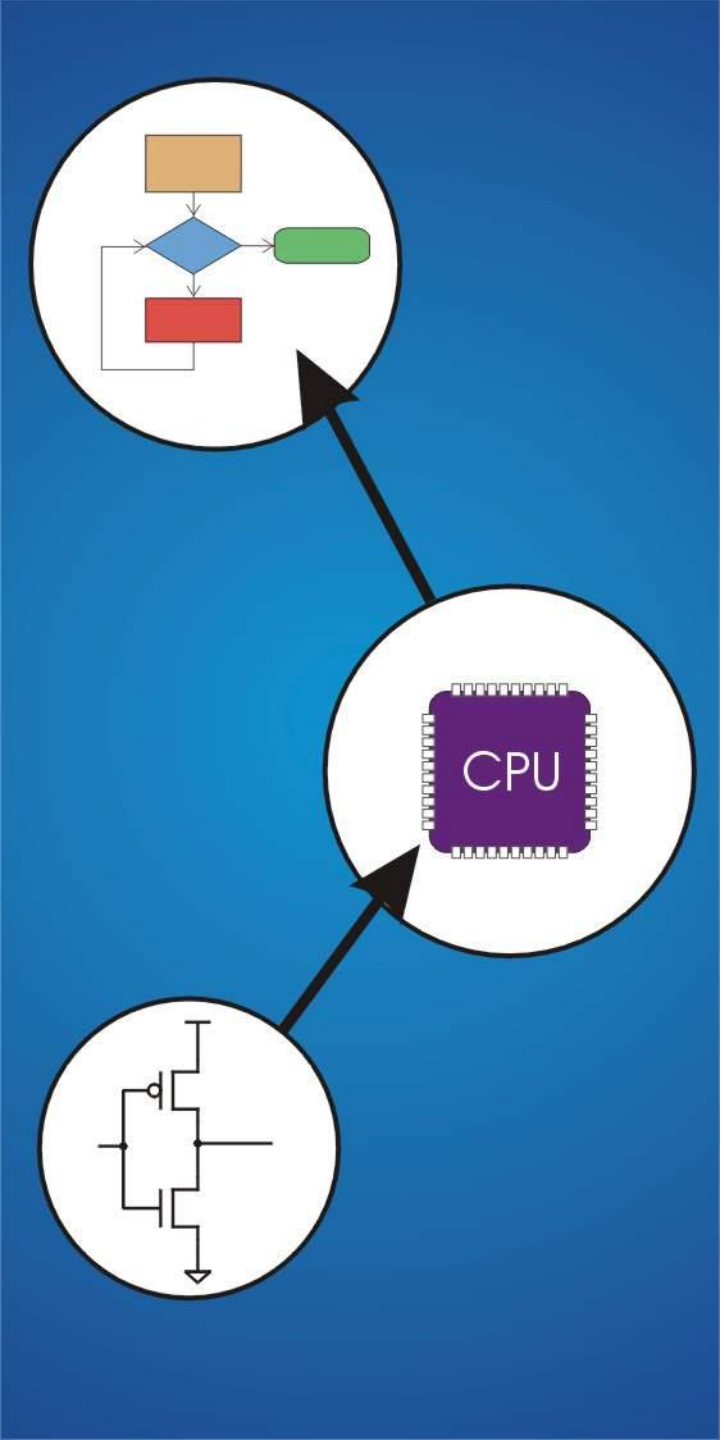


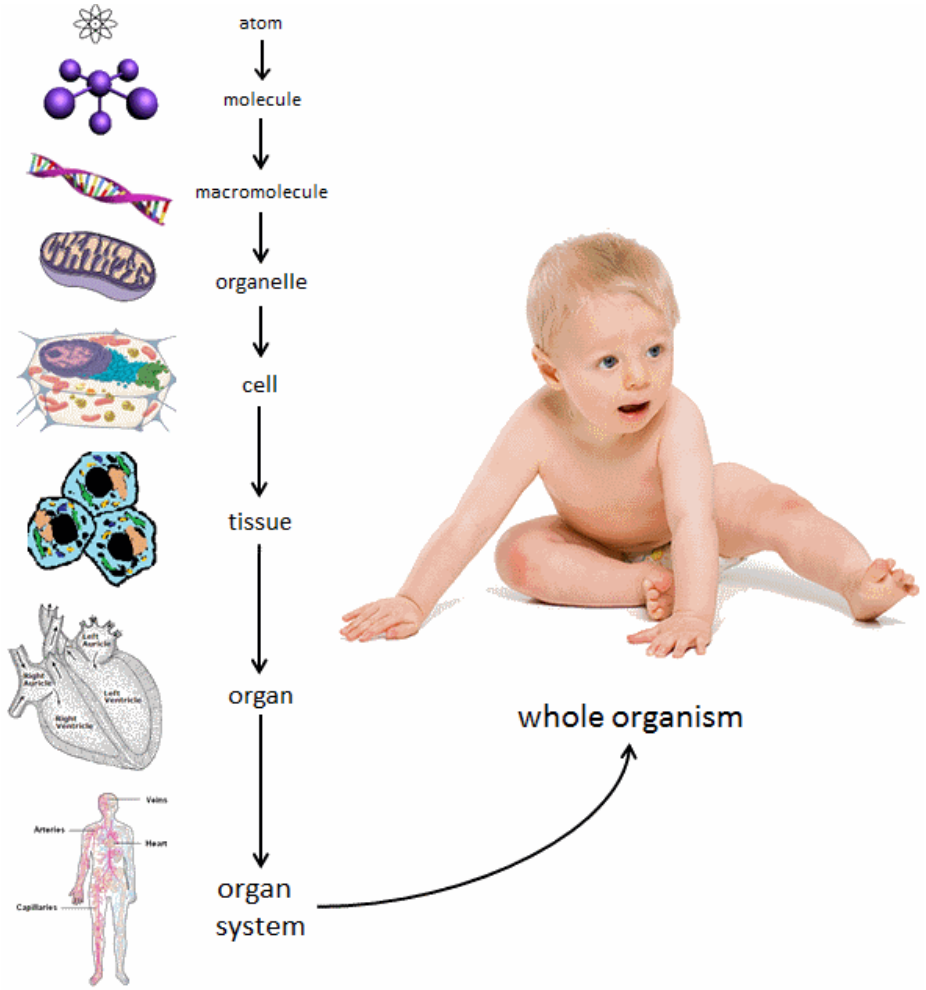
Chapter 1
Welcome Aboard

Abstraction

Interface

Source: http://static.usnews.rankingsandreviews.com/images/Auto/izmo/365609/2014_hyundai_elantra_gt_dashboard.jpg
http://www.ridelust.com/wp-content/uploads/2012/12/Engine2.jpg
http://www.plugincars.com/sites/default/files/Volkswagen-Golf-Variant-twinDRIVE-under-the-hood.jpg
http://wpmedia.driving.ca/2015/02/img_4777.jpg?quality=70&strip=all&w=960&h=480&crop=1

Levels of Abstraction (Biological System)



Source: https://media.studyisland.com/pics/40040_bio_organization.gif

Levels of Abstraction (Computer)

Problems

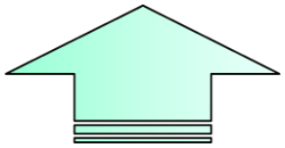
Algorithms

Language

Instruction Set Architecture – – – – – Hardware/Software Interface

Microarchitecture

Circuits



Devices

Universal Computing Device

All computers, given enough time and memory, are capable of computing exactly the same things.

Embedded Processor

=

=

Supercomputer

Then what is the simplest possible computing device?

Source: http://images.anandtech.com/reviews/storage/Samsung/SSD830/_DSC2164.jpg
http://www.miscw.com/wp-content/uploads/2016/09/Samsung-Smartphone.jpg
http://koreatechblog.com/wp-content/uploads/2015/02/lg-ultrapc-gram-14zd950-gx58k-1157540_2.jpg
https://4.imimg.com/data4/CO/YS/MY-29352968/samsung-desktop-computer-500x500.jpg
http://oiger.de/wp-content/uploads/Bull-Supercomputer.jpg

Turing Machine

Mathematical model of a device that can perform any computation – Alan Turing (1937)

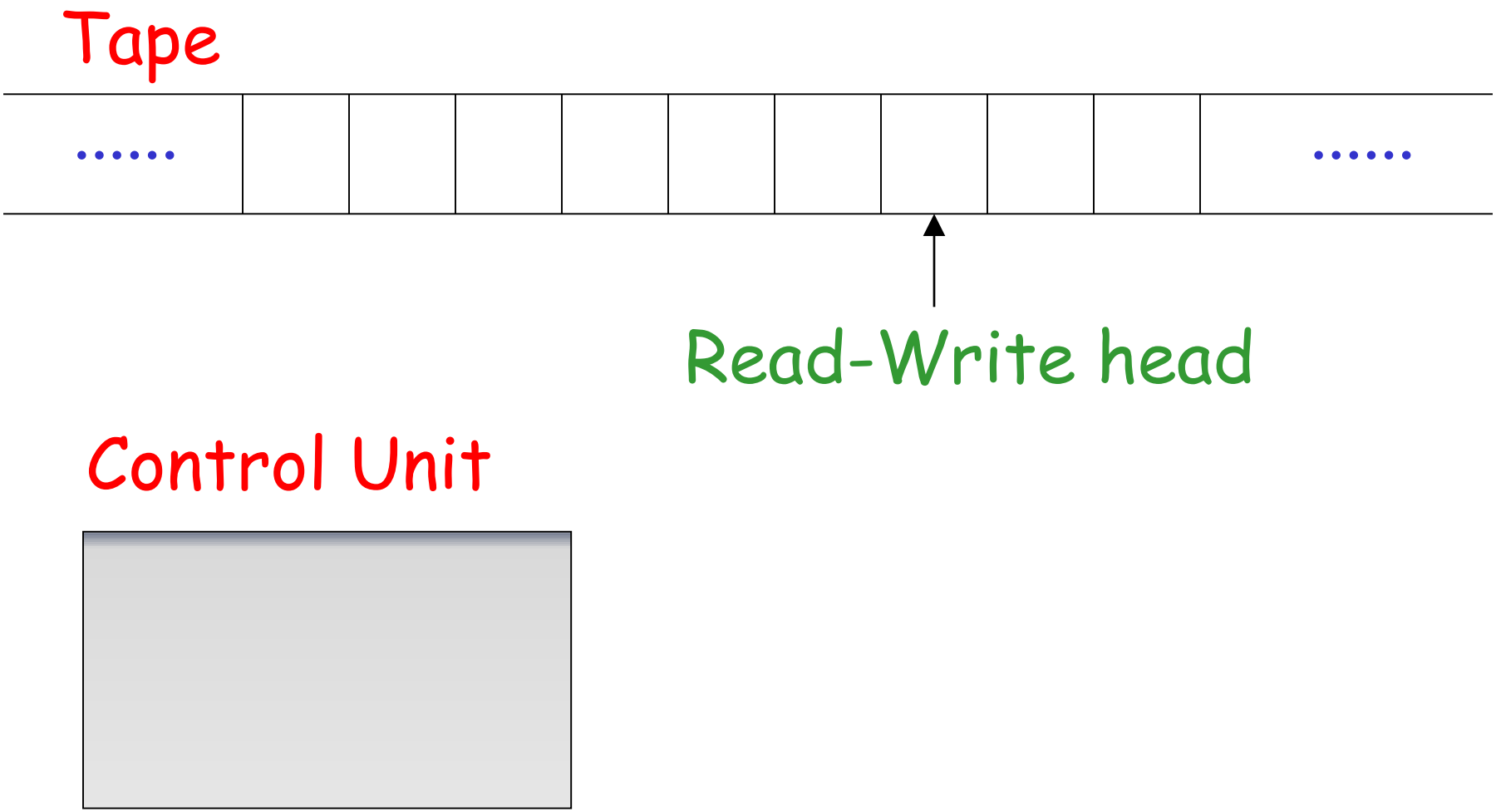
Every computation can be performed by some Turing machine. ***(Turing's thesis)***

(그럼 이 세상에 계산하지 못하는 것도 있나? Yes…Halting Problem…We'll discuss it later)

For more info about Turing machines, see http://www.wikipedia.org/wiki/Turing_machine/

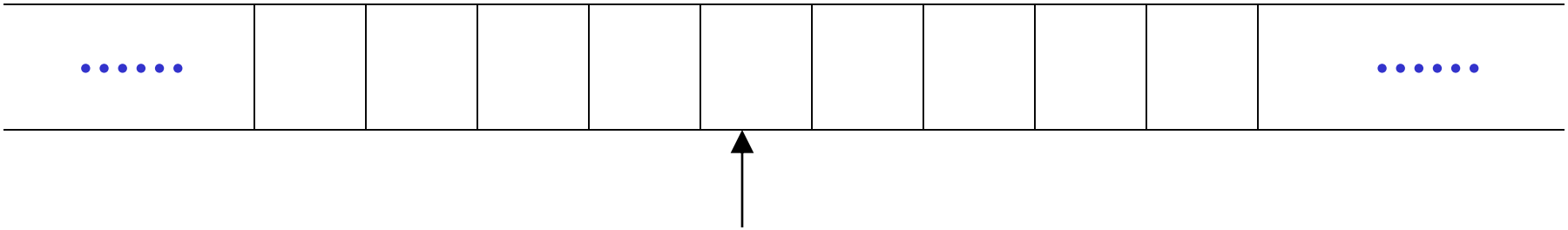
For more about Alan Turing, see http://www.turing.org.uk/turing/

A Turing Machine



The Tape

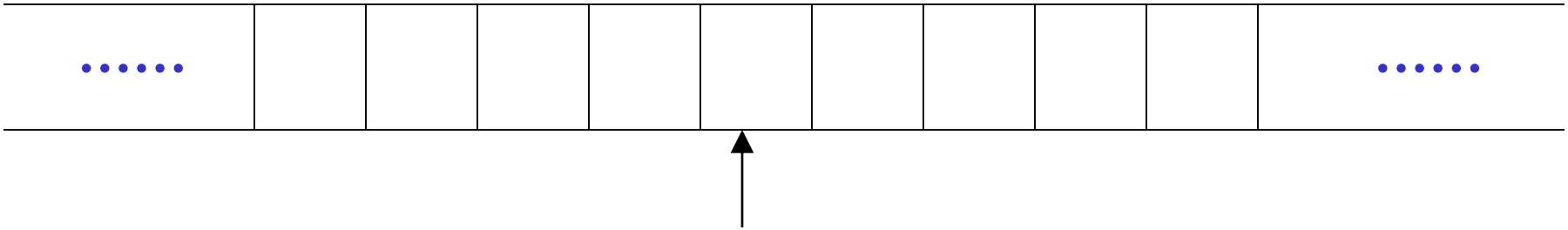
No boundaries -- infinite length



Read-Write head

The head moves Left or Right

Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/



Read-Write head

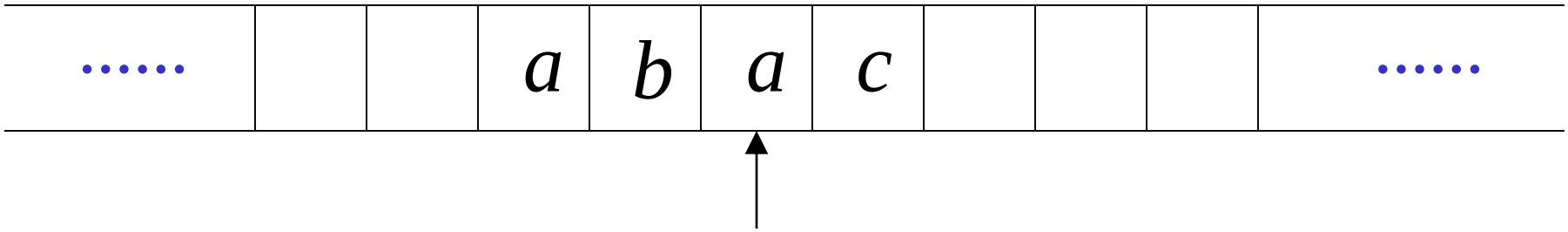
The head at each transition (time step):

1. Reads a symbol
2. Writes a symbol
3. Moves Left or Right

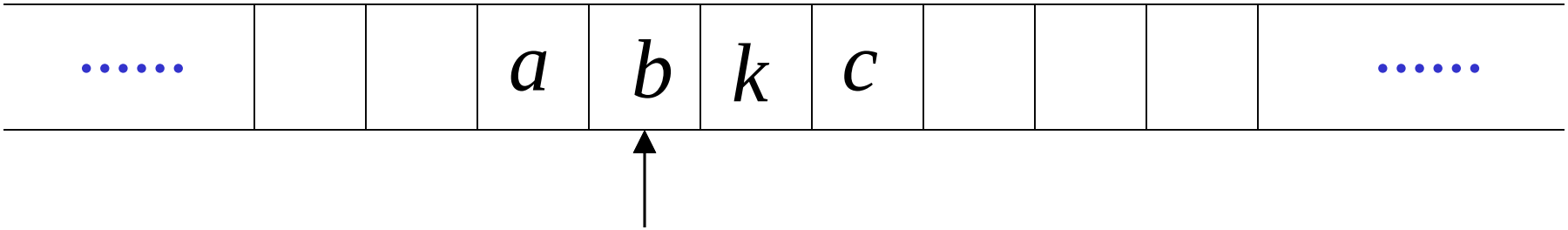
Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

Example:

Time 0



Time 1



1. Reads a
2. Writes k
3. Moves Left

Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

Computation Example

The function $f(x, y) = x + y$ is computable

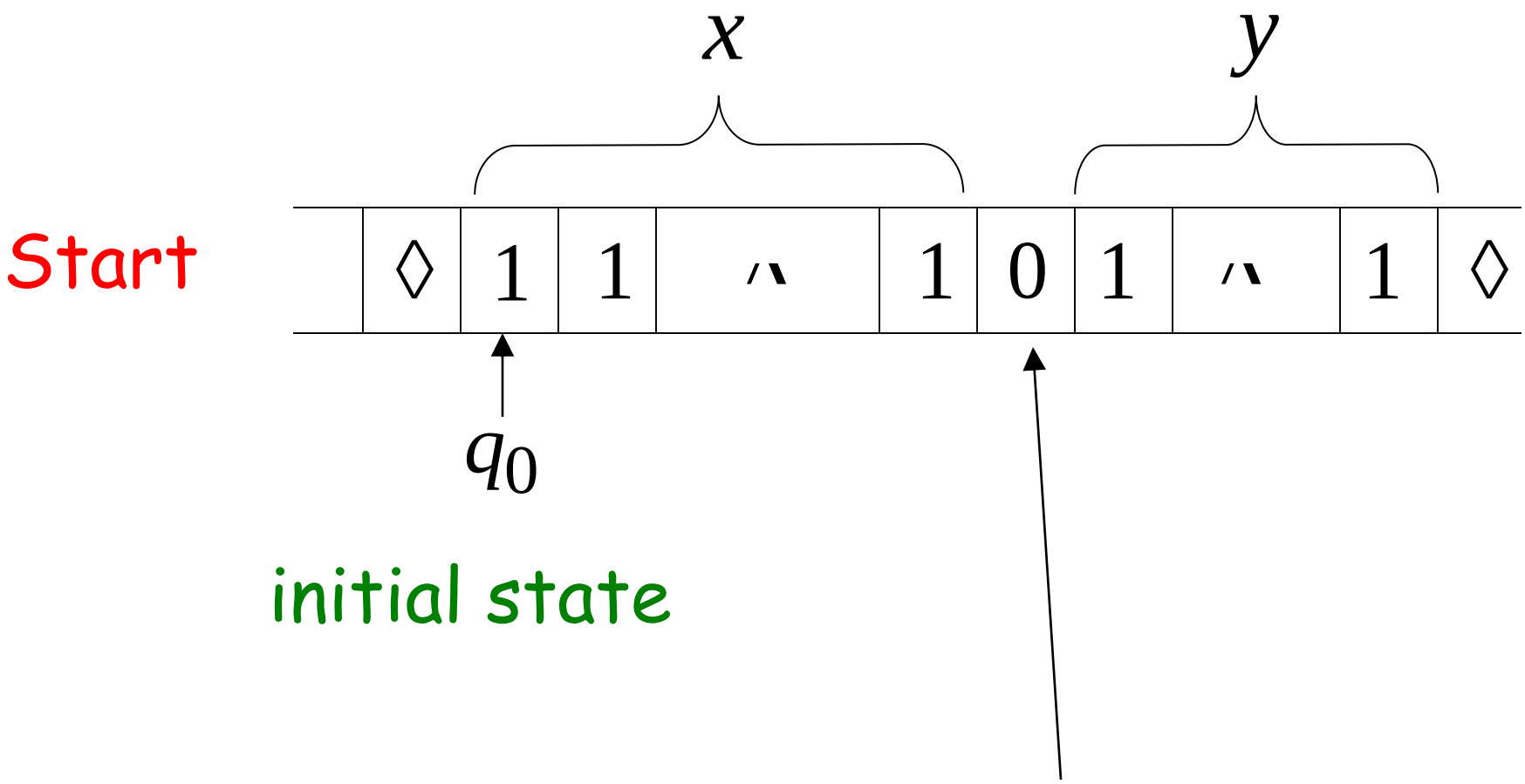
x, y are integers

Turing Machine:

Input string: $x0y$ unary

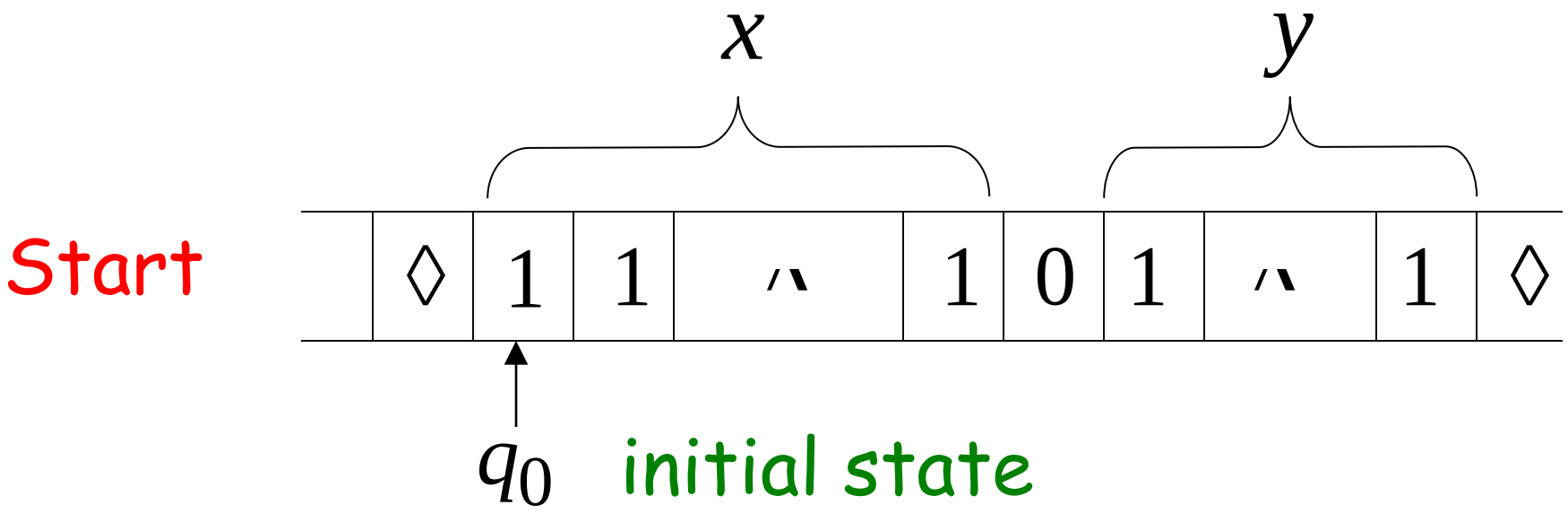
Output string: $xy0$ unary

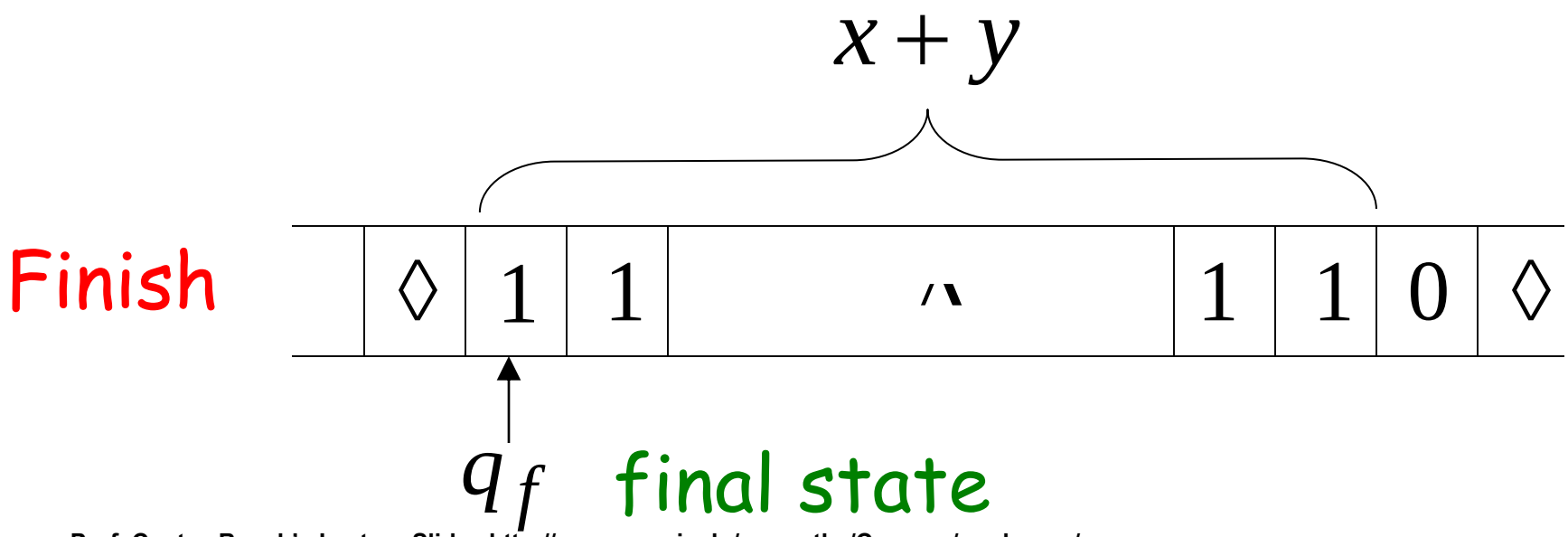
Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/



The 0 is the delimiter that separates the two numbers

Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/





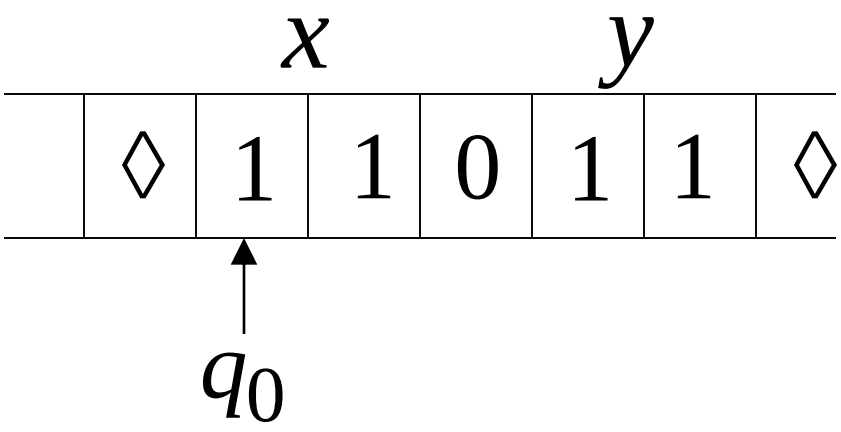
Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

Execution Example:

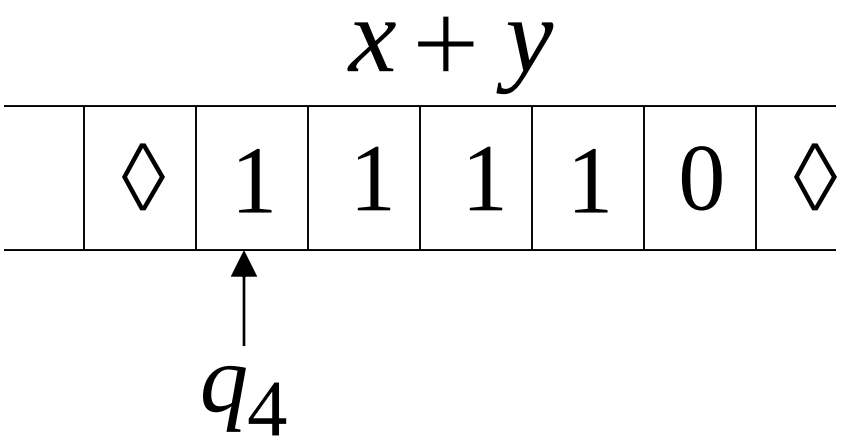
$x = 11$ (=2)

$y = 11$ (=2)

Time 0



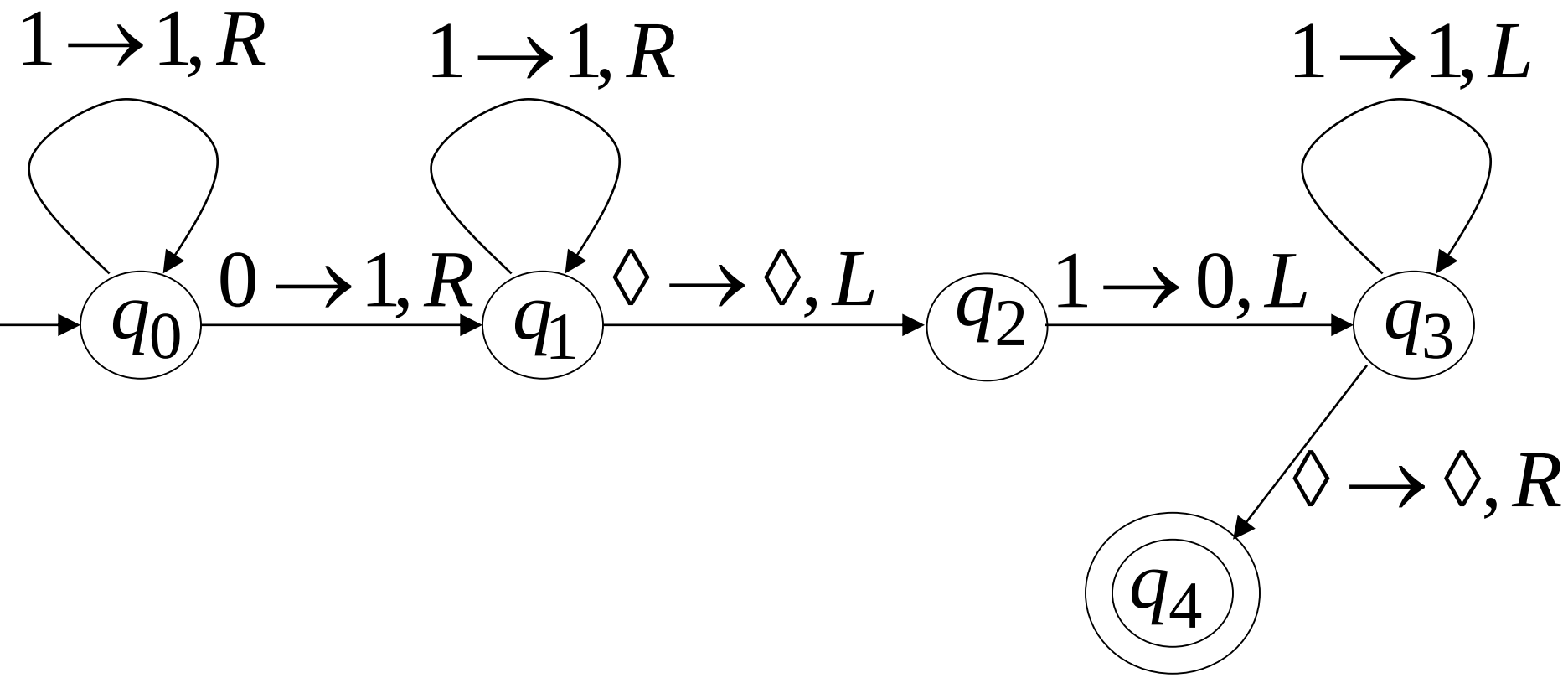
Final Result



Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

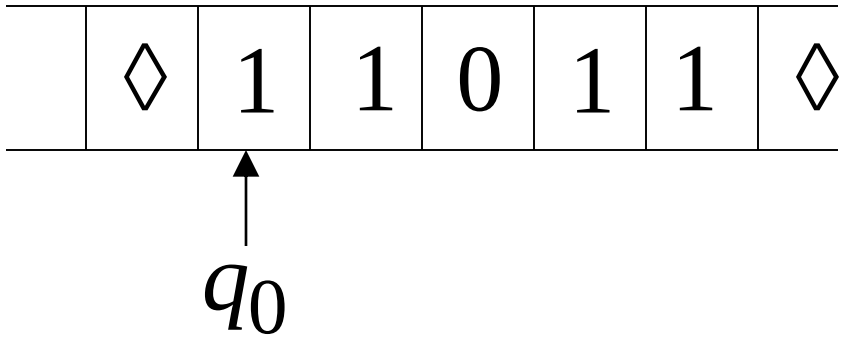
Turing machine for function $f(x, y) = x + y$

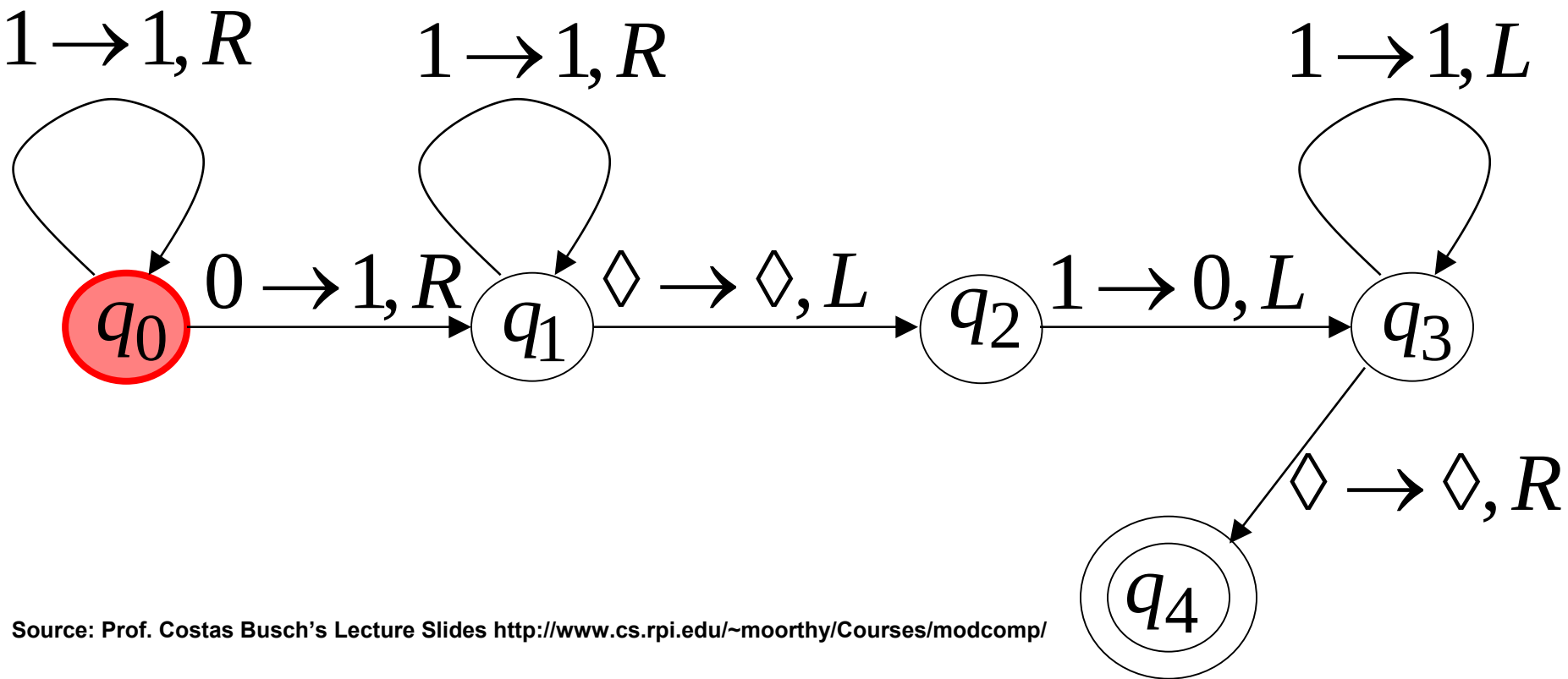
Control Unit



Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

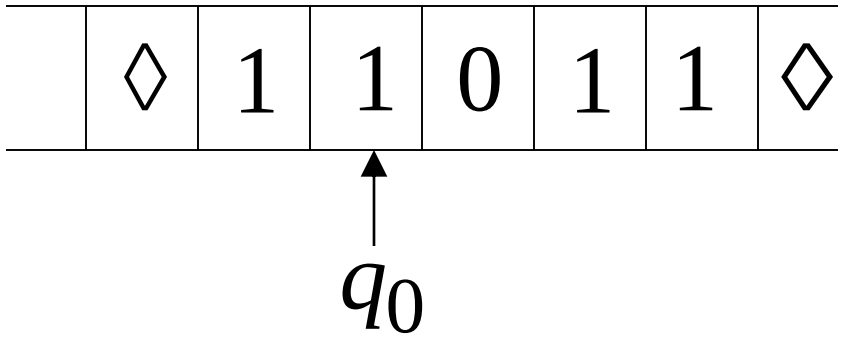
Time 0

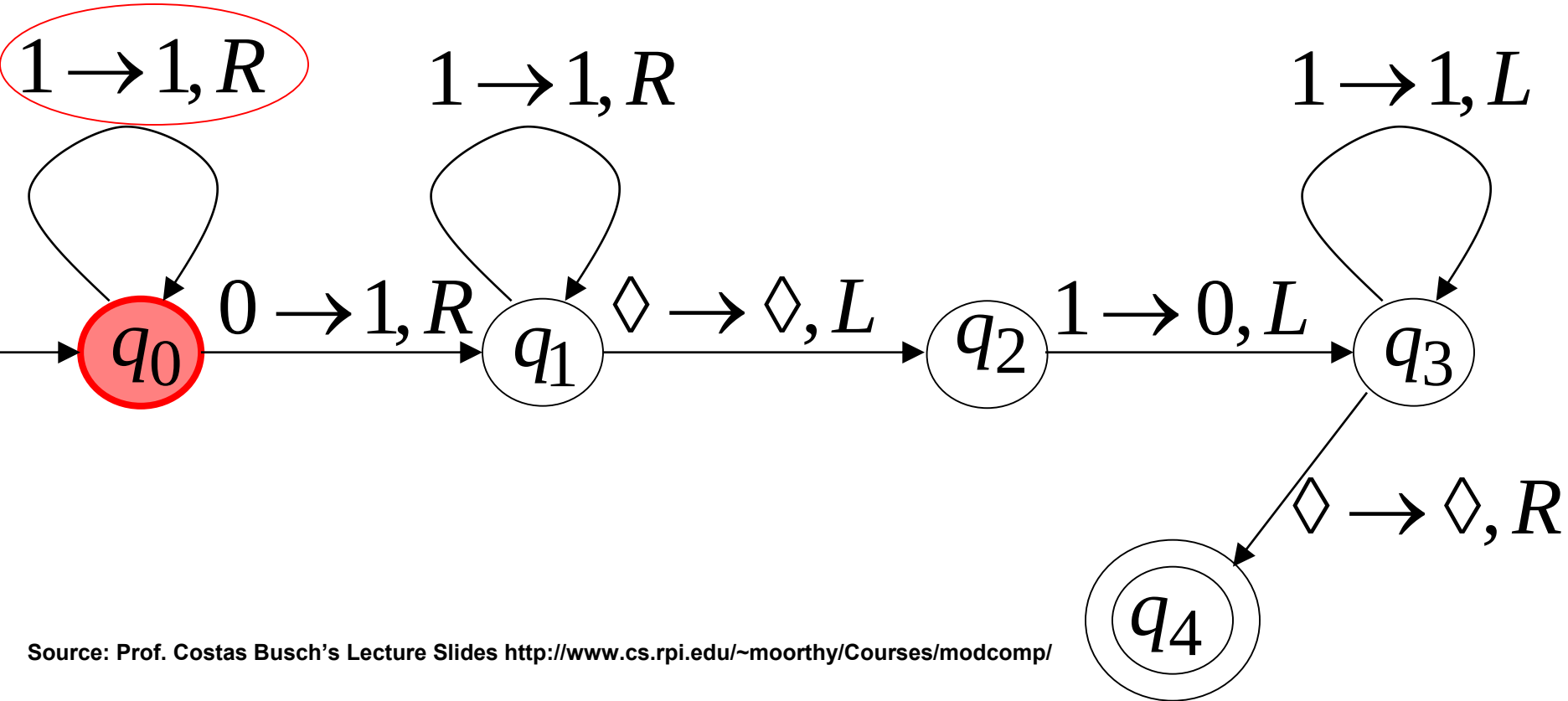




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

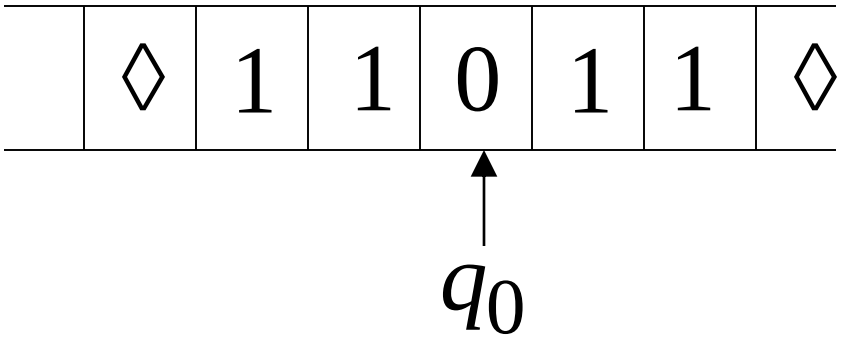
Time 1

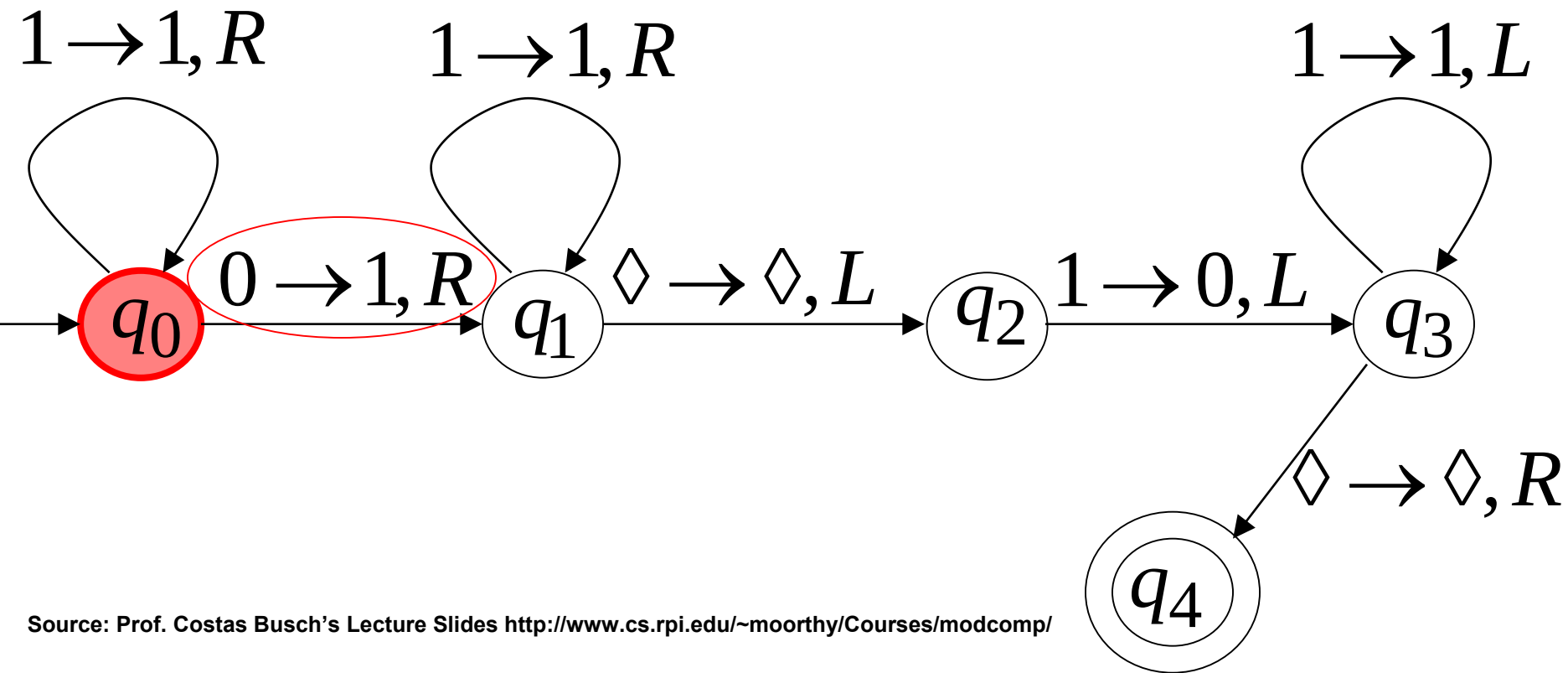




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

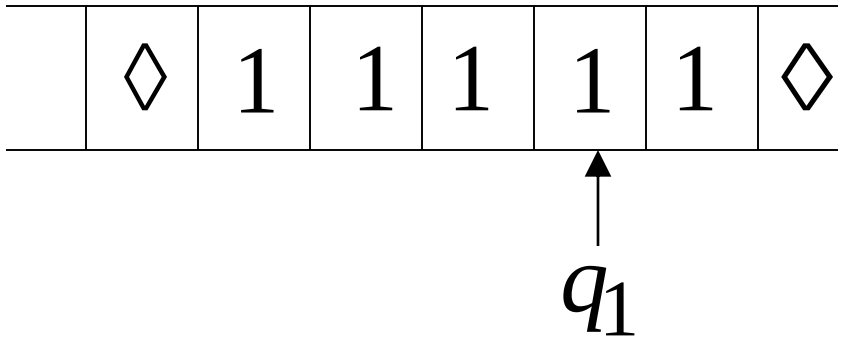
Time 2

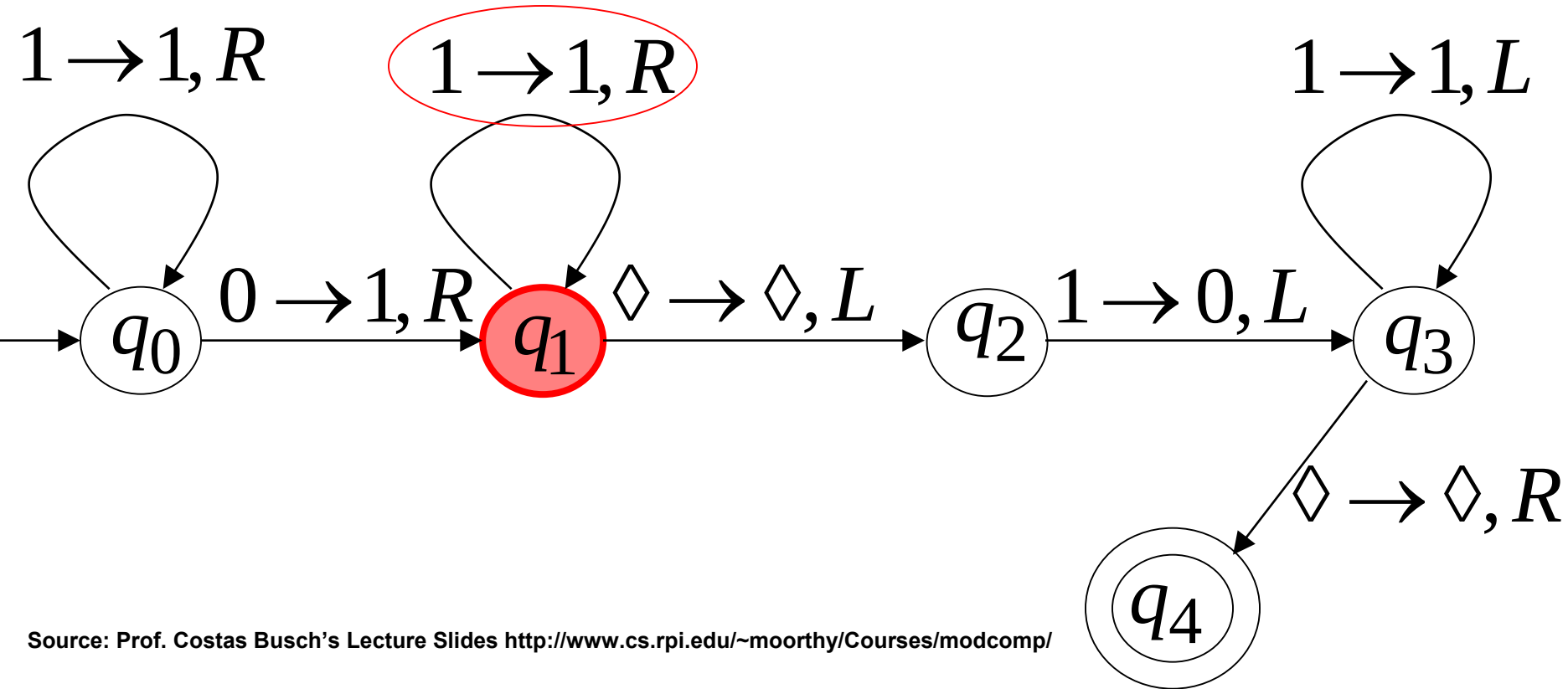




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

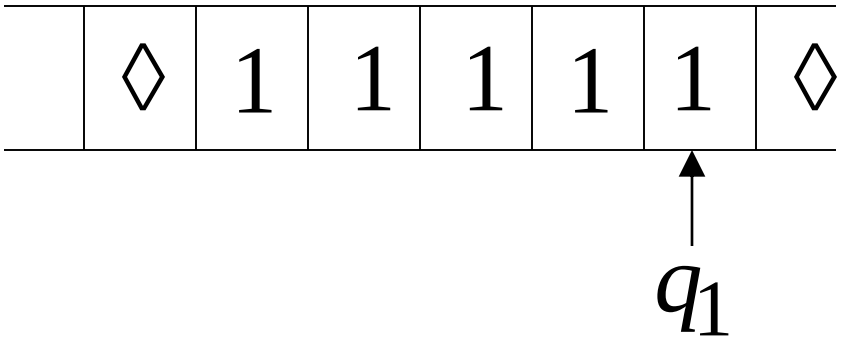
Time 3

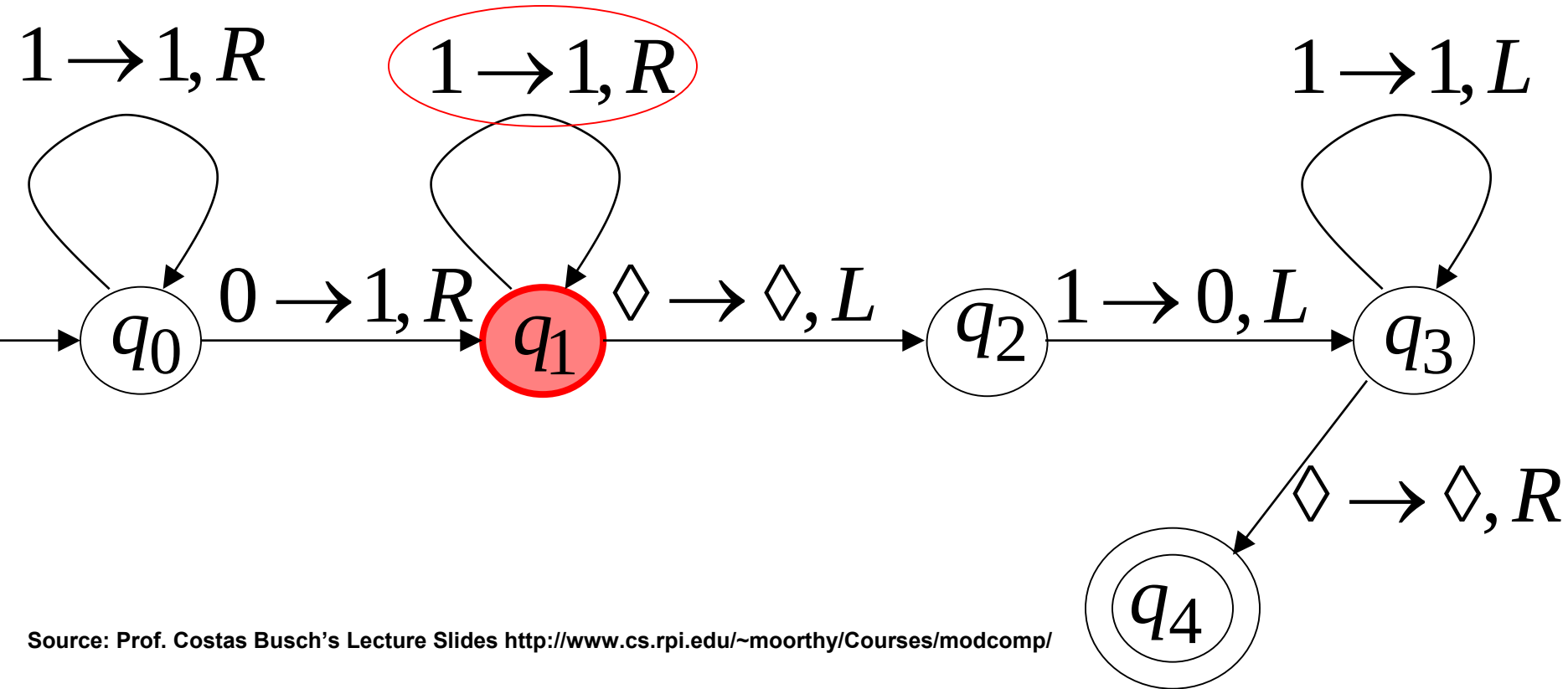




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

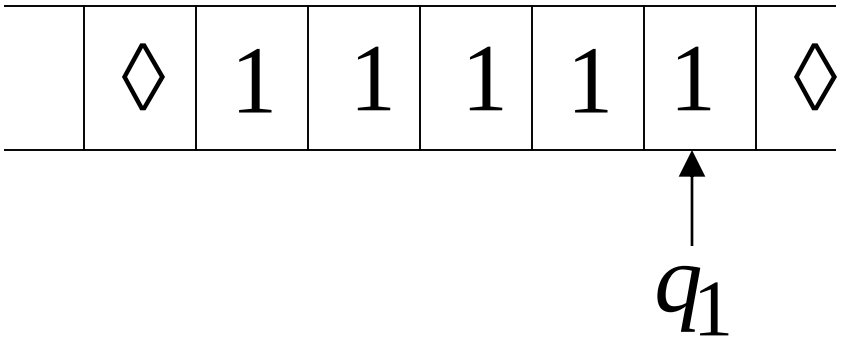
Time 4

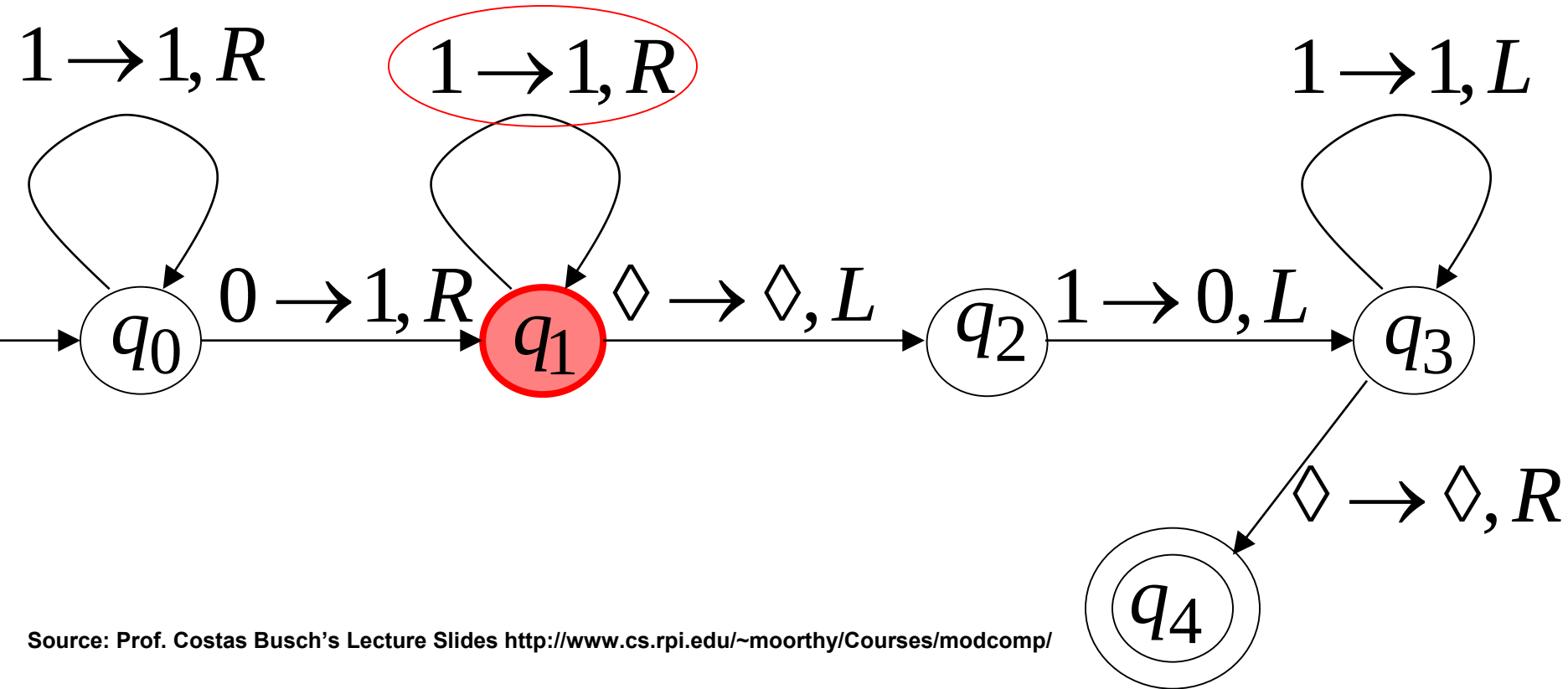




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

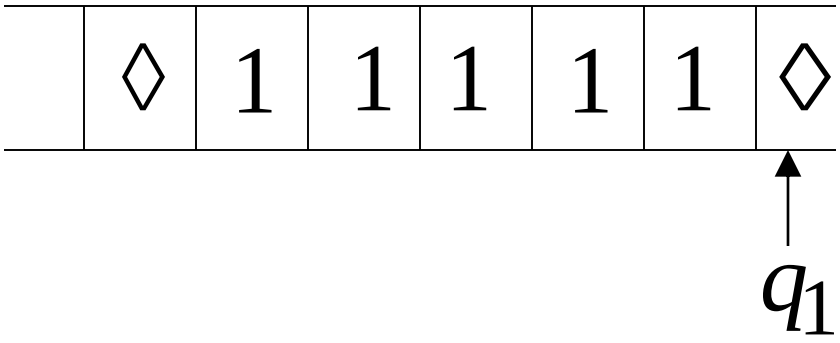
Time 4

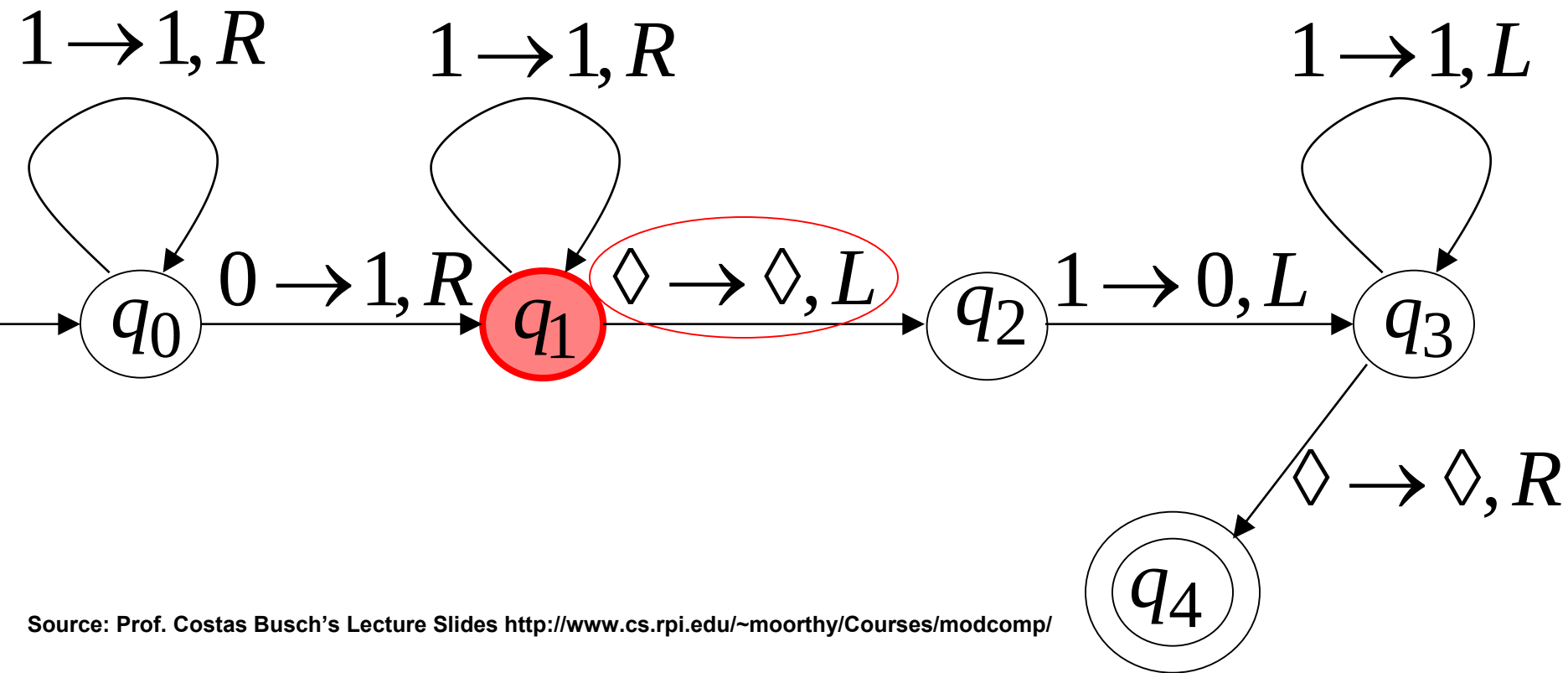




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

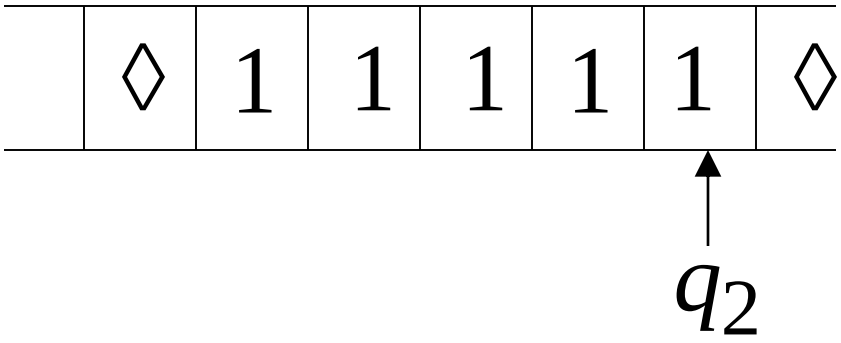
Time 5

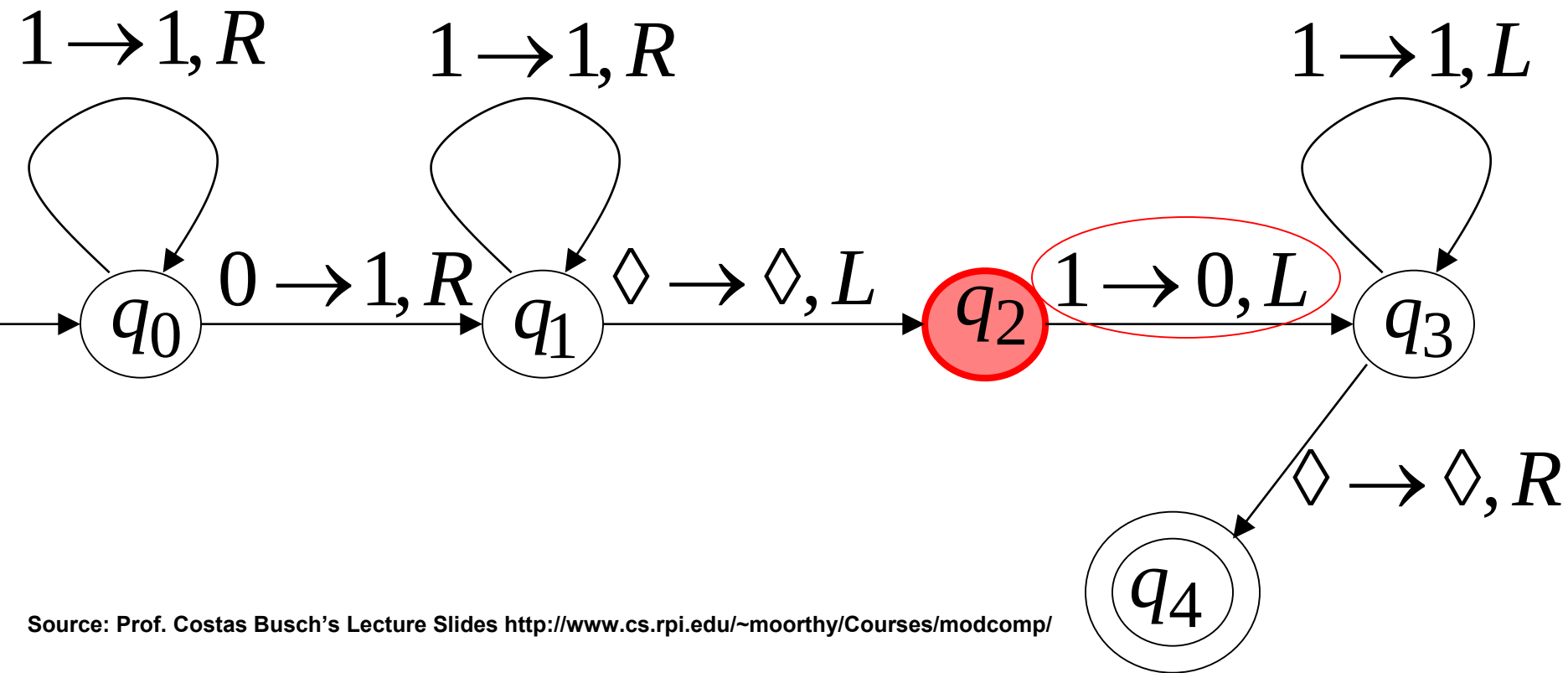




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

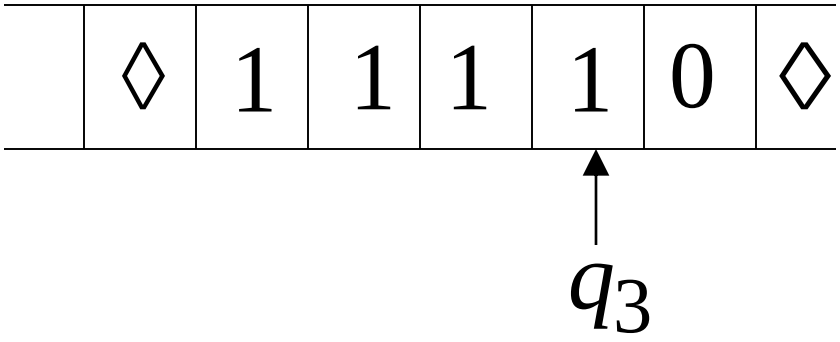
Time 6

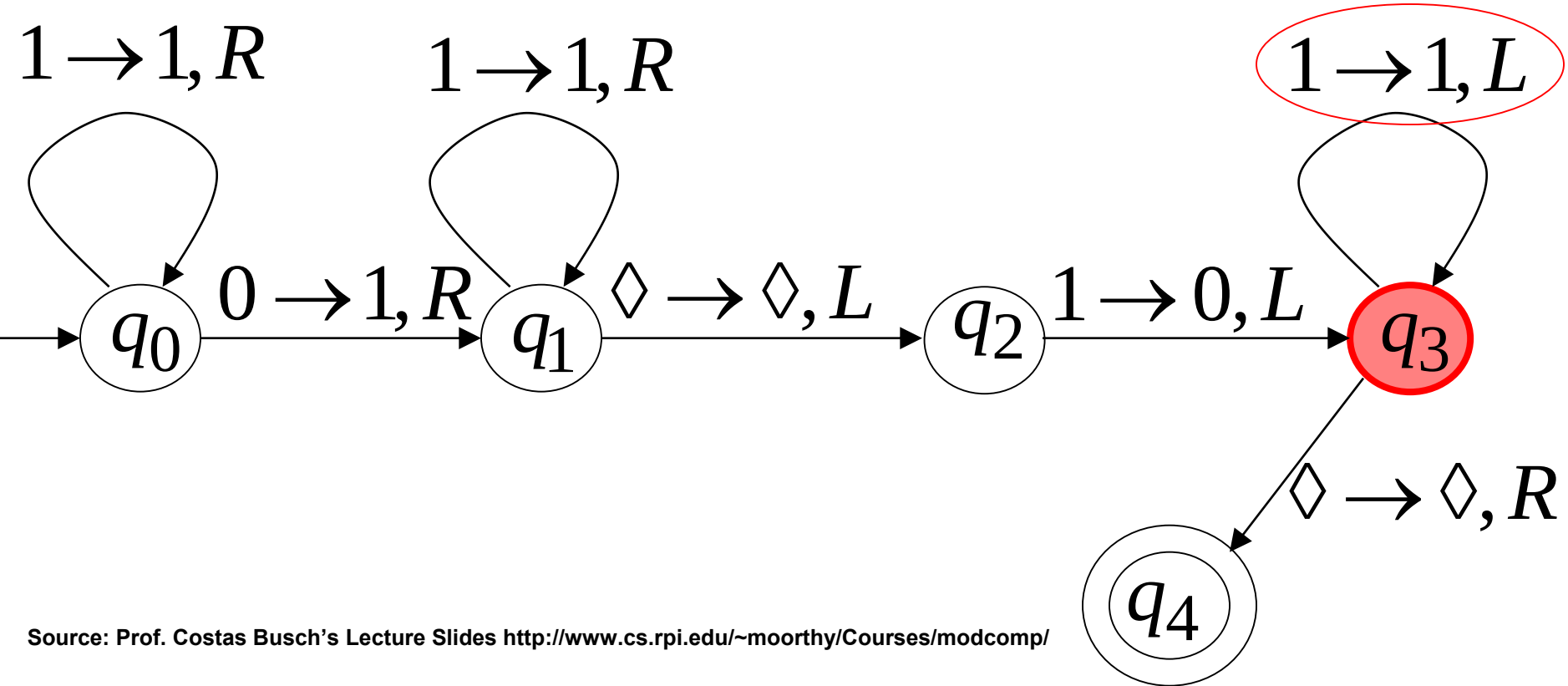




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

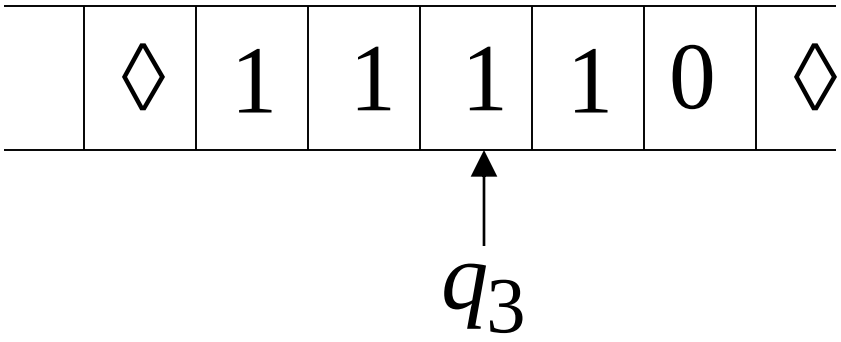
Time 7

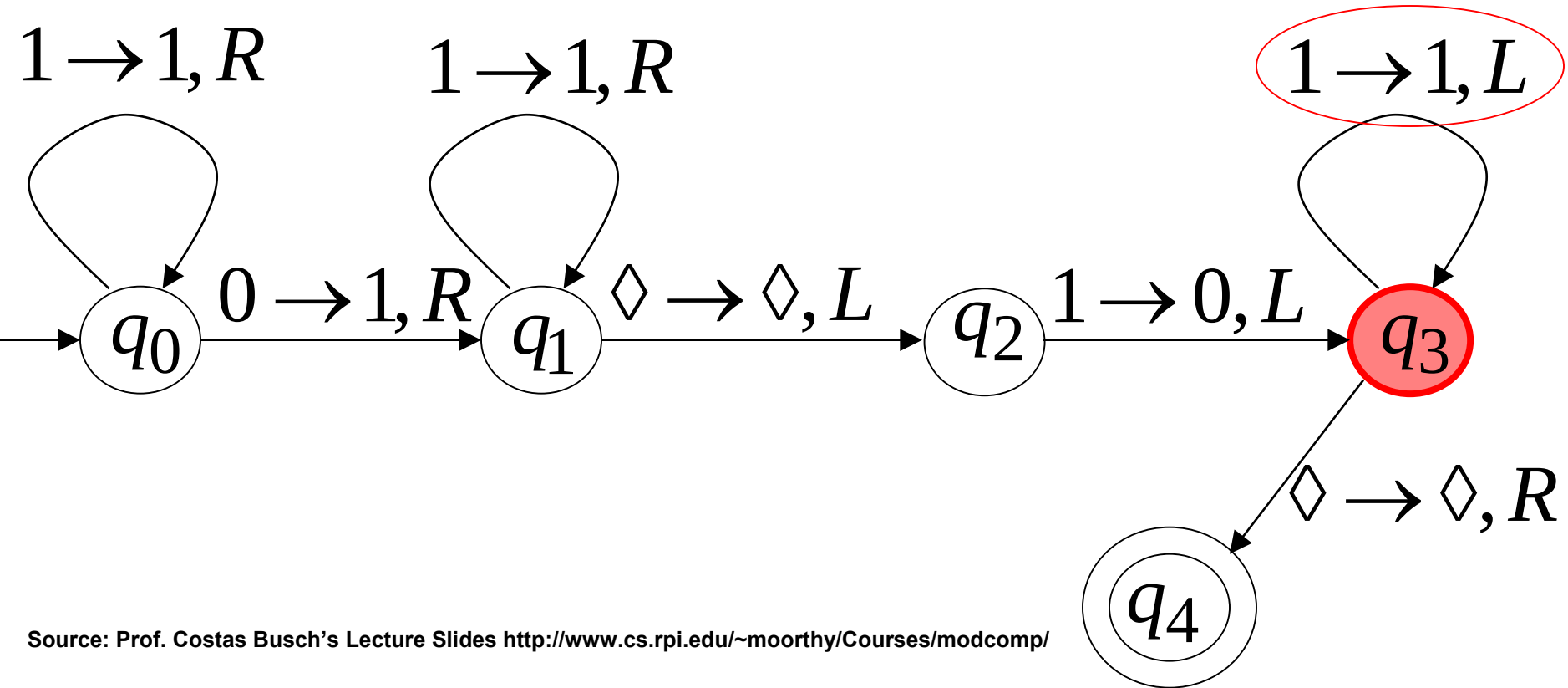




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

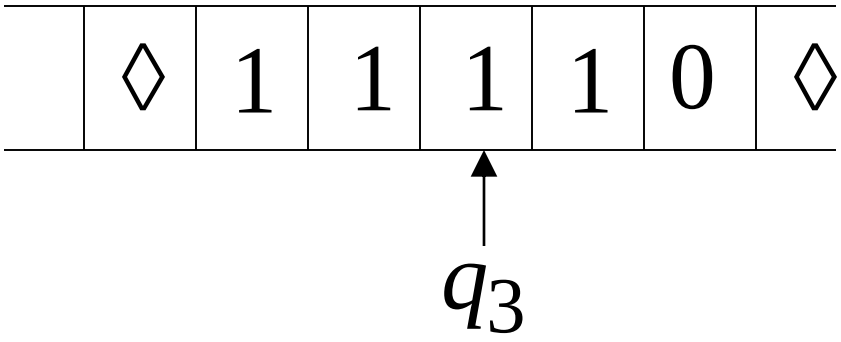
Time 8

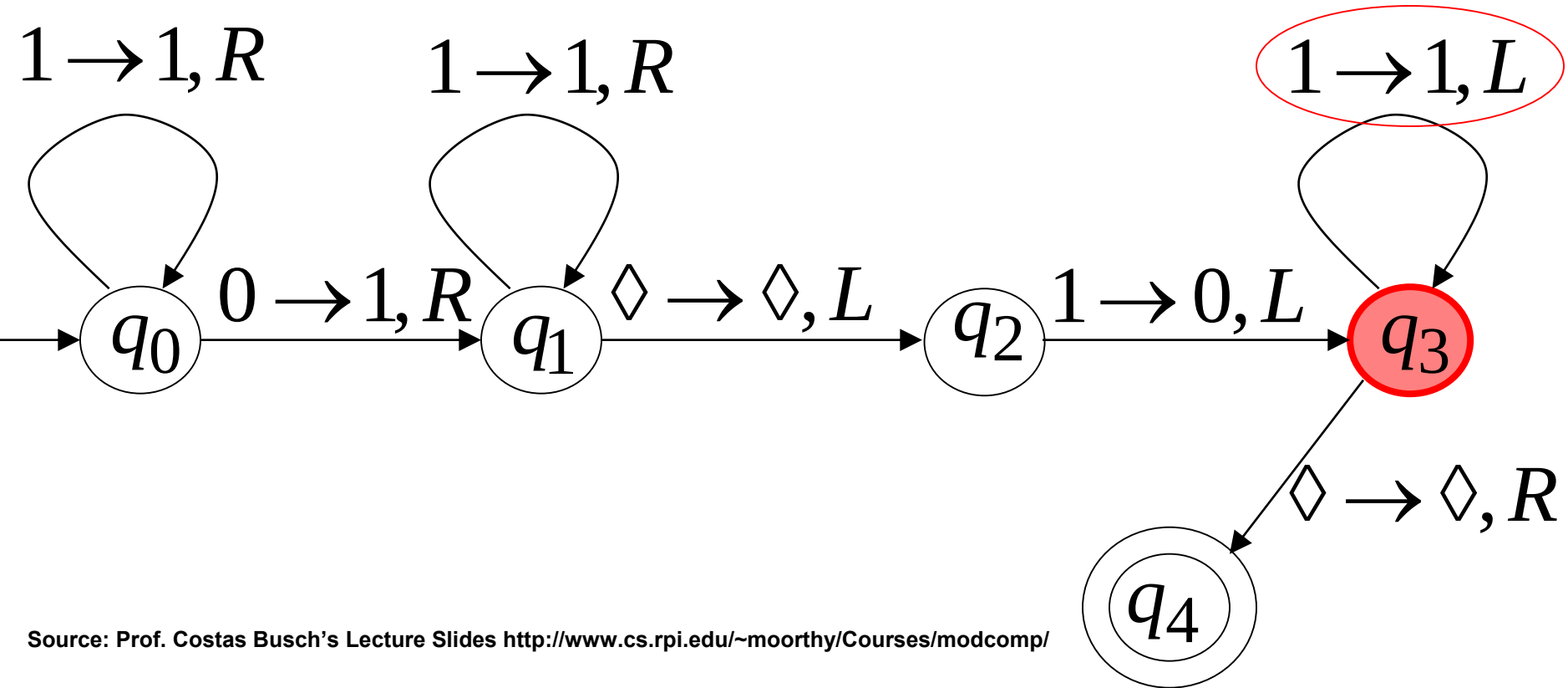




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

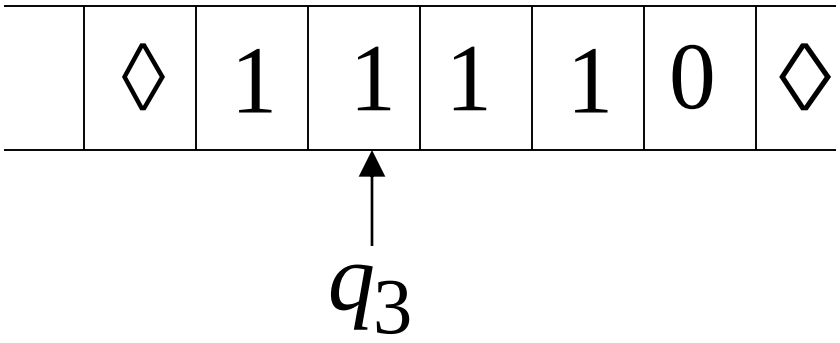
Time 8

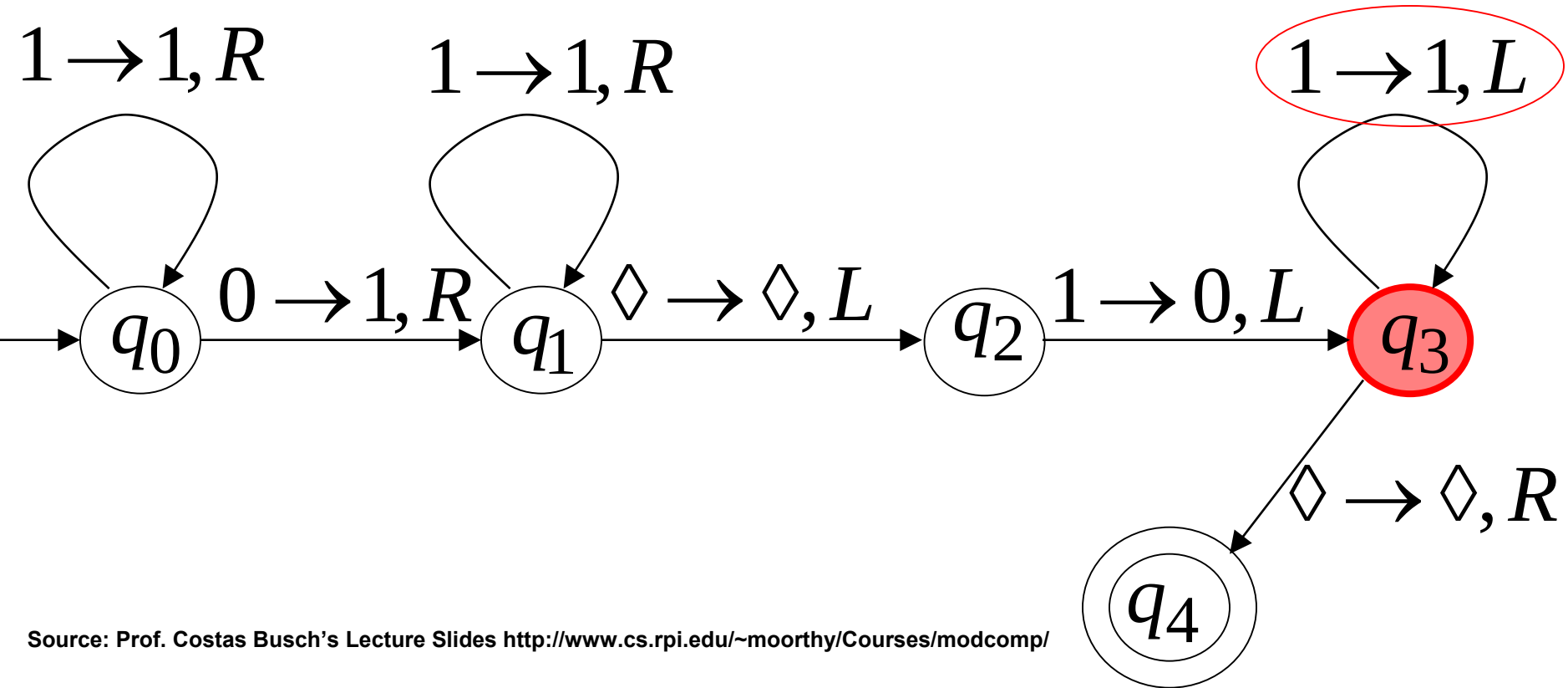




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

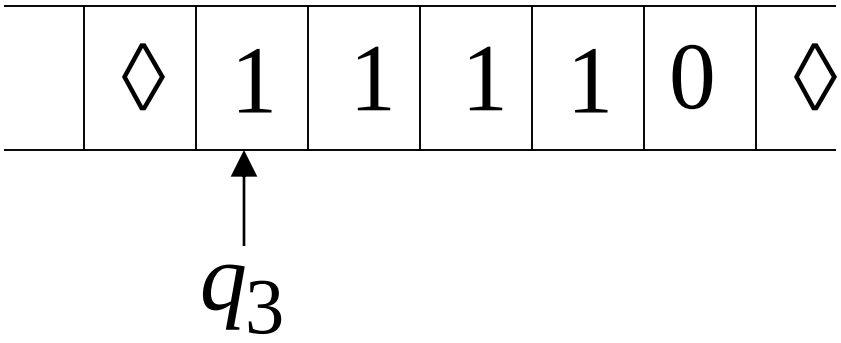
Time 9

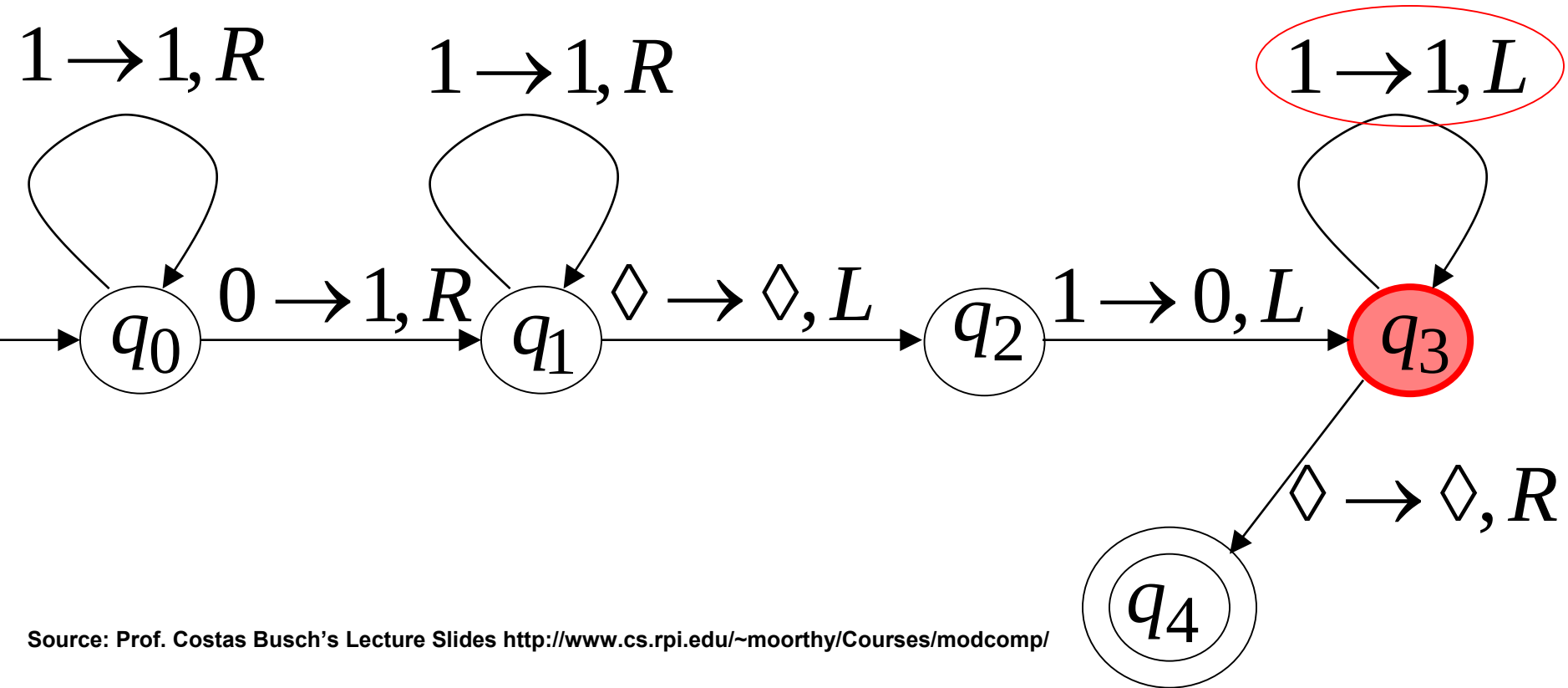




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

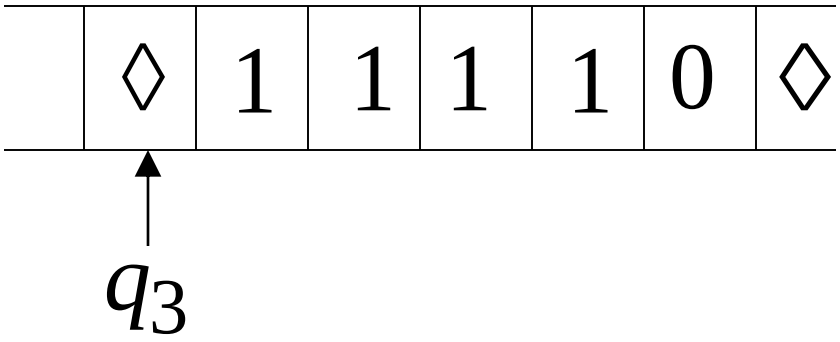
Time 10

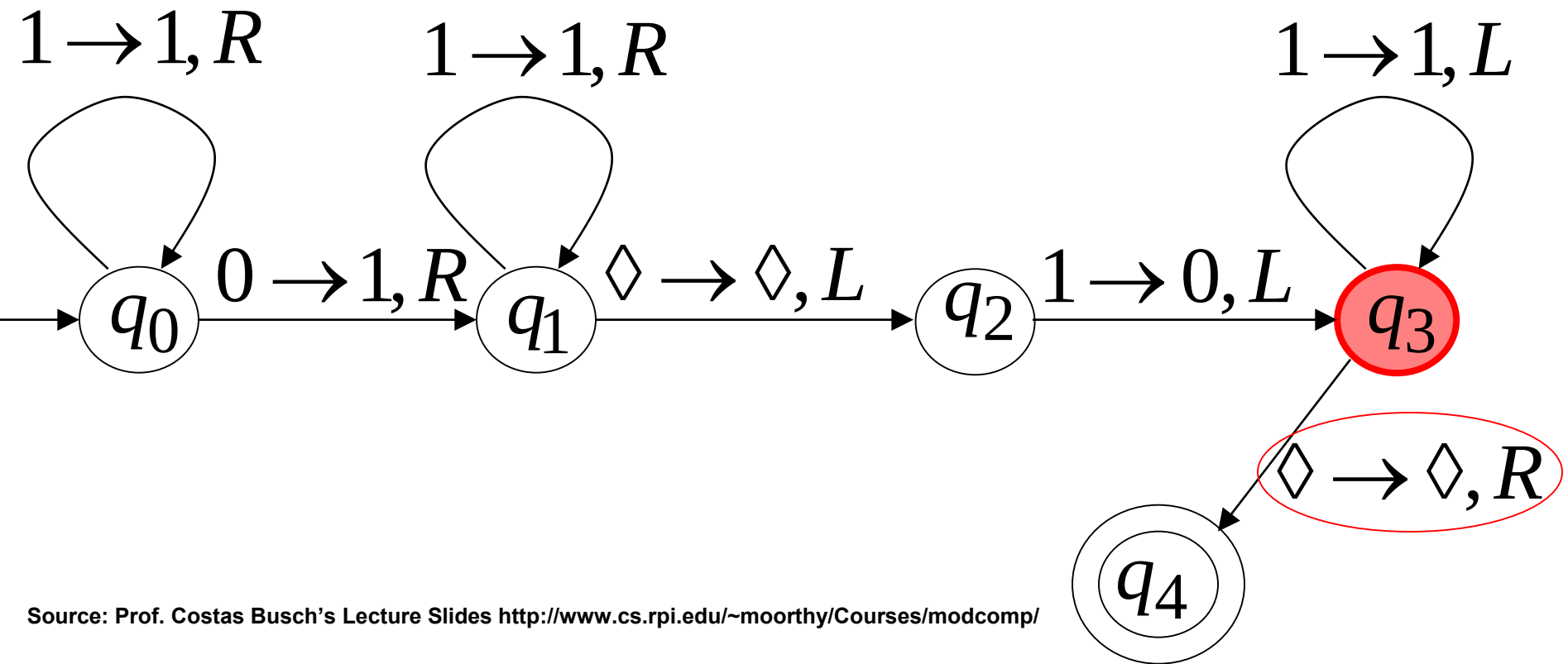




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

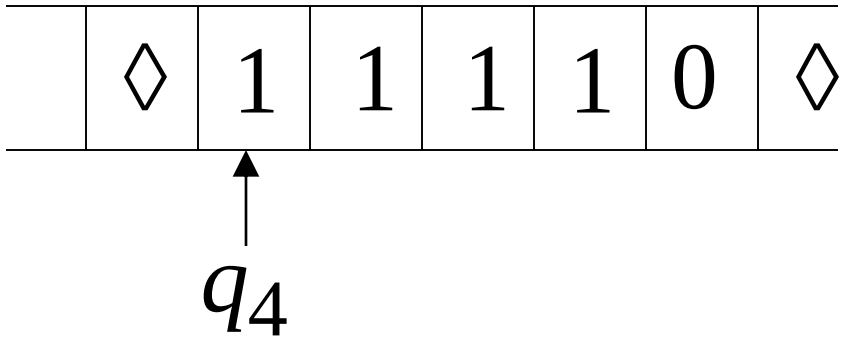
Time 11

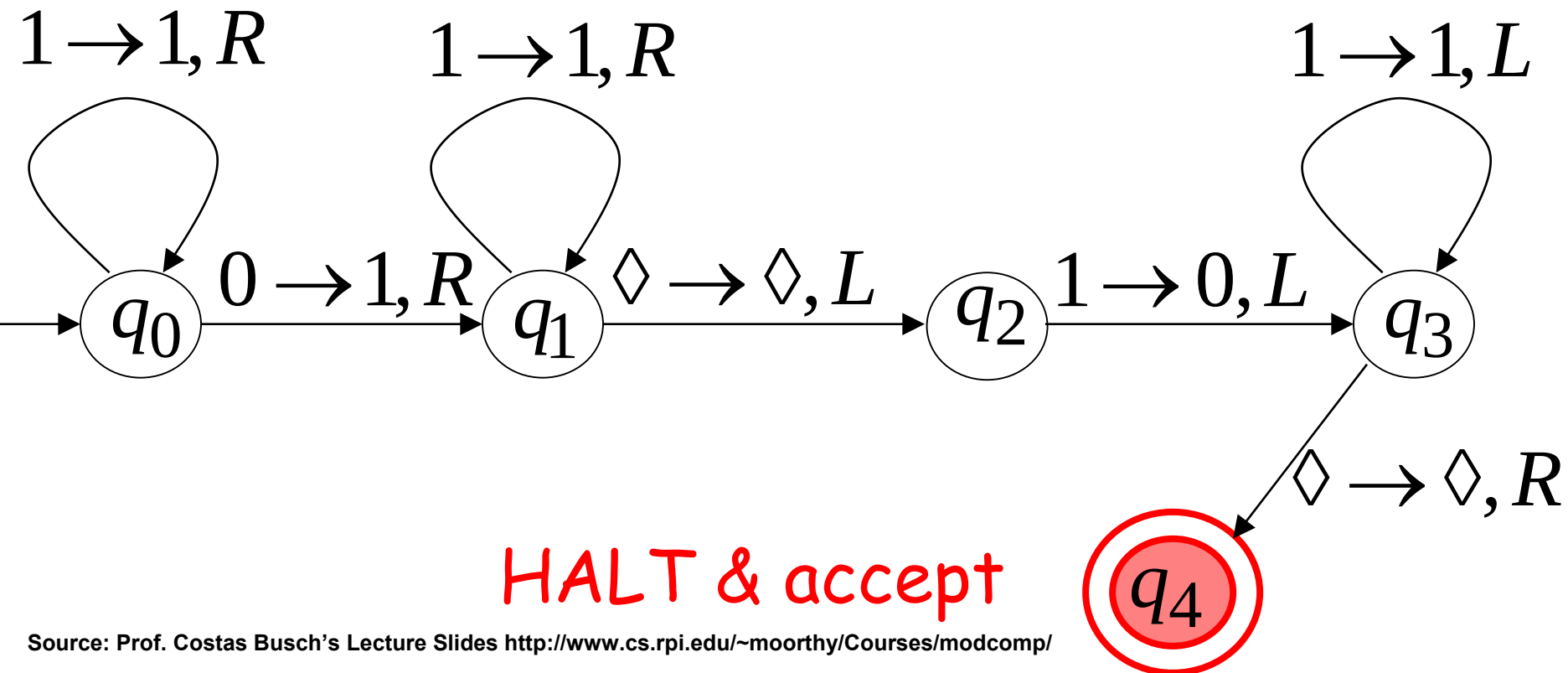




Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

Time 12





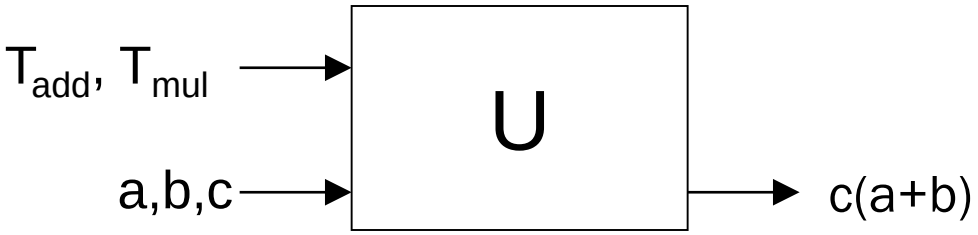
HALT & accept

Source: Prof. Costas Busch's Lecture Slides http://www.cs.rpi.edu/~moorthy/Courses/modcomp/

Universal Turing Machine

A machine that can implement all Turing machines -- this is also a Turing machine!

- **inputs: data, plus a description of computation (other TMs)**



Universal Turing Machine

U is programmable – so is a computer!

- **Program is part of the input data**
- **a computer can emulate a Universal Turing Machine and vice versa**

A computer is a universal computing device.

Halting Problem

- **Halting Problem**
  - **The problem of determining, from a description of an arbitrary computer program (i.e., Turing machine) and an input, whether he program will finish running (i.e., halts) or continue to run forever**
- **Halting problem is undecidable (not Turing machine solvable)**

  (Proof by an application of Cantor's diagonal argument)

  오토마타 교과목에서 더 깊게 다루어짐.

꼭 기억해야 할 것

- (Levels of) Abstraction
- Turing Equivalence
- Undecidable Problem (not Turing machine computable)