San José State University
Department of Computer Science

Ahmad Yazdankhah
ahmad.yazdankhah@sjsu.edu
www.cs.sjsu.edu/~yazdankhah

Turing Machines

(Part 1)

Lecture 15

Day 16/31

CS 154

Formal Languages and Computability

Fall 2019

Agenda of Day 16

- Collecting Quiz 5
- Summary of Lecture 14
- A Few Slides from the Past
  (the slides will be added to the Lecture 14)
- Lecture 15: Teaching ...
  - Turing Machines (Part 1)
- Team Formation

Summary of Lecture 14: We learned ...

NPDAs

- How can we create nondeterministic PDA?
  1. λ-transition
  2. Multifunction δ
- We can create a λ-transition by putting λ in the condition positions.
- For NPDAs, a λ-transition is labeled as:

  λ, λ; w

- But w is a string and can be λ.
- So, "λ, λ; λ" is a λ-transition and is used extensively.

- We took some examples for multifunction transitions.
- As usual, machines start parallel processing when they have multiple choices.
- The procedure of initiating a new process is exactly the same as NFAs.
- PDAs configuration ...
  1. Current state of transition graph
  2. Input string + Position of the cursor
  3. Stack and its content

Any question?

Summary of Lecture 14: We learned ...

NPDAs Formal Definition

- Formally, we defined NPDAs as:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$$

- We added Γ and Z to NFAs'.
- The other change is δ ...

$$\delta: Q \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \rightarrow 2^{Q \times \Gamma^*}$$

- Sub-rule example:

$$\delta\,(q_1 , a , x) = \{(q_2 , yx), (q_3 , \lambda)\}$$

Any question?

A Few Slides From the Past

Will be Added to the Lecture 14

Turing Machines

Template for Introducing a New Class of Automata

Repeated

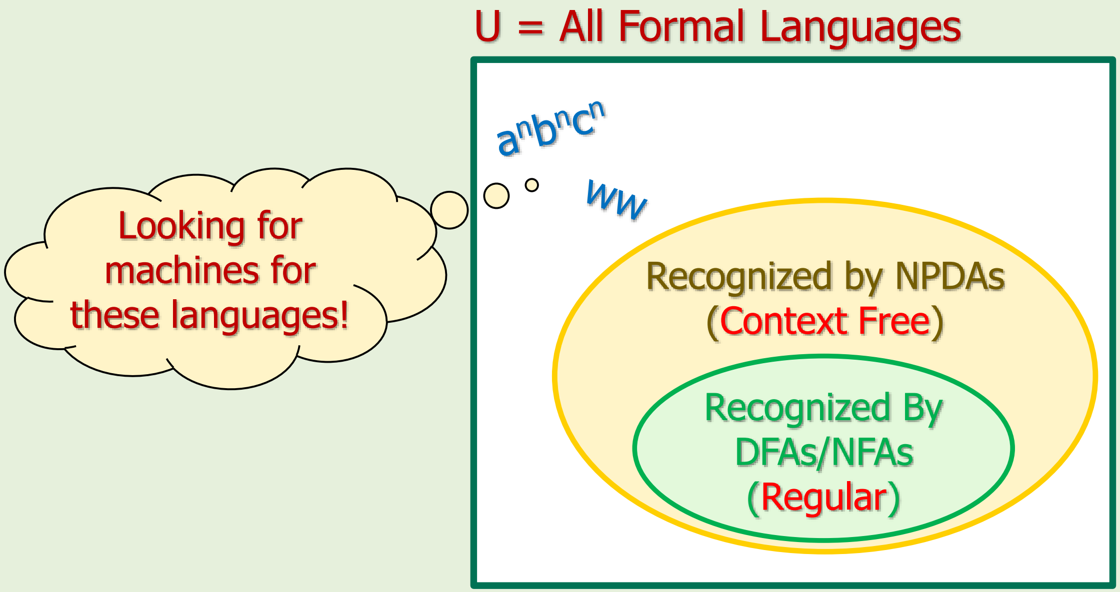
- To construct a new class of automata, we'll follow the following steps:

1. Why do we need this new class? (Justification)
2. Name of this new class
3. Building blocks of this new class
4. How do they work?
   - 4.1. What is the starting configuration?
   - 4.2. What would happen during a timeframe?
   - 4.3. When would the machine halt?
   - 4.4. How would a string be Accepted/Rejected?
5. The automata in action
6. Formal definition
7. Their power: this class versus previous class
8. What would be the next possible class?

1. Why do We Need a New Class?

- This was our last conclusion:

  There are some languages for which we cannot construct NPDAs!



1. Why do We Need a New Class?

What Was Missing in NPDAs?

- We had stack (a writable memory) for counting but ...
  1. ... stack is not so flexible in storing and retrieving data.
  2. ... we lose some data when we access the older data.

- We need more control on the memory.

- So, we are going to replace the stack with a more flexible memory.
- What is more flexible than stack?
- RAM (random access memory)!

2. Name of this New Class

- This machine was proposed by Alan M. Turing in 1936.
- That's why we call it:

  Turing machine (TM)

- Both deterministic and nondeterministic TMs can be defined.
- The deterministic TM is called:

  Standard TM

  – For convenience, we usually drop the standard and just call it Turing machine (TM).

- The nondeterministic one is called:

  Nondeterministic TM (NTM)

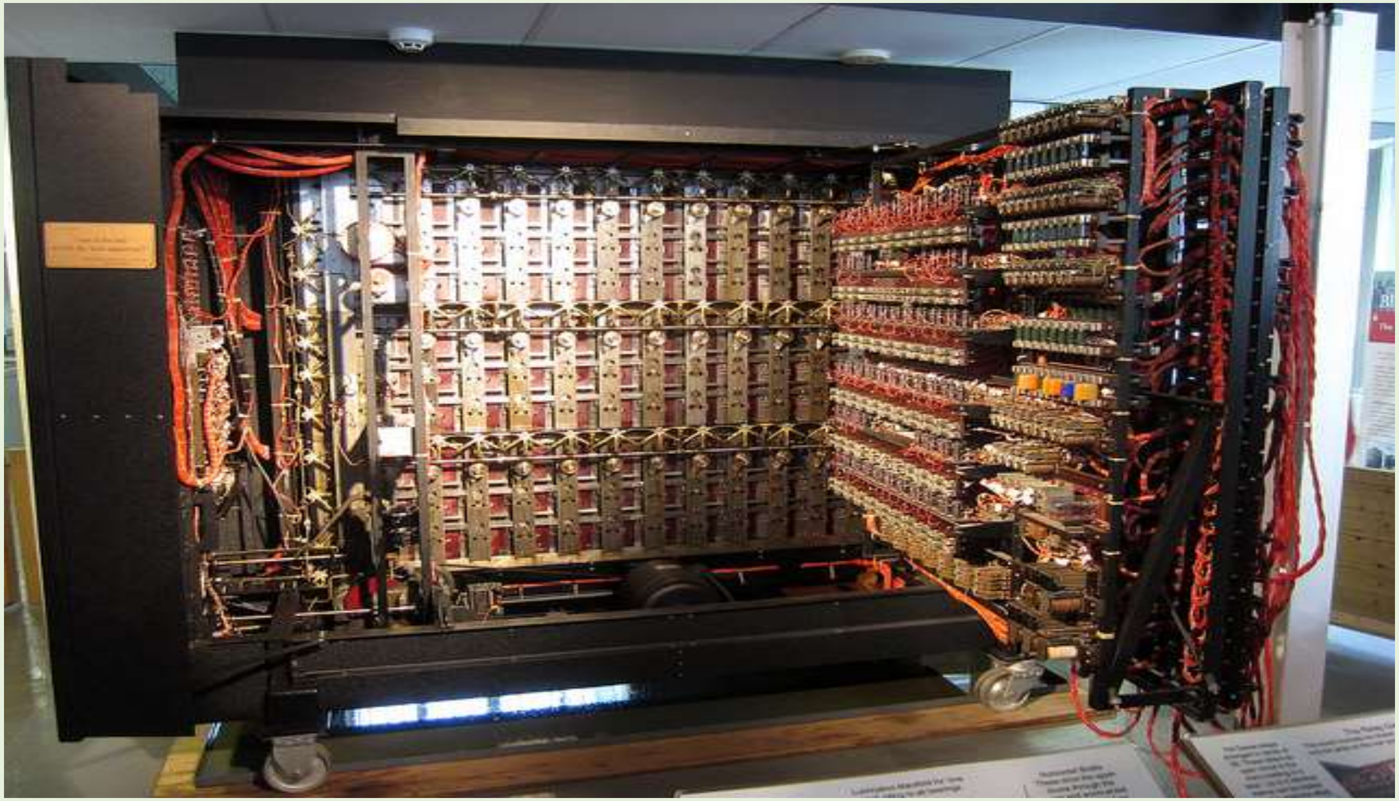
- Before introducing TMs, let's see who is Alan Turing?

Who is Turing?

- Alan Mathison Turing (1912 – 1954) born in Britain.
- He was:
  - mathematician
  - logician
  - cryptanalyst
  - theoretical biologist
- He is known as the father of the:
  - theory of computation (Computer Science Foundation)
  - artificial intelligence (AI)
- He is one of the most effective pioneering computer scientists.

Who is Turing?

- During World War II, he invented an electromechanical machine called "Bombe Machine" for cryptoanalysis of Hitler's cipher system.



- This is a reconstructed version of the original machine.

Who is Turing?

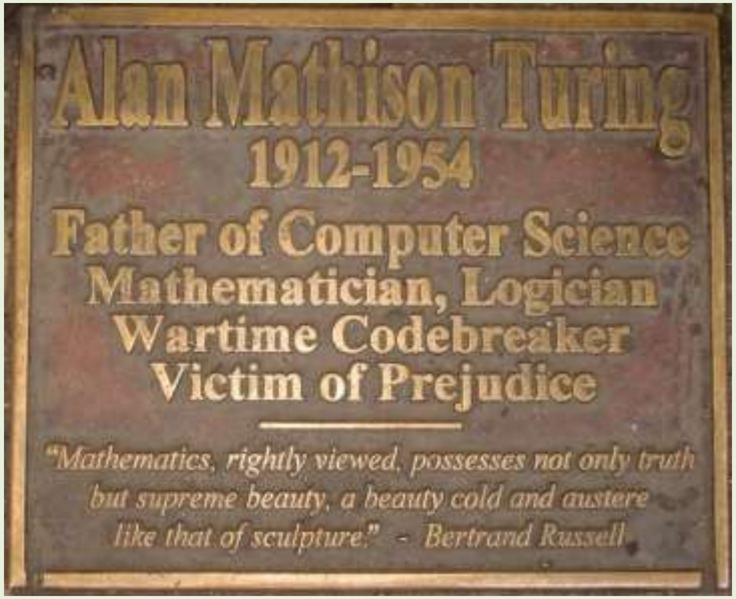
- His Bombe Machine could break Hitler's Enigma cipher system.

- It is said that his work shortened the war between 2 to 4 years.
- Based on some estimations, he saved 14 million lives.

- Hitler was so close to construct nuclear bomb.
- Just imagine what could happen if
  the war lasted two more years
  and Hitler could finish his nuclear bomb!

Who is Turing?

- He was prosecuted in 1952 for homosexual acts and died in 1954 when he was only 42!



- In 2013, 61 years after his death, he was granted a Royal pardon (!) by the British Queen.

Who is Turing?

- In 1966, Association for Computing Machinery (ACM) created an annual prize called "A. M. Turing Award".
- It is the highest award in computer science, given to an individual whose contribution in computer science is outstanding.
- It is called the "Nobel prize of computing".
- Since 2014, Google has been supporting the prize that is $1 million.

Who is Turing?

Documentary

- Produced by BBC:
  https://www.youtube.com/watch?v=GH1WYUKP3hk
- "Turing: Pioneer of the Information Age" by Stanford:
  https://www.youtube.com/watch?v=p7Lv9GxigYU
- "Turing the Man" by ACM:
  https://www.youtube.com/watch?v=KUaKrtF0-hQ

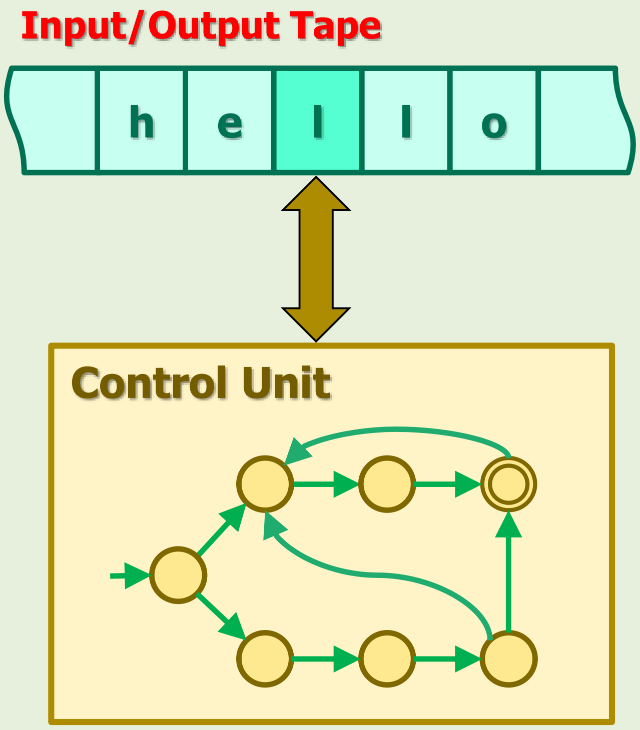
Movies

- Breaking the Code, Biography of Alan Turing (BBC 1996)
  https://www.youtube.com/watch?v=S23yie-779k
- The Imitation Game (2014) currently on Netflix
- Codebreaker: The Story of Alan Turing

3. TMs Building Blocks

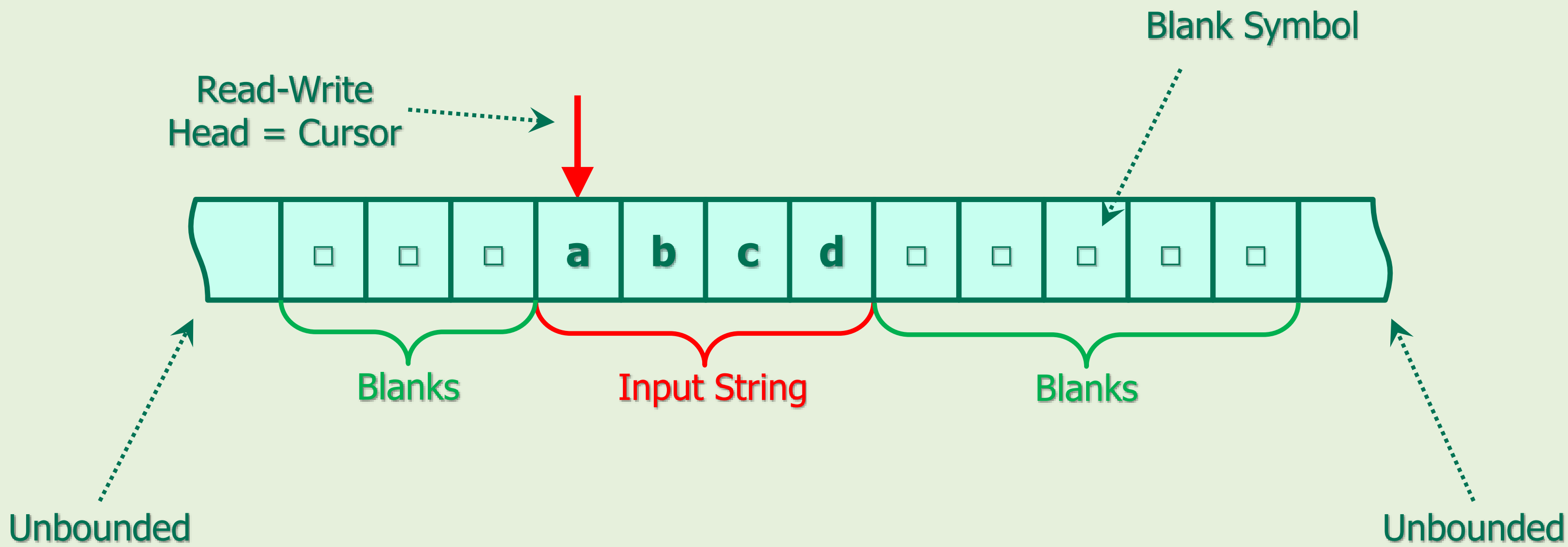
3. TMs Building Blocks

- TMs have 2 main blocks:
  1. Input / Output Tape
  2. Control Unit



- Let's see each block in detail.

3.1. Input / Output Tape: Structure

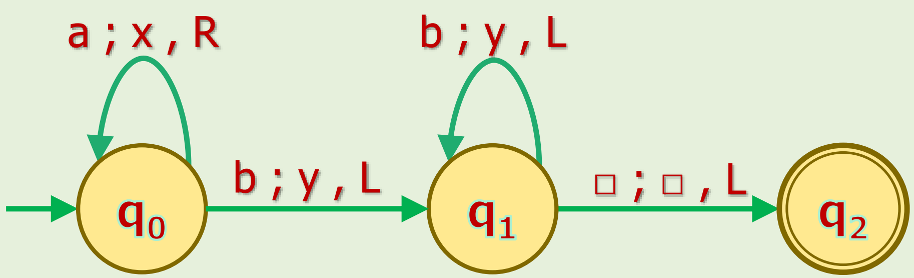


- The tape is unbounded from both sides.
- A read-write head (= cursor) reads a symbol, writes a symbol, and moves left or right.
- The input string can be written somewhere on the tape.
- The rest of the tape contains blank symbols, denoted by '□'.

3.2. Control Unit: Structure

- The control unit of TMs look pretty much like PDAs'.
  - They are represented by "transition graphs".

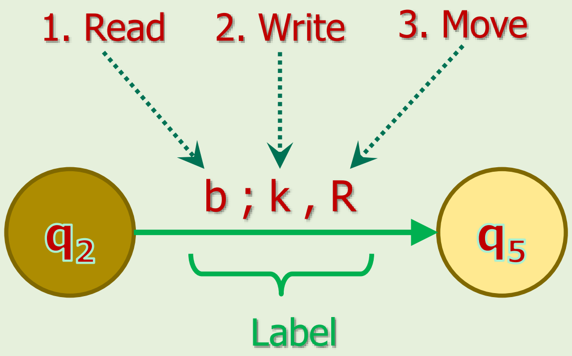
- This is an example of a TM's transition graph.



- The only difference is how the edges are labeled.

- Let's analyze a transition in detail.

3.2. Control Unit: Labels



- The label has 3 parts, delimited by semicolon and comma:
  - Note the difference with PDAs!

1. The input symbol (e.g. 'b') that should be read from the cell wherever the cursor is pointing at
2. The symbol (e.g. 'k') that should be written into the cell wherever the cursor is pointing at
3. The move direction of the cursor that can be "Left" or "Right". ('L' = Left or 'R' = Right).
   - 'L' and 'R' are called "move symbols".

4. How TMs Work

4. How TMs Work

Repeated

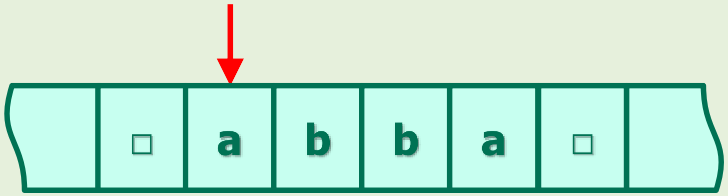
- To understand how TMS work, we should clearly respond to the following questions:

1. What is the "starting configuration"?
2. What would happen during a timeframe?
3. When would the machine halt (stop)?
4. How would a string be Accepted/Rejected?

4.1. TMs Starting Configuration

Clock

- The clock is set at timeframe 0.

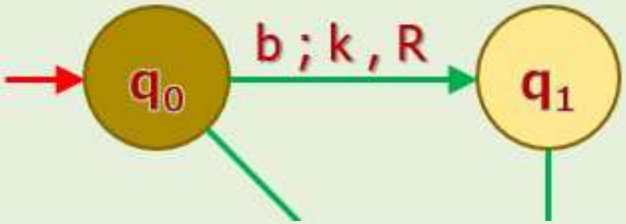


Input / Output Tape

- The input string has already been written somewhere on the tape.
- The rest of the tape will be initialized with blank symbols '□'.
- The cursor is pointing to the left-most symbol.

Control Unit

- The control unit is set to initial state.



4.2. What Happens During a Timeframe

- During a timeframe, the machine "transits" (aka "moves") from one configuration to another.

- Several tasks happen during a timeframe.
- The combination of these tasks is called "transition".

- Let's first visualize these tasks through some examples.
- Then, we'll summarize them in one slide.

4.2. What Happens During a Timeframe

Transition Examples

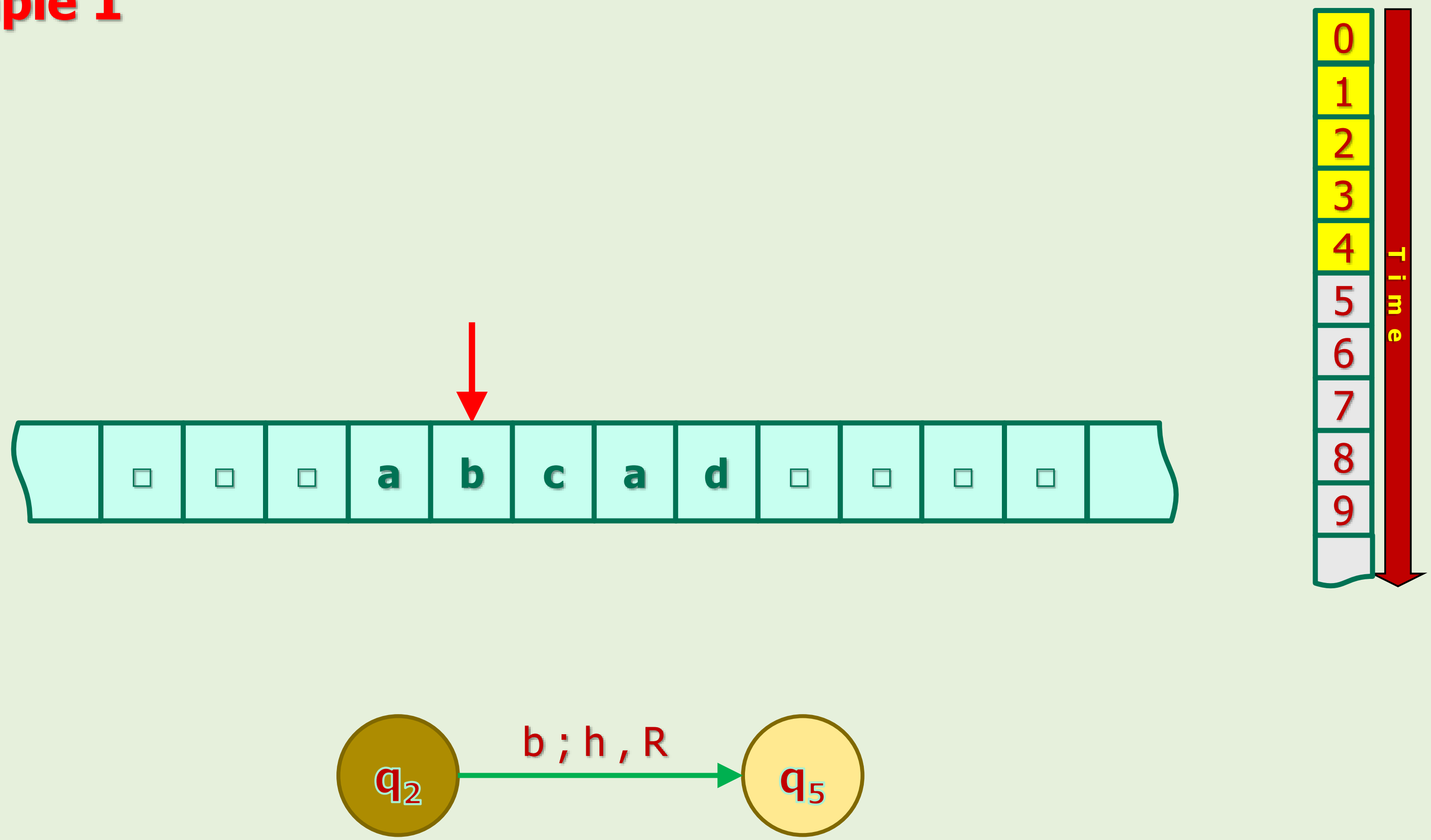
Transition Examples

- The next examples will show:
  - a partial transition graph
  - an input / output tape
  - a clock

- We assume that the machine is in the middle of its operation at timeframe n.
- The question is: in what configuration would the machine be at timeframe n+1?

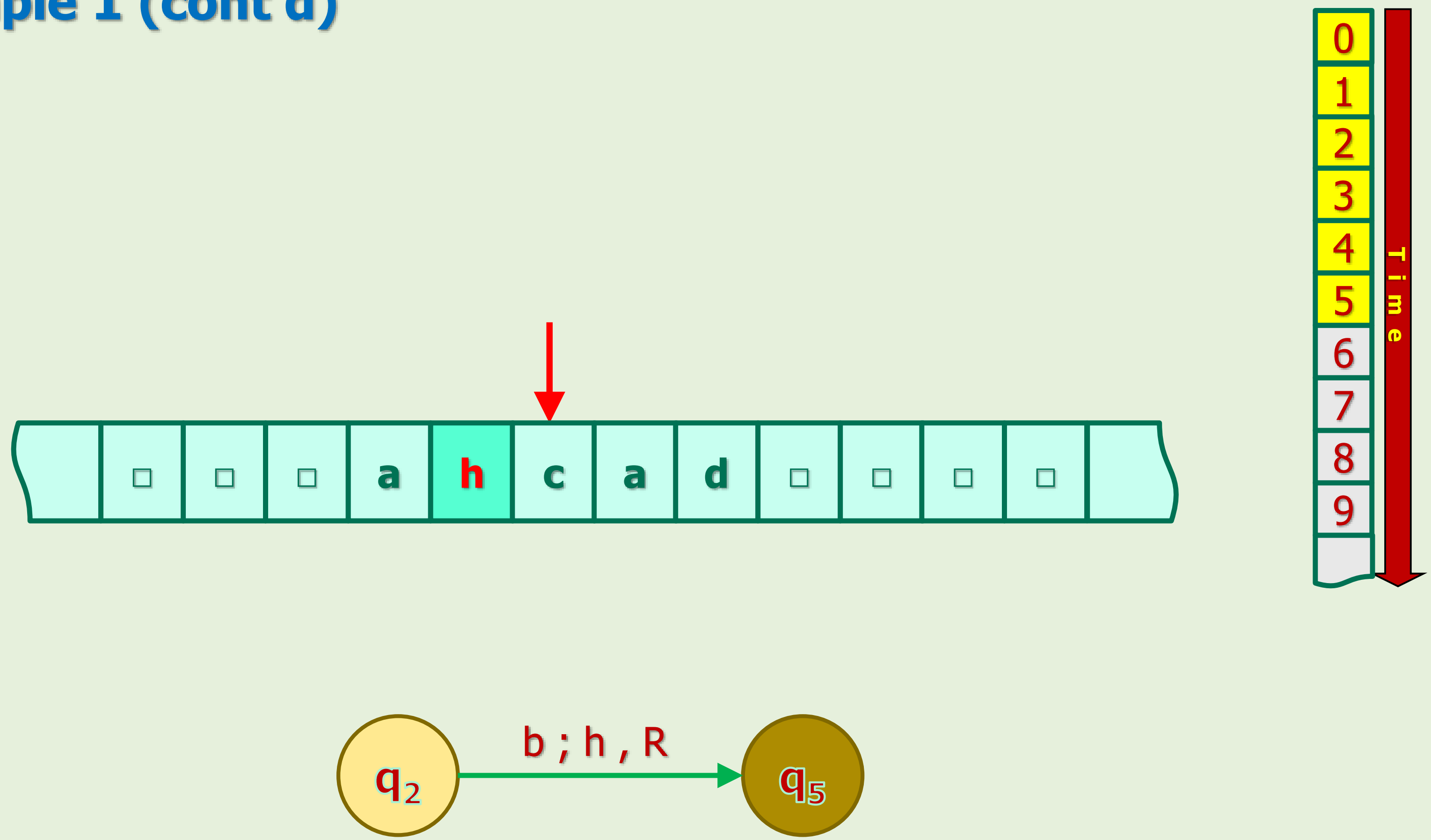
Transition **Example**

Example 1



Transition Example

Example 1 (cont'd)



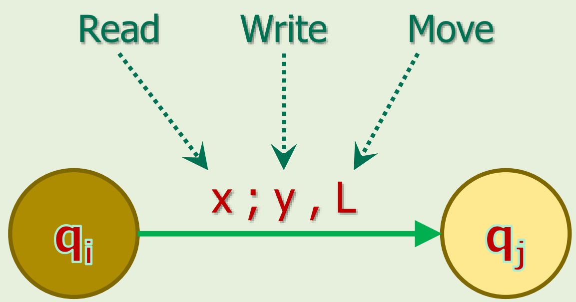
4.2. What Happens During a Timeframe

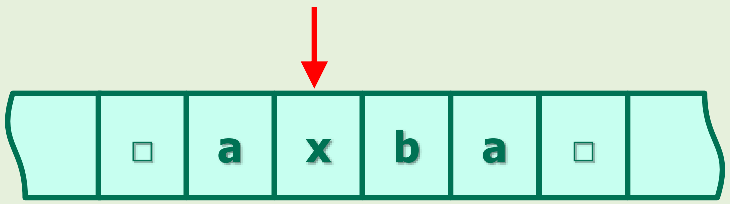
Transition

- The following tasks happen during a timeframe:
  1. A symbol at which the cursor is pointing, is read.
  2. A symbol is written into the same cell.
  3. The cursor is moved one cell to the left or right.
  4. The control unit makes its move based on the "logic of the transition".

- What is the "logic of the transition" of TMs?

TMs' Logic of Transitions





If (Condition)

in q_i

AND

the input symbol is 'x'

Then (Operation)

replace 'x' with 'y'

AND

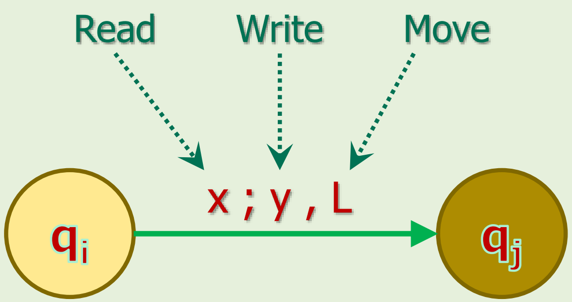
move the cursor to the Left

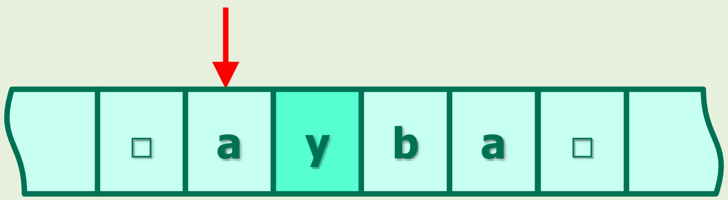
AND

transit to q_j

How does the machine look like after this transition?

TMs' Logic of Transitions





After the previous transition ...

- You might ask: what if the input is not 'x'?
- Good questions! We'll get back to this question later.

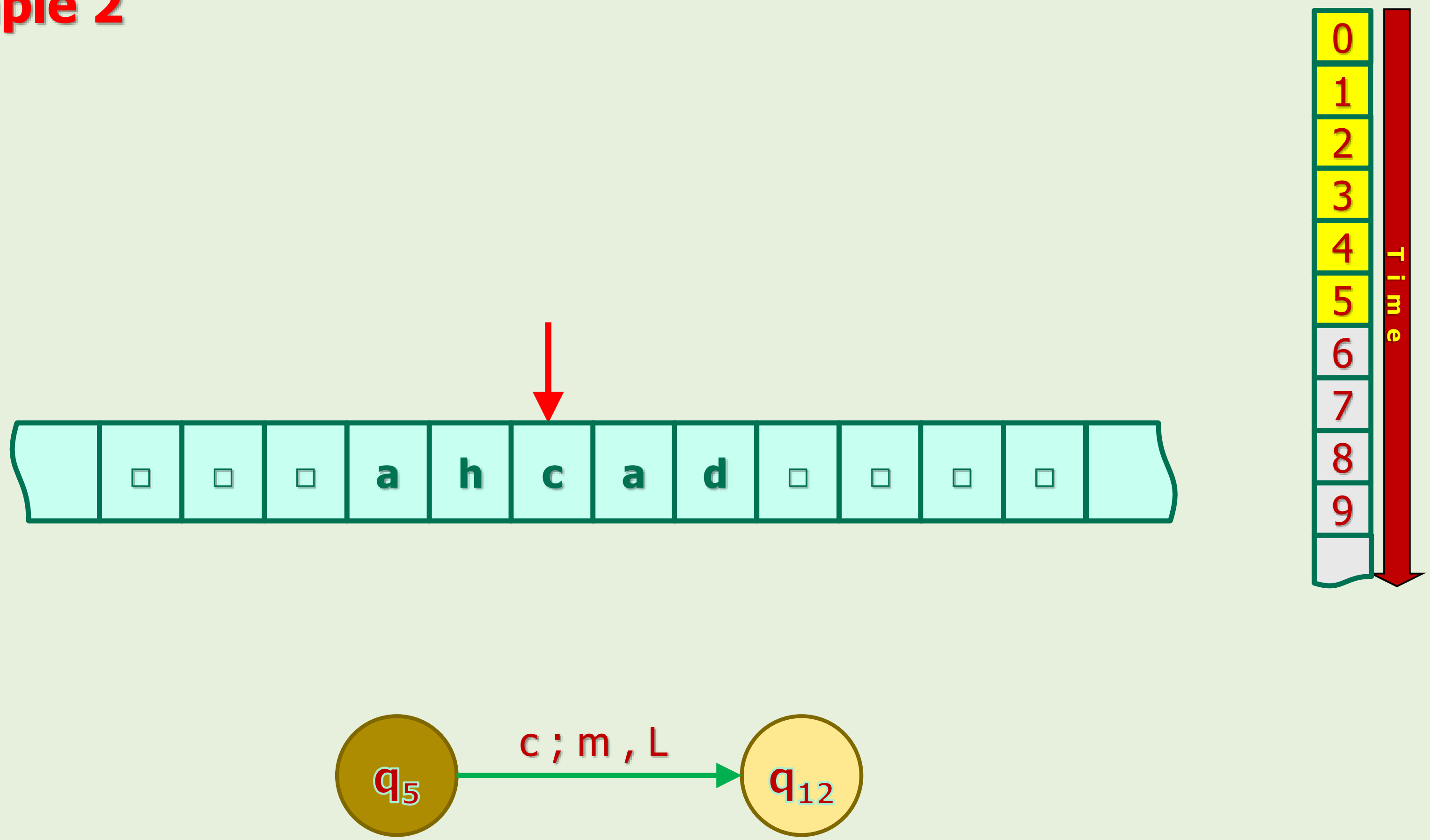
Multiple Labels

- A transition might have multiple labels.
- In that case, we stack them over the edge.

- Note that there is an OR between them.
- It means, in either condition, the machine transits and follows the label's operations.

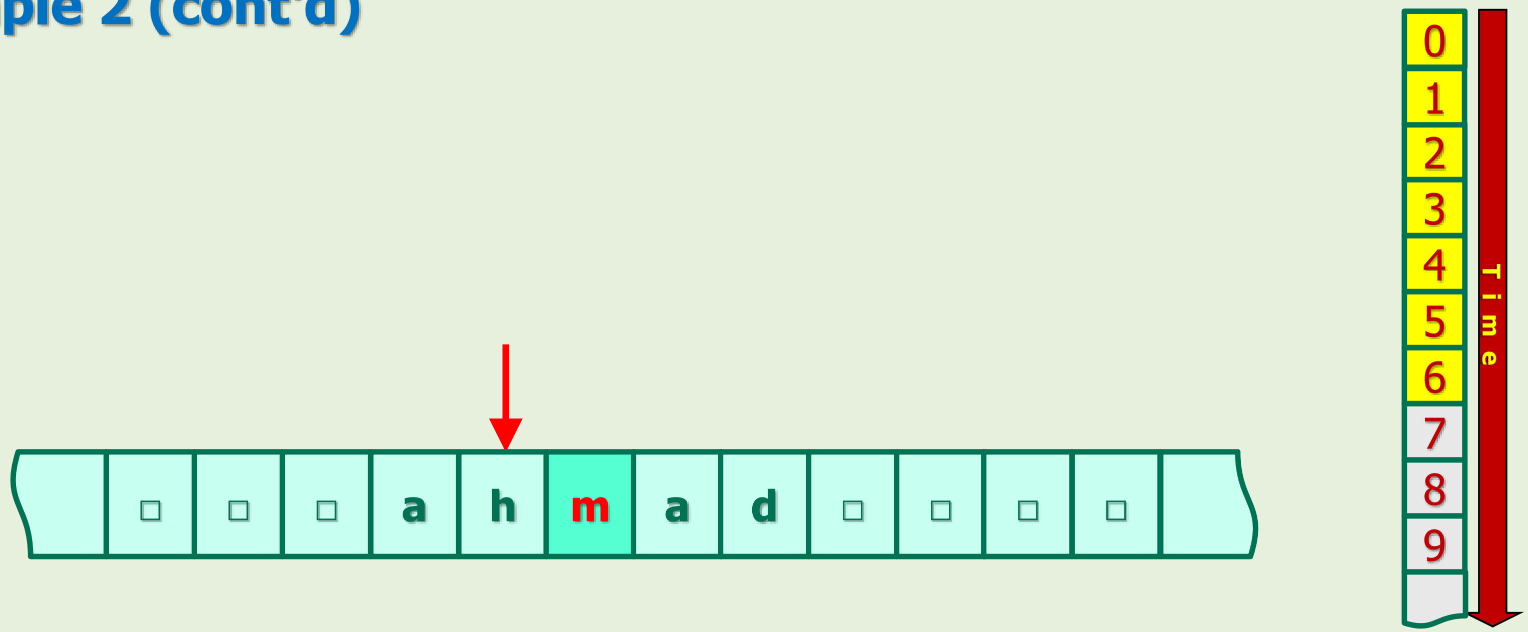
Transition Examples

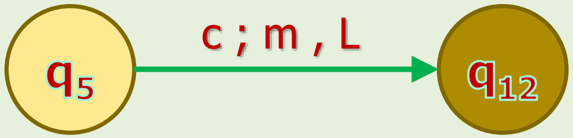
Example 2



Transition Examples

Example 2 (cont'd)

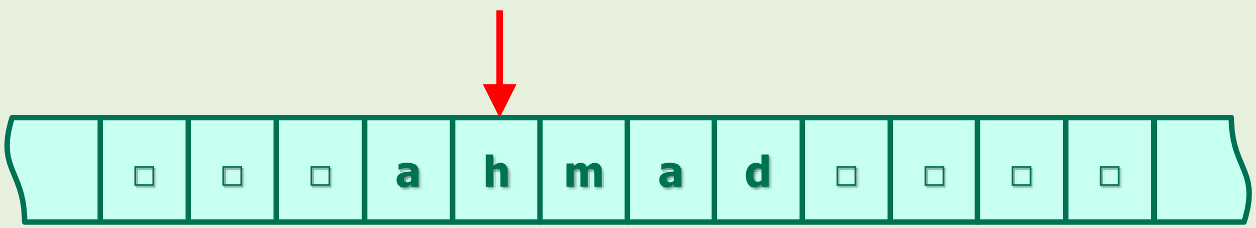




Transition Examples

Example 3

- No further transition ...
- Because the transition condition (input = 'b') is not satisfied.
- So, it "halts" in state q_{12}.



4.3. When TMs Halt

- From the previous example, we found out that:

  TMs halt when the next transition condition is NOT satisfied.

- Transition Conditions = input symbol

Halt Logical Representation

$$\left.\begin{array}{l}\text{TMs halt.} \equiv \mathbf{h} \\ \text{IFF} \\ \text{They have zero transition.} \equiv \mathbf{z}\end{array}\right\} \mathbf{z} \leftrightarrow \mathbf{h}$$

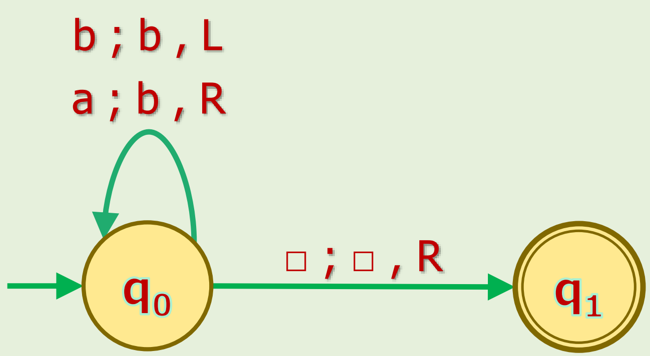
5. TMs in Action

Analysis Examples

5. TMs in Action

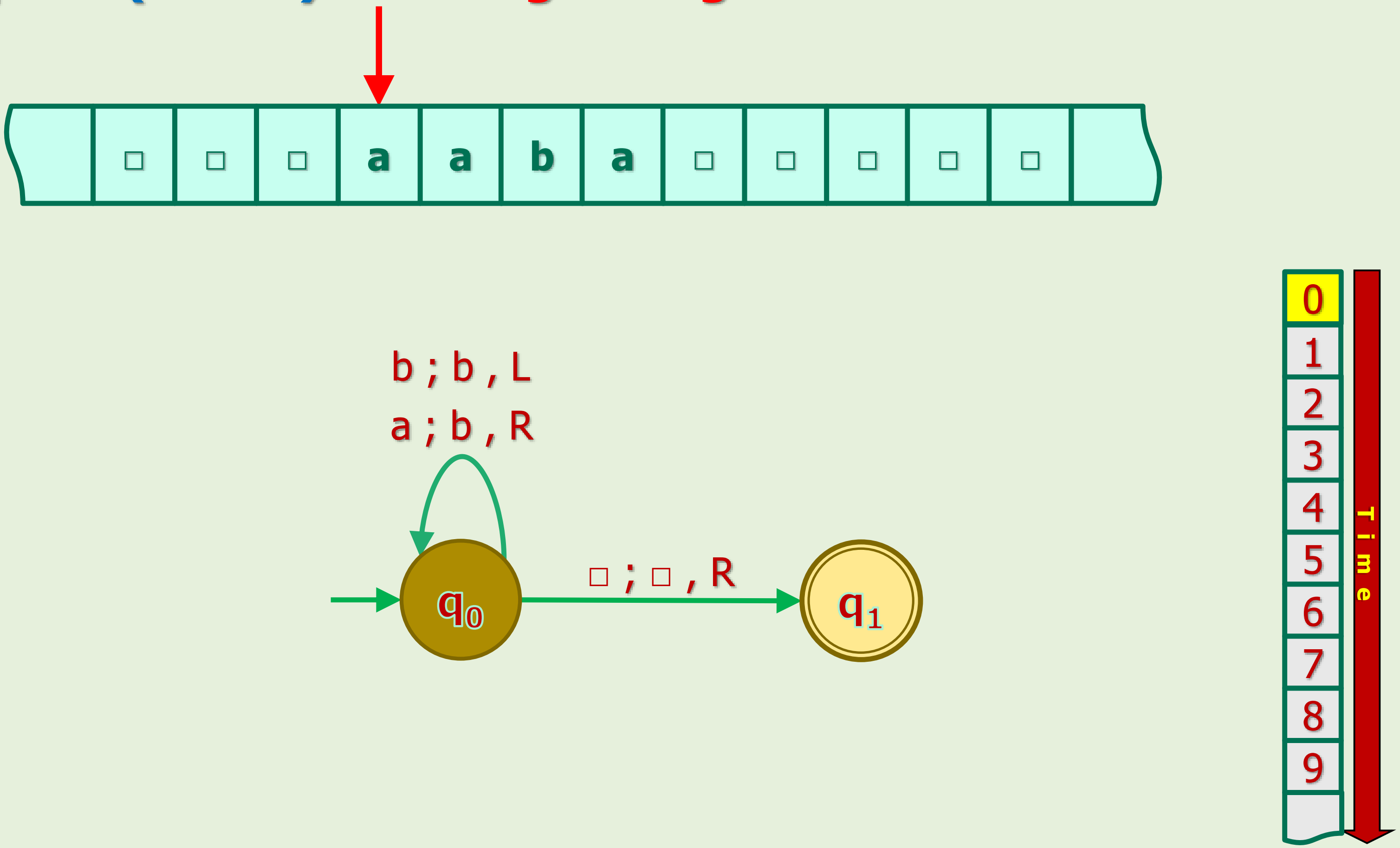
Example 4

- Consider the following TM over $\Sigma = \{a, b\}$.
- Trace the machine's operations for the input "aaba".



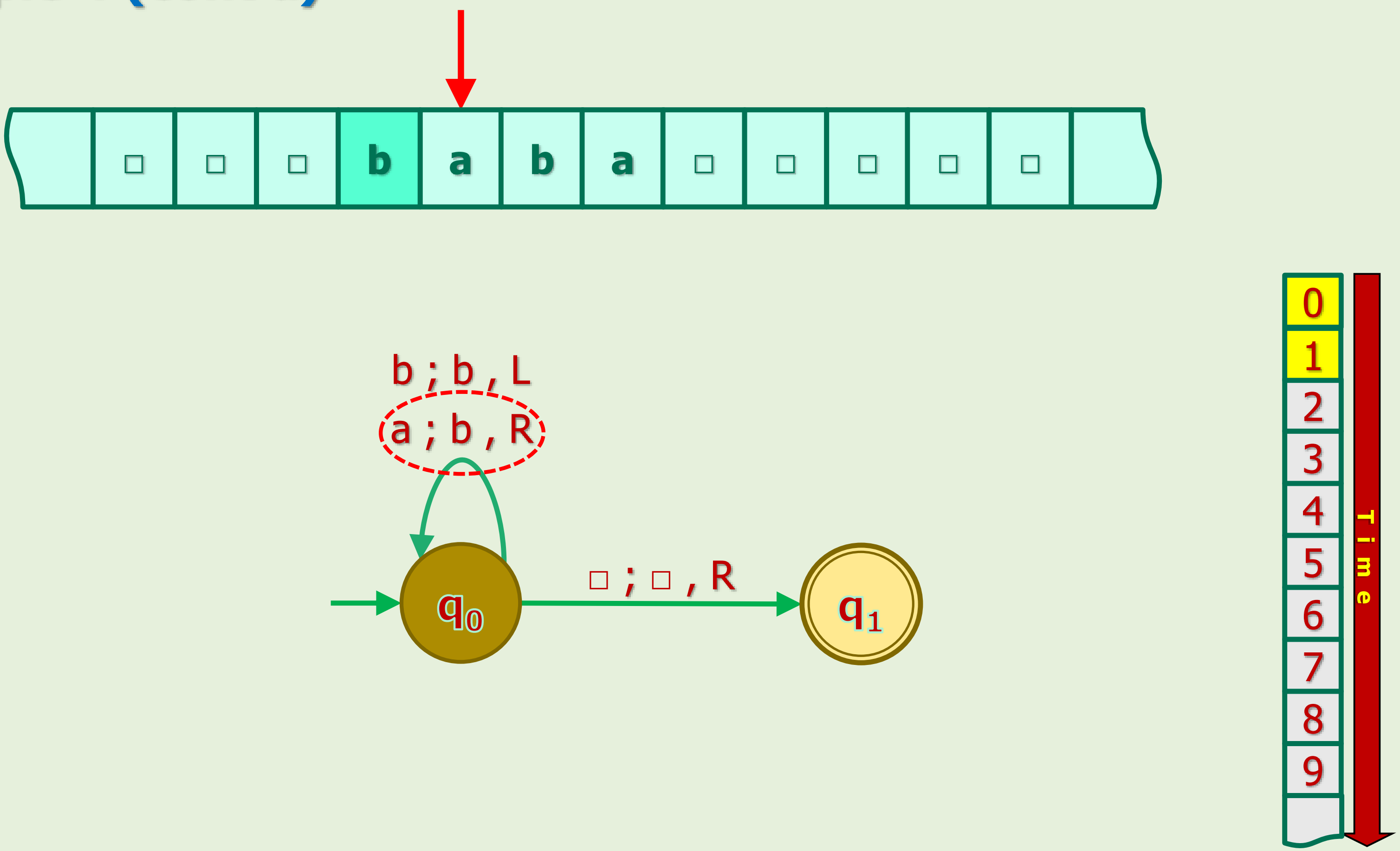
5. TMs in Action

Example 4 (cont'd): Starting Configuration



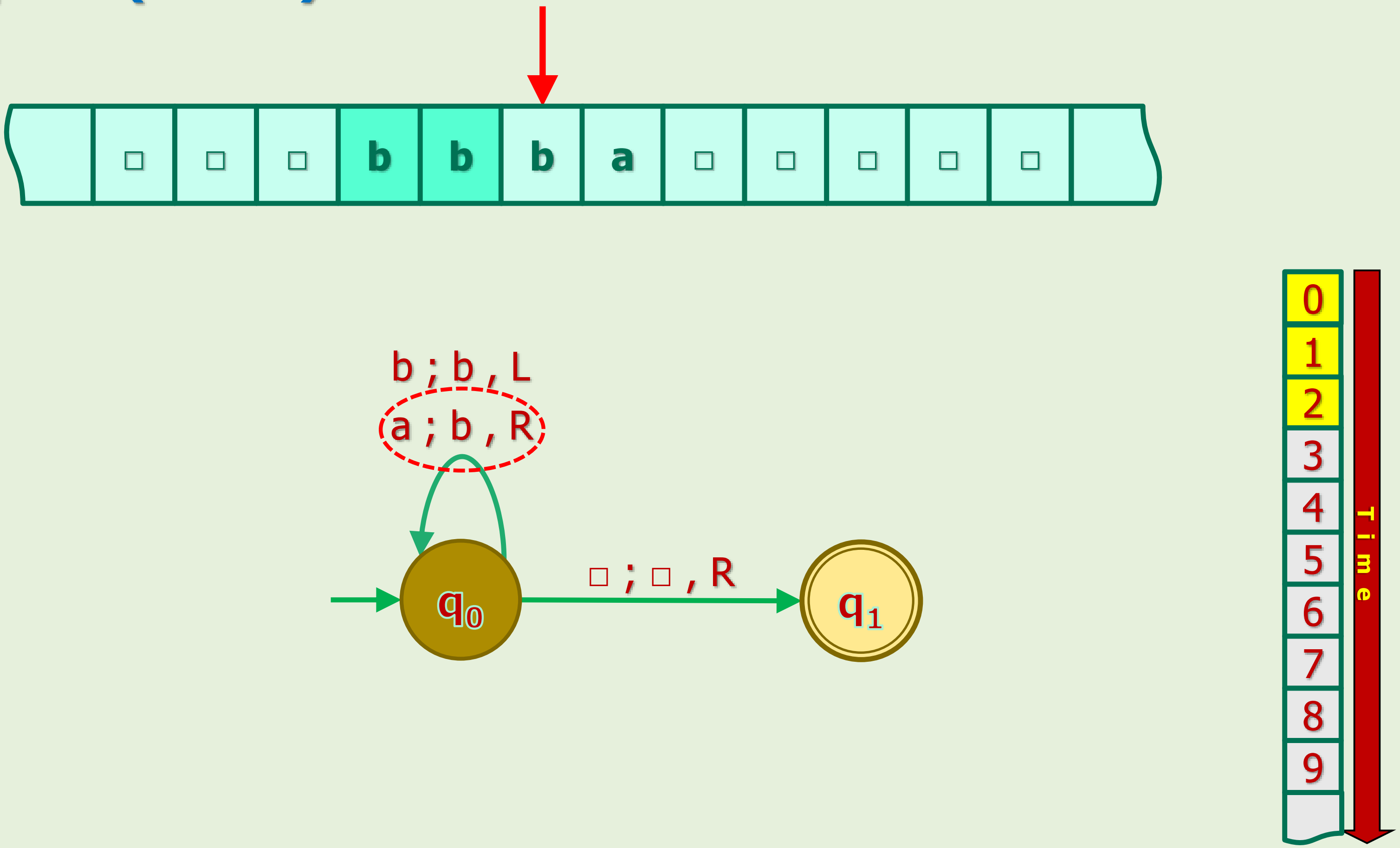
5. TMs in Action

Example 4 (cont'd)



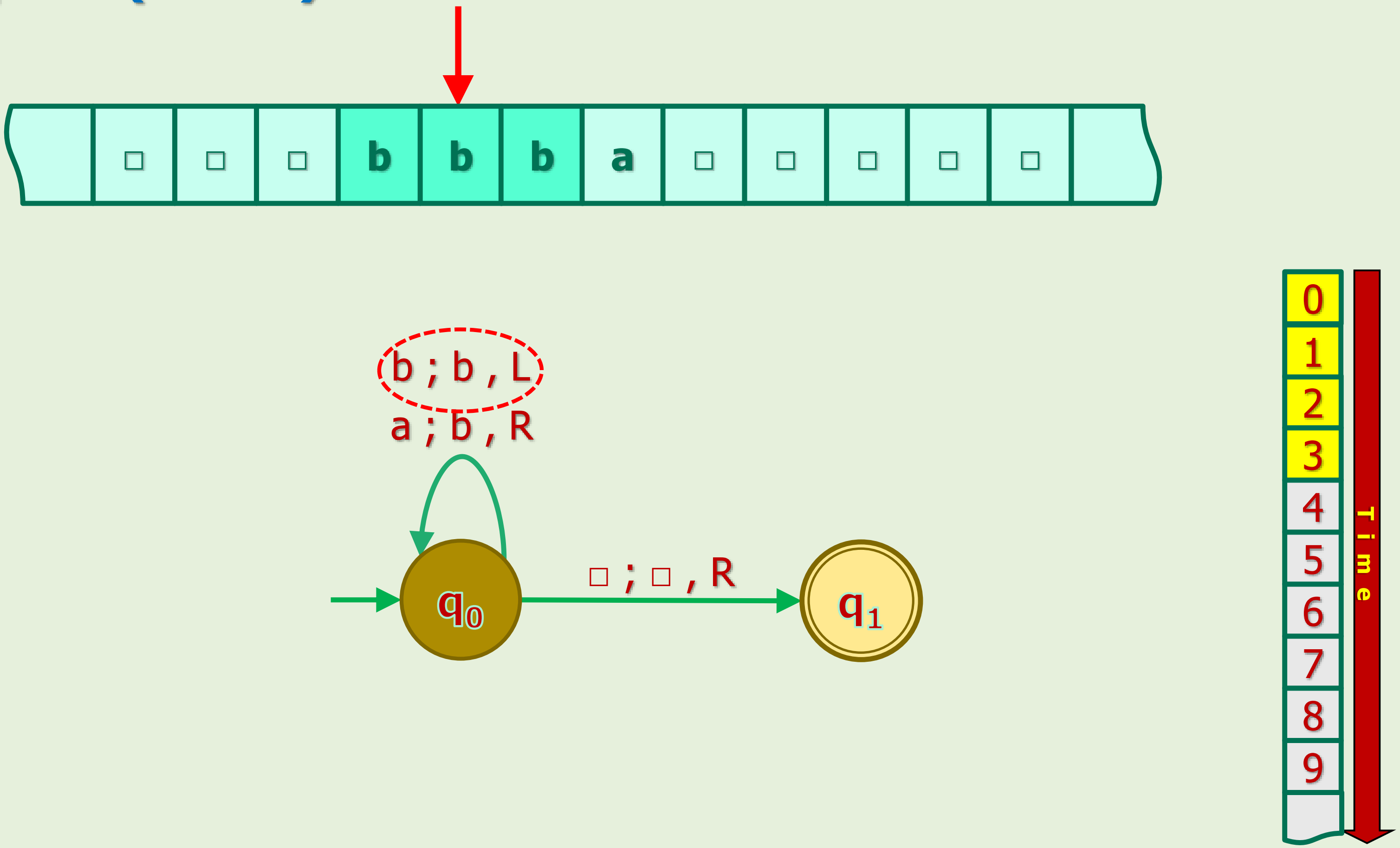
5. TMs in Action

Example 4 (cont'd)



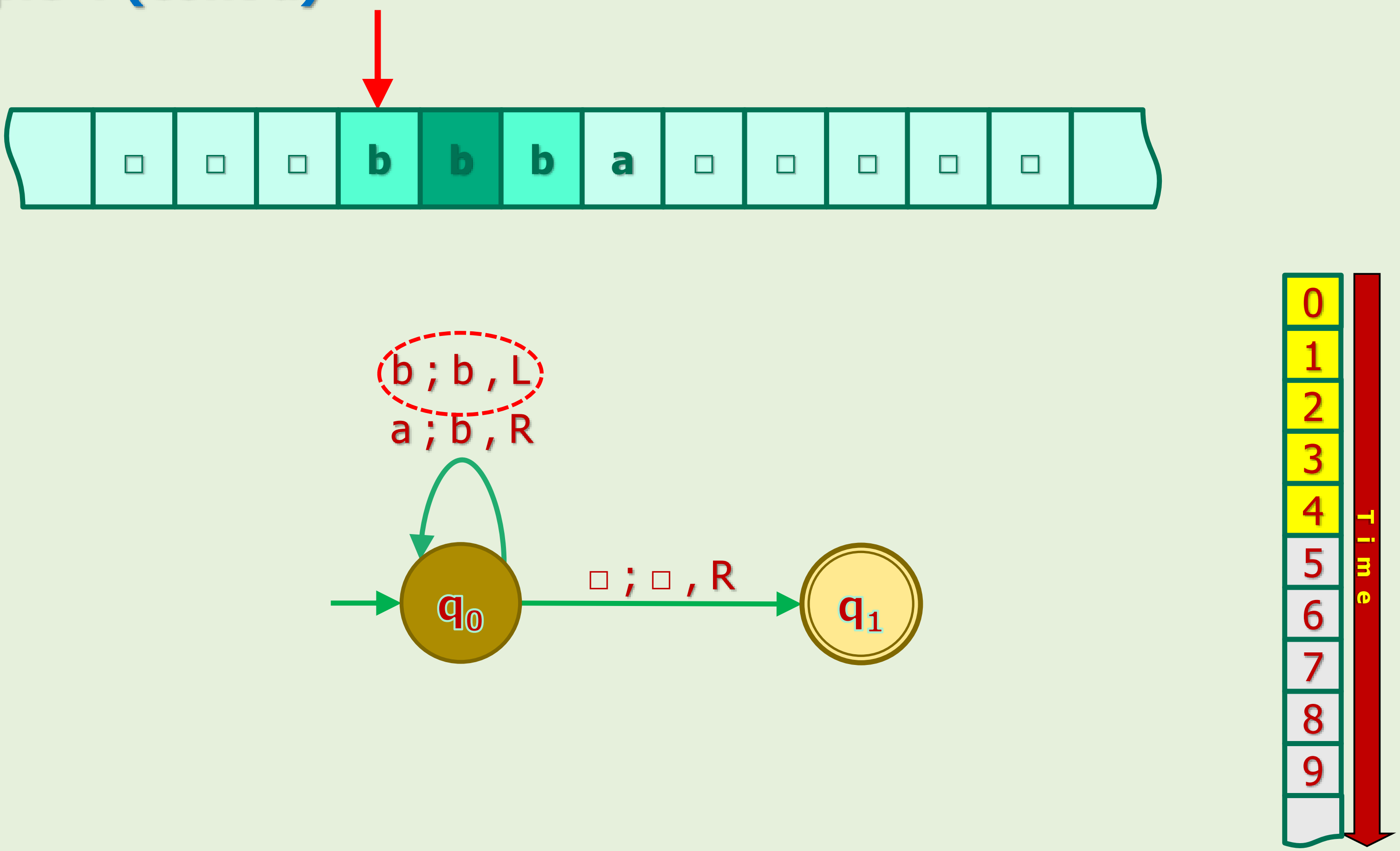
5. TMs in Action

Example 4 (cont'd)



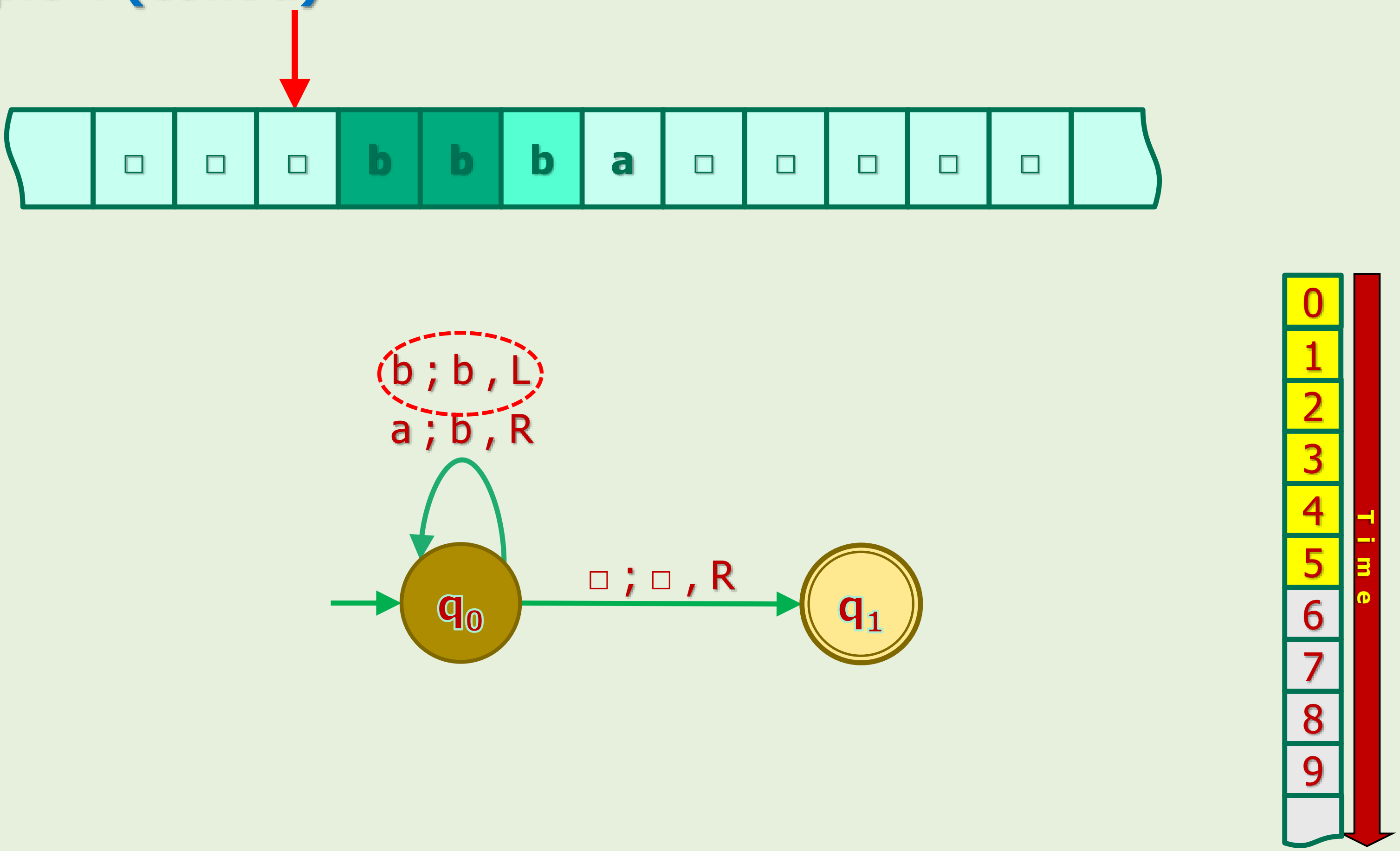
5. TMs in Action

Example 4 (cont'd)



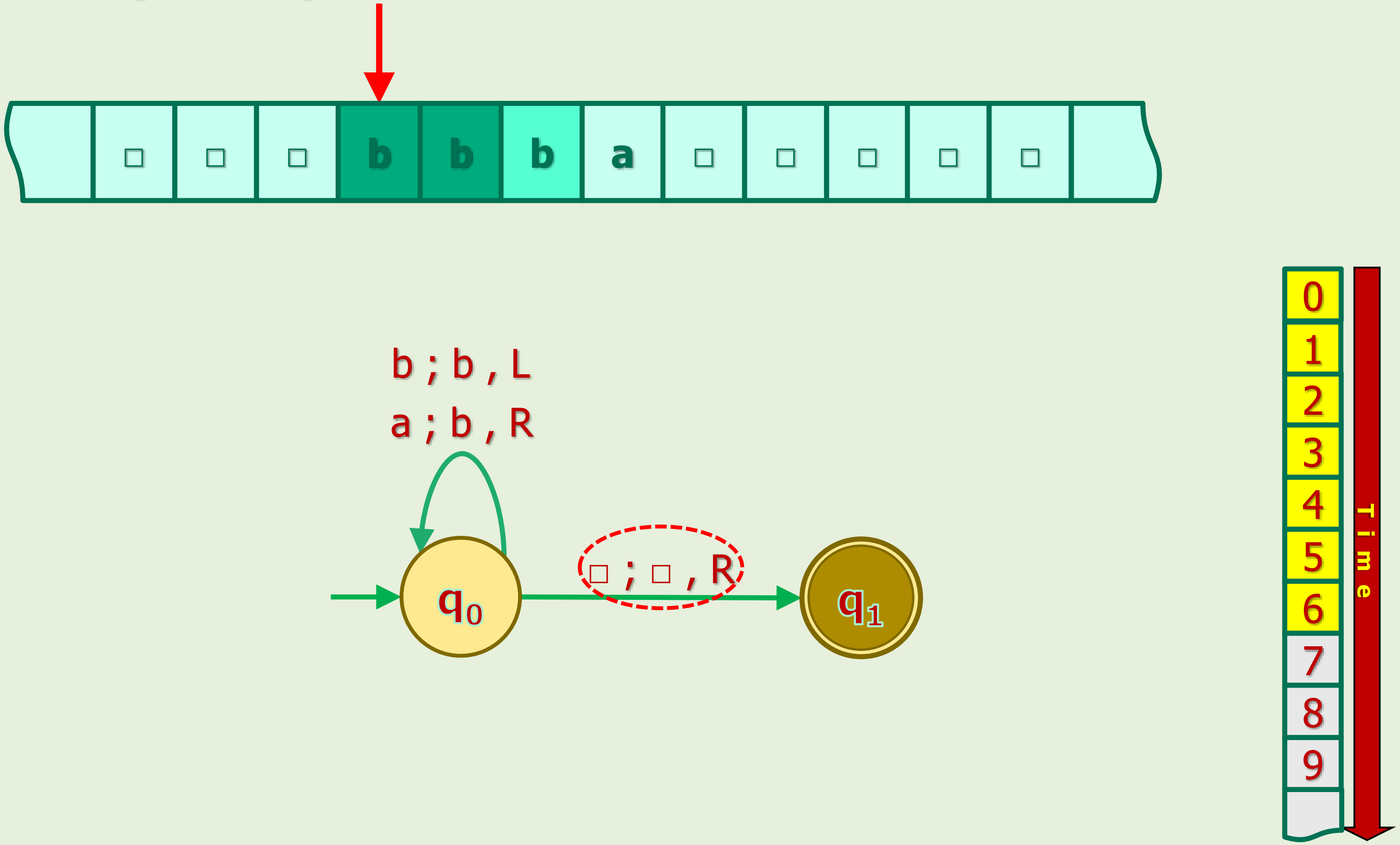
5. TMs in Action

Example 4 (cont'd)



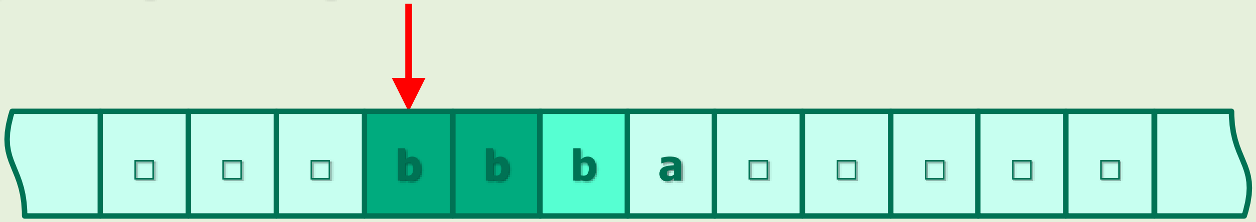
5. TMs in Action

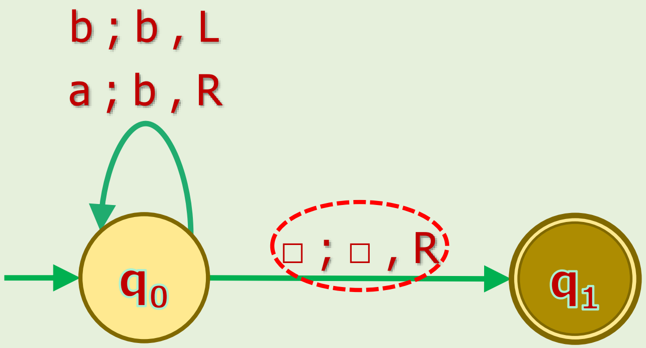
Example 4 (cont'd)

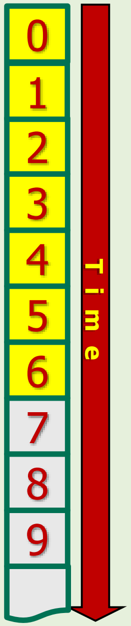


5. TMs in Action

Example 4 (cont'd)



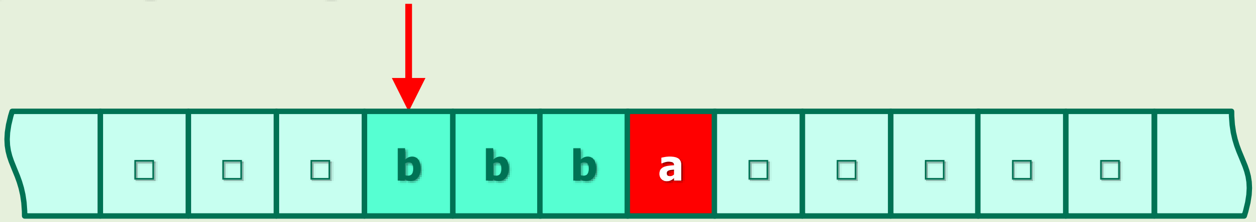




- The machine has no more transition.
- So, it halts.

Was The String Accepted?

Example 4 (cont'd)



- The machine halted in an accepting state.
- But the last symbol of the string (i.e. 'a') was never reached.

Question

- Was the string "aaba" accepted?

Answer

- It depends on how we define the string acceptance in TMs.

Was The String Accepted?

- If we judge based on the criteria of previous machines that were:

$$(\mathbf{h} \land \mathbf{c} \land \mathbf{f}) \leftrightarrow \mathbf{a}$$

- Then the answer would be "NO" because ...

  all symbols were not consumed.

- But in general, consuming the input symbols is meaningless for TMs. Why?
  - Because the cursor can move left or right.
  - So, some symbols might be visited several times while some other never reached.

Was The String Accepted?

- Therefore, it's the responsibility of the TMs designer (you) to define when a string is accepted/rejected.

- It could depend on your algorithm and the problem that you're solving.
- If your algorithm needs to consume all symbols, like other machines, then you need to design the TM accordingly.
- It means, you need to design it in such a way that it halts in an accepting state when all symbols are visited.

4.4. How TMs Accept/Reject Strings

Logical Representation of Accepting Strings

- If we remove **c** from the conditions, then theoretically, the logical representation of accepting strings is ...

$$\left.\begin{array}{l}\text{TMs accept a string w.} \equiv \mathbf{a} \\ \text{IFF} \\ \text{They halt.} \equiv \mathbf{h} \\ \text{AND} \\ \text{They are in an accepting (final) state.} \equiv \mathbf{f}\end{array}\right\} \quad (\mathbf{h} \wedge \mathbf{f}) \leftrightarrow \mathbf{a}$$

- Shorter version:
  The string w is accepted iff the TM halts in an accepting state.

4.4. How TMs Accept/Reject Strings

Logical Representation of Rejecting Strings

$$\sim(h \wedge f) \leftrightarrow \sim a$$

$$(\sim h \vee \sim f) \leftrightarrow \sim a$$

Translation

TMs reject a string w. ≡ $\sim$**a**

IFF

They do NOT halt. ≡ $\sim$**h**

OR

They are NOT in an accepting (final) state. ≡ $\sim$**f**

4.4. How TMs Accept/Reject Strings: Notes

JFLAP's Behavior

- JFLAP has two different settings for accepting a string that can be set in its preferences:

- Accept by Final State
  - In this mode, if the current state is accepting state, JFLAP forces to halt and accept the string regardless of having more transition or not!

- Accept by Halting
  - In this mode, when the TM halts, it accepts the string regardless of being in an accepting-state or not!

4.4. How TMs Accept/Reject Strings: Notes

JFLAP's Behavior

- You can also choose both.
- In this case, it accepts whichever comes first!

- For uniformity, we'll be using only "Accepting by Final State".

- Therefore, make sure that the flow of your design does not pass an accepting-state.
- In other words, always put your accepting-state where the machine should halt.

Team Formation

References

1. Linz, Peter, "An Introduction to Formal Languages and Automata, 5th ed.," Jones & Bartlett Learning, LLC, Canada, 2012
2. Kenneth H. Rosen, "Discrete Mathematics and Its Applications, 7th ed.," McGraw Hill, New York, United States, 2012
3. Michael Sipser, "Introduction to the Theory of Computation, 3rd ed.," CENGAGE Learning, United States, 2013
   ISBN-13: 978-1133187790
4. Wikimedia Commons, https://commons.wikimedia.org/wiki/Category:Animations_of_machinery
5. https://en.wikipedia.org/wiki/Turing_Award
6. https://en.wikipedia.org/wiki/Alan_Turing
7. https://www.turing.org.uk/

Appendix A: Alan Turing

Ref: https://www.awesomestories.com/asset/view/Alan-Turing-Early-Years

Appendix A: Alan Turing

Ref: https://www.histclo.com/essay/war/ww2/code/ncs/ger/ultra/ind/turn/tur-ed.html

Appendix A: Alan Turing

Ref: https://www.exploringsurreyspast.org.uk/themes/people/scientists/alan_turing/

Walton Athletics Club, going to a race

Appendix A: Alan Turing

Ref: https://cacm.acm.org/magazines/2017/8/219602-turings-pre-war-analog-computers/abstract

Walton Athletics Club: He was a professional runner.

Appendix A: Alan Turing

Ref: https://www.exploringsurreyspast.org.uk/themes/people/scientists/alan_turing/

Enigma machine at Bletchley Park