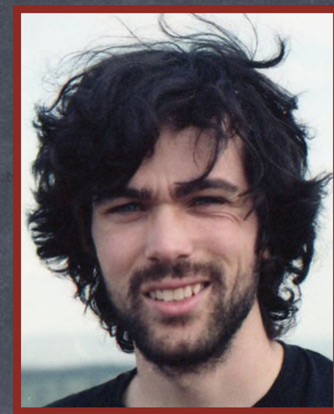
COMP-330
Theory of Computation

Fall 2019 -- Prof. Claude Crépeau

Lec. 4 : DFAs, NFAs + Kleene's theorem



TUTORIALS

Fri 13 sep, Mon 16 sep, Thu 19 sep

- **TUTORIALS Hours** :

Pouriya : Friday 13:00-14:00 ARTS W-215 (cap 105)

Pierre-William : Monday 15:00-16:00 ARTS 150 (cap 88)

Anirudha : Monday 16:00-17:00 ENGTR 3090

Justin : Tuesday 15:00-16:00 ENGTR 3110

Yanjia : Friday 10:00-11:00 ENGTR 3110

Shiquan : Thursday 15:00-16:00 WILSON 105 (cap 70)

COMP-330 Fall 2019 — Weekly Schedule

Mon 10:00	**Tue** 10:00	**Wed** 10:00	**Thu** 10:00	Yanjia TR-3110
Mon 10:30	**Tue** 10:30	**Wed** 10:30	**Thu** 10:30	
Mon 11:00	**Tue** 11:00	**Wed** 11:00	**Thu** 11:00	**Fri** 11:00
Mon 11:30	**Tue** 11:30	**Wed** 11:30	**Thu** 11:30	**Fri** 11:30
Mon 12:00	**Tue** 12:00	**Wed** 12:00	**Thu** 12:00	**Fri** 12:00
Mon 12:30	**Tue** 12:30	**Wed** 12:30	**Thu** 12:30	**Fri** 12:30
Mon 13:00	Claude MA-112 course	Claude MC-110N office hours	Claude MA-112 course	Pouriya ARTW-215
Mon 13:30				
Mon 14:00				**Fri** 14:00
Mon 14:30	**Tue** 14:30		**Thu** 14:30	**Fri** 14:30
Pierre-W. ARTS-150	Justin TR-3110		Shiquan WIL-105	**Fri** 15:00
				Fri 15:30
Anirudha TR-3090	**Tue** 16:00	**Wed** 16:00	TA meeting ?	**Fri** 16:00
	Tue 16:30	**Wed** 16:30		**Fri** 16:30

MC = MCENG = McConnell • TR = ENGTR = Trottier

COMP 330 Fall 2019: Lectures Schedule

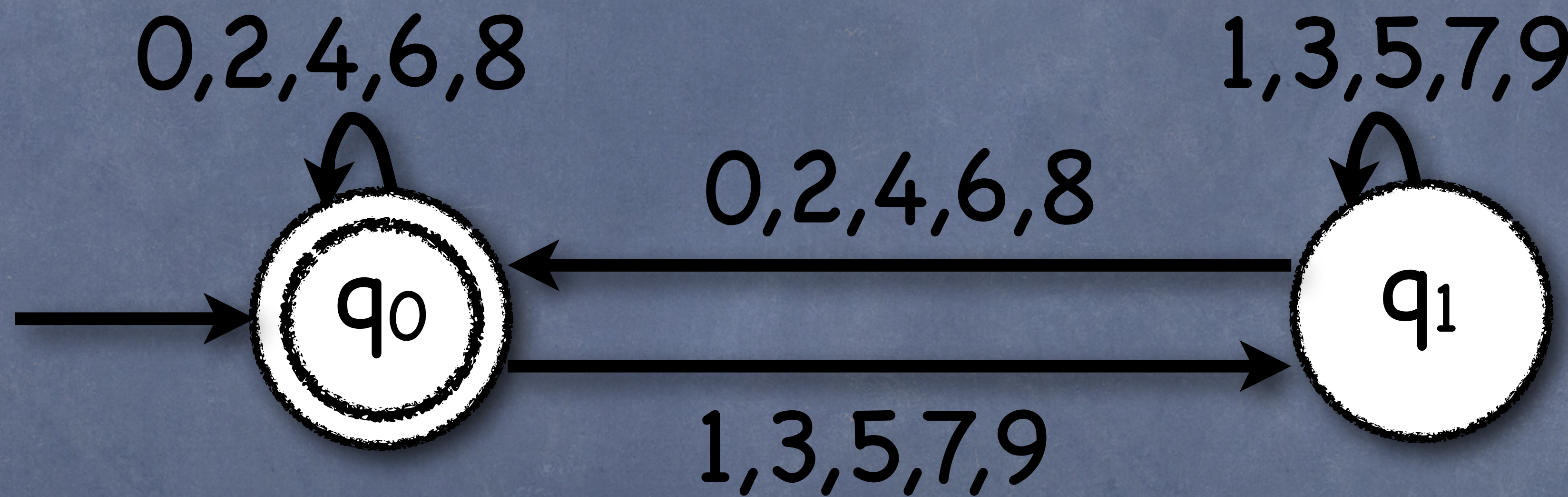
1-2. Introduction
    1.5. Some basic mathematics
2-3. Deterministic finite automata
    +Closure properties,
4. Nondeterministic finite automata
5. Minimization+ Myhill-Nerode theorem
6. Determinization+Kleene's theorem
7. Regular Expressions+GNFA
8. Regular Expressions and Languages
9-10. The pumping lemma
11. Duality
12. Labelled transition systems
13. MIDTERM
14. Context-free languages
15. Pushdown automata
16. Parsing
17. The pumping lemma for CFLs
18. Introduction to computability
19. Models of computation
    Basic computability theory
20. Reducibility, undecidability and Rice's theorem
21. Undecidable problems about CFGs
22. Post Correspondence Problem
23. Validity of FOL is RE / Gödel's and Tarski's thms
24. Universality / The recursion theorem
25. Degrees of undecidability
26. Introduction to complexity

Examples: automata for multiples of N base B

- automata for multiples of N = 0 mod N
- examples mod 2, mod 3, mod 7

0 MOD 2 (base 10)

$M_{2,10}$

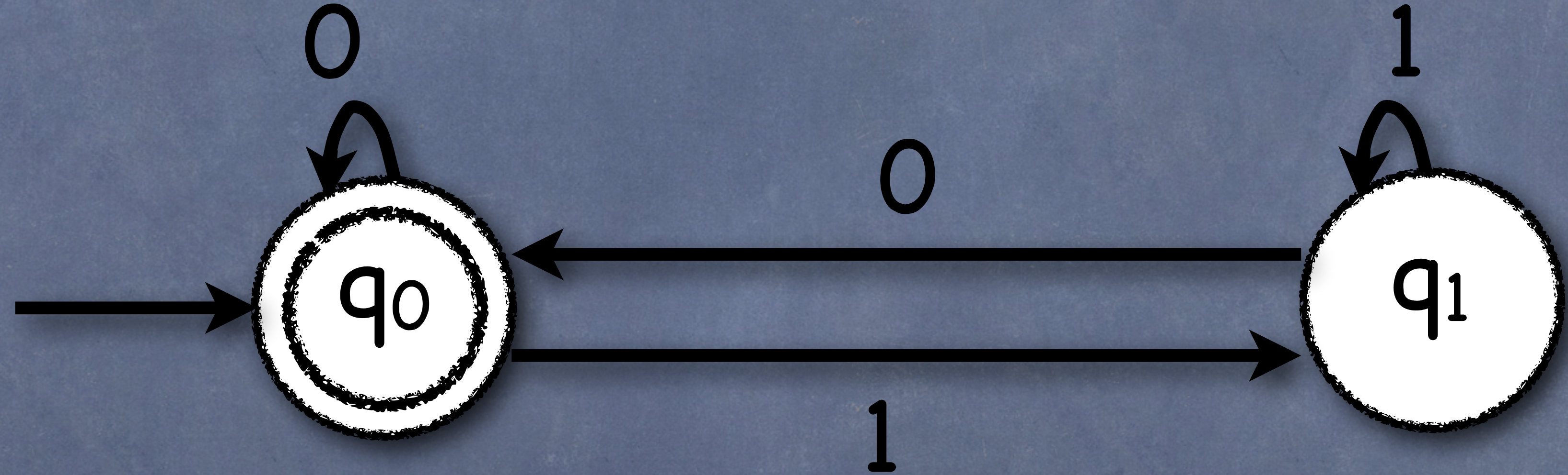


- Remember what you learned in elementary school: N is a multiple of 2 iff it ends by 0,2,4,6 or 8.

$M_{2,10}$ stops in state $q_r \Leftrightarrow w = r \bmod 2$

0 MOD 2 (base 2)

$M_{2,2}$

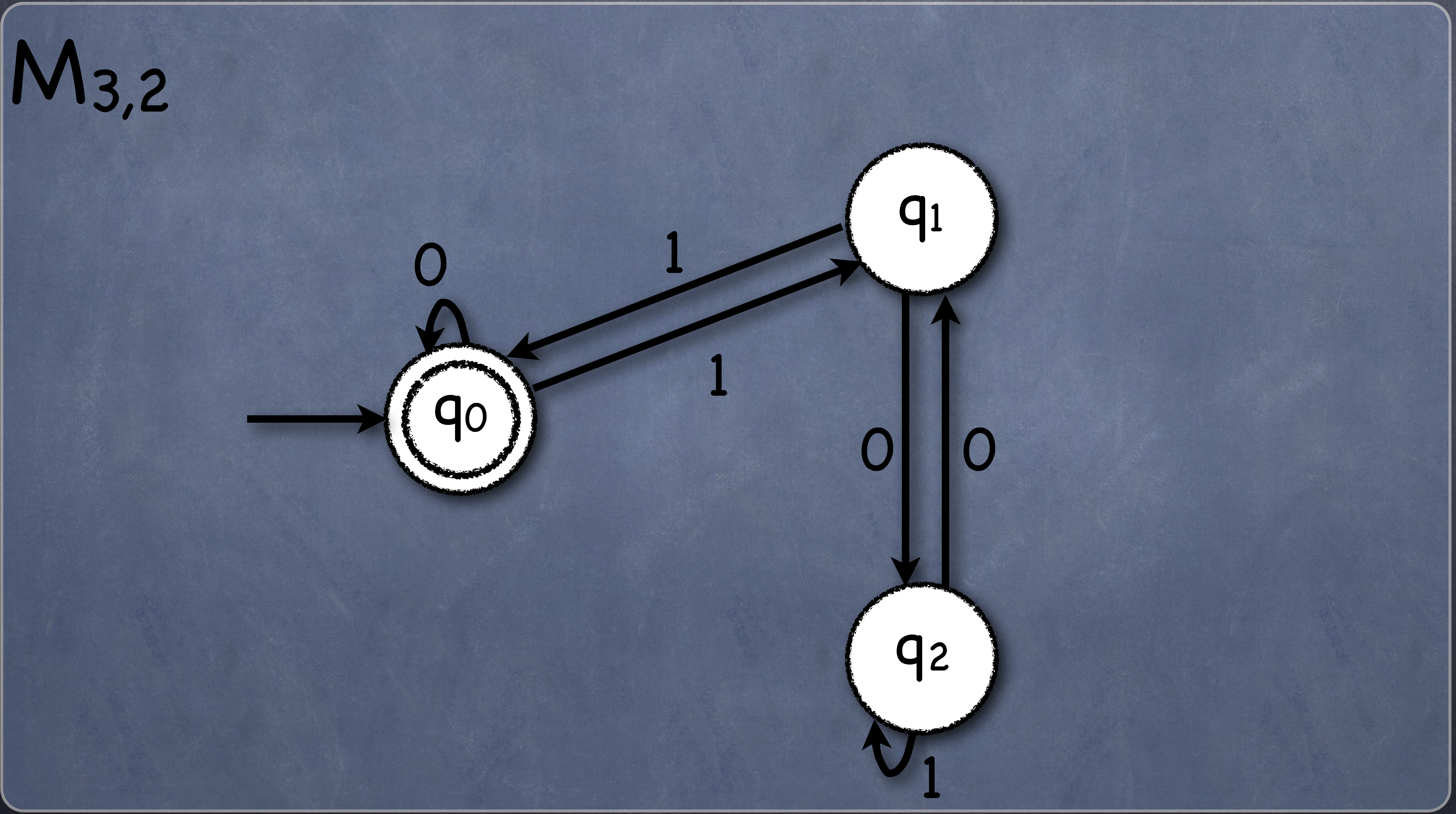


- Remember what you learned in school of CS: N (in binary) is a multiple of 2 iff it ends by 0.

$M_{2,2}$ stops in state $q_r \Leftrightarrow w = r \bmod 2$

$gcd(B,N) = 1$

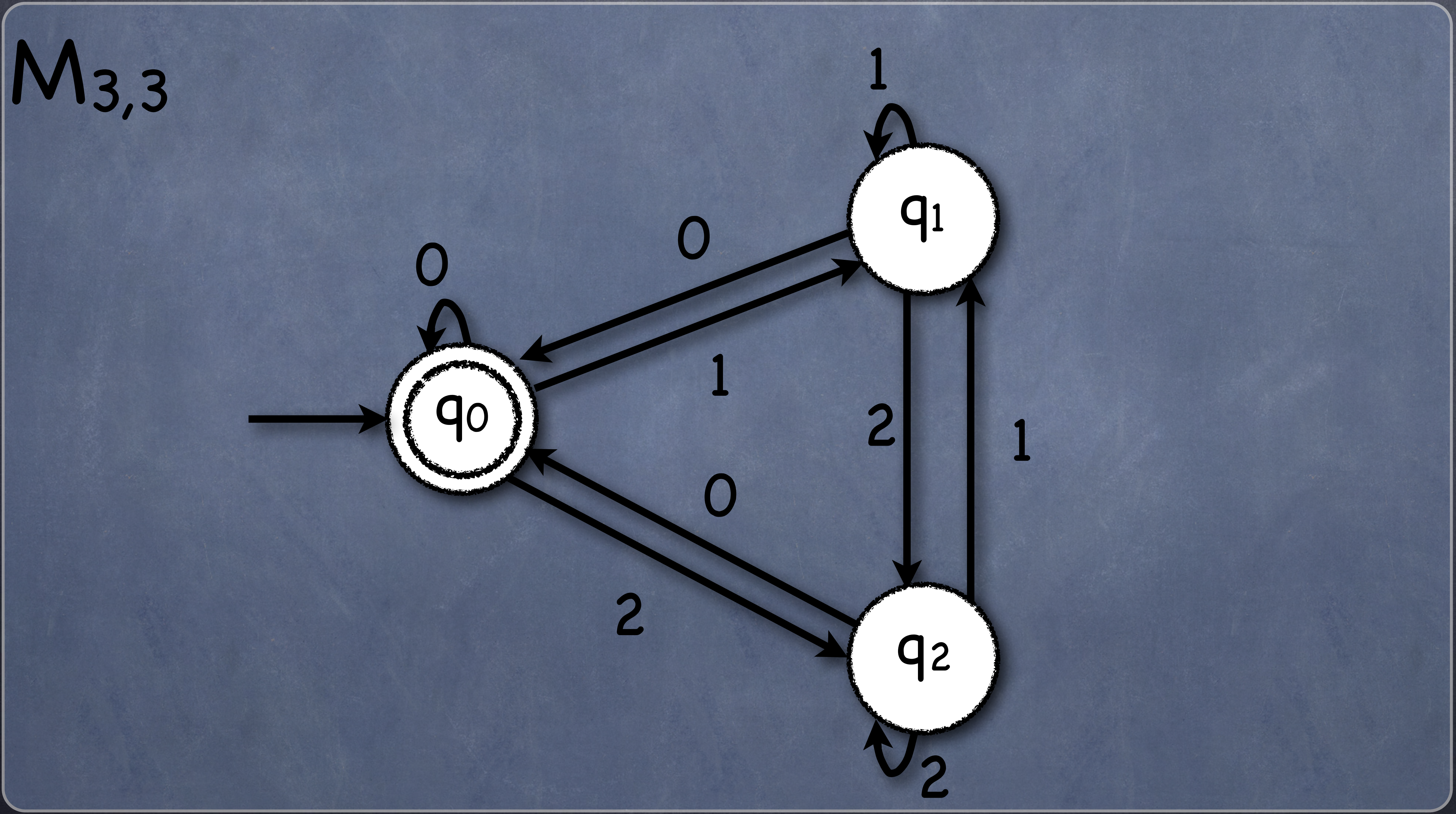
0 MOD 3 (base 2)



$M_{3,2}$ stops in state $q_r \Leftrightarrow w = r \bmod 3$

gcd(B,N) > 1

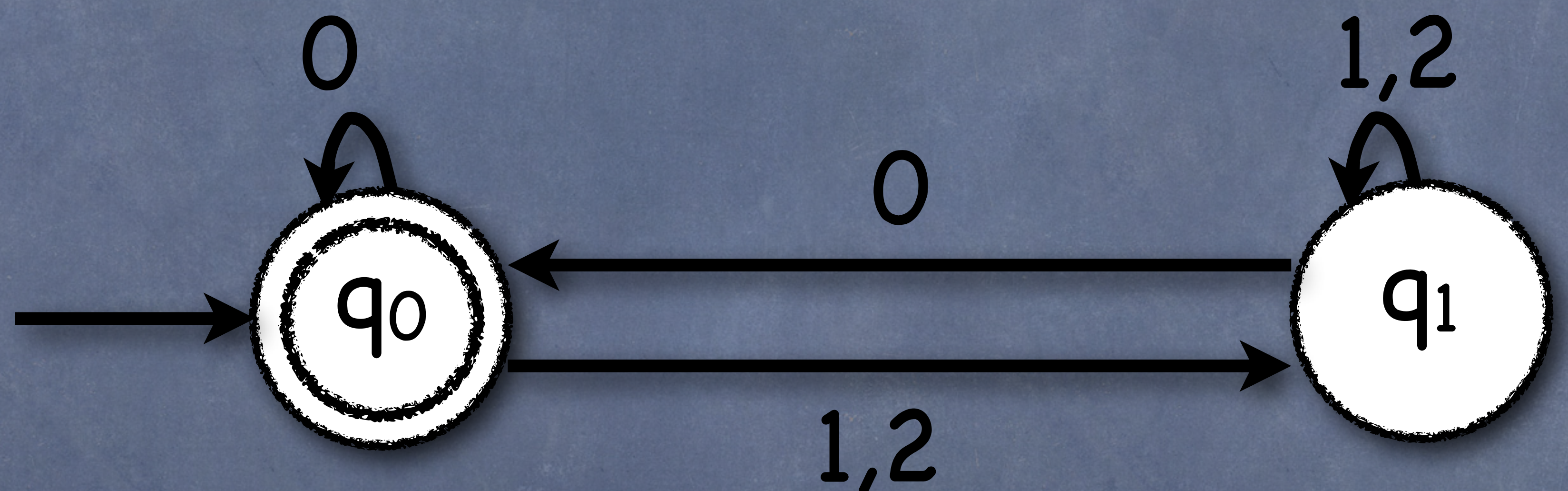
0 MOD 3 (base 3)



$M_{3,3}$ stops in state $q_r \Leftrightarrow w = r \bmod 3$

0 MOD 3 (base 3)

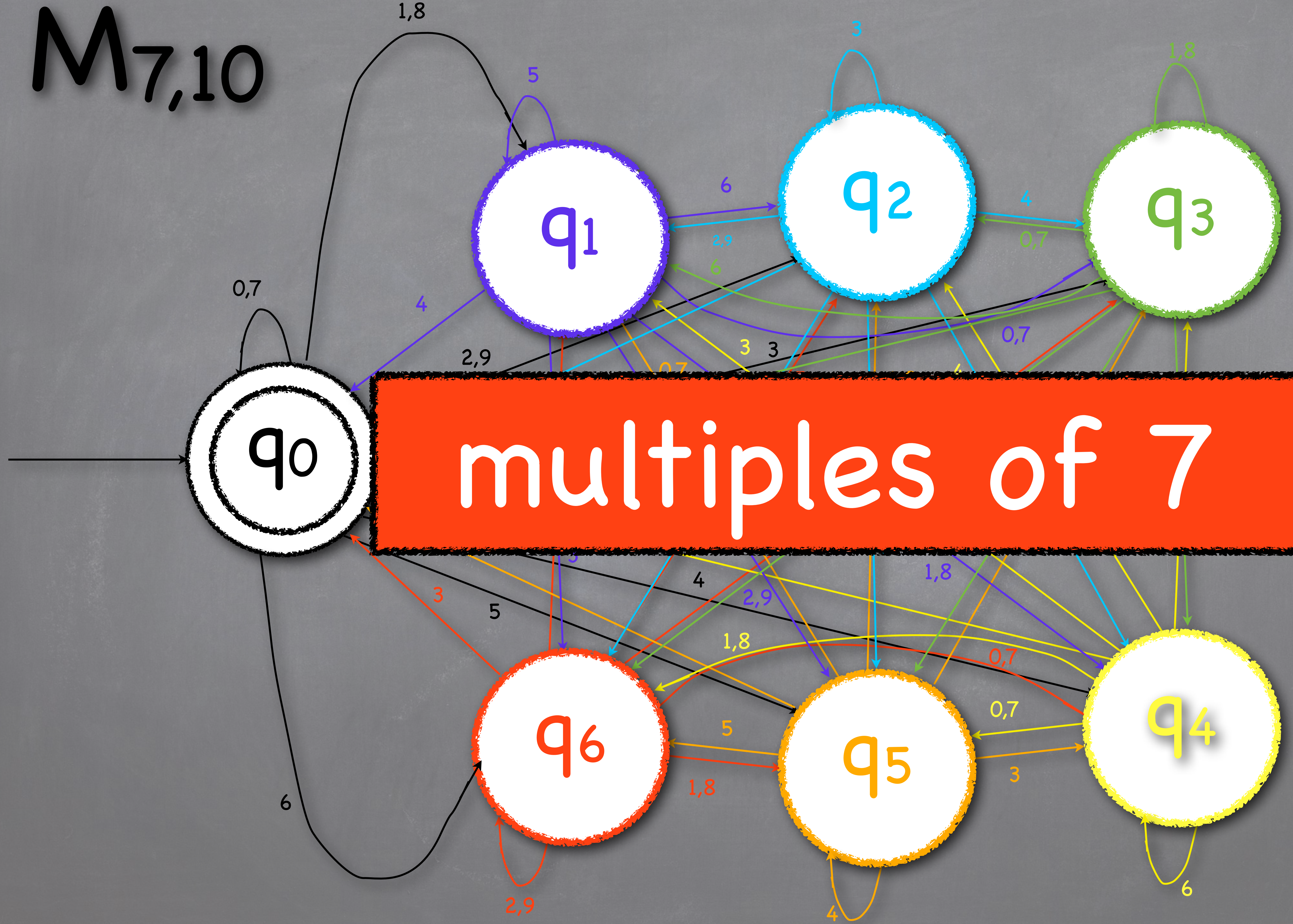
$M'_{3,3}$



- Remember what you learned in school of CS: N (in ternary) is a multiple of 3 iff it ends by 0.

$M'_{3,3}$ stops in state $q_0 \Leftrightarrow w = 0 \bmod 3$

$M_{7,10}$



multiples of 7

Another example: multiples of 7…

Another example: multiples of 7...

- Remember forever what you are learning in COMP-330 today : N is a multiple of 7 if $N \in L(M_{7,10})$.
- Example: 54705 is a multiple of 7 because
- 5 = (10x0+5) = 5 = 5 mod 7,
- 54 = (10x5+4) = 54 = 5 mod 7,
- 547 = (10x5+7) = 57 = 1 mod 7,
- 5470 = (10x1+0) = 10 = 3 mod 7 and
- 54705 = (10x3+5) = 35 = 0 mod 7

Another example: multiples of 7…

Another example: multiples of 7...

- Remember forever what you learn in COMP-330 today : N is a multiple of 7 if $N \in L(M_{7,10})$.

Another example: multiples of 7...

- Remember forever what you learn in COMP-330 today : N is a multiple of 7 if $N \in L(M_{7,10})$.
- Example: 54705 is a multiple of 7 because

Another example: multiples of 7...

- Remember forever what you learn in COMP-330 today : N is a multiple of 7 if $N \in L(M_{7,10})$.

- Example: 54705 is a multiple of 7 because
- 5 = (10x0+5) = 5 = 5 mod 7,

Another example: multiples of 7...

- Remember forever what you learn in COMP-330 today : N is a multiple of 7 if $N \in L(M_{7,10})$.

- Example: 54705 is a multiple of 7 because
- 5 = (10x0+5) = 5 = 5 mod 7,
- 54 = (10x5+4) = 54 = 5 mod 7,

Another example: multiples of 7...

- Remember forever what you learn in COMP-330 today : N is a multiple of 7 if $N \in L(M_{7,10})$.

- Example: 54705 is a multiple of 7 because
- 5 = (10x0+5) = 5 = 5 mod 7,
- 54 = (10x5+4) = 54 = 5 mod 7,
- 547 = (10x5+7) = 57 = 1 mod 7,

Another example: multiples of 7…

- Remember forever what you learn in COMP-330 today : N is a multiple of 7 if $N \in L(M_{7,10})$.

- Example: 54705 is a multiple of 7 because
- 5 = (10x0+5) = 5 = 5 mod 7,
- 54 = (10x5+4) = 54 = 5 mod 7,
- 547 = (10x5+7) = 57 = 1 mod 7,
- 5470 = (10x1+0) = 10 = 3 mod 7 and

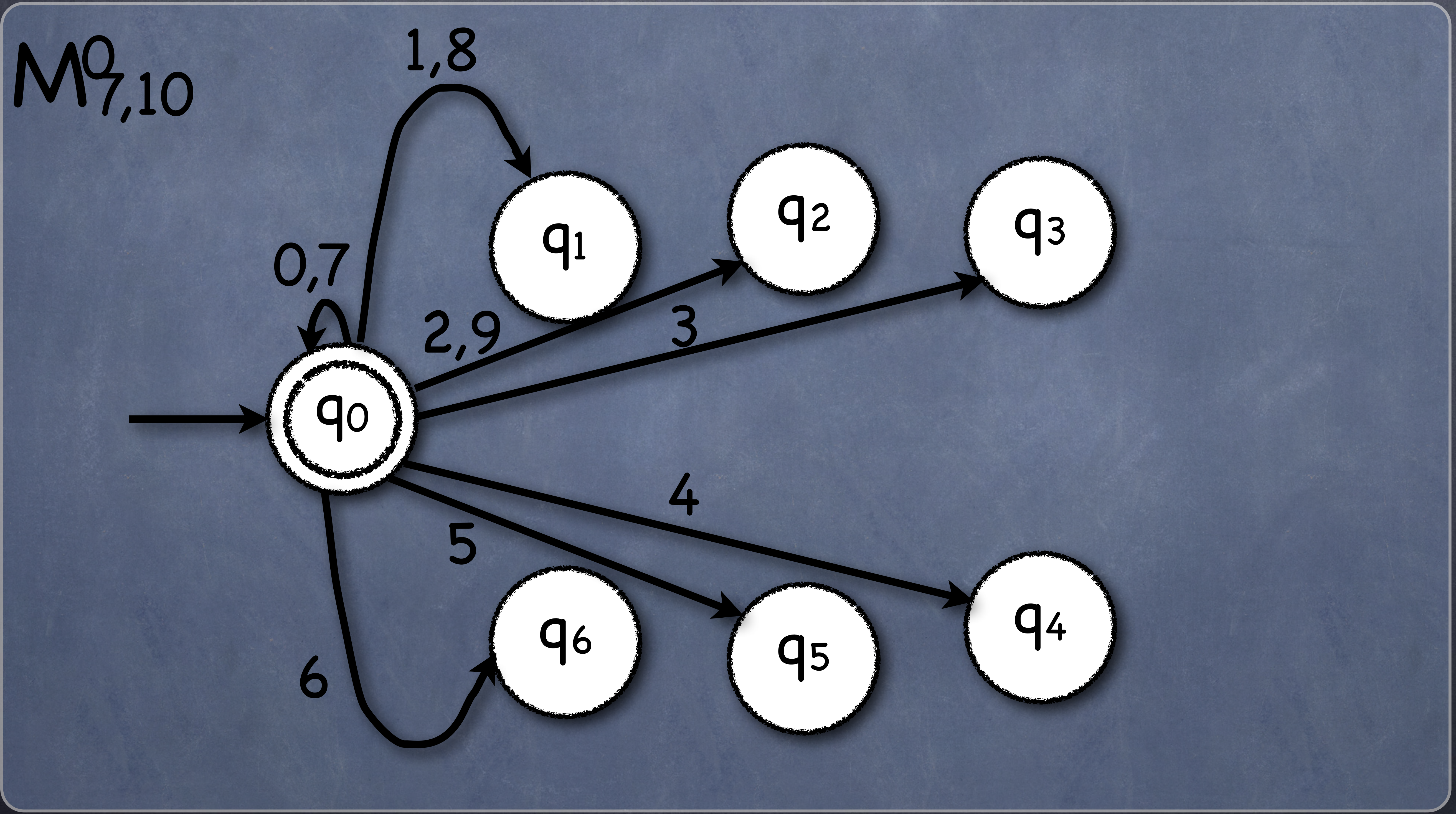
Another example: multiples of 7...

- Remember forever what you learn in COMP-330 today : N is a multiple of 7 if $N \in L(M_{7,10})$.

- Example: 54705 is a multiple of 7 because
- 5 = (10x0+5) = 5 = 5 mod 7,
- 54 = (10x5+4) = 54 = 5 mod 7,
- 547 = (10x5+7) = 57 = 1 mod 7,
- 5470 = (10x1+0) = 10 = 3 mod 7 and
- 54705 = (10x3+5) = 35 = 0 mod 7.

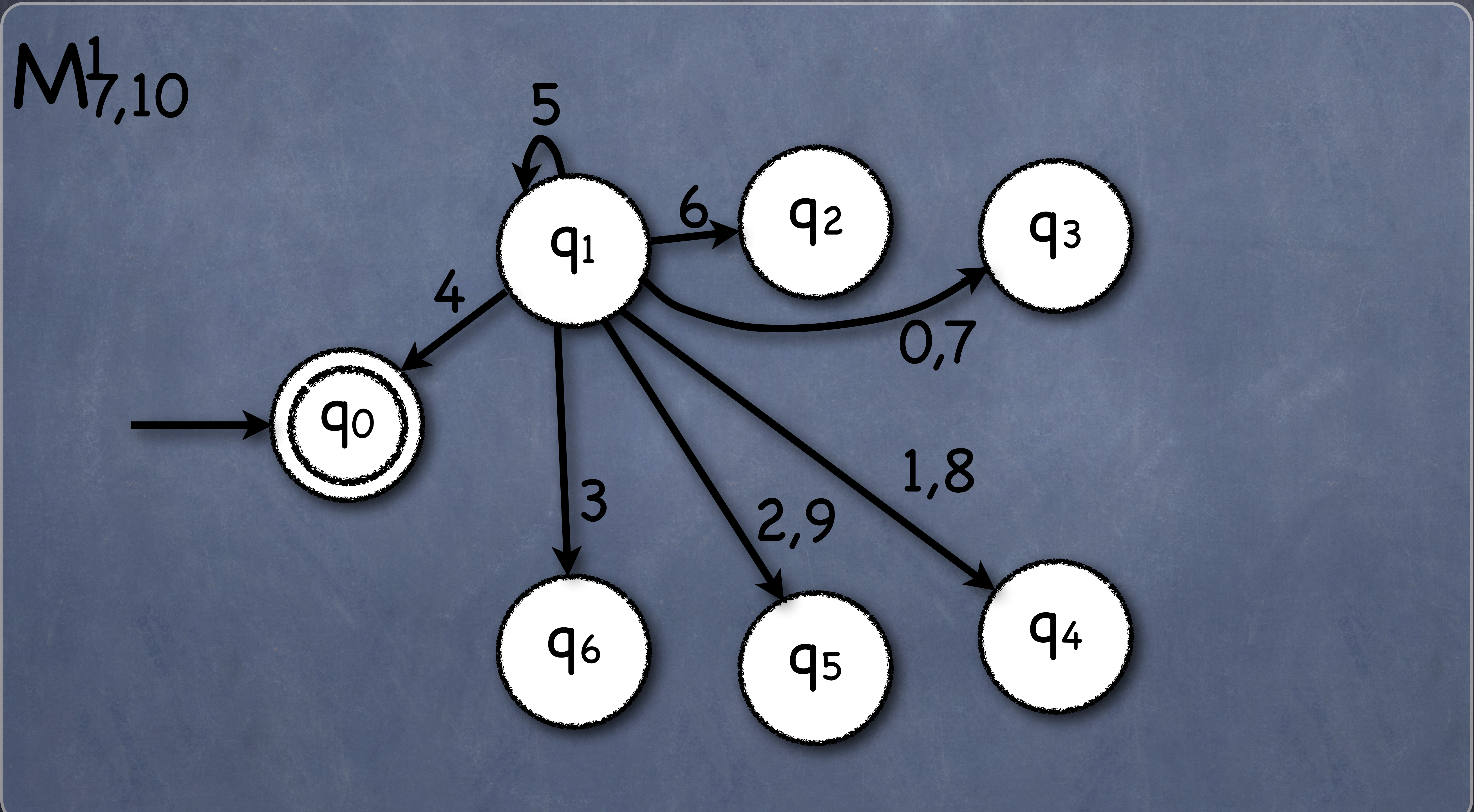
gcd(B,N) = 1

0 MOD 7 (base 10)



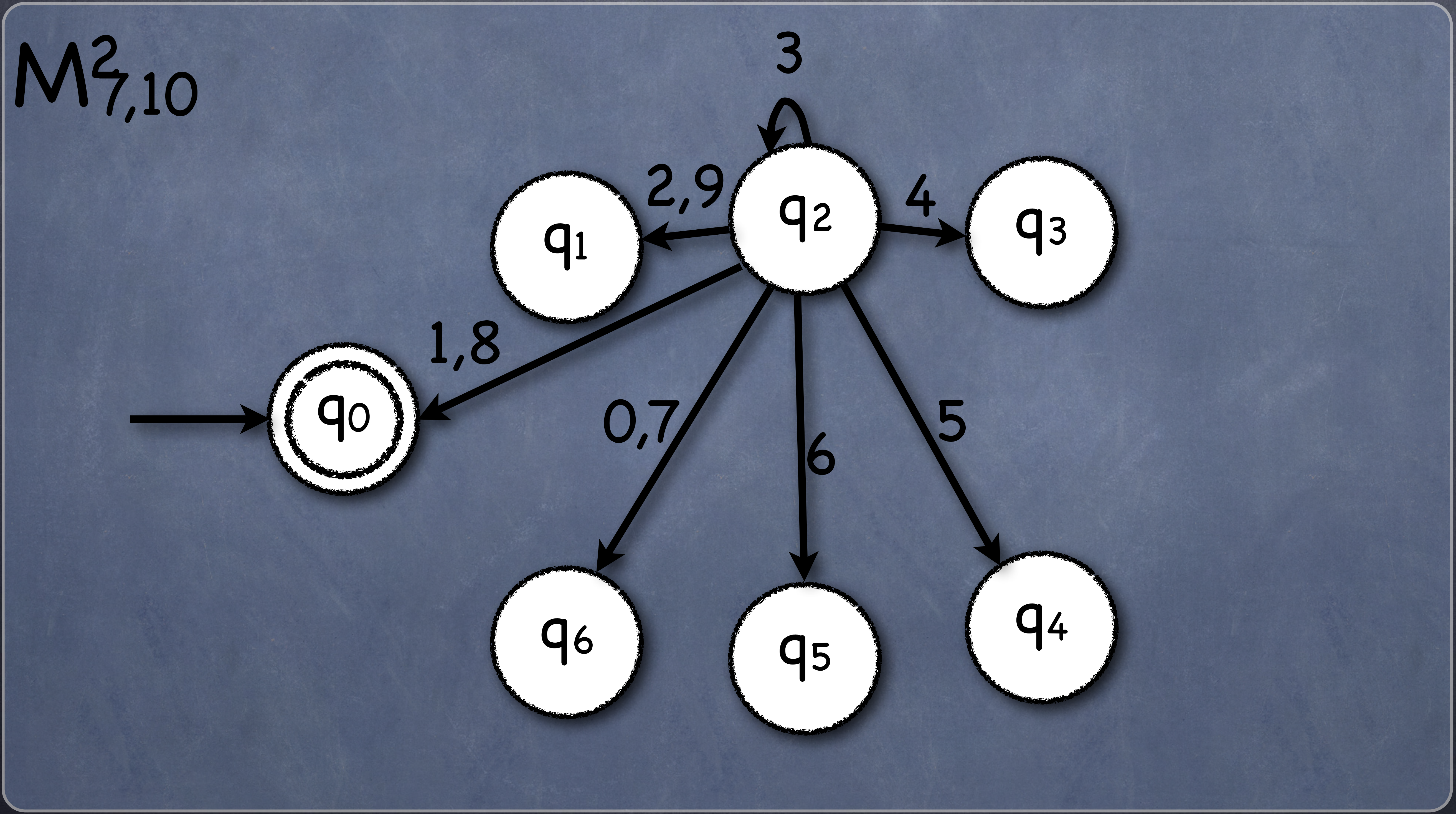
$M_{7,10}$ stops in state $q_r \Leftrightarrow w = r \bmod 7$

1 MOD 7 (base 10)



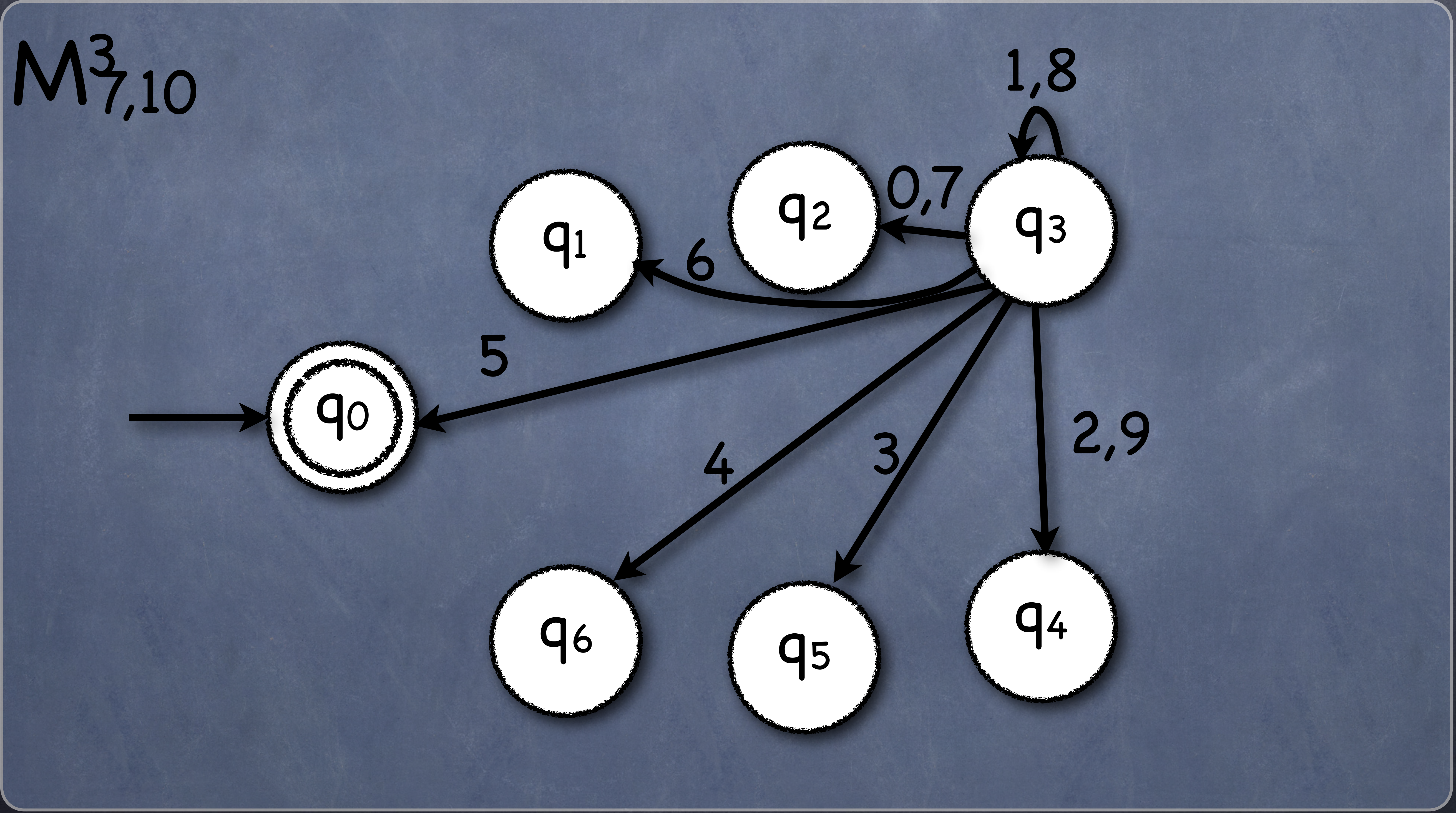
$M_{7,10}$ stops in state $q_r \Leftrightarrow w = r \bmod 7$

2 MOD 7 (base 10)



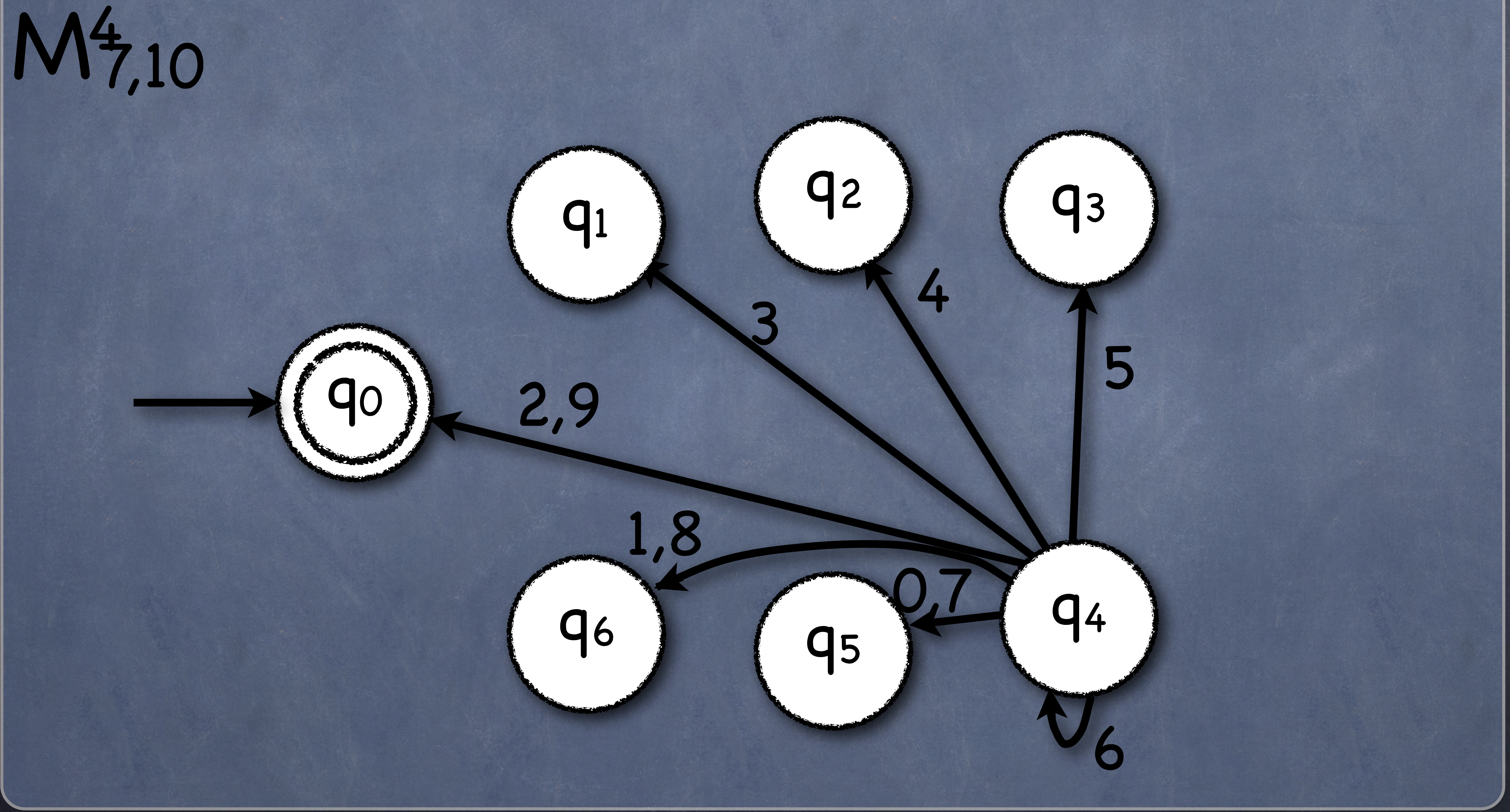
$M_{7,10}$ stops in state $q_r \Leftrightarrow w = r \bmod 7$

3 MOD 7 (base 10)



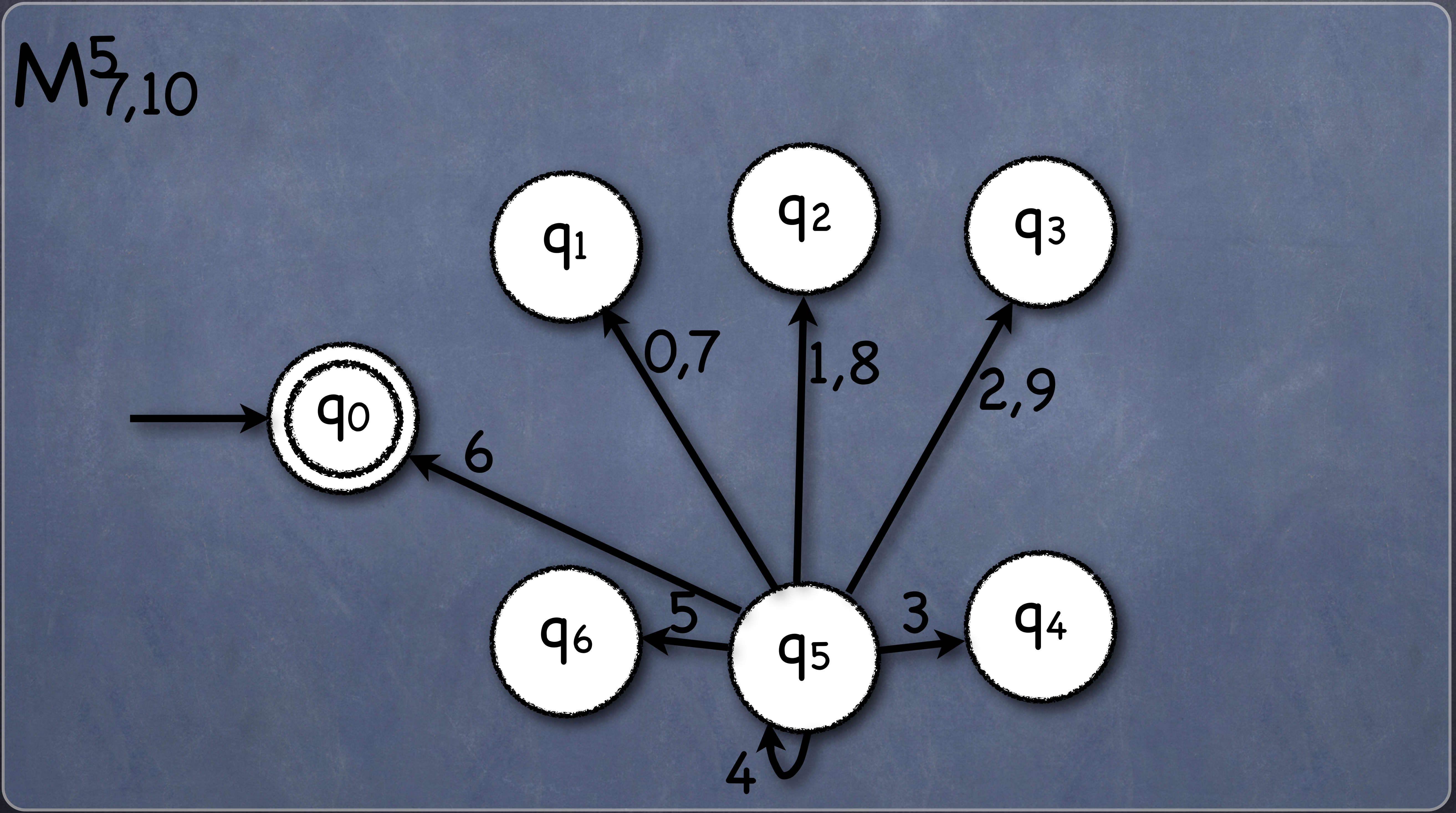
$M_{7,10}$ stops in state $q_r \Leftrightarrow w = r \bmod 7$

4 MOD 7 (base 10)



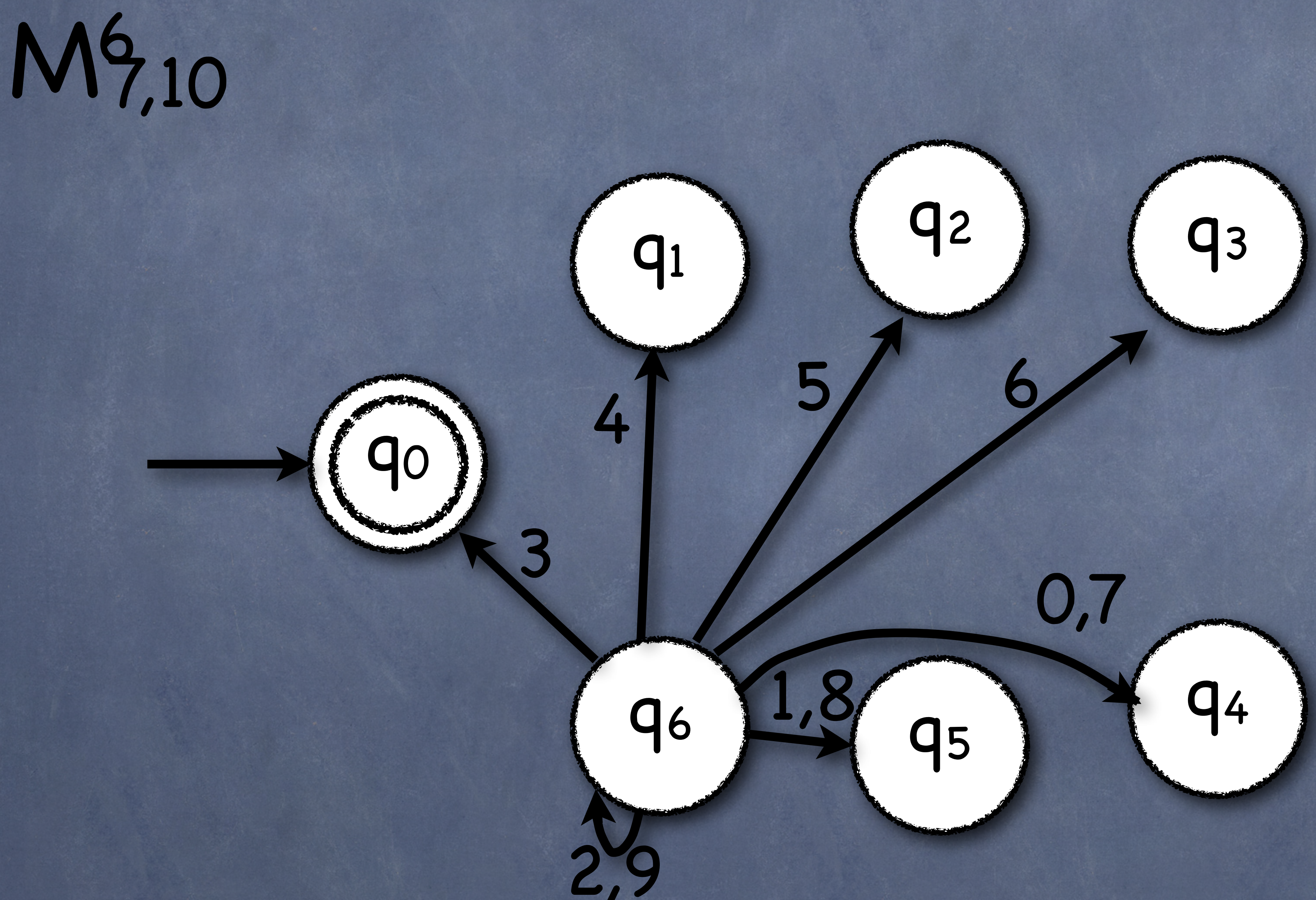
$M_{7,10}$ stops in state $q_r \Leftrightarrow w = r \bmod 7$

5 MOD 7 (base 10)



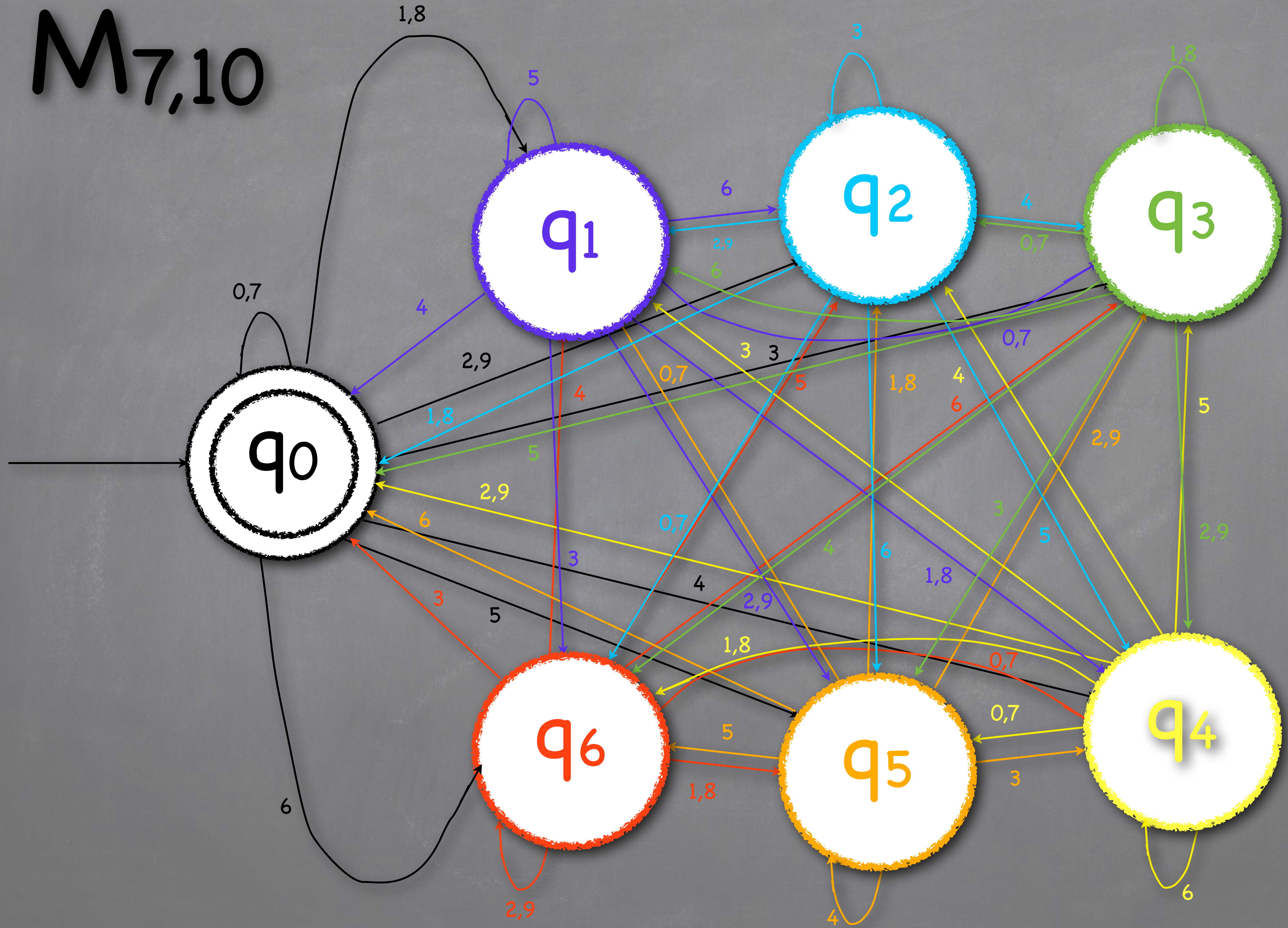
$M_{7,10}$ stops in state q_r $\Leftrightarrow$ w = r mod 7

6 MOD 7 (base 10)



$M_{7,10}$ stops in state $q_r \Leftrightarrow w = r \bmod 7$

$M_{7,10}$



Regular Operations

Regular Operations

DEFINITION 1.23

Let A and B be languages. We define the regular operations ***union***, ***concatenation***, and ***star*** as follows.

- **Union**: $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$.
- **Concatenation**: $A \circ B = \{xy \mid x \in A \text{ and } y \in B\}$.
- **Star**: $A^* = \{x_1 x_2 \ldots x_k \mid k \geq 0 \text{ and each } x_i \in A\}$.

Regular Operations

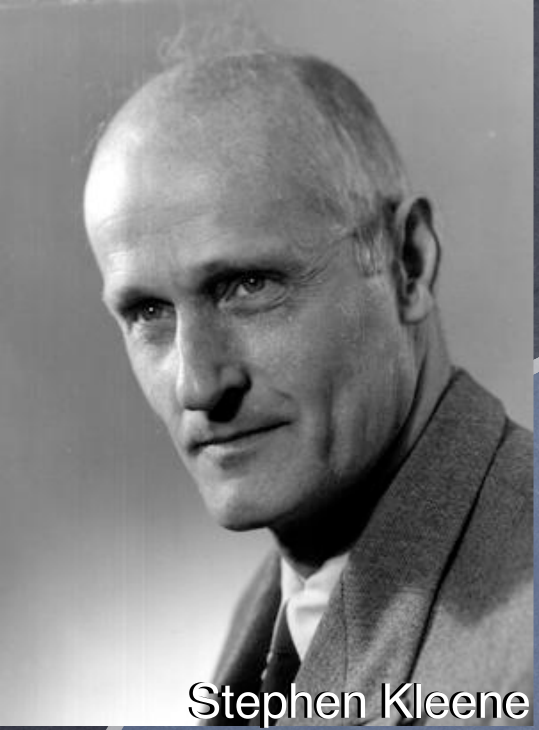
EXAMPLE 1.24

Let the alphabet Σ be the standard 26 letters $\{\texttt{a}, \texttt{b}, \ldots, \texttt{z}\}$. If $A = \{\texttt{good}, \texttt{bad}\}$ and $B = \{\texttt{boy}, \texttt{girl}\}$, then

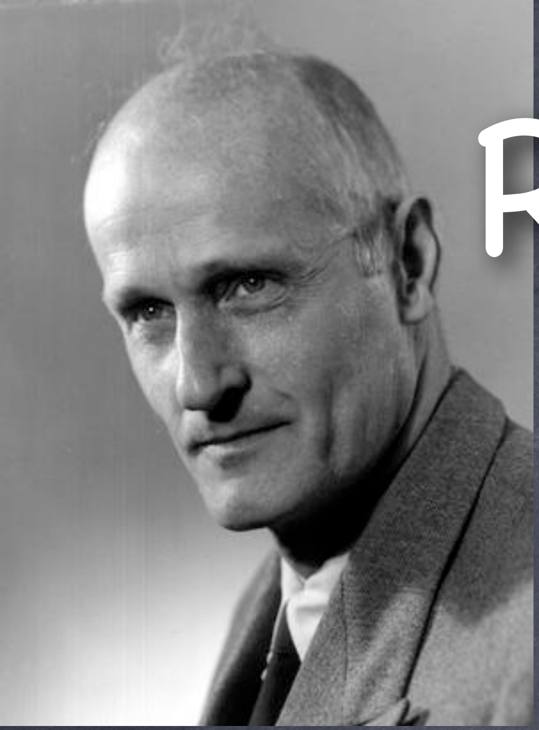
$$A \cup B = \{\texttt{good}, \texttt{bad}, \texttt{boy}, \texttt{girl}\},$$

$$A \circ B = \{\texttt{goodboy}, \texttt{goodgirl}, \texttt{badboy}, \texttt{badgirl}\}, \text{ and}$$

$$A^* = \{\varepsilon, \text{good, bad, goodgood, goodbad, badgood, badbad, goodgoodgood, goodgoodbad, goodbadgood, goodbadbad}, \ldots\}.$$



Regular Operations : Kleene's theorem



Regular Operations : Kleene's theorem

THEOREM 1.25

The class of regular languages is closed under the union operation.

In other words, if A_1 and A_2 are regular languages, so is $A_1 \cup A_2$.

Regular Operations : Kleene's theorem

Regular Operations : Kleene's theorem

- Let $M_A=(Q_A,\Sigma,\delta_A,q_{0A},F_A)$ be a DFA accepting L_A and $M_B=(Q_B,\Sigma,\delta_B,q_{0B},F_B)$ be a DFA accepting L_B.

Regular Operations : Kleene's theorem

- Let $M_A=(Q_A,\Sigma,\delta_A,q_{0A},F_A)$ be a DFA accepting L_A and $M_B=(Q_B,\Sigma,\delta_B,q_{0B},F_B)$ be a DFA accepting L_B.
- Consider $M_U=(Q_A \times Q_B,\Sigma,\delta_U,(q_{0A},q_{0B}),F_U)$ where

  $\delta_U((q,q'),s) = (\delta_A(q,s), \delta_B(q',s))$ for all q,q',s

  and

  $F_U = \{ (q,q') \mid q \in F_A \text{ or } q' \in F_B \}$.

Regular Operations : Kleene's theorem

- Let $M_A=(Q_A,\Sigma,\delta_A,q_{0A},F_A)$ be a DFA accepting L_A and $M_B=(Q_B,\Sigma,\delta_B,q_{0B},F_B)$ be a DFA accepting L_B.

- Consider $M_U=(Q_A \times Q_B,\Sigma,\delta_U,(q_{0A},q_{0B}),F_U)$ where

  $\delta_U((q,q'),s) = (\delta_A(q,s), \delta_B(q',s))$ for all q,q',s

  and

  $F_U = \{ (q,q') \mid q \in F_A \text{ or } q' \in F_B \}$.

- $L_U = L_A \cup L_B$.

- Let $M_A=(Q_A,\Sigma,\delta_A,q_{0A},F_A)$ be a DFA accepting L_A and $M_B=(Q_B,\Sigma,\delta_B,q_{0B},F_B)$ be a DFA accepting L_B.

- Let $M_A=(Q_A,\Sigma,\delta_A,q_{0A},F_A)$ be a DFA accepting L_A and $M_B=(Q_B,\Sigma,\delta_B,q_{0B},F_B)$ be a DFA accepting L_B.
- Consider $M_U=(Q_A \times Q_B,\Sigma,\delta_U,(q_{0A},q_{0B}),F_U)$ where

  $\delta_U((q,q'),s) = (\delta_A(q,s), \delta_B(q',s))$ for all q,q',s

  and

  $F_U = \{ (q,q') \mid q \in F_A \text{ or } q' \in F_B \}$.

- Let $M_A=(Q_A,\Sigma,\delta_A,q_{0A},F_A)$ be a DFA accepting L_A and $M_B=(Q_B,\Sigma,\delta_B,q_{0B},F_B)$ be a DFA accepting L_B.
- Consider $M_U=(Q_A \times Q_B,\Sigma,\delta_U,(q_{0A},q_{0B}),F_U)$ where

  $\delta_U((q,q'),s) = (\delta_A(q,s), \delta_B(q',s))$ for all q,q',s

  and

  $F_U = \{ (q,q') \mid q \in F_A \text{ or } q' \in F_B \}$.
- $L_U = L_A \cup L_B$.

- Let $M_A=(Q_A,\Sigma,\delta_A,q_{0A},F_A)$ be a DFA accepting L_A and $M_B=(Q_B,\Sigma,\delta_B,q_{0B},F_B)$ be a DFA accepting L_B.

- Consider $M_U=(Q_A \times Q_B,\Sigma,\delta_U,(q_{0A},q_{0B}),F_U)$ where

  $\delta_U((q,q'),s) = (\delta_A(q,s), \delta_B(q',s))$ for all q,q',s

  and

  $F_U = \{ (q,q') \mid q \in F_A \text{ or } q' \in F_B \}$.

- $L_U = L_A \cup L_B$.

- We can write it as $F_U = (F_A \times Q_B) \cup (Q_A \times F_B)$.

  (Not the same as $F_A \times F_B$.)

- Let $M_A=(Q_A,\Sigma,\delta_A,q_{0A},F_A)$ be a DFA accepting L_A and $M_B=(Q_B,\Sigma,\delta_B,q_{0B},F_B)$ be a DFA accepting L_B.
- Consider $M_U=(Q_A \times Q_B,\Sigma,\delta_U,(q_{0A},q_{0B}),F_U)$ where

  $\delta_U((q,q'),s) = (\delta_A(q,s), \delta_B(q',s))$ for all q,q',s

  and

  $F_U = \{ (q,q') \mid q \in F_A$ or $q' \in F_B \}$.
- $L_U = L_A \cup L_B$.
- We can write it as $F_U = (F_A \times Q_B) \cup (Q_A \times F_B)$.

  (Not the same as $F_A \times F_B$.)
- The resulting language would be the **intersection** and not the **union**. This proves that the class of regular languages is closed under intersection.

- Let $M_A=(Q_A,\Sigma,\delta_A,q_{0A},F_A)$ be a DFA accepting L_A and $M_B=(Q_B,\Sigma,\delta_B,q_{0B},F_B)$ be a DFA accepting L_B.
- Consider $M_U=(Q_A{\times}Q_B,\Sigma,\delta_U,(q_{0A},q_{0B}),F_U)$ where

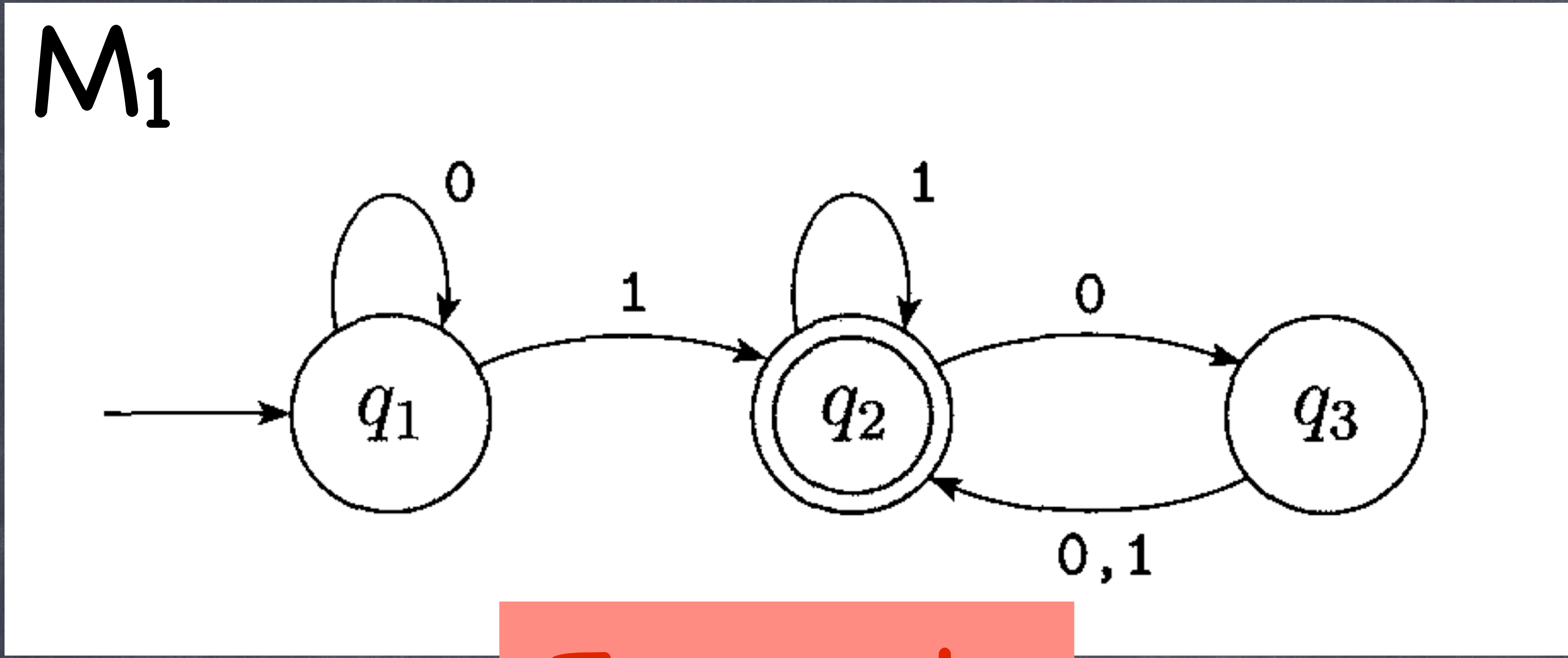
  $\delta_U((q,q'),s) = (\ \delta_A(q,s),\ \delta_B(q',s)\ )$ for all q,q',s

  and

  $F_U = \{\ (q,q') \mid q{\in}F_A \text{ or } q'{\in}F_B\ \}$.
- $L_U = L_A{\cup}L_B$.
- We can write it as $F_U = (F_A \times Q_B) \cup (Q_A \times F_B)$.

  (Not the same as $F_A \times F_B$.)
- The resulting language would be the **intersection** and not the **union**. This proves that the class of regular languages is closed under intersection.

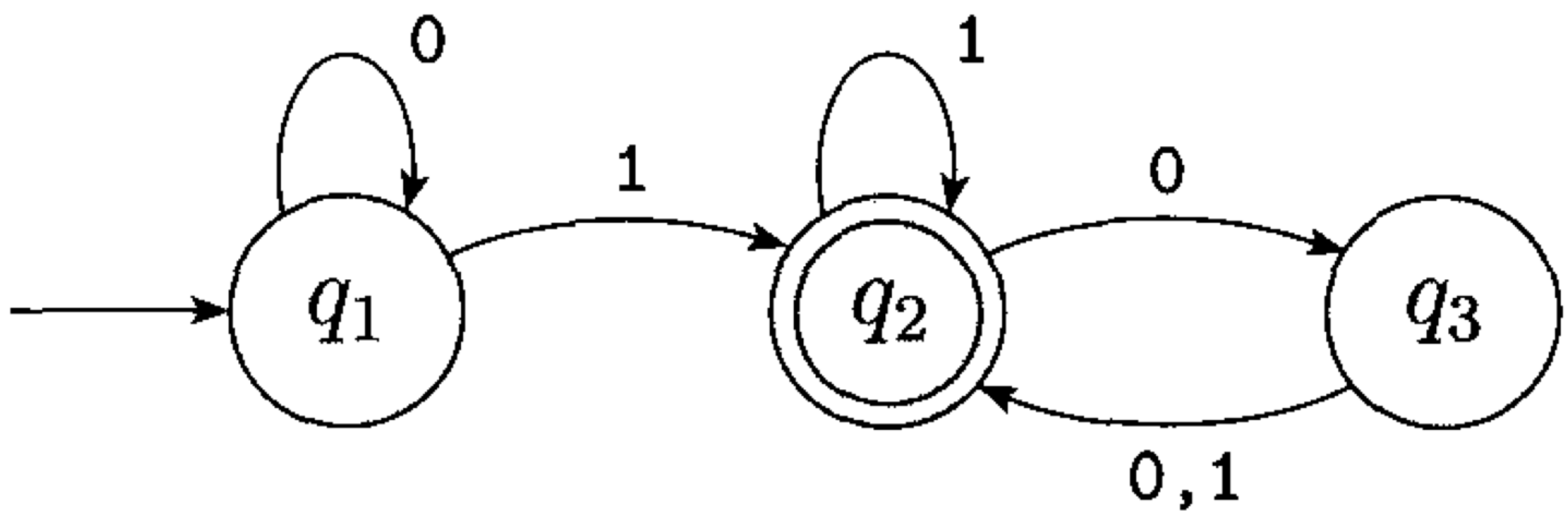
Example



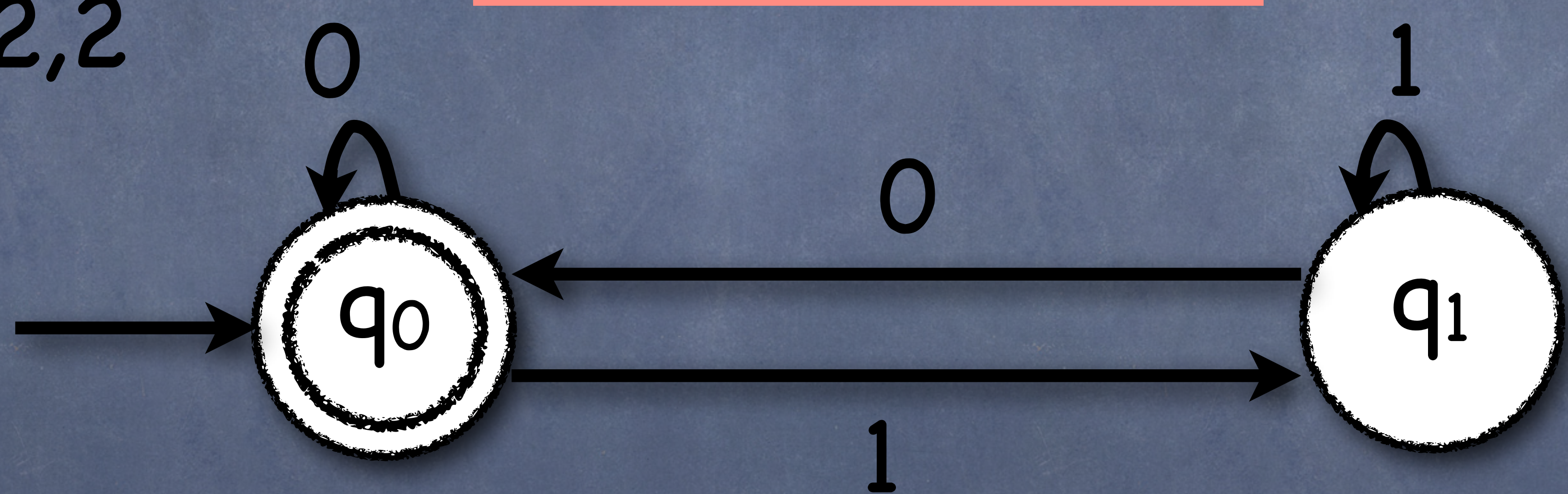
Example

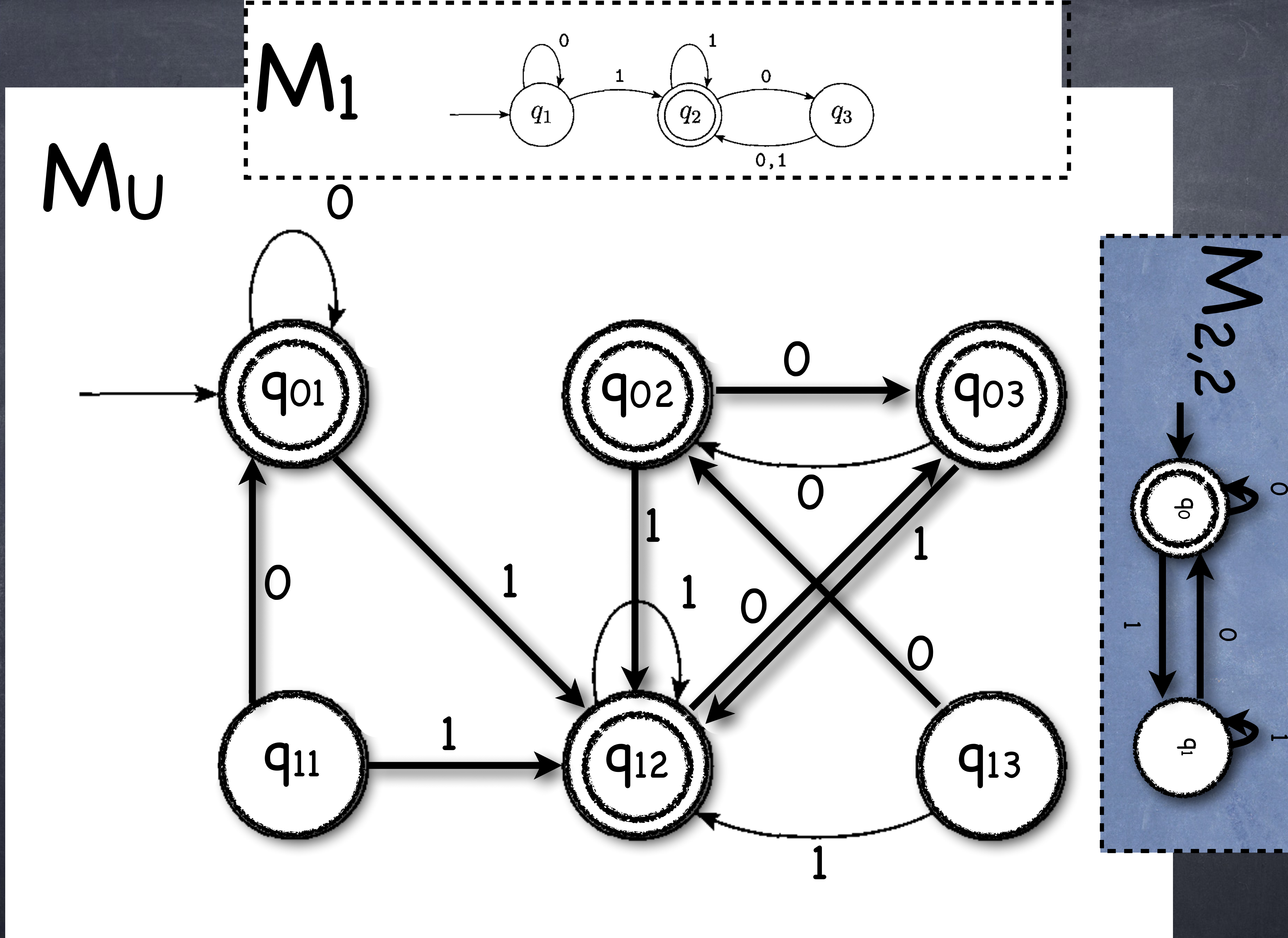
Example

M_1

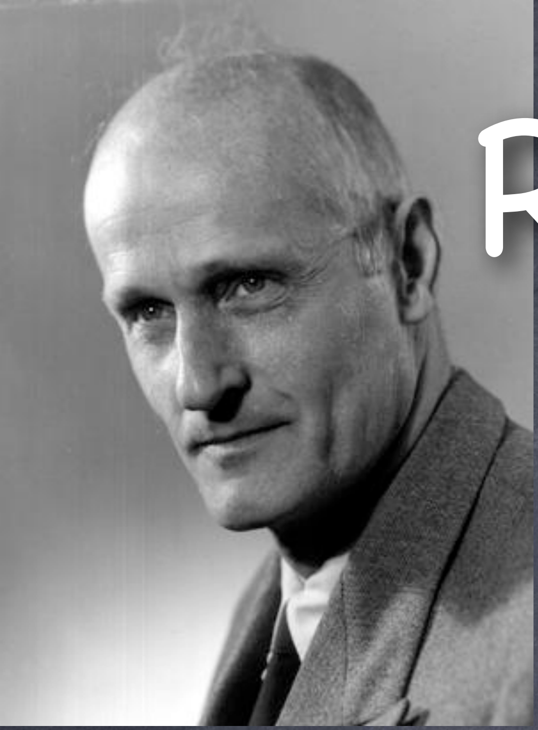


$M_{2,2}$





$L(M_U) = L(M_1) \cup L(M_{2,2})$



Regular Operations : Kleene's theorem

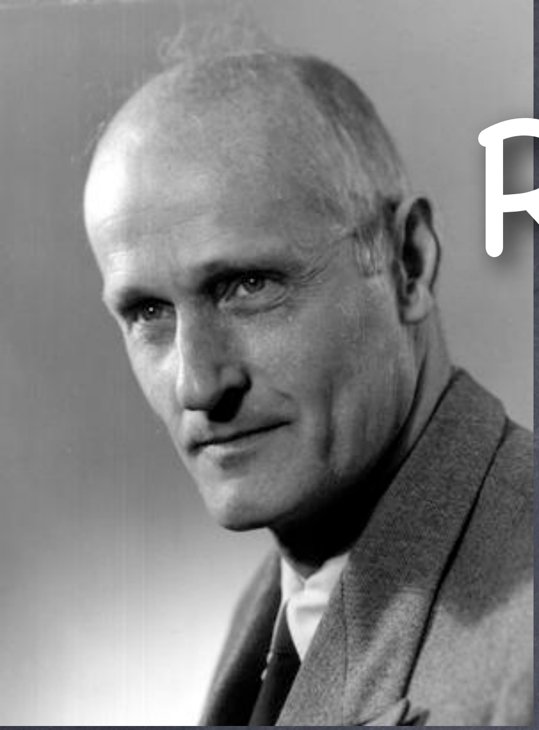
(DFA)

THEOREM 1.26

The class of regular languages is closed under the concatenation operation.

In other words, if A_1 and A_2 are regular languages then so is $A_1 \circ A_2$.

(NFA)



Regular Operations : Kleene's theorem

(DFA)

THEOREM 1.26

The class of regular languages is closed under the concatenation operation.

In other words, if A_1 and A_2 are regular languages then so is $A_1 \circ A_2$.

(NFA)

THEOREM 1.47

The class of regular languages is closed under the concatenation operation.

Regular Operations : Kleene's theorem (NFA)

Regular Operations : Kleene’s theorem (NFA)

THEOREM 1.45

The class of regular languages is closed under the union operation.

Regular Operations : Kleene’s theorem (NFA)

THEOREM 1.47

The class of regular languages is closed under the concatenation operation.

THEOREM 1.45

The class of regular languages is closed under the union operation.

Regular Operations : Kleene's theorem (NFA)

THEOREM 1.49

The class of regular languages is closed under the star operation.

THEOREM 1.47

The class of regular languages is closed under the concatenation operation.

THEOREM 1.45

The class of regular languages is closed under the union operation.

Non-Deterministic Finite Automata

Non-Deterministic Finite Automata

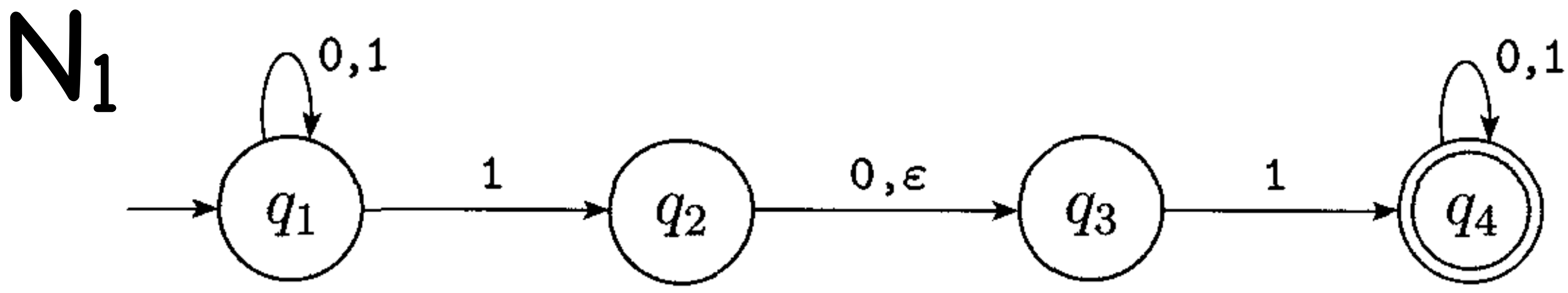


FIGURE 1.27
The nondeterministic finite automaton N_1

Non-Deterministic Finite Automata

FIGURE 1.29
The computation of N_1 on input 010110

Non-Deterministic Finite Automata

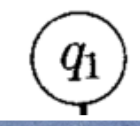


FIGURE 1.29
The computation of N_1 on input 010110

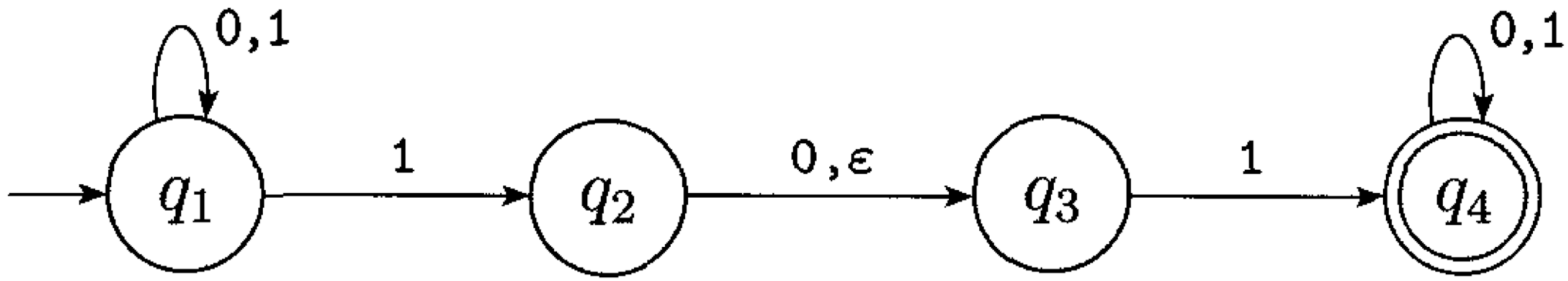
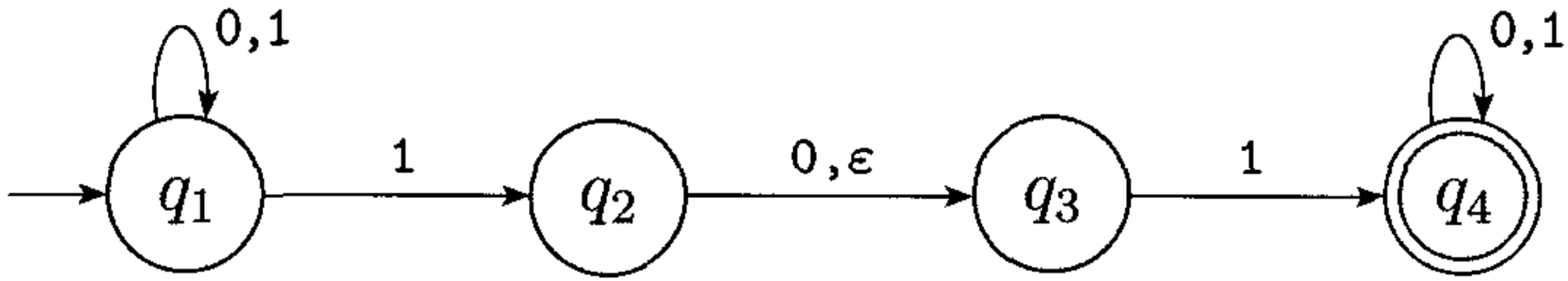
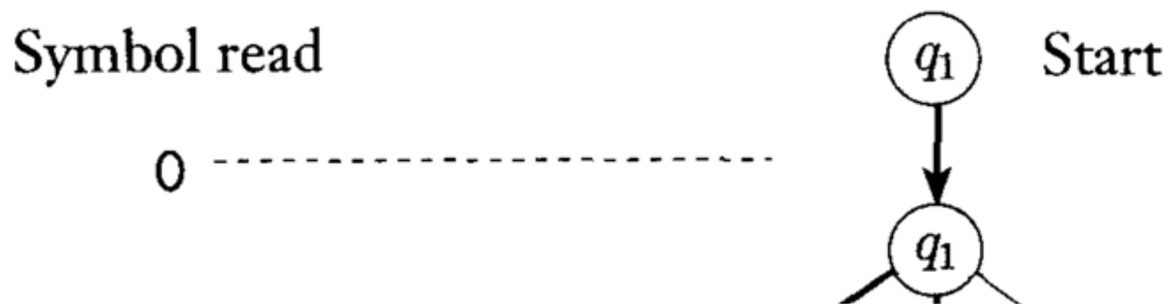
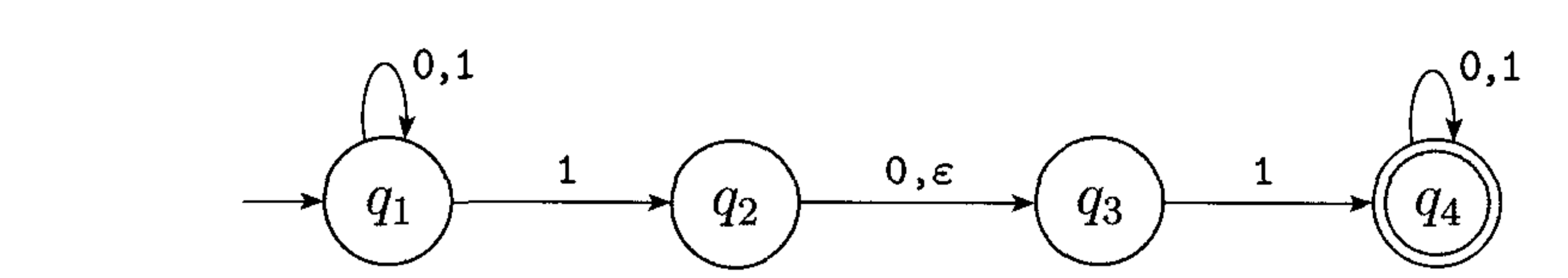
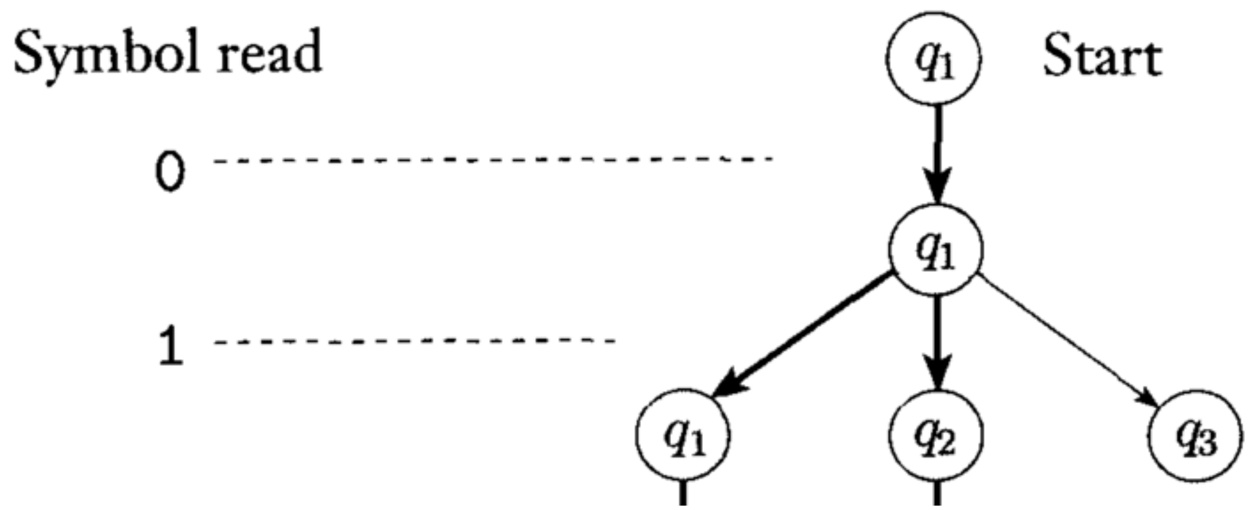


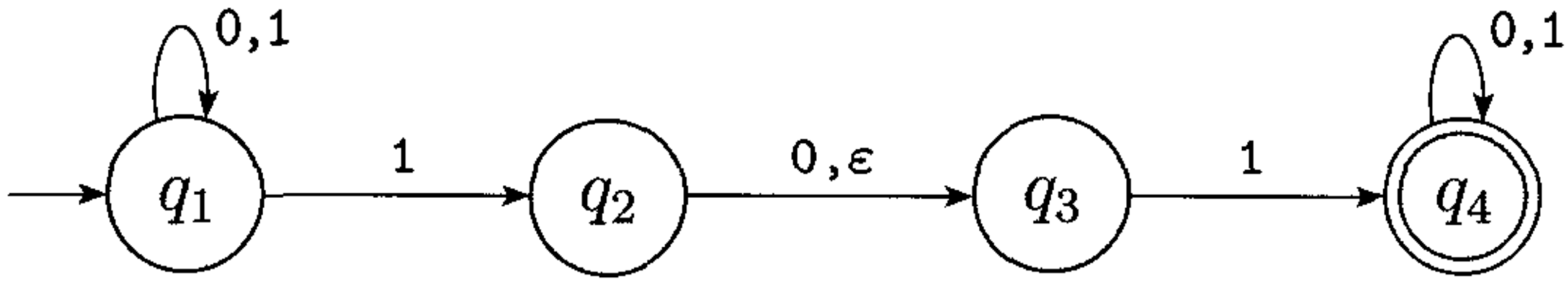
FIGURE 1.29
The computation of N_1 on input 010110

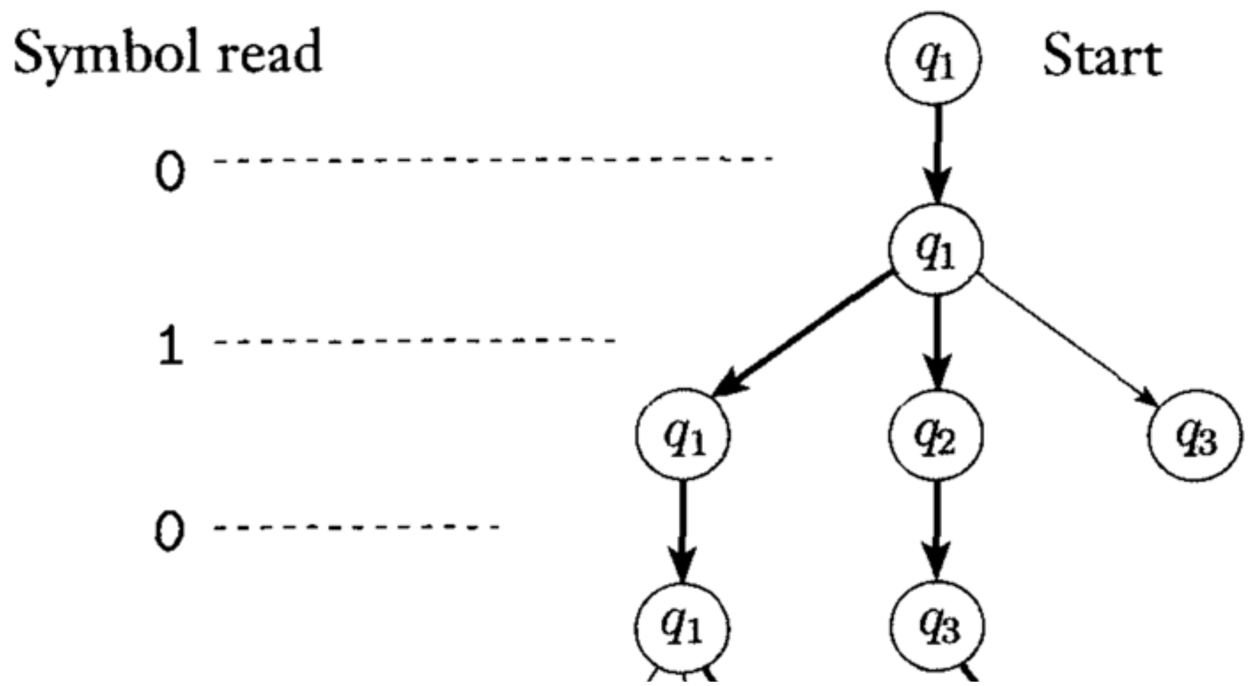

0,1
1
0,ε
1
0,1
q_1
q_2
q_3
q_4

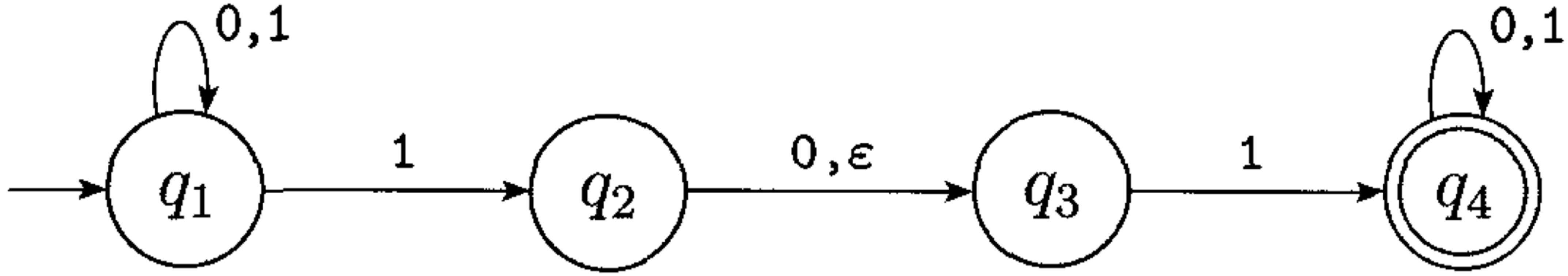

Symbol read
0
q_1
Start
q_1

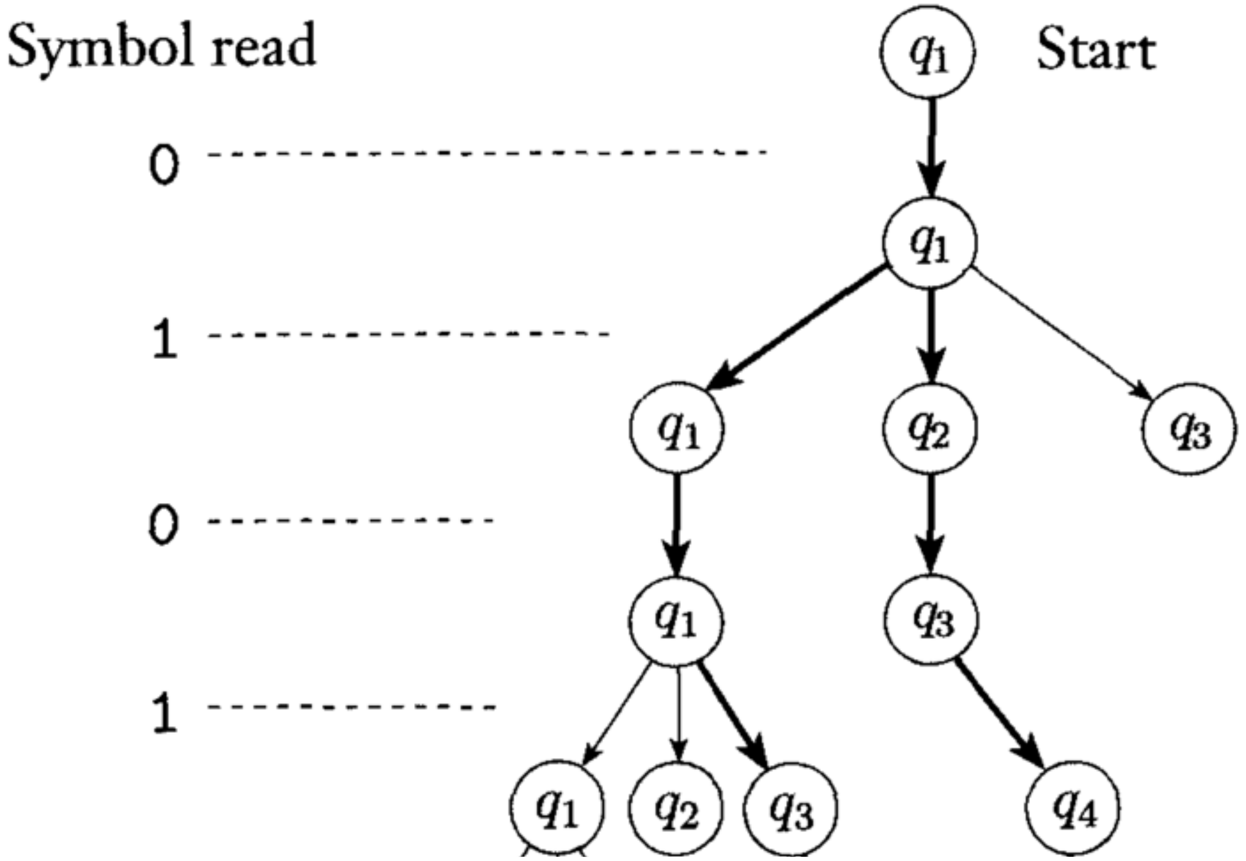

0,1
1
0,ε
1
0,1
q_1
q_2
q_3
q_4

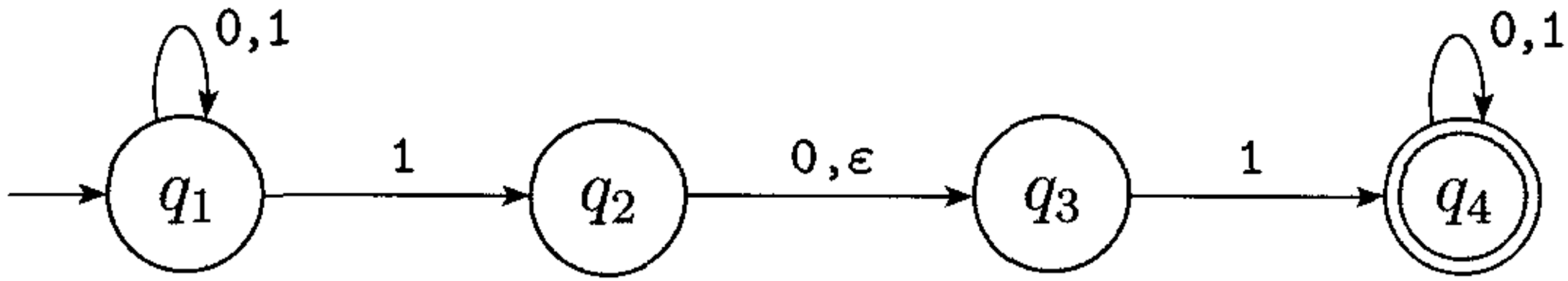

Symbol read
Start
0
1
q_1
q_1
q_1
q_2
q_3


0,1
1
0,ε
1
0,1
q_1
q_2
q_3
q_4


Symbol read
Start
0
1
0
q_1
q_1
q_1
q_2
q_3
q_1
q_3


0,1
1
0,ε
1
0,1
q_1
q_2
q_3
q_4


Symbol read
Start
0
1
0
1
q_1
q_2
q_3
q_4



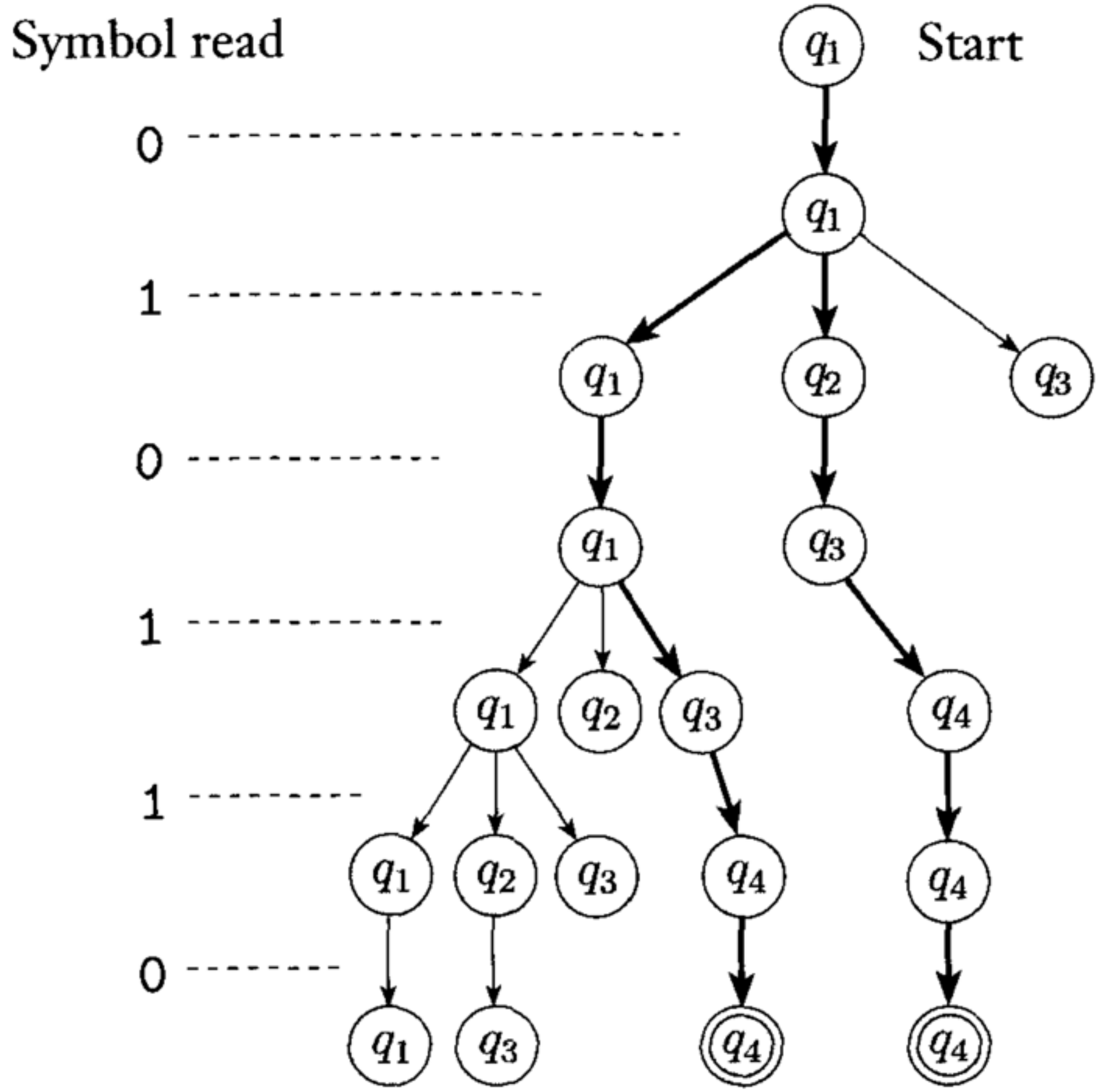
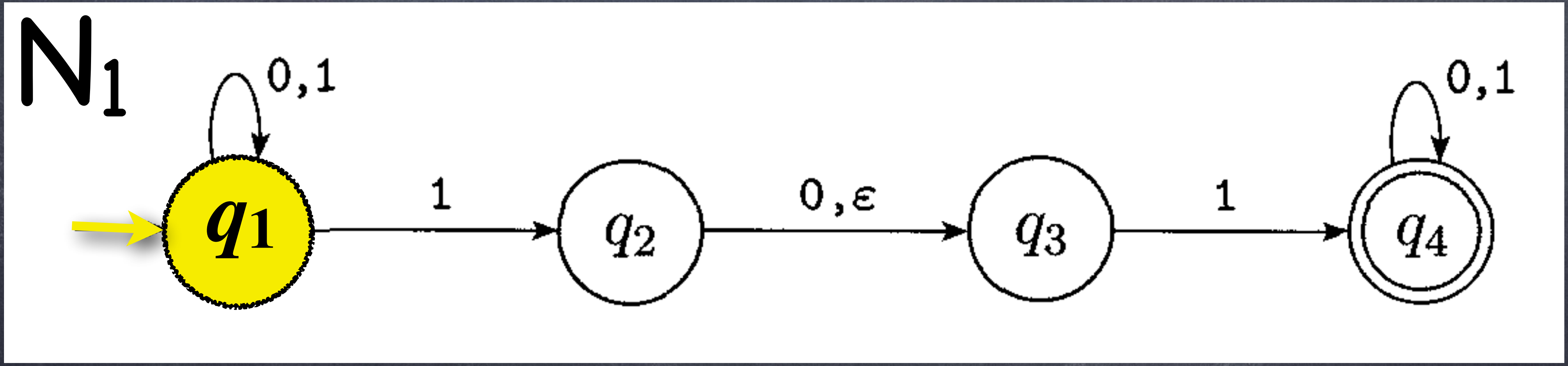


FIGURE 1.29
The computation of N_1 on input 010110

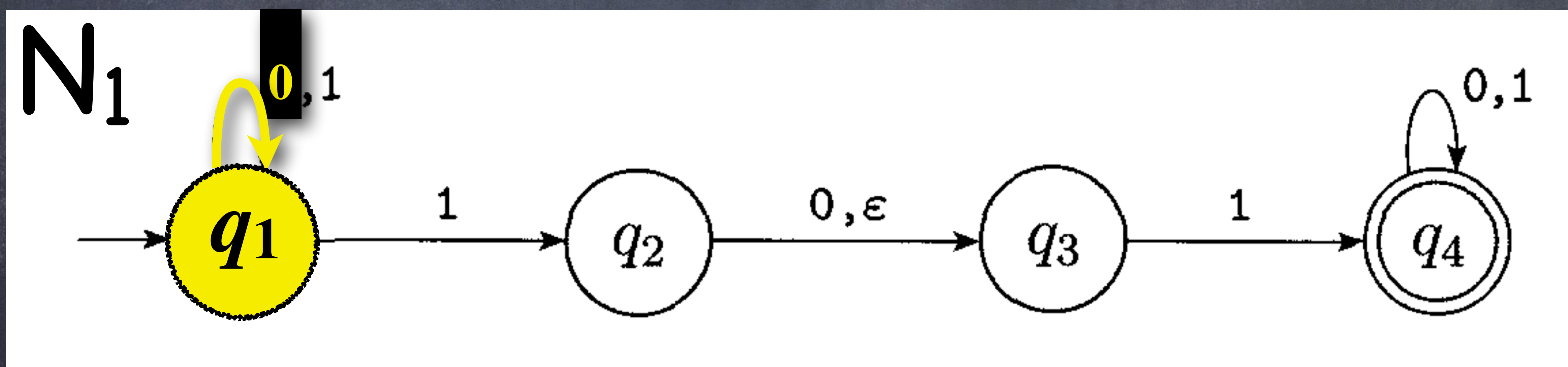
Non-Deterministic Finite Automata

010110



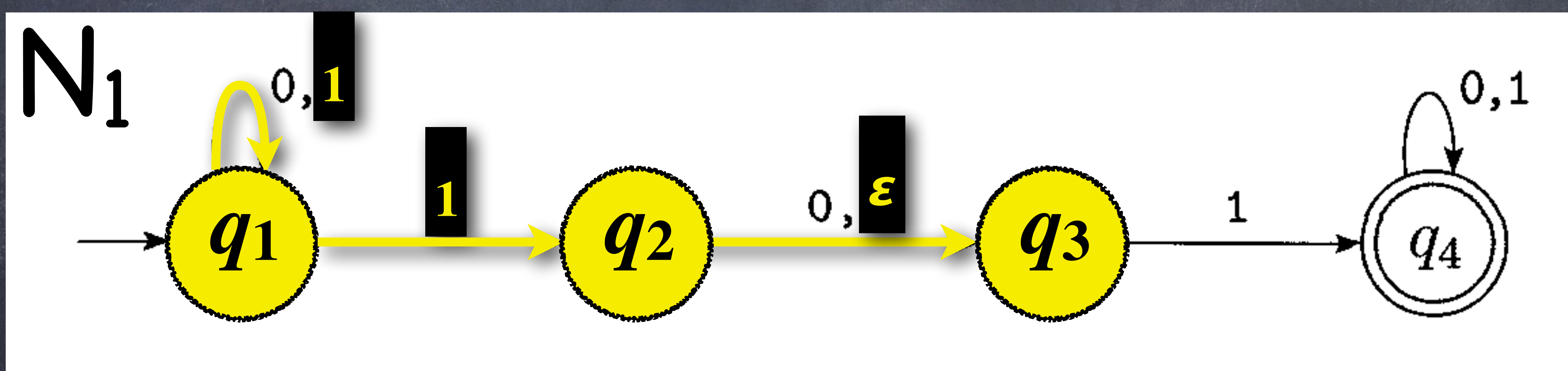
Non-Deterministic Finite Automata

010110



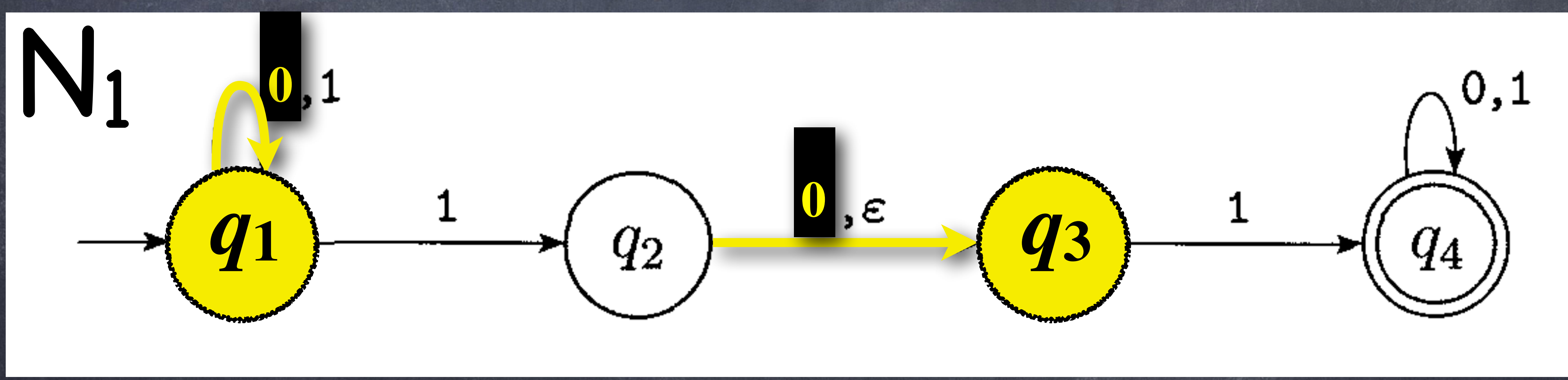
Non-Deterministic Finite Automata

010110



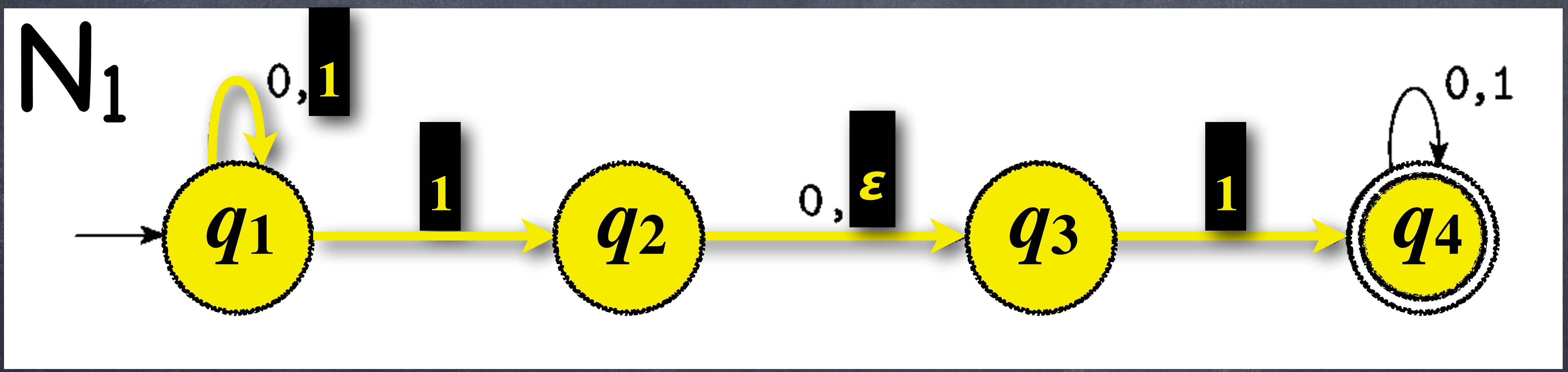
Non-Deterministic Finite Automata

010110



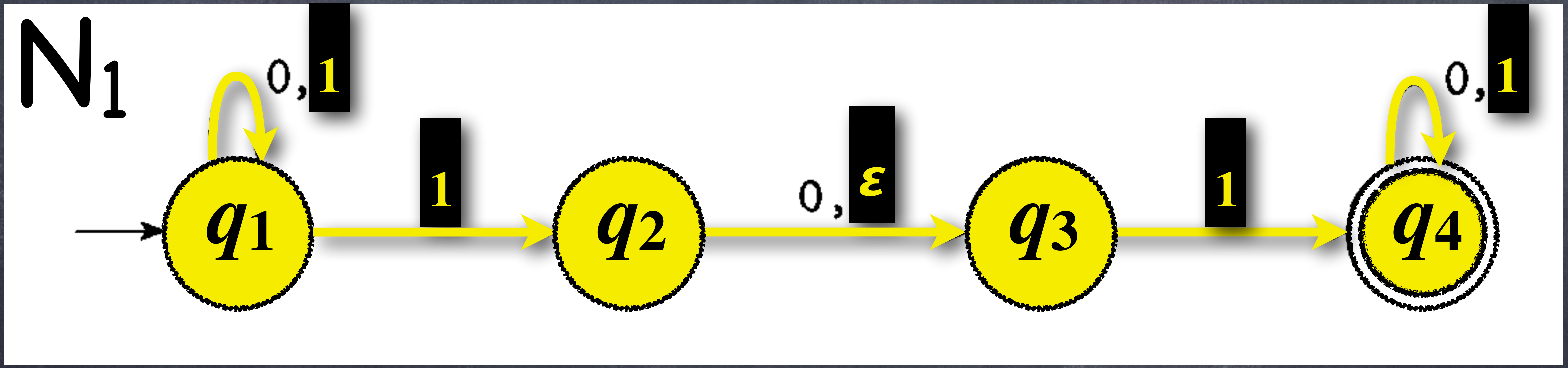
Non-Deterministic Finite Automata

010110



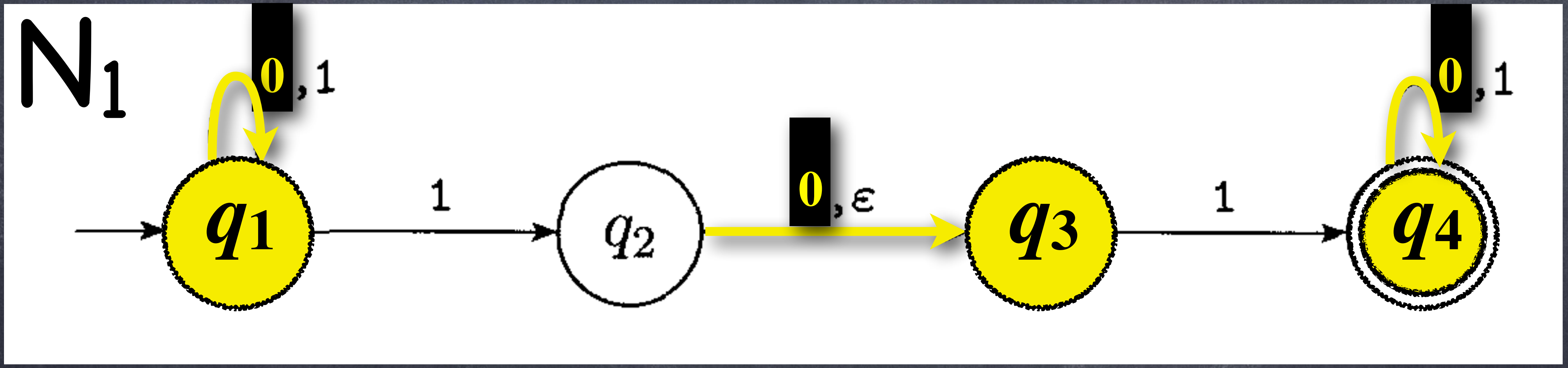
Non-Deterministic Finite Automata

010110



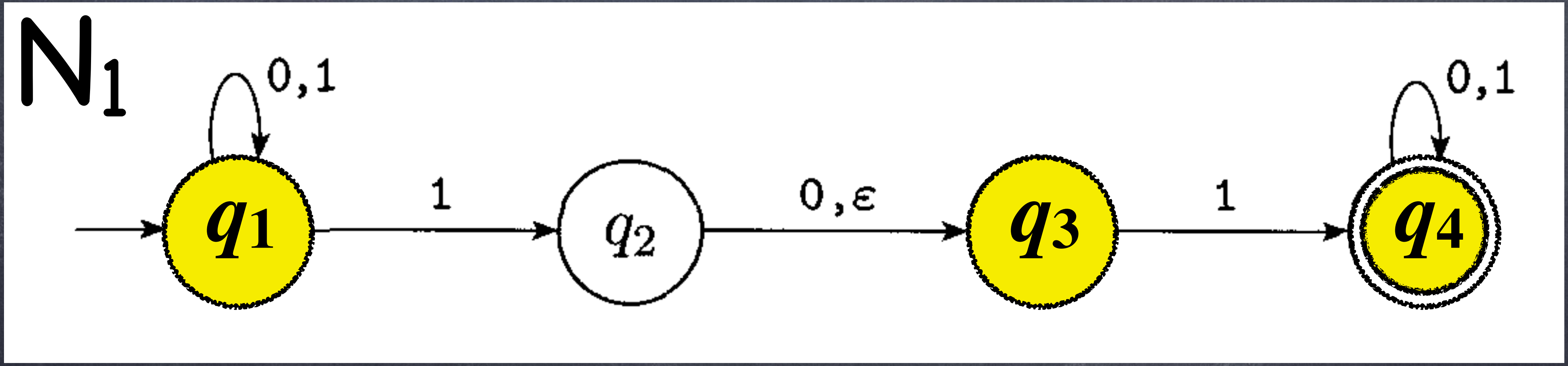
Non-Deterministic Finite Automata

010110



Nondeterministic Finite Automata

010110



$010110 \in L_{N_1} \Leftrightarrow$

$\{q_1, q_3, q_4\} \cap F = \{q_4\} \neq \varnothing$

Definition of NFA

DEFINITION 1.37

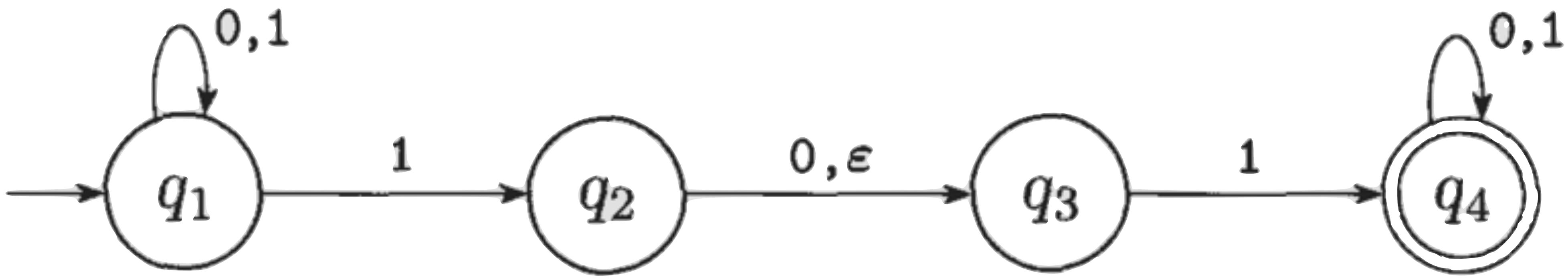
A ***nondeterministic finite automaton*** is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

1. Q is a finite set of states,
2. Σ is a finite alphabet,
3. $\delta \colon Q \times \Sigma_{\varepsilon} \longrightarrow \mathcal{P}(Q)$ is the transition function,
4. $q_0 \in Q$ is the start state, and
5. $F \subseteq Q$ is the set of accept states.

$\Sigma_{\varepsilon} = \Sigma \cup \{\varepsilon\}$ $\quad$ $\mathcal{P}(Q) = \{ S : S \subseteq Q \}$

EXAMPLE 1.38

Recall the NFA N_1:



The formal description of N_1 is $(Q, \Sigma, \delta, q_1, F)$, where

1. $Q = \{q_1, q_2, q_3, q_4\}$,
2. $\Sigma = \{0,1\}$,
3. δ is given as

	0	1	ε
q_1	$\{q_1\}$	$\{q_1, q_2\}$	$\emptyset$
q_2	$\{q_3\}$	$\emptyset$	$\{q_3\}$
q_3	$\emptyset$	$\{q_4\}$	$\emptyset$
q_4	$\{q_4\}$	$\{q_4\}$	$\emptyset$

,

4. q_1 is the start state, and
5. $F = \{q_4\}$.

Definition of NFA

- Let $N = (Q,\Sigma,\delta,q_0,F)$ be a nondeterministic finite state automaton and let $w=w_1w_2...w_n$ ($n\geq 0$) be a string where each symbol $w_i \in \Sigma$.

- N <u>**accepts**</u> w if $\exists\ m\geq n$, $\exists\ s_0,s_1,\ldots,s_m$ and $\exists\ y_1y_2...y_m = w$, with each $y_i \in \Sigma_\varepsilon$ s.t.
  1. $s_0 = q_0$
  2. $s_{i+1} \in \delta(s_i,y_{i+1})$ for $i = 0 \ldots m-1$, and
  3. $s_m \in F$

COMP-330
Theory of Computation

Fall 2019 -- Prof. Claude Crépeau

Lec. 4 : DFAs, NFAs + Kleene's theorem