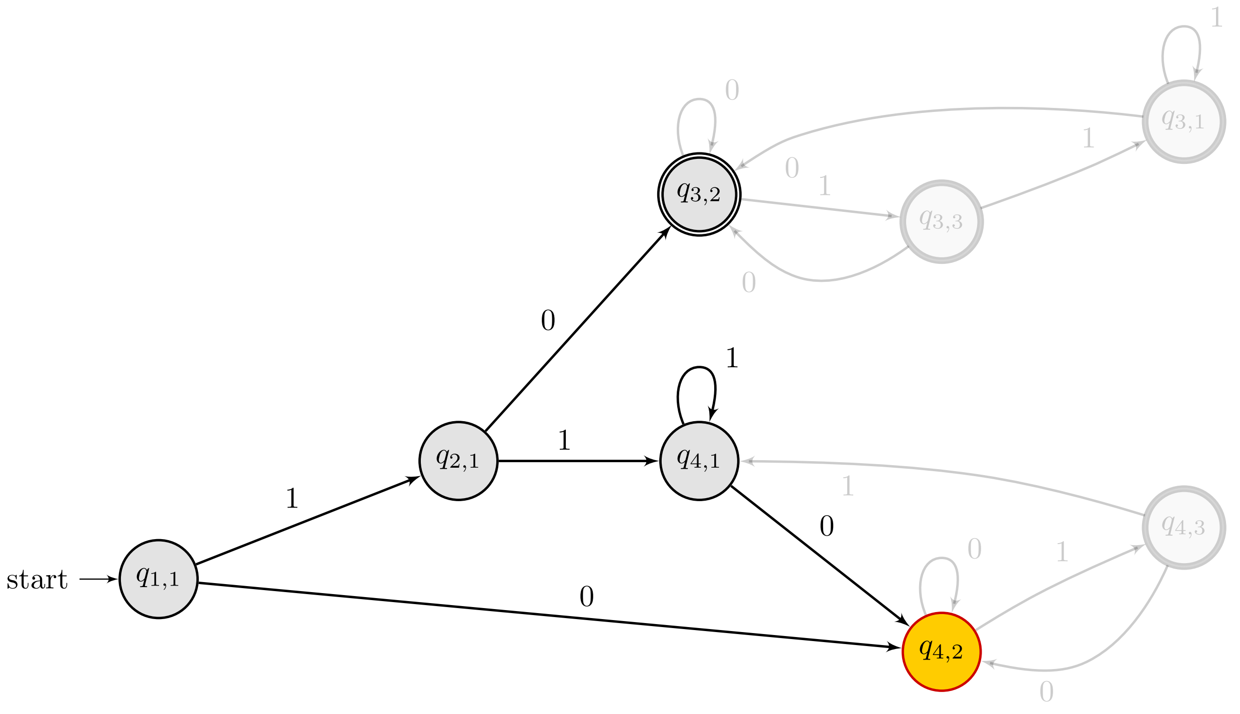
Drawing state diagram

Source	**Edge**	**Target**	**Done**
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	x
(q_2, q_1)	1	(q_4, q_1)	x
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	x
(q_3, q_2)	0	(q_3, q_2)	x
(q_3, q_2)	1	(q_3, q_3)	x
(q_4, q_1)	0	(q_4, q_2)	x
(q_4, q_1)	1	(q_4, q_1)	x
(q_3, q_3)	0	(q_3, q_2)	x
(q_3, q_3)	1	(q_3, q_1)	x
(q_3, q_1)	0	(q_3, q_2)	x
(q_3, q_1)	1	(q_3, q_1)	x



Source	Edge	Target	Done
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	x
(q_2, q_1)	1	(q_4, q_1)	x
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	x
(q_3, q_2)	0	(q_3, q_2)	x
(q_3, q_2)	1	(q_3, q_3)	x
(q_4, q_1)	0	(q_4, q_2)	x
(q_4, q_1)	1	(q_4, q_1)	x
(q_3, q_3)	0	(q_3, q_2)	x
(q_3, q_3)	1	(q_3, q_1)	x
(q_3, q_1)	0	(q_3, q_2)	x
(q_3, q_1)	1	(q_3, q_1)	x

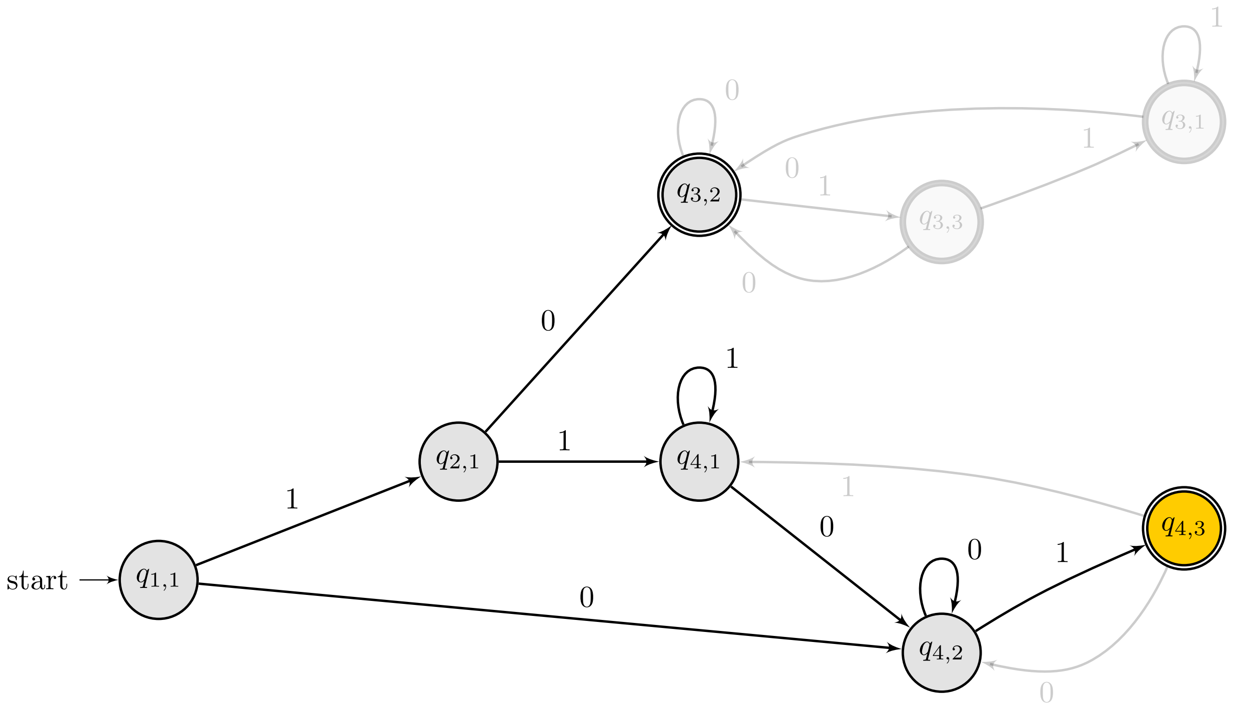
Drawing state diagram



> Pick one state without outgoing edges, say $q_{4,2}$, and draw its outgoing edges. Lookup the edges in the transition table.

Source	Edge	Target	Done
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	x
(q_2, q_1)	1	(q_4, q_1)	x
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	x
(q_3, q_2)	0	(q_3, q_2)	x
(q_3, q_2)	1	(q_3, q_3)	x
(q_4, q_1)	0	(q_4, q_2)	x
(q_4, q_1)	1	(q_4, q_1)	x
(q_3, q_3)	0	(q_3, q_2)	x
(q_3, q_3)	1	(q_3, q_1)	x
(q_3, q_1)	0	(q_3, q_2)	x
(q_3, q_1)	1	(q_3, q_1)	x

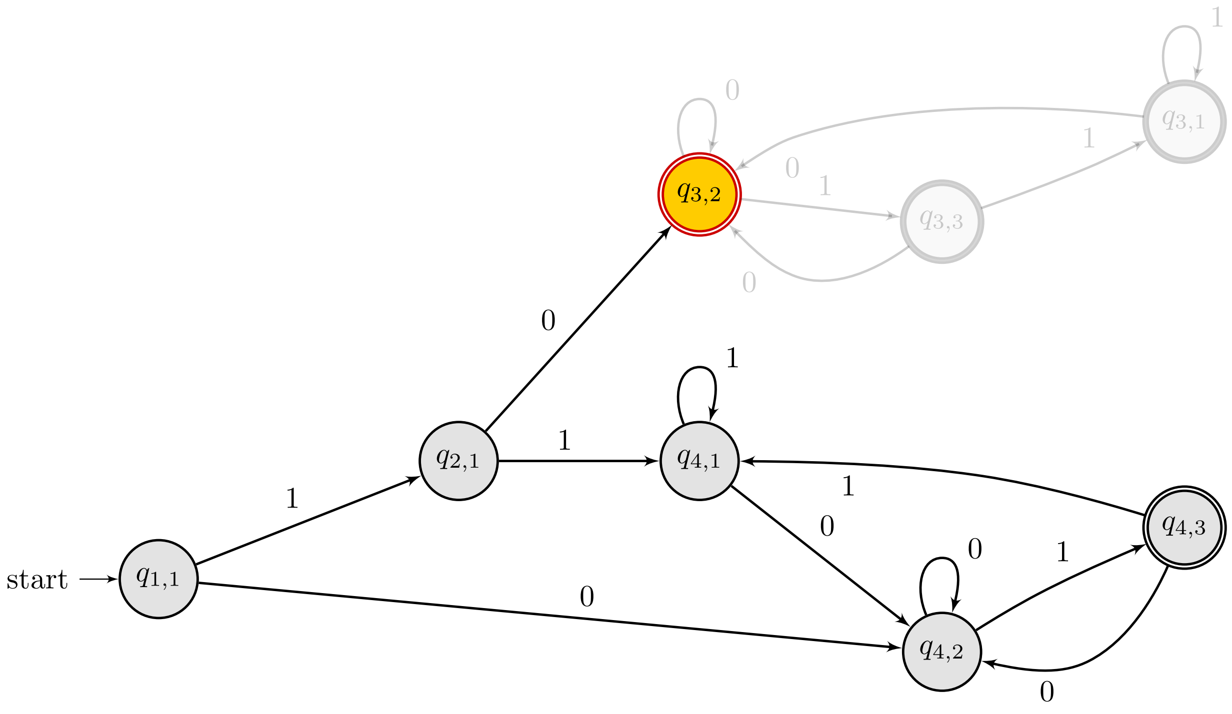
Drawing state diagram



Pick one state without outgoing edges, say $q_{3,2}$, and draw its outgoing edges. Lookup the edges in the transition table.

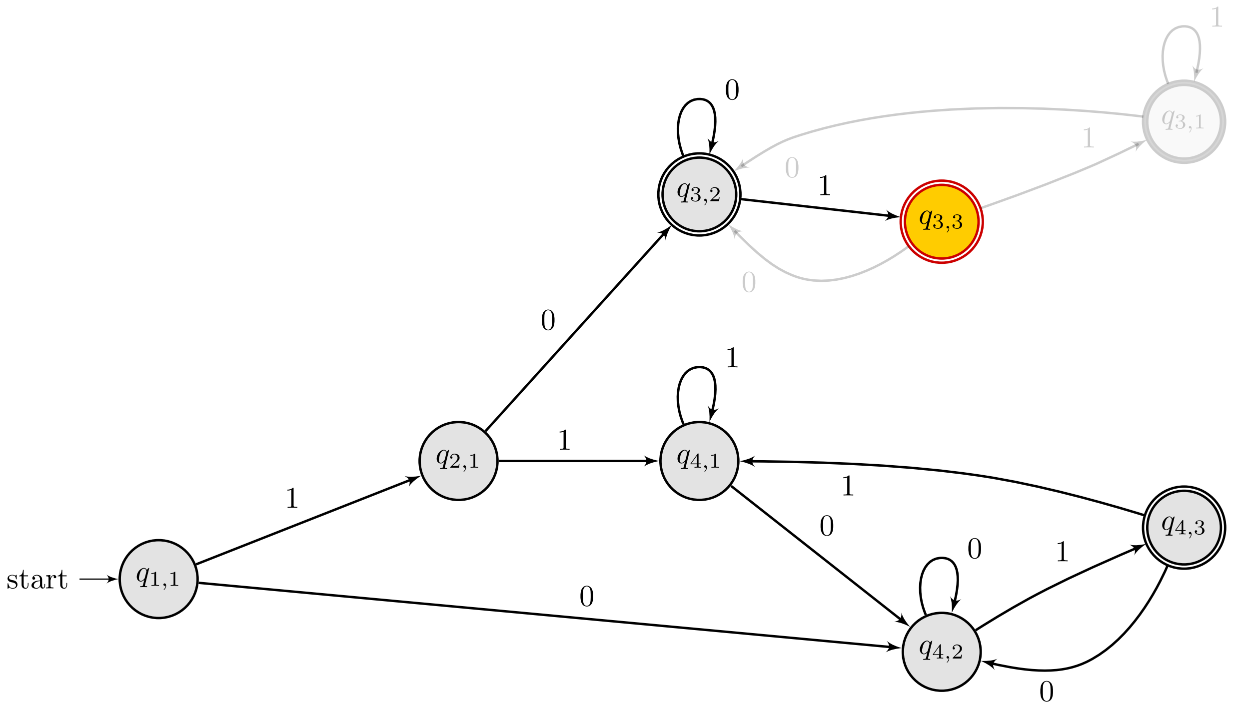
Drawing state diagram

Source	Edge	Target	Done
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	x
(q_2, q_1)	1	(q_4, q_1)	x
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	x
(q_3, q_2)	0	(q_3, q_2)	x
(q_3, q_2)	1	(q_3, q_3)	x
(q_4, q_1)	0	(q_4, q_2)	x
(q_4, q_1)	1	(q_4, q_1)	x
(q_3, q_3)	0	(q_3, q_2)	x
(q_3, q_3)	1	(q_3, q_1)	x
(q_3, q_1)	0	(q_3, q_2)	x
(q_3, q_1)	1	(q_3, q_1)	x



Drawing state diagram

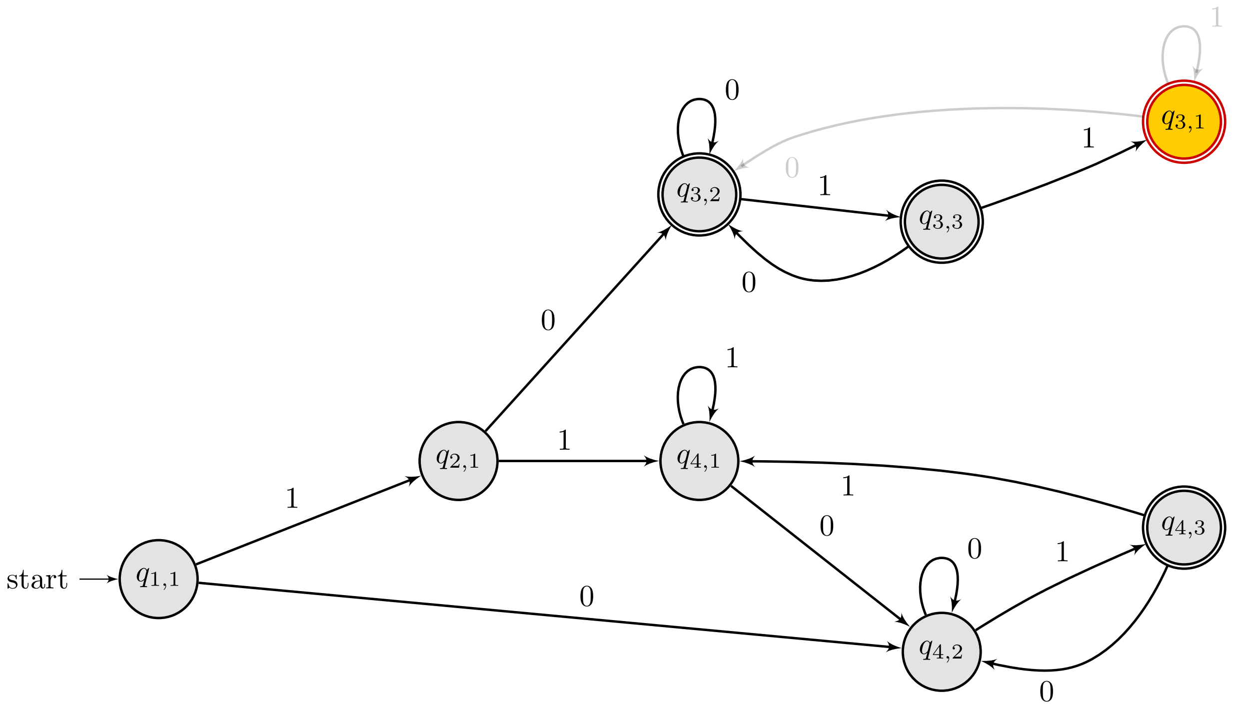
Source	Edge	Target	Done
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	x
(q_2, q_1)	1	(q_4, q_1)	x
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	x
(q_3, q_2)	0	(q_3, q_2)	x
(q_3, q_2)	1	(q_3, q_3)	x
(q_4, q_1)	0	(q_4, q_2)	x
(q_4, q_1)	1	(q_4, q_1)	x
(q_3, q_3)	0	(q_3, q_2)	x
(q_3, q_3)	1	(q_3, q_1)	x
(q_3, q_1)	0	(q_3, q_2)	x
(q_3, q_1)	1	(q_3, q_1)	x



Pick one state without outgoing edges, say $q_{3,3}$, and draw its outgoing edges. Lookup the edges in the transition table.

Source	Edge	Target	Done
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	x
(q_2, q_1)	1	(q_4, q_1)	x
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	x
(q_3, q_2)	0	(q_3, q_2)	x
(q_3, q_2)	1	(q_3, q_3)	x
(q_4, q_1)	0	(q_4, q_2)	x
(q_4, q_1)	1	(q_4, q_1)	x
(q_3, q_3)	0	(q_3, q_2)	x
(q_3, q_3)	1	(q_3, q_1)	x
(q_3, q_1)	0	(q_3, q_2)	x
(q_3, q_1)	1	(q_3, q_1)	x

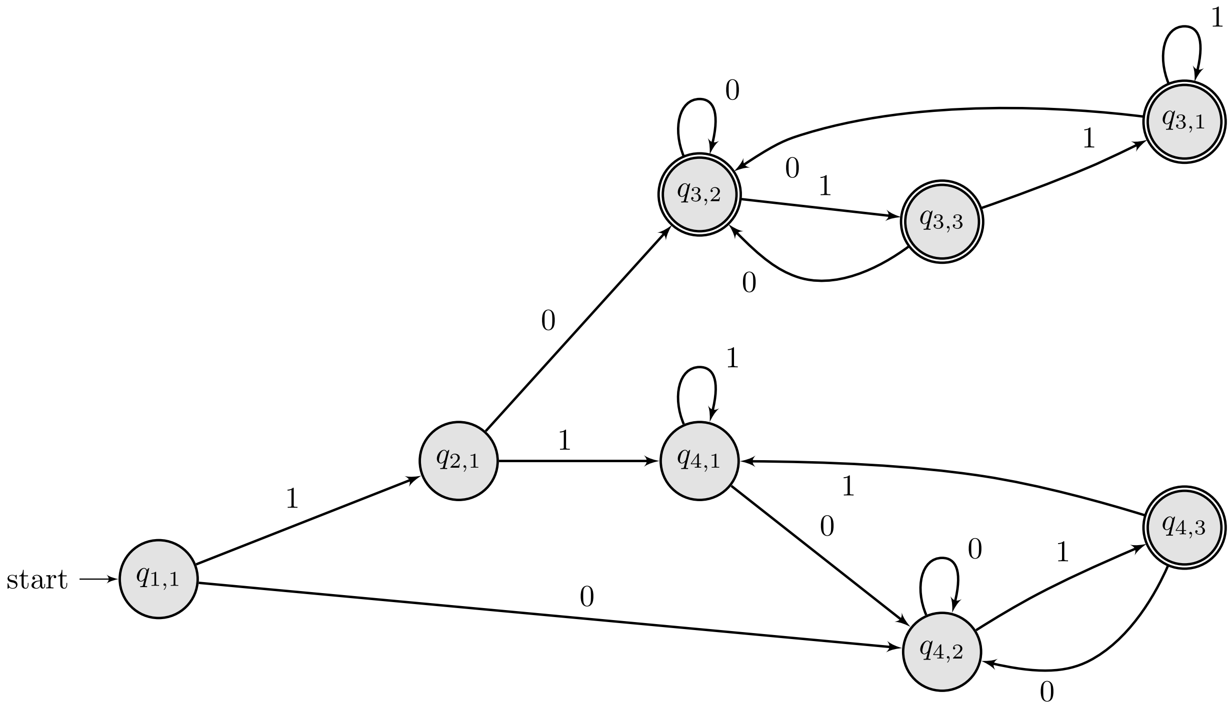
Drawing state diagram



Pick one state without outgoing edges, say $q_{3,1}$, and draw its outgoing edges. Lookup the edges in the transition table.

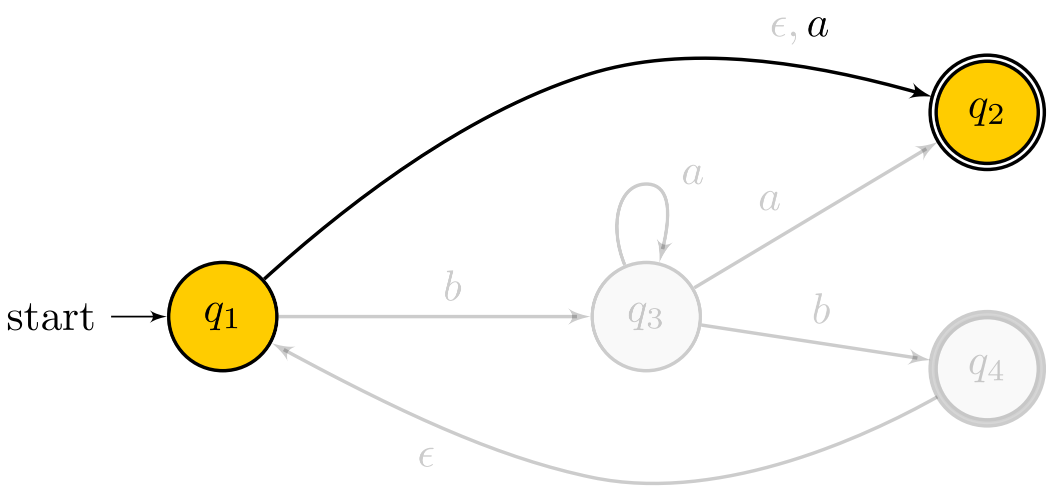
Source	Edge	Target	Done
(q_1, q_1)	0	(q_4, q_2)	x
(q_1, q_1)	1	(q_2, q_1)	x
(q_4, q_2)	0	(q_4, q_2)	x
(q_4, q_2)	1	(q_4, q_3)	x
(q_2, q_1)	0	(q_3, q_2)	x
(q_2, q_1)	1	(q_4, q_1)	x
(q_4, q_3)	0	(q_4, q_2)	x
(q_4, q_3)	1	(q_4, q_1)	x
(q_3, q_2)	0	(q_3, q_2)	x
(q_3, q_2)	1	(q_3, q_3)	x
(q_4, q_1)	0	(q_4, q_2)	x
(q_4, q_1)	1	(q_4, q_1)	x
(q_3, q_3)	0	(q_3, q_2)	x
(q_3, q_3)	1	(q_3, q_1)	x
(q_3, q_1)	0	(q_3, q_2)	x
(q_3, q_1)	1	(q_3, q_1)	x

Drawing state diagram



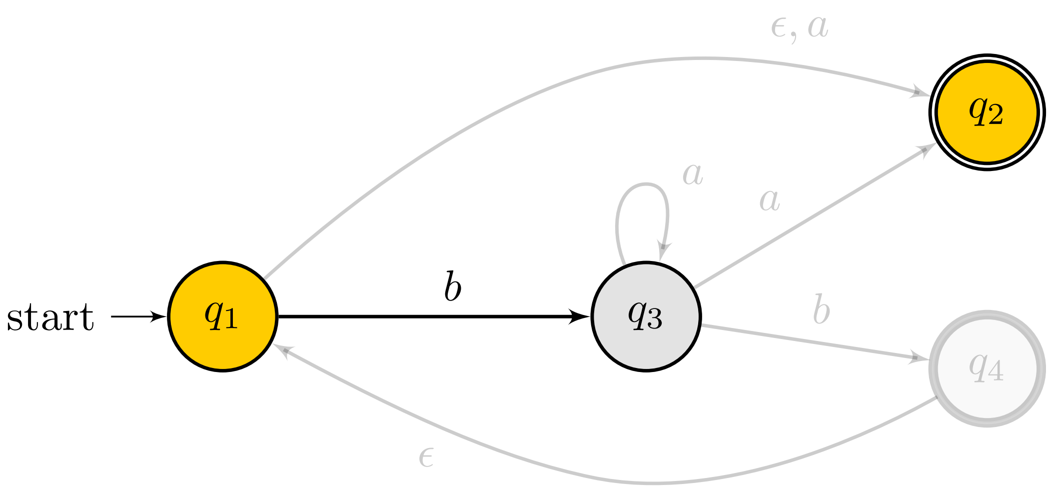
Converting an NFA into a DFA

Converting an NFA into a DFA



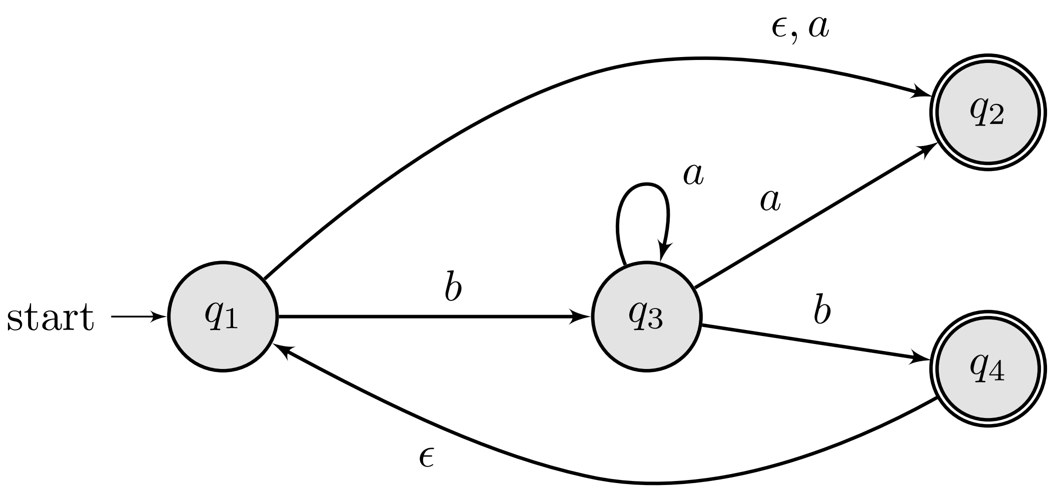
Source	Edge	Target	Done
$\{q_1, q_2\}$	a		
$\{q_1, q_2\}$	b		

Converting an NFA into a DFA



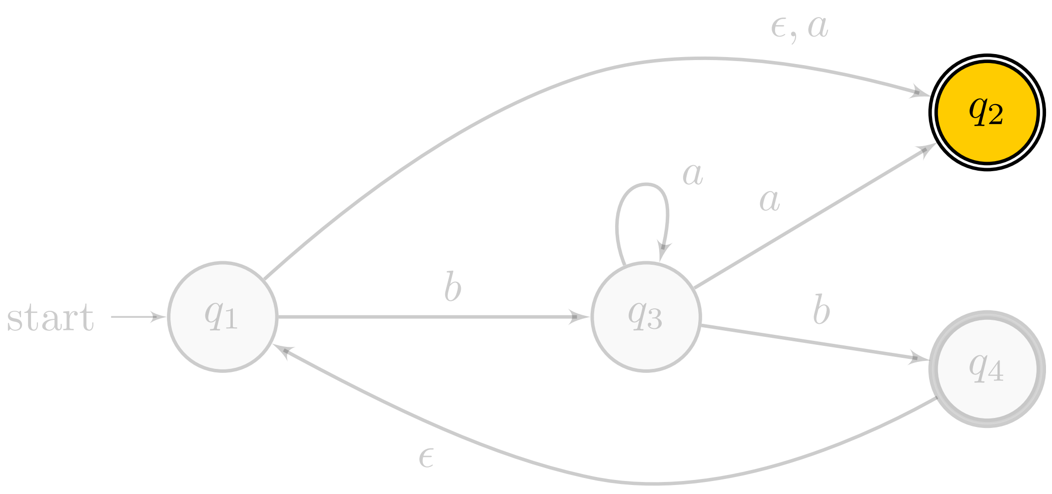
Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	
$\{q_1, q_2\}$	b		

Converting an NFA into a DFA



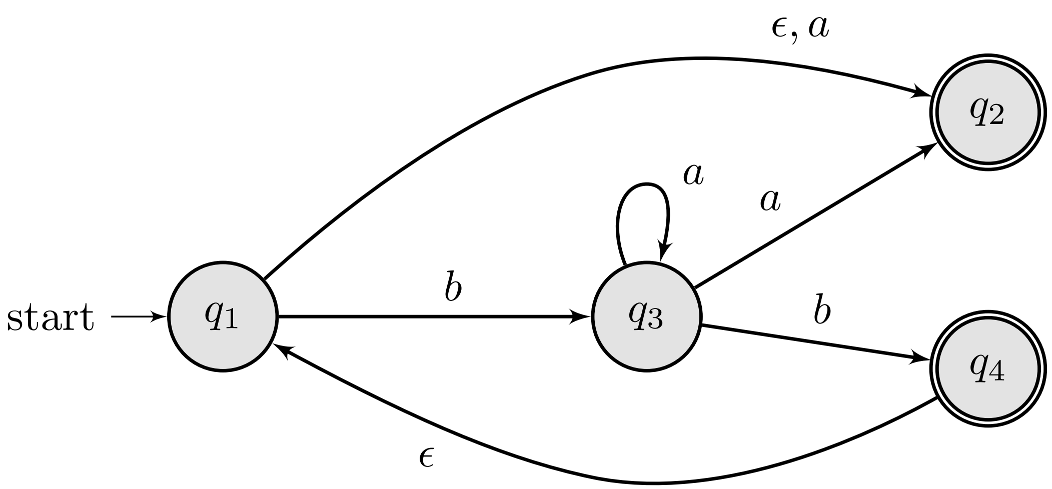
Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	
$\{q_1, q_2\}$	b	$\{q_3\}$	
$\{q_2\}$	a		
$\{q_2\}$	b		
$\{q_3\}$	a		
$\{q_3\}$	b		

Converting an NFA into a DFA



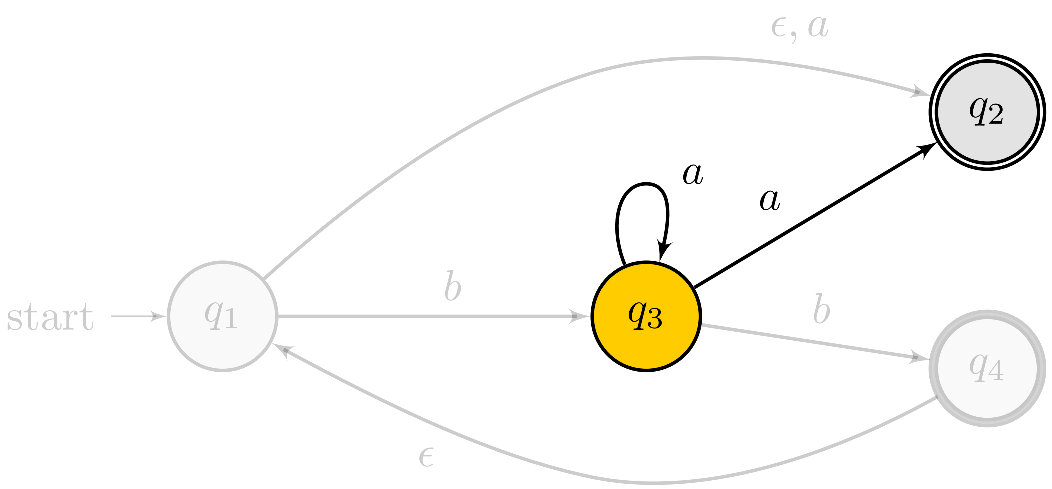
Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	
$\{q_1, q_2\}$	b	$\{q_3\}$	
$\{q_2\}$	a		
$\{q_2\}$	b		
$\{q_3\}$	a		
$\{q_3\}$	b		

Converting an NFA into a DFA



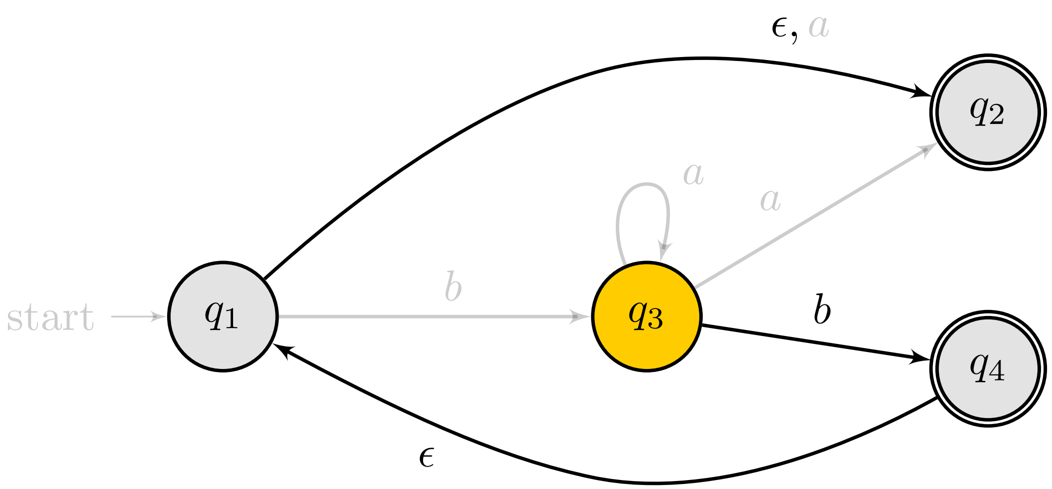
Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	x
$\{q_1, q_2\}$	b	$\{q_3\}$	
$\{q_2\}$	a	$\{\}$	
$\{q_2\}$	b	$\{\}$	
$\{q_3\}$	a		
$\{q_3\}$	b		
$\{\}$	a		
$\{\}$	b		

Converting an NFA into a DFA



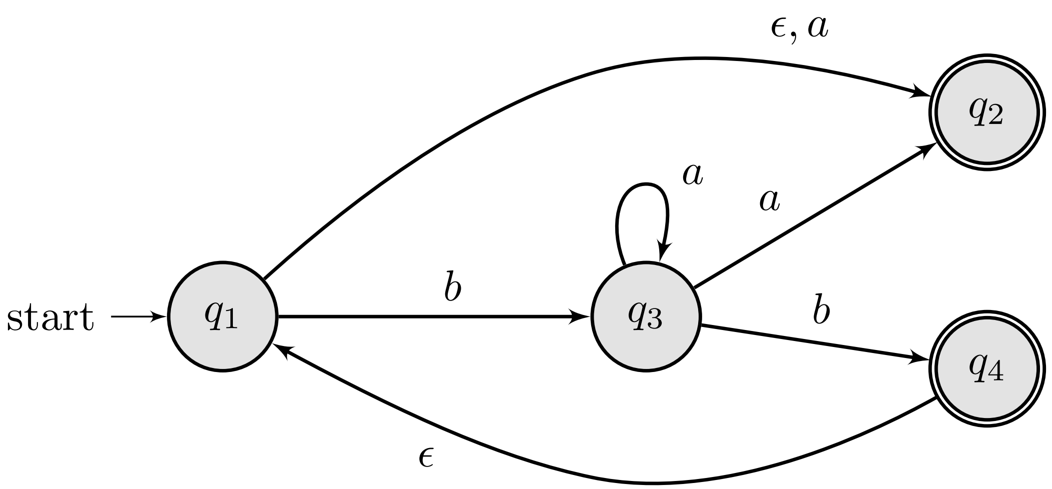
Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	x
$\{q_1, q_2\}$	b	$\{q_3\}$	
$\{q_2\}$	a	$\{\}$	
$\{q_2\}$	b	$\{\}$	
$\{q_3\}$	a		
$\{q_3\}$	b		
$\{\}$	a		
$\{\}$	b		

Converting an NFA into a DFA



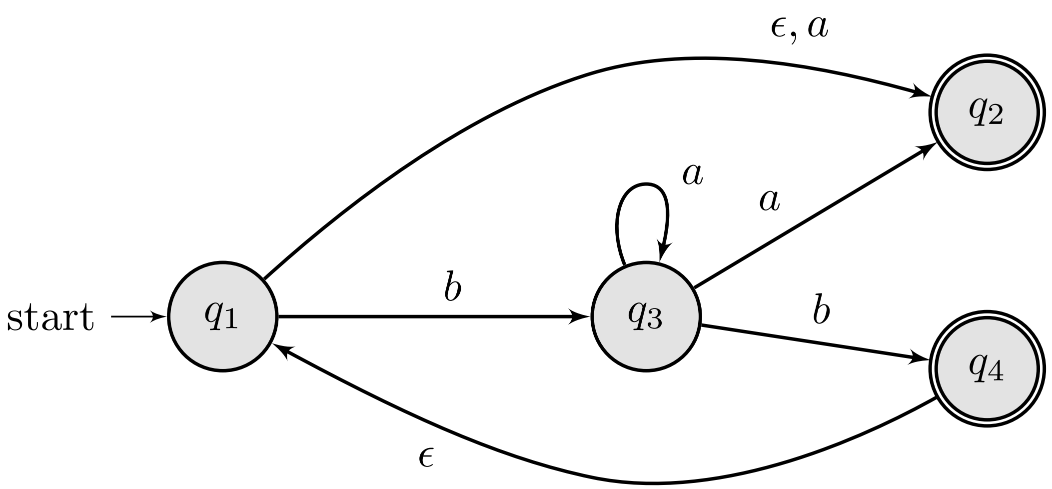
Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	x
$\{q_1, q_2\}$	b	$\{q_3\}$	
$\{q_2\}$	a	$\{\}$	
$\{q_2\}$	b	$\{\}$	
$\{q_3\}$	a		
$\{q_3\}$	b		
$\{\}$	a		
$\{\}$	b		

Converting an NFA into a DFA



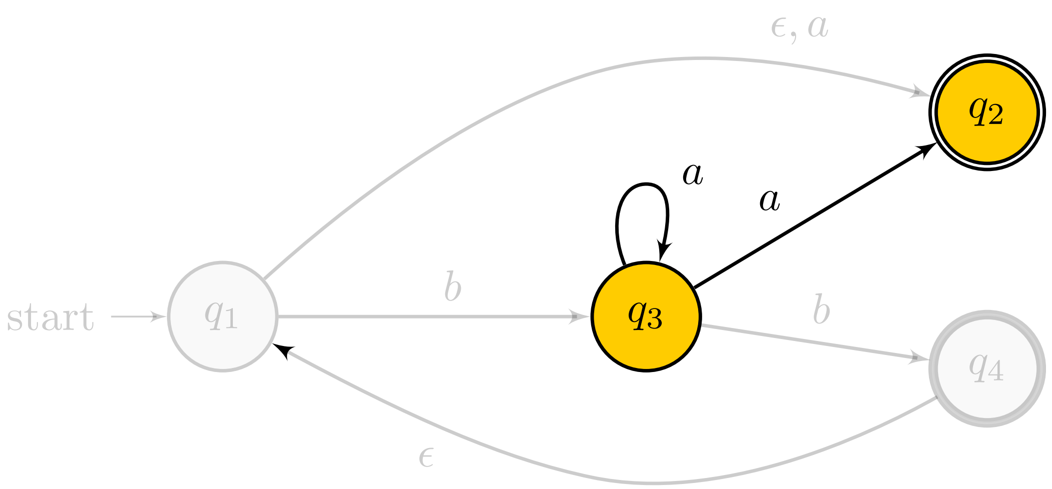
Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	x
$\{q_1, q_2\}$	b	$\{q_3\}$	x
$\{q_2\}$	a	$\{\}$	
$\{q_2\}$	b	$\{\}$	
$\{q_3\}$	a	$\{q_2, q_3\}$	
$\{q_3\}$	b	$\{q_1, q_2, q_4\}$	
$\{\}$	a		
$\{\}$	b		
$\{q_2, q_3\}$	a		
$\{q_2, q_3\}$	b		
$\{q_1, q_2, q_4\}$	a		
$\{q_1, q_2, q_4\}$	b		

Converting an NFA into a DFA



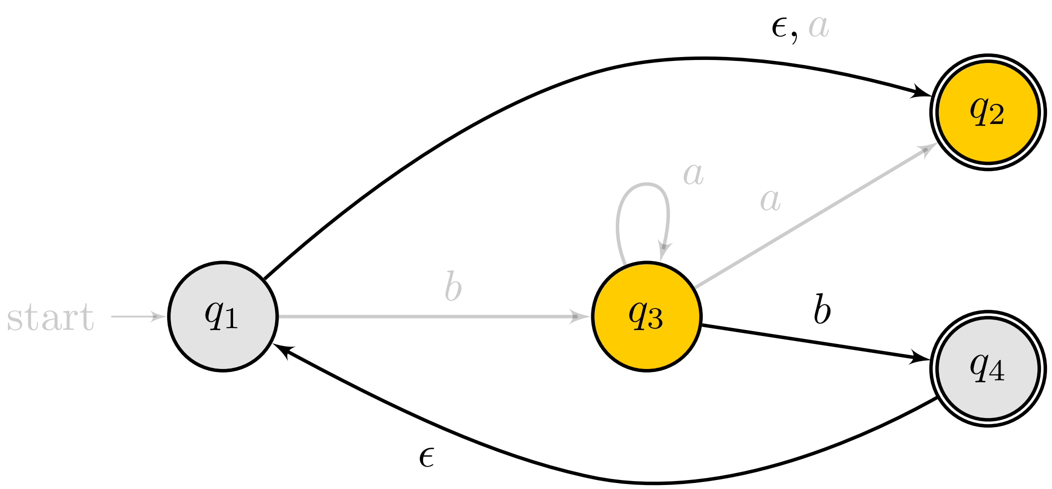
Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	x
$\{q_1, q_2\}$	b	$\{q_3\}$	x
$\{q_2\}$	a	$\{\}$	
$\{q_2\}$	b	$\{\}$	
$\{q_3\}$	a	$\{q_2, q_3\}$	
$\{q_3\}$	b	$\{q_1, q_2, q_4\}$	
$\{\}$	a	$\{\}$	x
$\{\}$	b	$\{\}$	x
$\{q_2, q_3\}$	a		
$\{q_2, q_3\}$	b		
$\{q_1, q_2, q_4\}$	a		
$\{q_1, q_2, q_4\}$	b		

Converting an NFA into a DFA



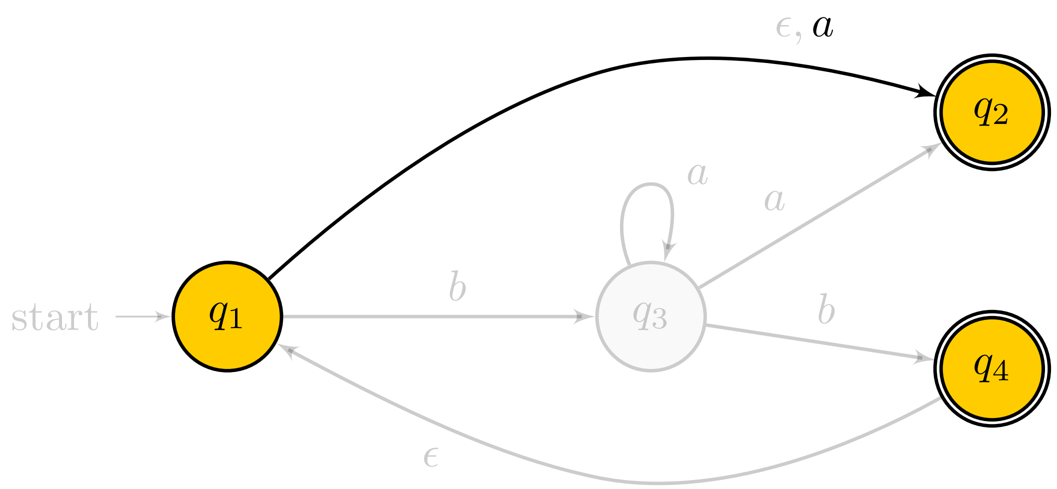
Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	x
$\{q_1, q_2\}$	b	$\{q_3\}$	x
$\{q_2\}$	a	$\{\}$	
$\{q_2\}$	b	$\{\}$	
$\{q_3\}$	a	$\{q_2, q_3\}$	
$\{q_3\}$	b	$\{q_1, q_2, q_4\}$	
$\{\}$	a	$\{\}$	x
$\{\}$	b	$\{\}$	x
$\{q_2, q_3\}$	a		
$\{q_2, q_3\}$	b		
$\{q_1, q_2, q_4\}$	a		
$\{q_1, q_2, q_4\}$	b		

Converting an NFA into a DFA



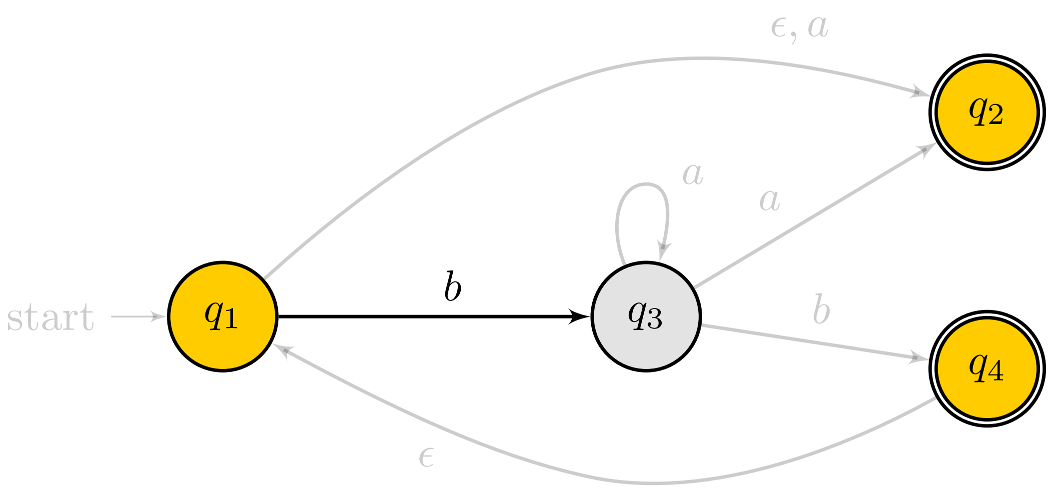
Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	x
$\{q_1, q_2\}$	b	$\{q_3\}$	x
$\{q_2\}$	a	$\{\}$	
$\{q_2\}$	b	$\{\}$	
$\{q_3\}$	a	$\{q_2, q_3\}$	
$\{q_3\}$	b	$\{q_1, q_2, q_4\}$	
$\{\}$	a	$\{\}$	x
$\{\}$	b	$\{\}$	x
$\{q_2, q_3\}$	a		
$\{q_2, q_3\}$	b		
$\{q_1, q_2, q_4\}$	a		
$\{q_1, q_2, q_4\}$	b		

Converting an NFA into a DFA



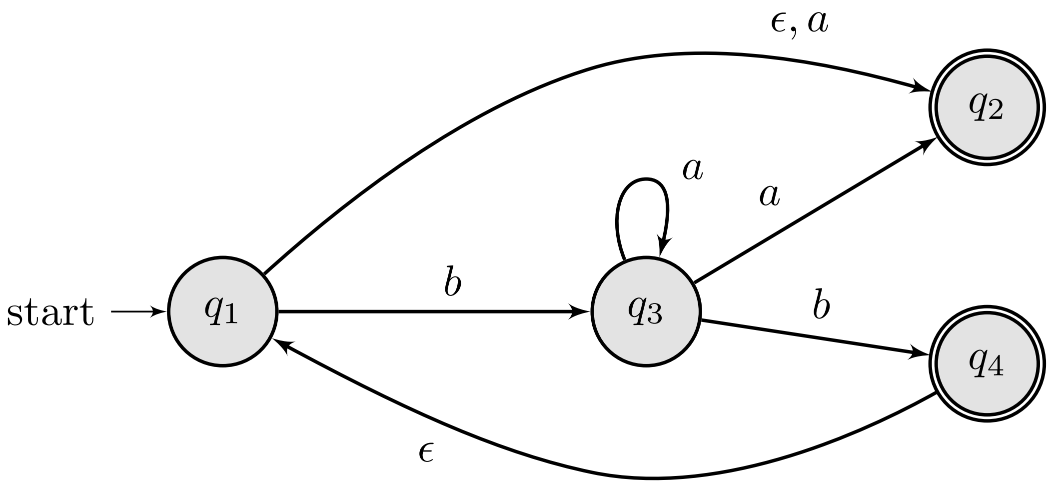
Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	x
$\{q_1, q_2\}$	b	$\{q_3\}$	x
$\{q_2\}$	a	$\{\}$	x
$\{q_2\}$	b	$\{\}$	x
$\{q_3\}$	a	$\{q_2, q_3\}$	x
$\{q_3\}$	b	$\{q_1, q_2, q_4\}$	
$\{\}$	a	$\{\}$	x
$\{\}$	b	$\{\}$	x
$\{q_2, q_3\}$	a	$\{q_2, q_3\}$	x
$\{q_2, q_3\}$	b	$\{q_1, q_2, q_4\}$	
$\{q_1, q_2, q_4\}$	a		
$\{q_1, q_2, q_4\}$	b		

Converting an NFA into a DFA



Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	x
$\{q_1, q_2\}$	b	$\{q_3\}$	x
$\{q_2\}$	a	$\{\}$	x
$\{q_2\}$	b	$\{\}$	x
$\{q_3\}$	a	$\{q_2, q_3\}$	x
$\{q_3\}$	b	$\{q_1, q_2, q_4\}$	
$\{\}$	a	$\{\}$	x
$\{\}$	b	$\{\}$	x
$\{q_2, q_3\}$	a	$\{q_2, q_3\}$	x
$\{q_2, q_3\}$	b	$\{q_1, q_2, q_4\}$	
$\{q_1, q_2, q_4\}$	a		
$\{q_1, q_2, q_4\}$	b		

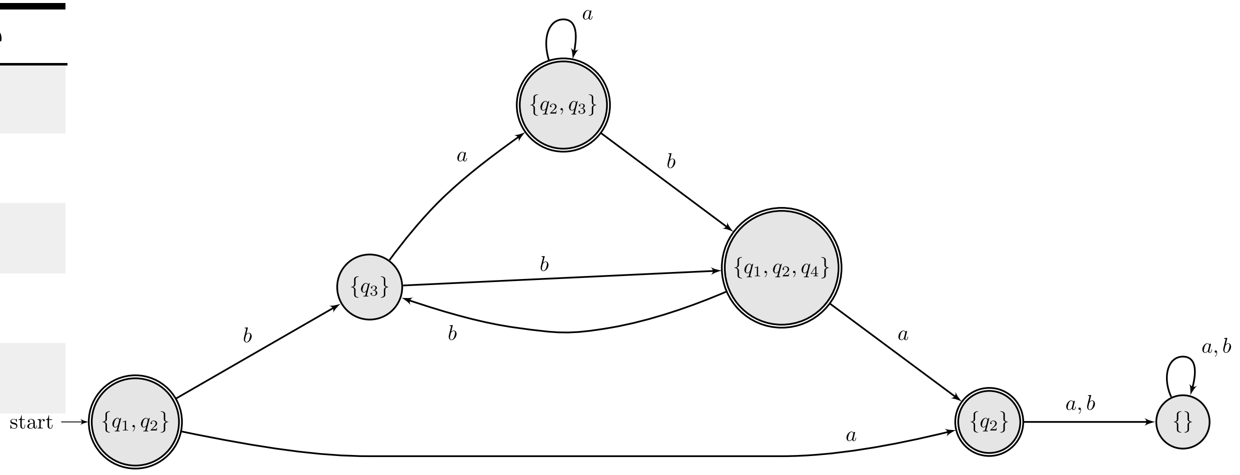
Converting an NFA into a DFA



Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	x
$\{q_1, q_2\}$	b	$\{q_3\}$	x
$\{q_2\}$	a	$\{\}$	x
$\{q_2\}$	b	$\{\}$	x
$\{q_3\}$	a	$\{q_2, q_3\}$	x
$\{q_3\}$	b	$\{q_1, q_2, q_4\}$	x
$\{\}$	a	$\{\}$	x
$\{\}$	b	$\{\}$	x
$\{q_2, q_3\}$	a	$\{q_2, q_3\}$	x
$\{q_2, q_3\}$	b	$\{q_1, q_2, q_4\}$	x
$\{q_1, q_2, q_4\}$	a	$\{q_2\}$	x
$\{q_1, q_2, q_4\}$	b	$\{q_3\}$	x

Converting an NFA into a DFA

Source	Edge	Target	Done
$\{q_1, q_2\}$	a	$\{q_2\}$	x
$\{q_1, q_2\}$	b	$\{q_3\}$	x
$\{q_2\}$	a	$\{\}$	x
$\{q_2\}$	b	$\{\}$	x
$\{q_3\}$	a	$\{q_2, q_3\}$	x
$\{q_3\}$	b	$\{q_1, q_2, q_4\}$	x
$\{\}$	a	$\{\}$	x
$\{\}$	b	$\{\}$	x
$\{q_2, q_3\}$	a	$\{q_2, q_3\}$	x
$\{q_2, q_3\}$	b	$\{q_1, q_2, q_4\}$	x
$\{q_1, q_2, q_4\}$	a	$\{q_2\}$	x
$\{q_1, q_2, q_4\}$	b	$\{q_3\}$	x

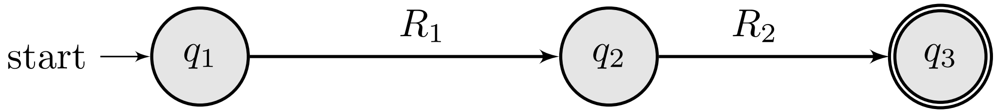


Converting an NFA into a REGEX

Converting an NFA into a REGEX

The simplest example.

Before

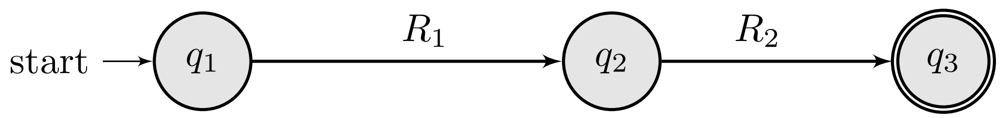


After

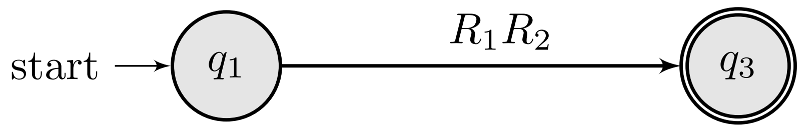
Converting an NFA into a REGEX

- The simplest example.

Before



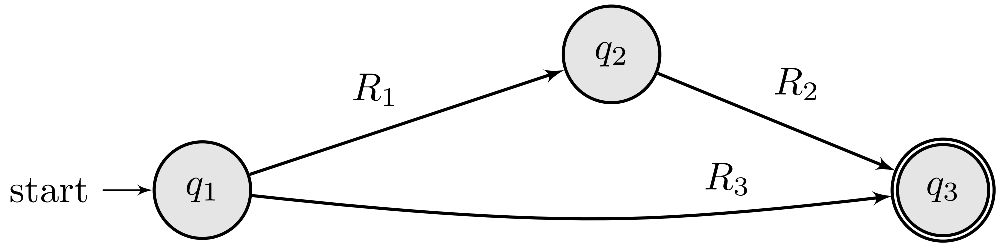
After



Converting an NFA into a REGEX

When there are existing edges, the two overlapping edges are joined with $+$.

Before

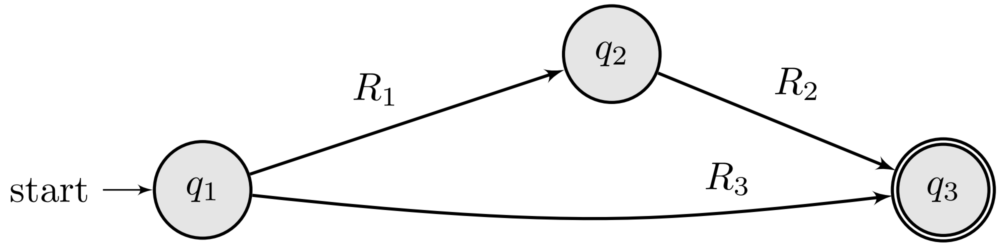


After

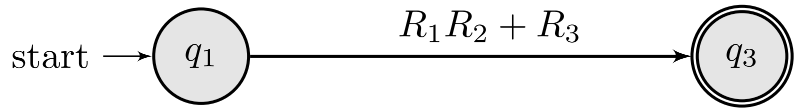
Converting an NFA into a REGEX

- When there are existing edges, the two overlapping edges are joined with $+$.

Before



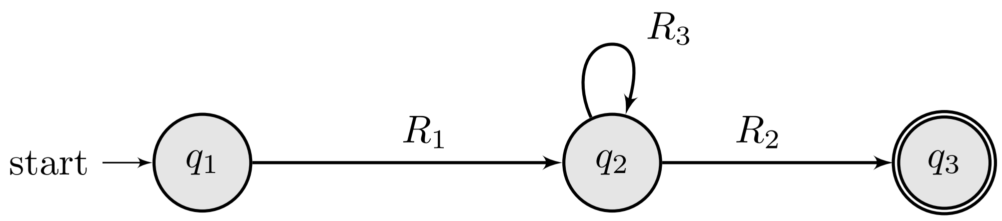
After



Converting an NFA into a REGEX

When there is a self loop we convert it to *.

Before

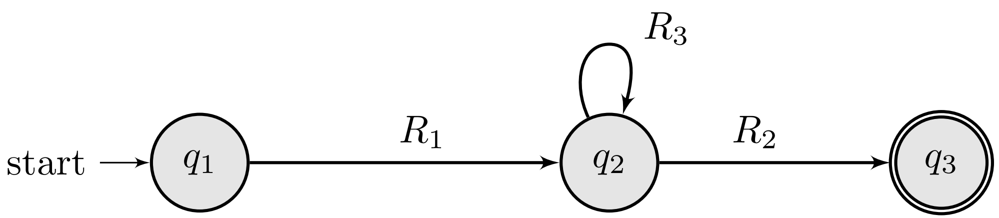


After

Converting an NFA into a REGEX

When there is a self loop we convert it to *.

Before

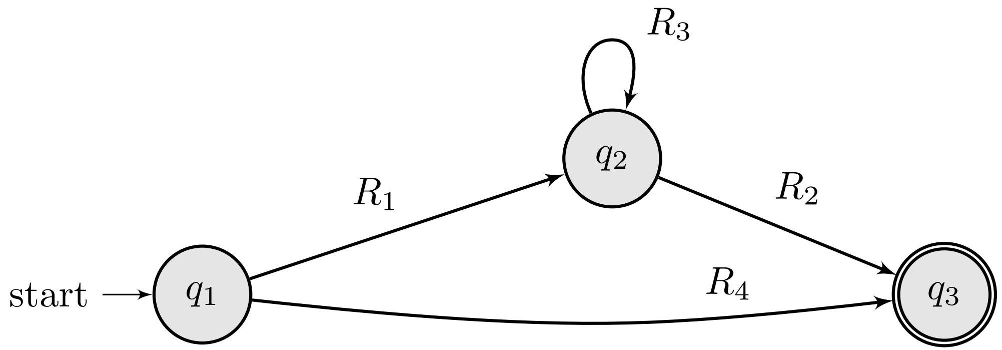


After

Converting an NFA into a REGEX

When there are existing edges, the two overlapping edges are joined with $+$.

Before

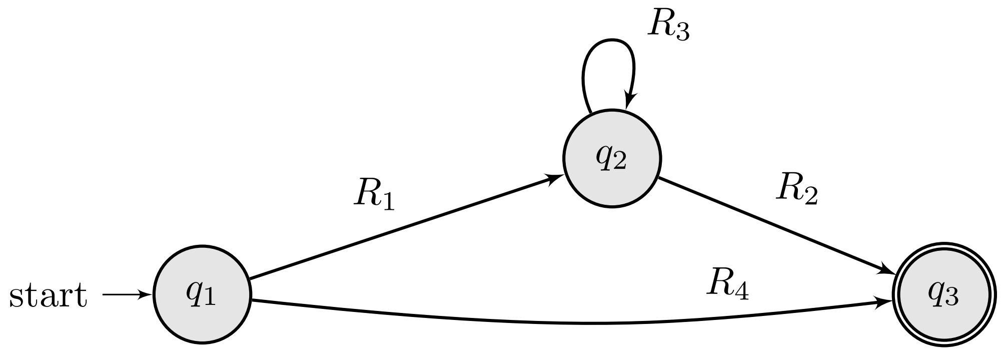


After

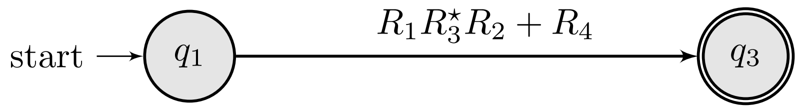
Converting an NFA into a REGEX

When there are existing edges, the two overlapping edges are joined with $+$.

Before



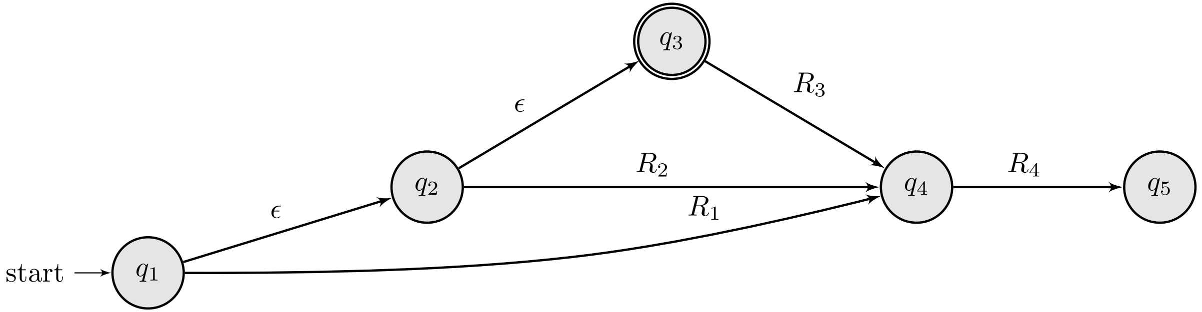
After



Converting an NFA into a REGEX

Every incoming state becomes connected to every outgoing state.

Before

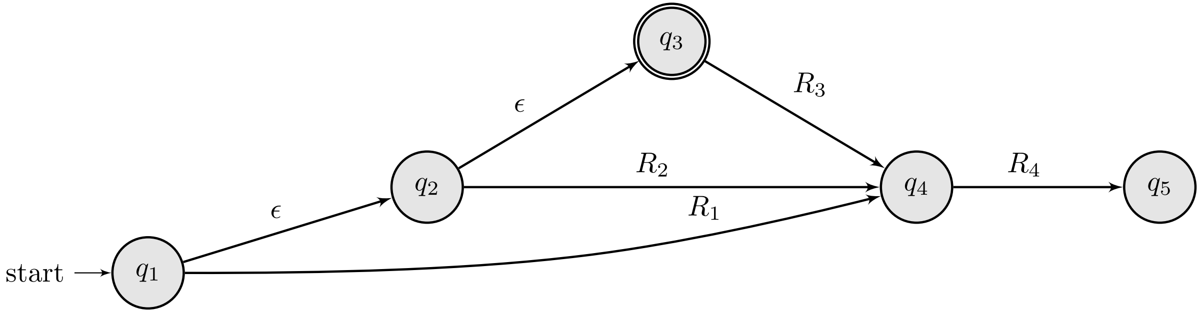


After

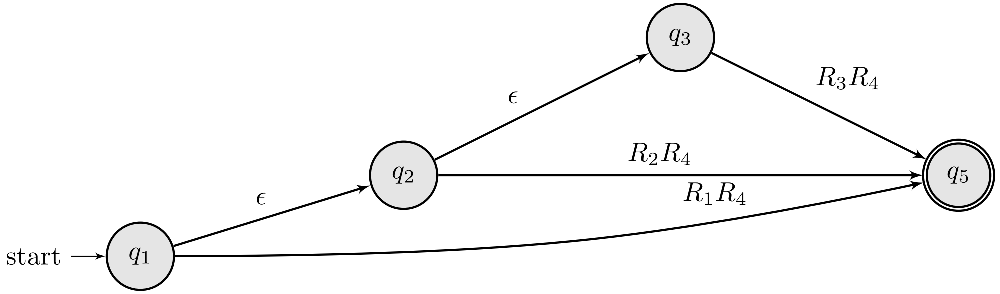
Converting an NFA into a REGEX

Every incoming state becomes connected to every outgoing state.

Before



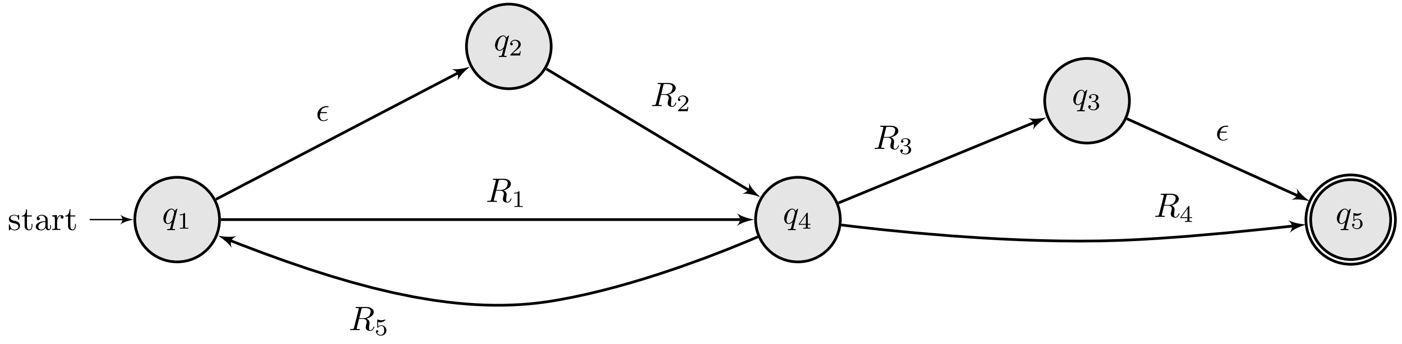
After



Converting an NFA into a REGEX

Every incoming state becomes connected to every outgoing state.

Before

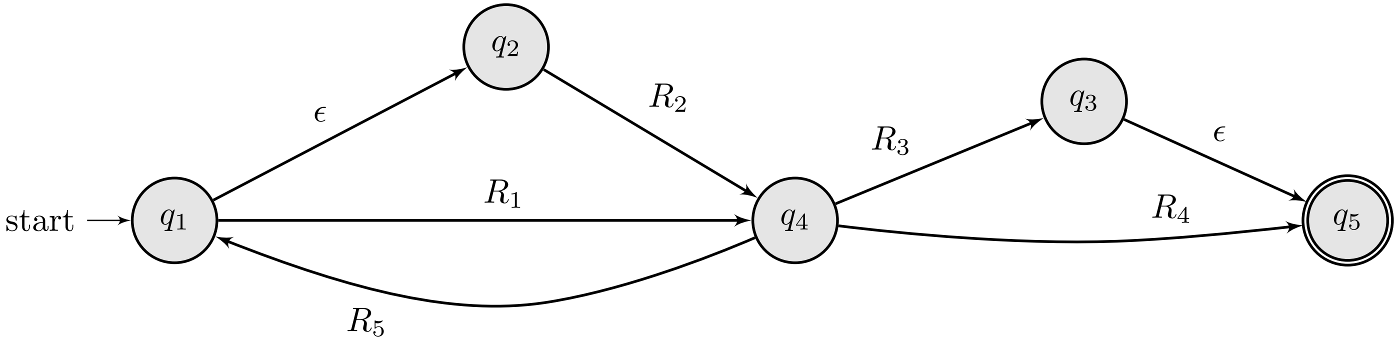


After

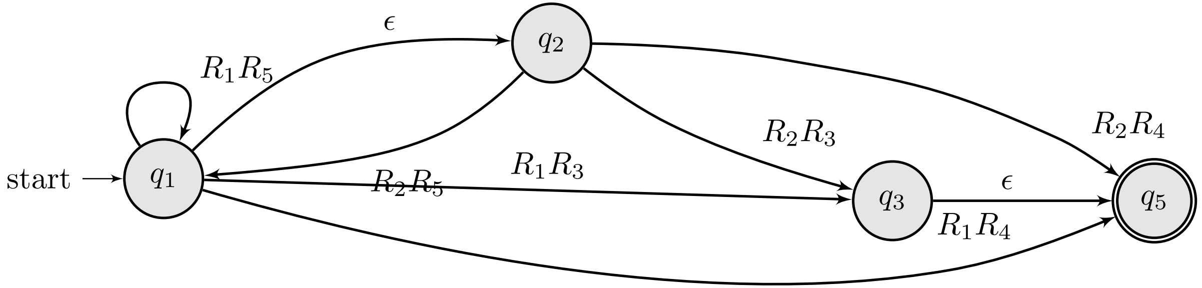
Converting an NFA into a REGEX

- Every incoming state becomes connected to every outgoing state.

Before



After



Pumping lemma for regular languages

Any proofs showing that a language is not regular should show:

- Assumption 1: p is the pumping length
- Goal 1: $w \in L$
- Goal 2: $|w| \leq p$
- Assumption 2: $w = xyz$
- Assumption 3: $|xy| \leq p$
- Assumption 4: $|y| > 0$
- Goal 3: $\exists i, xy^i z \notin L$

Theorem $L_1 = \{0^n 1^n \mid \forall n \colon n \geq 0\}$ is not regular.

We prove that the language above does not satisfy the pumping property, thus the language is not regular.

Let p be the pumping length, we pick string $w = 0^p 1^p$.

We must show that

- **(Goal 1)** $w \in 0^n 1^n \mid \forall n \colon n \geq 0$ **Proof:** holds by replacing n by p.
- **(Goal 2)** $|w| \geq p$ **Proof:** holds since $|w| = 2p \geq p$.

Theorem $L_1 = \{0^n 1^n \mid \forall n\colon n \geq 0\}$ is not regular.

Given some x, y, z, our assumptions are:

- $H_1\colon w = xyz$
- $H_2\colon |xy| \leq p$
- $H_3\colon |y| > 0$

We must show **(Goal 3)** that

$$\exists i, xy^i z \notin L_1$$

Assumptions

- H_1: $w = xyz$
- H_2: $|xy| \leq p$
- H_3: $|y| > 0$

Goals

$\exists i, xy^i z \notin L_1$

Recall that (H_2) $|xy| \leq p$, thus let $a + b = p$ and $a = |xy|$. We can rewrite assumption (H_1) $w = xyz$ such that, since for any w, n, and m we have that $w^{n+m} = w^n w^m$

$$(H_1) \quad w = \underbrace{0^p 1^p}_{xyz} = \underbrace{0^a}_{xy}\underbrace{0^b 1^{a+b}}_{z}$$

Assumptions

- H_1: $w = xyz$
- H_2: $|xy| \leq p$
- H_3: $|y| > 0$

Goals

$\exists i, xy^i z \notin L_1$

Recall that (H_2) $|xy| \leq p$, thus let $a + b = p$ and $a = |xy|$. We can rewrite assumption (H_1) $w = xyz$ such that, since for any w, n, and m we have that $w^{n+m} = w^n w^m$

$$(H_1) \quad w = \underbrace{0^p 1^p}_{xyz} = \underbrace{0^a}_{xy} \underbrace{0^b 1^{a+b}}_{z}$$

Or, simply,

$$(H_1) \quad \underbrace{0^a}_{xy} \underbrace{0^b 1^{a+b}}_{z} = \underbrace{0^{|xy|}}_{xy} \underbrace{0^b 1^{|xy|+b}}_{z}$$

Assumptions

- H_1: $w = xyz$
- H_2: $|xy| \leq p$
- H_3: $|y| > 0$

Goals

$\exists i, xy^i z \notin L_1$

Recall that (H_2) $|xy| \leq p$, thus let $a + b = p$ and $a = |xy|$.

We can rewrite assumption (H_1) $w = xyz$ such that, since for any w, n, and m we have that $w^{n+m} = w^n w^m$

$$(H_1) \quad w = \underbrace{0^p 1^p}_{xyz} = \underbrace{0^a}_{xy} \underbrace{0^b 1^{a+b}}_{z}$$

Or, simply,

$$(H_1) \quad \underbrace{0^a}_{xy} \underbrace{0^b 1^{a+b}}_{z} = \underbrace{0^{|xy|}}_{xy} \underbrace{0^b 1^{|xy|+b}}_{z}$$

We note that

$$xy^2 z = \underbrace{0^{|xy|}}_{xy} \underbrace{0^{|y|}}_{y} \underbrace{0^b 1^{|xy|+b}}_{z} = 0^{|xyy|+b} 1^{|xy|+b}$$

Assumptions

- H_1: $w = xyz$
- H_2: $|xy| \leq p$
- H_3: $|y| > 0$

Goals

$$0^{|xyy|+b}1^{|xy|+b} \notin L_1$$

We restate our goal (Goal 3)

$$\exists i, xy^i z \notin L_1$$

Assumptions

- $H_1\colon w = xyz$
- $H_2\colon |xy| \leq p$
- $H_3\colon |y| > 0$

Goals

$$0^{|xyy|+b}1^{|xy|+b} \notin L_1$$

We restate our goal (Goal 3)

$$\exists i, xy^i z \notin L_1$$

We pick $i = 2$, so our goal is to show that

$$0^{|xyy|+b}1^{|xy|+b} \notin \{0^n 1^n \mid \forall n\colon n \geq 0\}$$

Assumptions

- $H_1 : w = xyz$
- $H_2 : |xy| \leq p$
- $H_3 : |y| > 0$

Goals

$0^{|xyy|+b} 1^{|xy|+b} \notin L_1$

We restate our goal (Goal 3)

$$\exists i, xy^i z \notin L_1$$

We pick $i = 2$, so our goal is to show that

$$0^{|xyy|+b} 1^{|xy|+b} \notin \{0^n 1^n \mid \forall n \colon n \geq 0\}$$

By unfolding the definition of $\notin$ we have

$$\neg\left(0^{|xyy|+b} 1^{|xy|+b} \in \{0^n 1^n \mid \forall n \colon n \geq 0\}\right)$$

Assumptions

- H_1: $w = xyz$
- H_2: $|xy| \leq p$
- H_3: $|y| > 0$

Goals

$0^{|xyy|+b}1^{|xy|+b} \notin L_1$

We restate our goal (Goal 3)

$$\exists i, xy^i z \notin L_1$$

We pick $i = 2$, so our goal is to show that

$$0^{|xyy|+b}1^{|xy|+b} \notin \{0^n 1^n \mid \forall n \colon n \geq 0\}$$

By unfolding the definition of $\notin$ we have

$$\neg(0^{|xyy|+b}1^{|xy|+b} \in \{0^n 1^n \mid \forall n \colon n \geq 0\})$$

We apply the definition of set membership:

$$\neg(0^{|xyy|+b}1^{|xy|+b} = 0^n 1^n)$$

(Continuation… restate Goal 3)

Assumptions

- $H_1 \colon w = xyz$
- $H_2 \colon |xy| \le p$
- $H_3 \colon |y| > 0$

Goals

$0^{|xyy|+b} 1^{|xy|+b} \notin L_1$

$$\neg\left(0^{|xyy|+b} 1^{|xy|+b} = 0^n 1^n\right)$$

Assumptions

- H_1: $w = xyz$
- H_2: $|xy| \leq p$
- H_3: $|y| > 0$

Goals

$0^{|xyy|+b} 1^{|xy|+b} \notin L_1$

(Continuation... restate Goal 3)

$$\neg\left(0^{|xyy|+b} 1^{|xy|+b} = 0^n 1^n\right)$$

We isolate each exponent:

$$\neg\left(|xyy| + b = n \wedge |xy| + b = n\right)$$

Assumptions

- H_1: $w = xyz$
- H_2: $|xy| \leq p$
- H_3: $|y| > 0$

Goals

$0^{|xyy|+b}1^{|xy|+b} \notin L_1$

(Continuation... restate Goal 3)

$$\neg(0^{|xyy|+b}1^{|xy|+b} = 0^n 1^n)$$

We isolate each exponent:

$$\neg(|xyy| + b = n \wedge |xy| + b = n)$$

We replace the left-hand side on the right-hand side of $\wedge$.

$$\neg(|xyy| + b = |xy| + b)$$

Assumptions

- $H_1 \colon w = xyz$
- $H_2 \colon |xy| \leq p$
- $H_3 \colon |y| > 0$

Goals

$0^{|xyy|+b}1^{|xy|+b} \notin L_1$

(Continuation... restate Goal 3)

$$\neg(0^{|xyy|+b}1^{|xy|+b} = 0^n 1^n)$$

We isolate each exponent:

$$\neg(|xyy| + b = n \land |xy| + b = n)$$

We replace the left-hand side on the right-hand side of $\land$.

$$\neg(|xyy| + b = |xy| + b)$$

We now apply the negation operator and simplify the equation:

$$|xyy| + b \neq |xy| + b \iff |y| \neq 0$$

Assumptions

- $H_1 \colon w = xyz$
- $H_2 \colon |xy| \leq p$
- $H_3 \colon |y| > 0$

Goals

$0^{|xyy|+b}1^{|xy|+b} \notin L_1$

(Continuation... restate Goal 3)

$$\neg(0^{|xyy|+b}1^{|xy|+b} = 0^n 1^n)$$

We isolate each exponent:

$$\neg(|xyy| + b = n \wedge |xy| + b = n)$$

We replace the left-hand side on the right-hand side of $\wedge$.

$$\neg(|xyy| + b = |xy| + b)$$

We now apply the negation operator and simplify the equation:

$$|xyy| + b \neq |xy| + b \iff |y| \neq 0$$

Which holds since (H_3) $|y| > 0$.

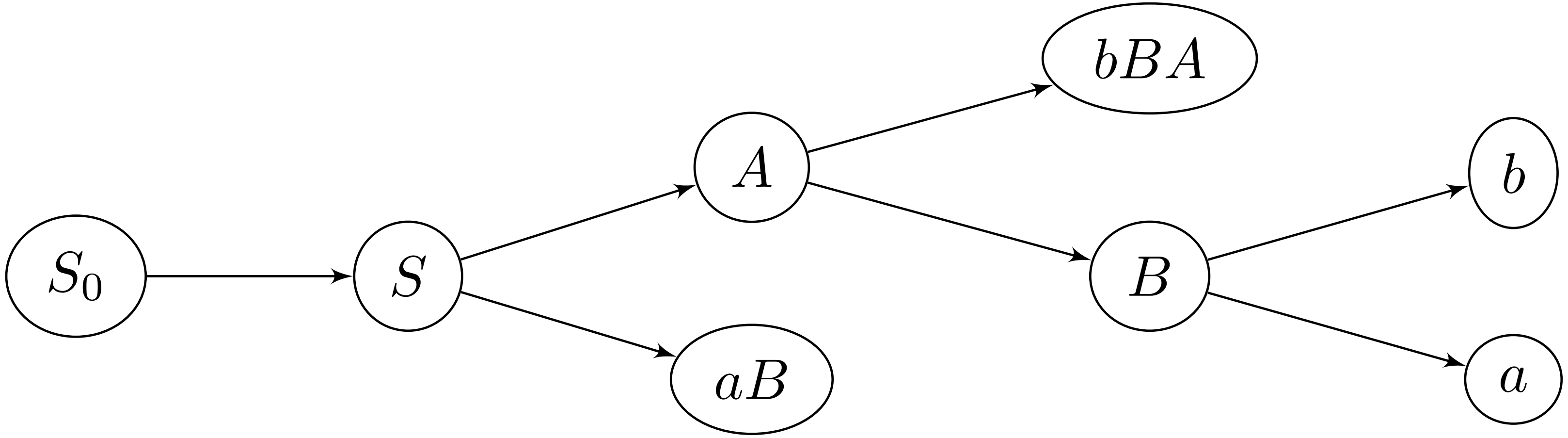
Chomsky Normal Form

More on removing unit transitions

On unit transitions and transitivity

Example 2

$$S_0 \to S$$
$$S \to A \mid aB$$
$$A \to bBA \mid B$$
$$B \to b \mid a$$

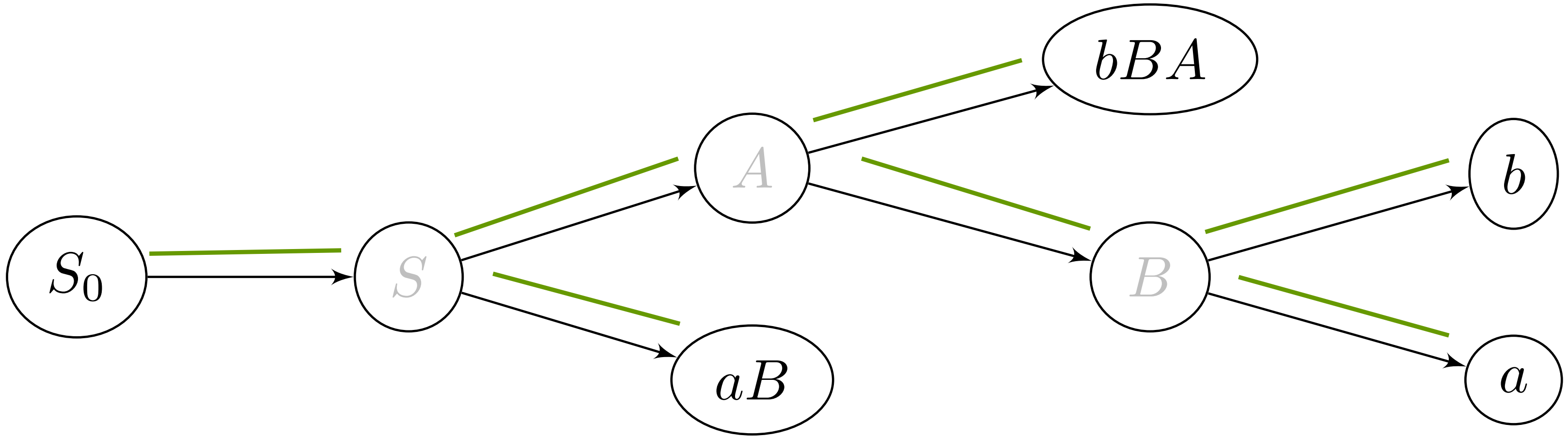


What do you think is the resulting grammar?

More on removing unit transitions

On unit transitions and transitivity

Example 2 (S_0)

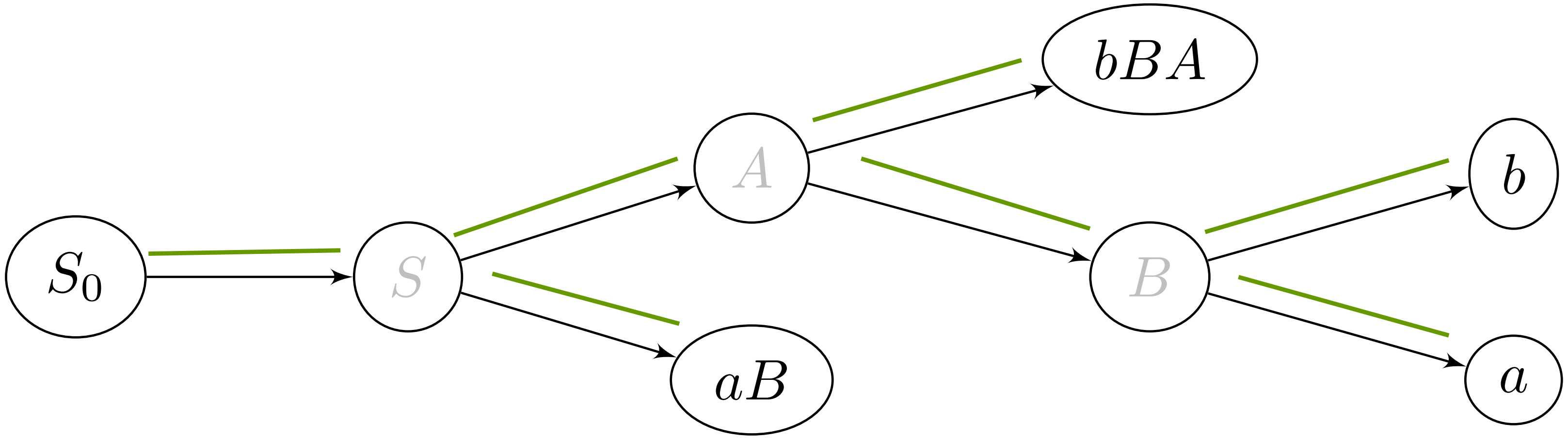


$S_0 \to$

More on removing unit transitions

On unit transitions and transitivity

Example 2 (S_0)



$$S_0 \to \underbrace{bBA}_{A} \mid \underbrace{b \mid a}_{B} \mid \underbrace{aB}_{S}$$

$S \to$

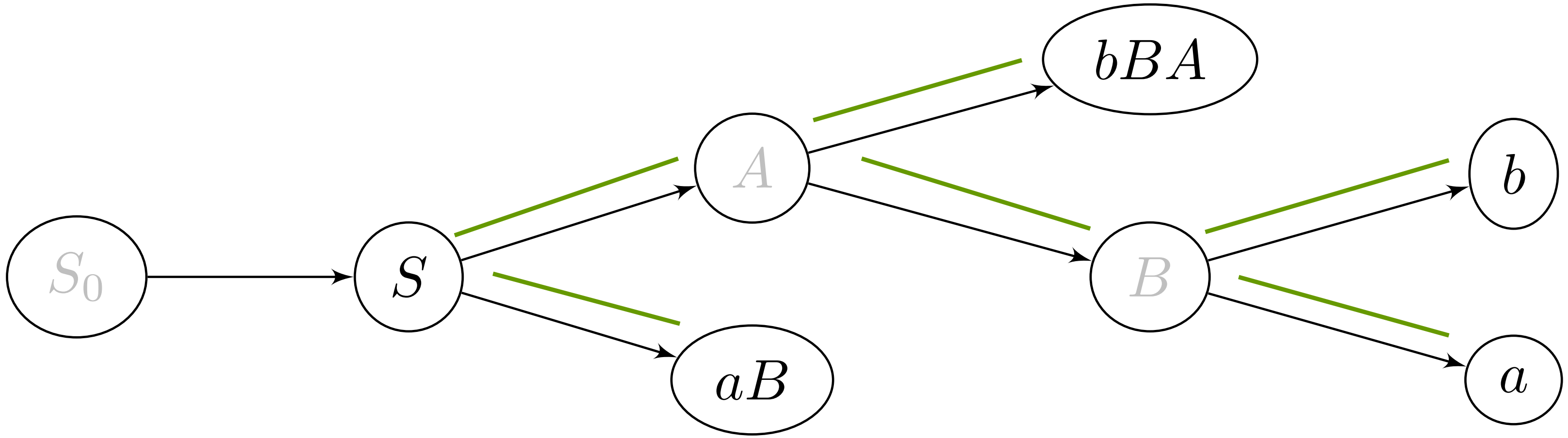
$A \to$

$B \to$

More on removing unit transitions

On unit transitions and transitivity

Example 2 (S)



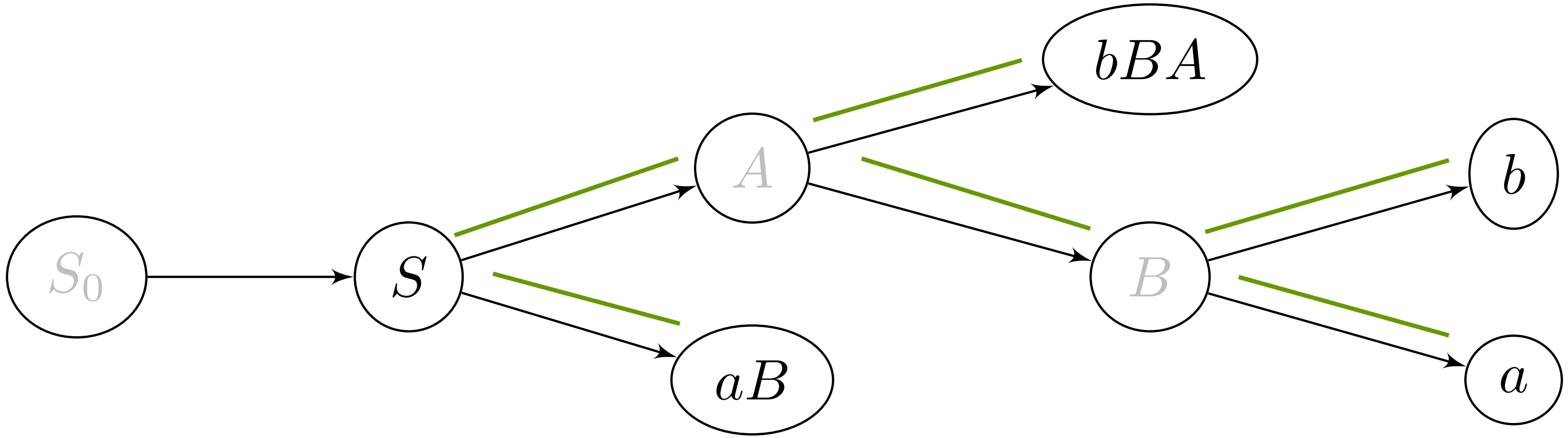
$S_0 \to A \mid b \mid a \mid aB$

$S \to$

More on removing unit transitions

On unit transitions and transitivity

Example 2 (S)



$S_0 \to A \mid b \mid a \mid aB$

$S \to \underbrace{bBA}_{A} \mid \underbrace{b \mid a}_{B} \mid \underbrace{aB}_{S}$

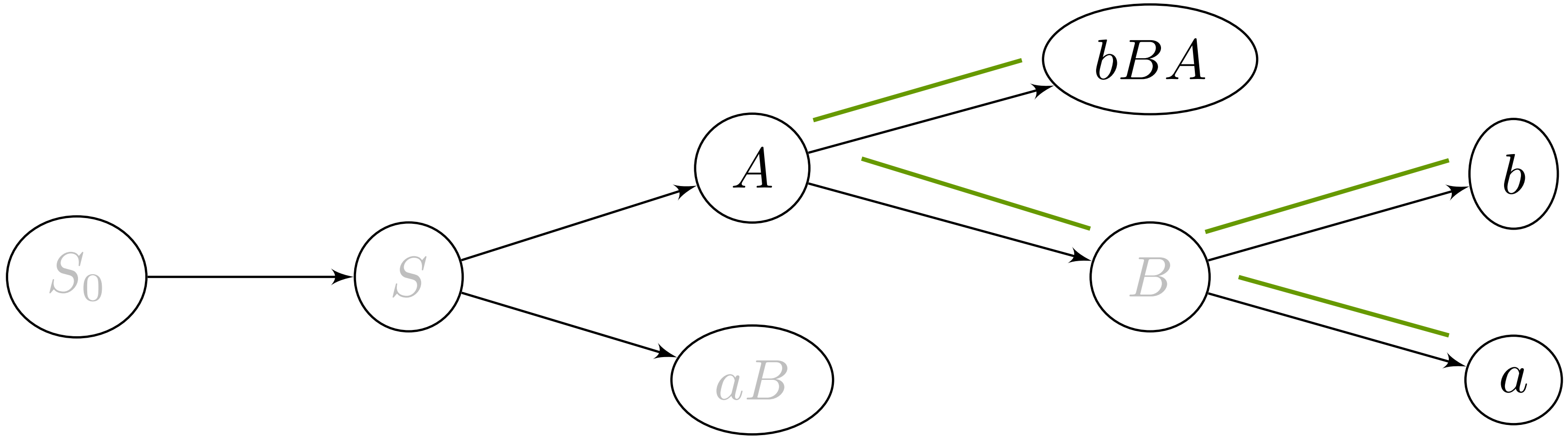
$A \to$

$B \to$

More on removing unit transitions

On unit transitions and transitivity

Example 2 ($\boldsymbol{A}$)

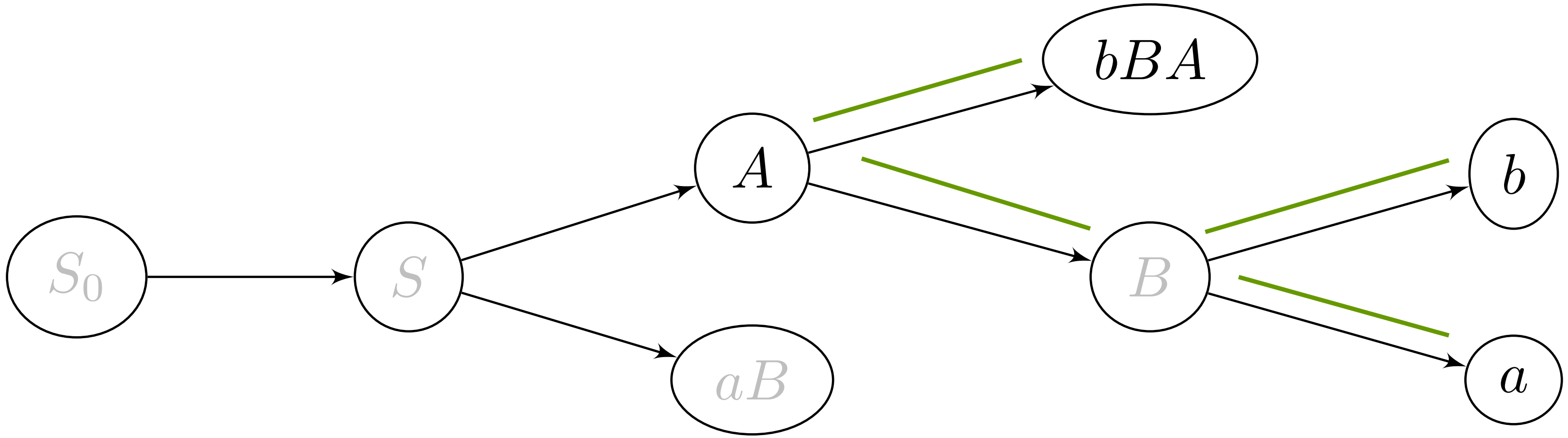


$S_0 \to A \mid b \mid a \mid aB$
$S \to A \mid b \mid a \mid aB$
$\boldsymbol{A} \to$

More on removing unit transitions

On unit transitions and transitivity

Example 2 (A)



$S_0 \to A \mid b \mid a \mid aB$

$S \to A \mid b \mid a \mid aB$

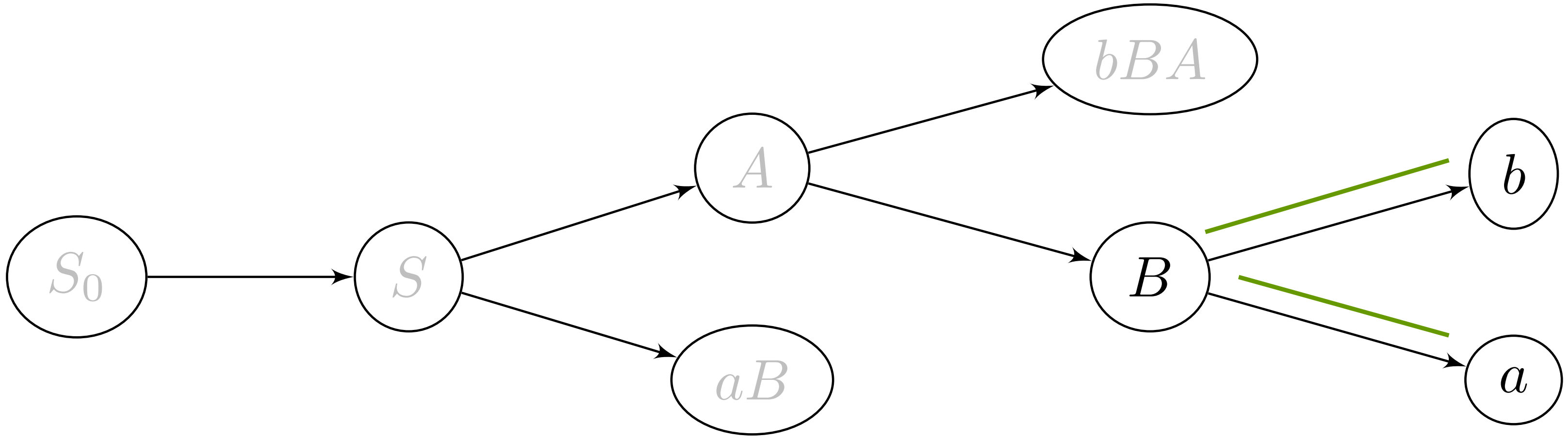
$A \to \underbrace{bBA}_{A} \mid \underbrace{b \mid a}_{B}$

$B \to$

More on removing unit transitions

On unit transitions and transitivity

Example 2 (B)

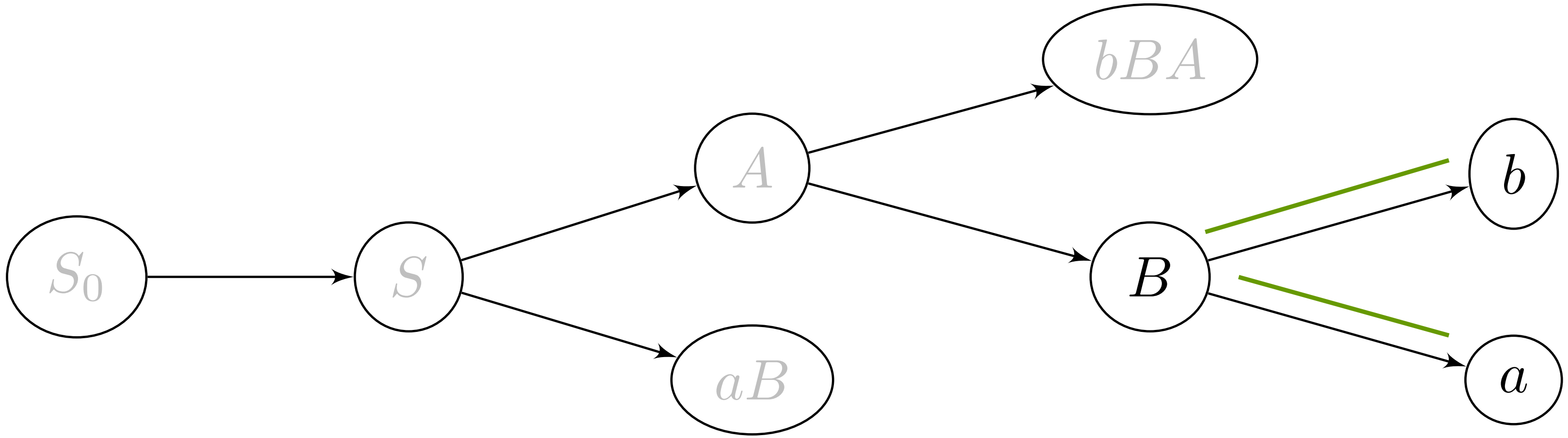


$S_0 \to A \mid b \mid a \mid aB$
$S \to A \mid b \mid a \mid aB$
$A \to bBA \mid b \mid a$
$B \to$

More on removing unit transitions

On unit transitions and transitivity

Example 2 (B)



$S_0 \to A \mid b \mid a \mid aB$
$S \to A \mid b \mid a \mid aB$
$A \to bBA \mid b \mid a$
$B \to b \mid a$

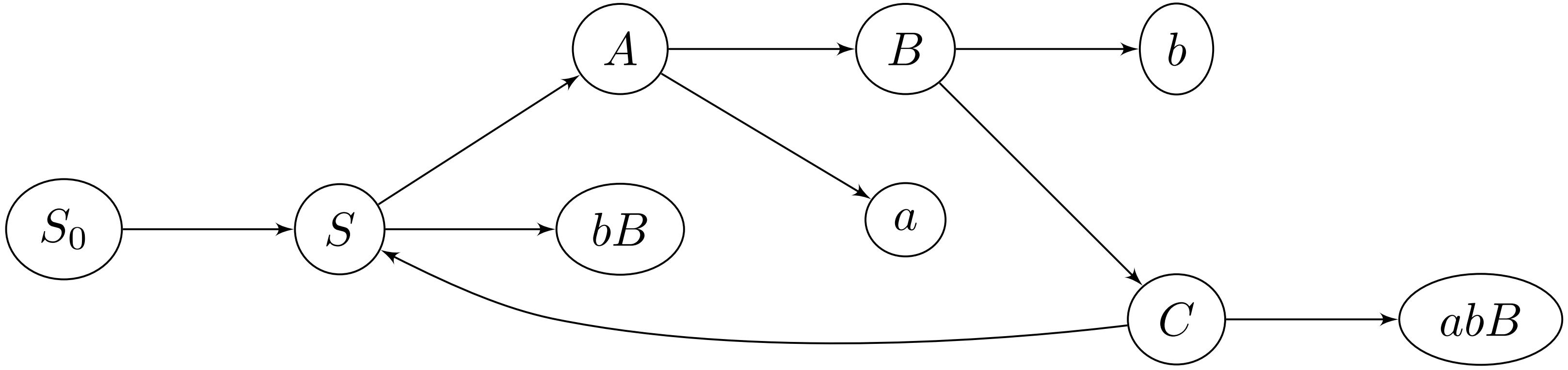
> We must take into consideration all possible paths via unit-edges.

More on removing unit transitions

On unit transitions with loops

Example 3

$$S_0 \to S$$
$$S \to A \mid bB$$
$$A \to B \mid a$$
$$B \to b \mid C$$
$$C \to abB \mid S$$

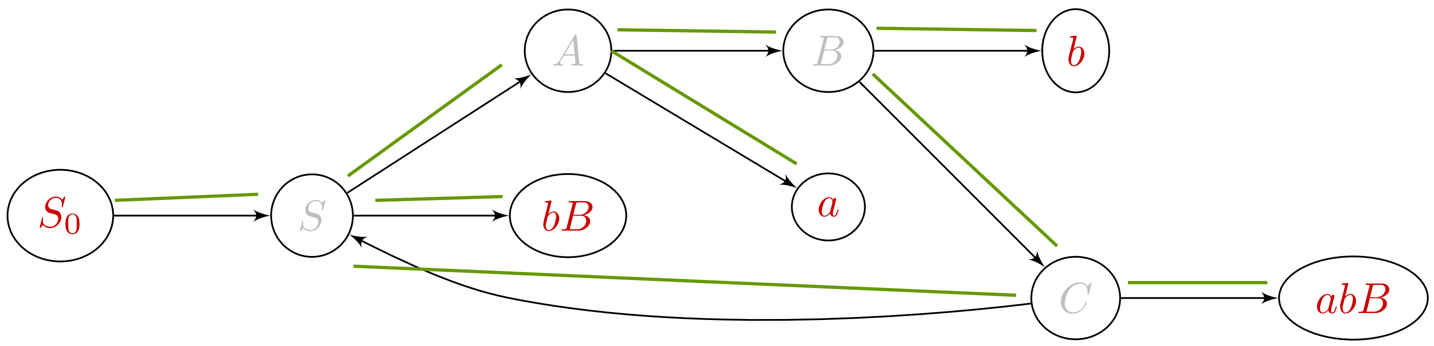


What do you think is the resulting grammar?

More on removing unit transitions

On unit transitions with loops

Example 3 (S_0)

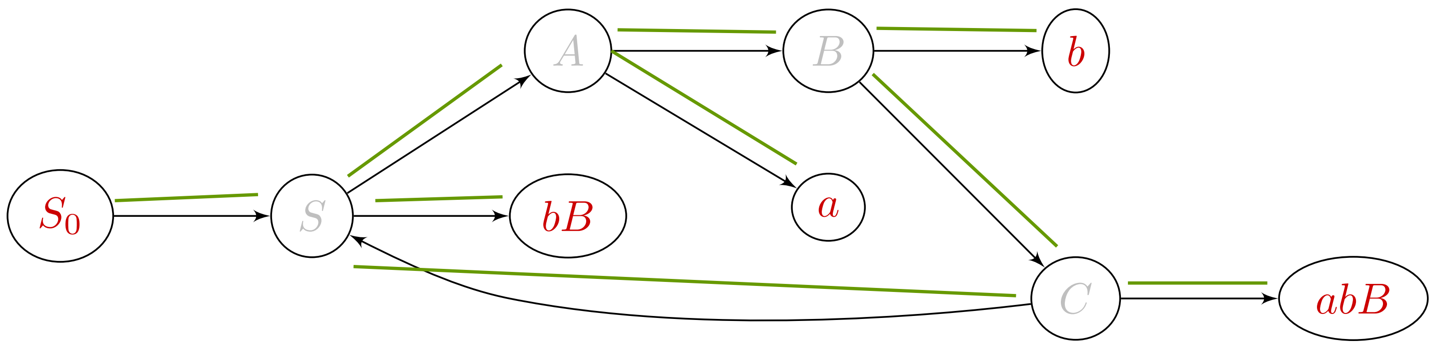


$S_0 \rightarrow$

More on removing unit transitions

On unit transitions with loops

Example 3 (S_0)



$$S_0 \to \underbrace{bB}_{S} \mid \underbrace{a}_{A} \mid \underbrace{b}_{B} \mid \underbrace{abB}_{C}$$

$S \to$

$A \to$

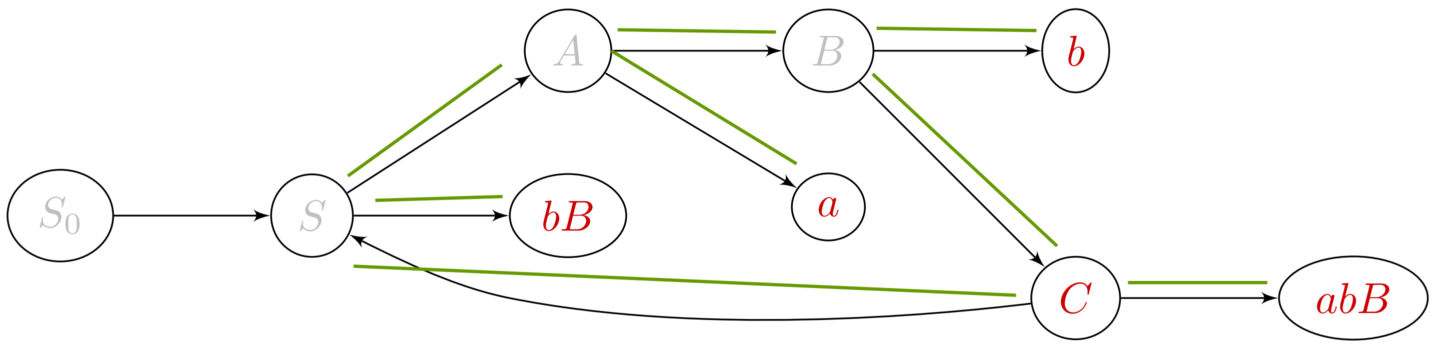
$B \to$

$C \to$

More on removing unit transitions

On unit transitions with loops

Example 3 (C)



$$S_0 \to \underbrace{bB}_{S} \mid \underbrace{a}_{A} \mid \underbrace{b}_{B} \mid \underbrace{abB}_{C}$$

$S \to$

$A \to$

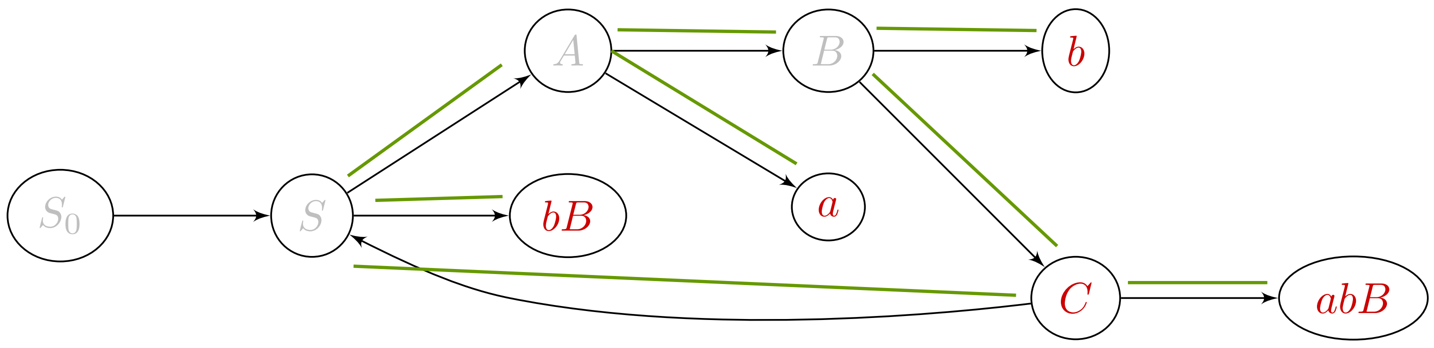
$B \to$

$C \to$

More on removing unit transitions

On unit transitions with loops

Example 3 (C)



$$S_0 \to \underbrace{bB}_{S} \mid \underbrace{a}_{A} \mid \underbrace{b}_{B} \mid \underbrace{abB}_{C}$$

$$S \to$$

$$A \to$$

$$B \to$$

$$C \to \underbrace{bB}_{S} \mid \underbrace{a}_{A} \mid \underbrace{b}_{B} \mid \underbrace{abB}_{C}$$

> Note that we musth handle loops. All variables in the loop have the same productions.