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

Spring 2018

Agenda of Day 16

- Solution and Feedback of Quiz 5
- Summary of Lecture 14
- Lecture 15: Teaching ...
  - Turing Machines (Part 1)
- About Teams

Solution and Feedback of Quiz 5 (Out of 23)

Metrics	Section 1	Section 2	Section 3
Average	18.5	18.6	17.6
High Score	22	22	23
Low Score	12	13	7

Summary of Lecture 14: We learned ...

NPDAs: λ-transition

- We create a λ-transition by putting λ in the condition positions.
- For NPDAs, a λ-transition is labeled as:

  $\lambda, \lambda; w$

- Note that w is a string and can be λ.
- So, "λ, λ; λ" is a λ-transition and is used extensively.
- As usual, machines may start parallel processing when they encounter λ-transition.

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 ...
  - δ: $Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow$ a finite subset of $Q \times \Gamma^*$
  - $\delta(q_1, a, x) = \{(q_2, yx), (q_3, \lambda)\}$

Any question?

Summary of Lecture 14: We learned ...

NPDAs vs NFAs

- NPDAs are more powerful because ...
  - ... they can accept all languages accepted by NFAs.
  - ... but there are some languages that NFAs cannot accept but NPDAs can such as: $a^n b^n$ and ww^R.

- There are still non-regular languages that cannot be accepted by NPDAs, such as:

  ww and $a^n b^n c^n$

Any question?

Template for Constructing a New Class of Automata

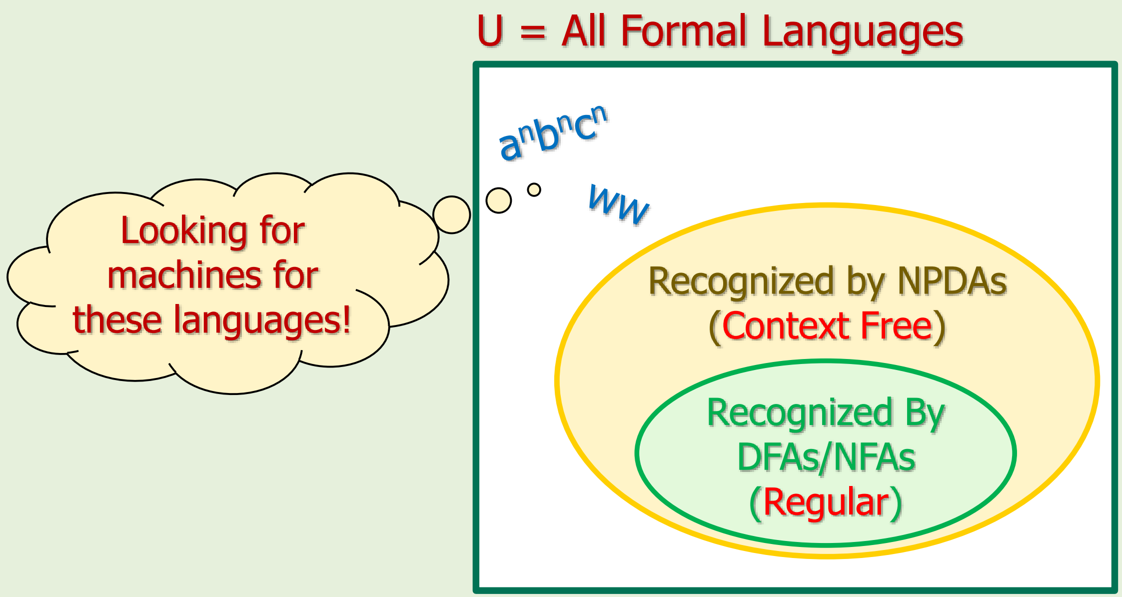
Repeated

- To construct a new class of automata, we need to respond the following questions:

1. Why do we need a new class of machines? (Justification)
2. Name of the new class
3. Building blocks of the new class
4. How they work
   4.1. What is the starting configuration?
   4.2. What would happen during a timeframe?
   4.3. When would the machines halts?
   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 We Need a New Class

- This was our last conclusion:
  - There are some languages for which we cannot construct NPDAs!



1. Why We Need a New Class

What Was Missing in NPDAs?

- We had stack for counting but ...
  - ... stack is not so flexible in storing and retrieving data.
  - ... we lose some data when we access the older data.

- We need more control on the memory.

- So, we are going to add some flexibility to the memory and construct a new class of automata.

2. Name of the New Class

- This machine was proposed by Alan M. Turing in 1936.
- That's why we call this new class as "Turing machine (TM)".
  - Both deterministic and nondeterministic TMs can be defined.
  - But the deterministic ones are more interesting.
- The deterministic TM is called as "standard TM".
- For convenience, we usually drop the "standard" and call it "Turing machine".
- Before introducing Turing machines, let's see who is "Turing"?

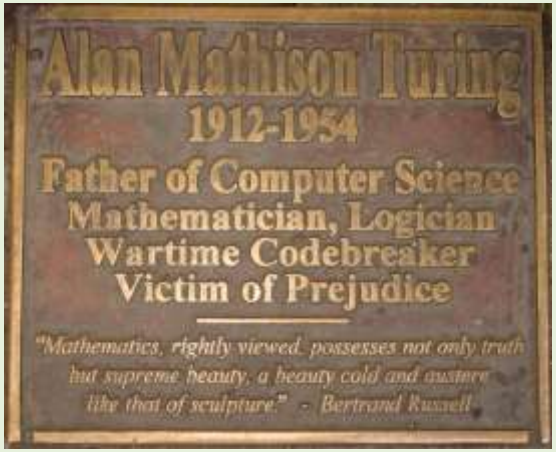
Who is Turing?

- Alan M. Turing (1912 – 1954) born in Britain.
- He was a:
  - mathematician,
  - logician,
  - cryptanalyst, and
  - theoretical biologist.
- He is known as the father of the:
  - theory of computation (Computer Science Foundation)
  - artificial intelligence
- 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" that could break Hitler's Enigma machine encryptions.

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



- 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

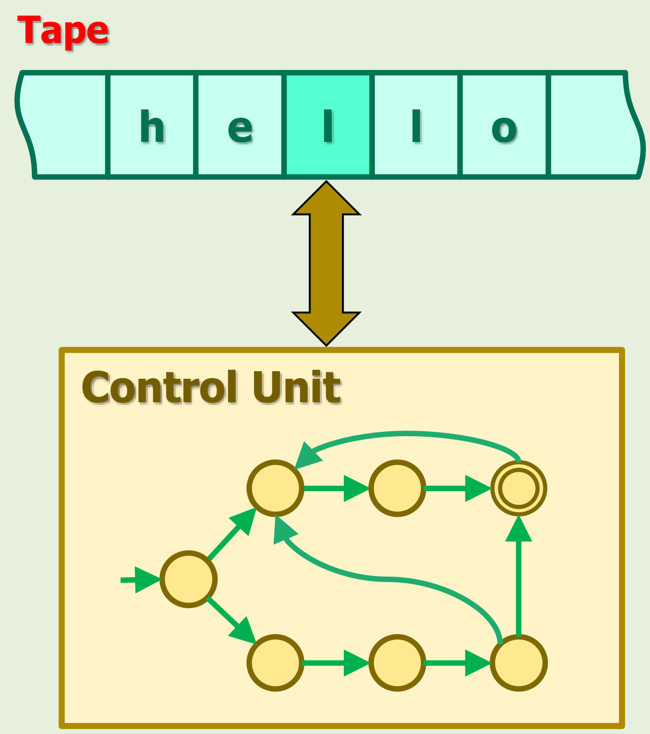
Movies

- 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

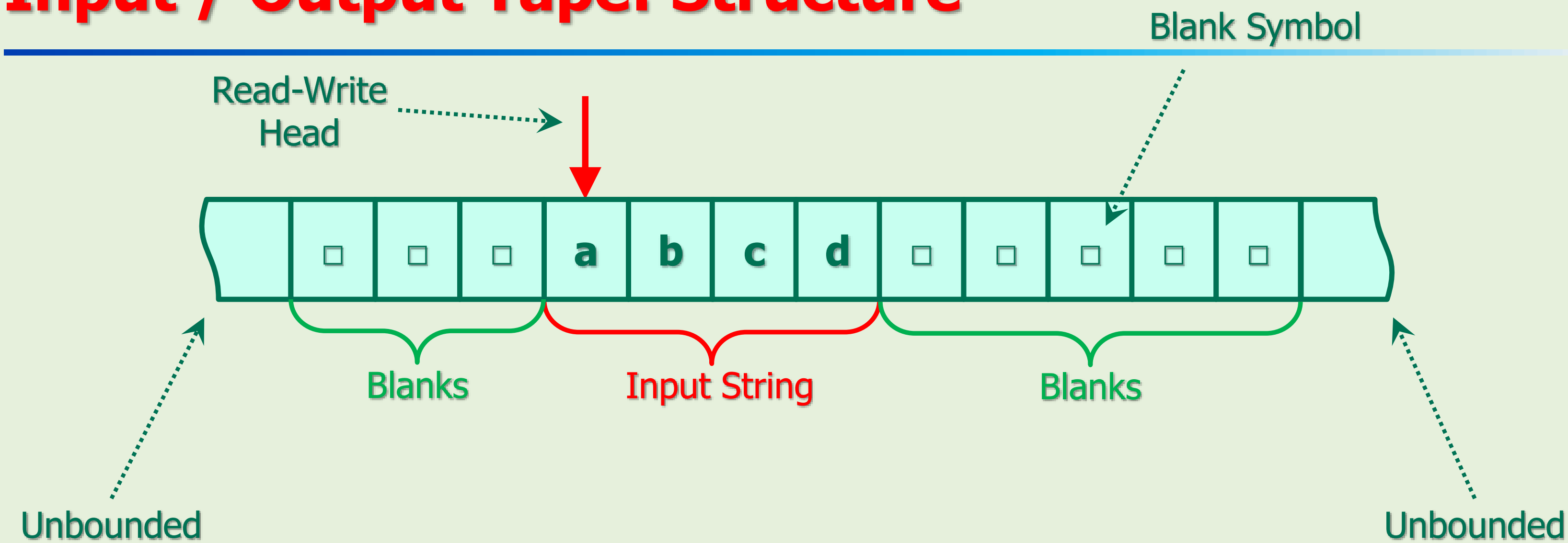
3. TMs Building Blocks

- A TM has 2 main blocks:
  1. Input / output tape
  2. Control unit



- Let's see each block in detail.

Input / Output Tape: Structure

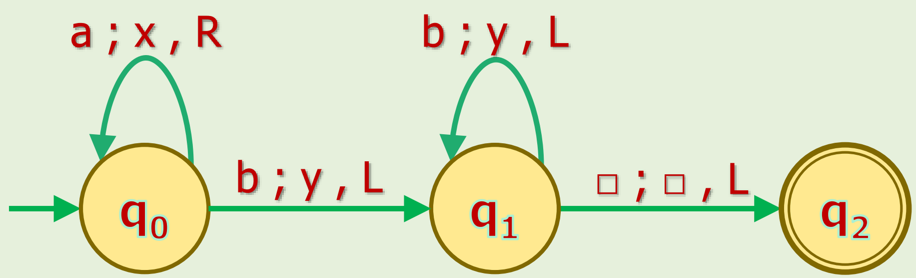


- The tape is unbounded from both sides.
- A read-write head reads a symbol, writes a symbol, and moves left or right.
- The input string can be written somewhere on the tape.
- The rest of the tape contains blank symbols '□'.

Control Unit: Structure

- The control unit (transition graph) of TMs look pretty much like NPDAs'.

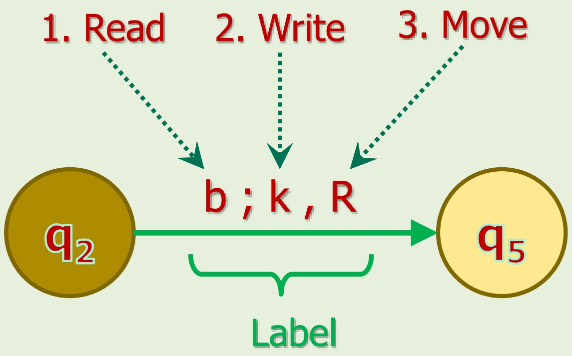
Example 1



- The only difference is how the edges are labeled.

- To understand the meaning of the 3 parts of the labels, let's analyze one transition.

Control Unit: Labels



- The label of an edge has 3 parts delimited by semicolon and comma:
  1. The input symbol (e.g. 'b') that should be read from the tape
  2. The symbol (e.g. 'k') that should be written into the tape
  3. The move direction of the read-write head ('L' = Left or 'R' = Right)

- We'll explain in detail what they mean shortly!

4. How TMs Work

4. How TMs Work

Repeated

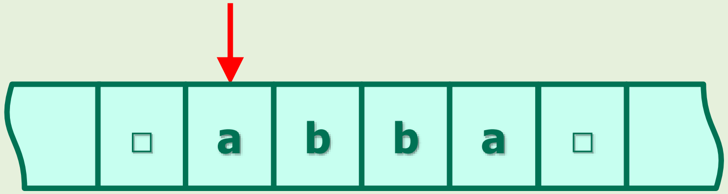
- To understand the operations of TMs (and any other class of machines), we should clearly respond to the following questions:

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.

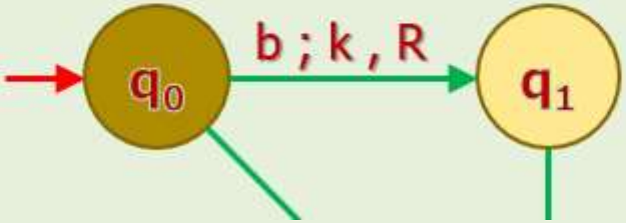


Input / Output Tape

- The tape has already been initialized with blank symbols '□'.
- The input string has already been written somewhere on the tape.
- The read-write head is pointing to the left-most symbol.
- The string can be partially the input and partially other contents.

Control Unit

- The control unit is set to initial state.



4.2. What Happens During a Timeframe

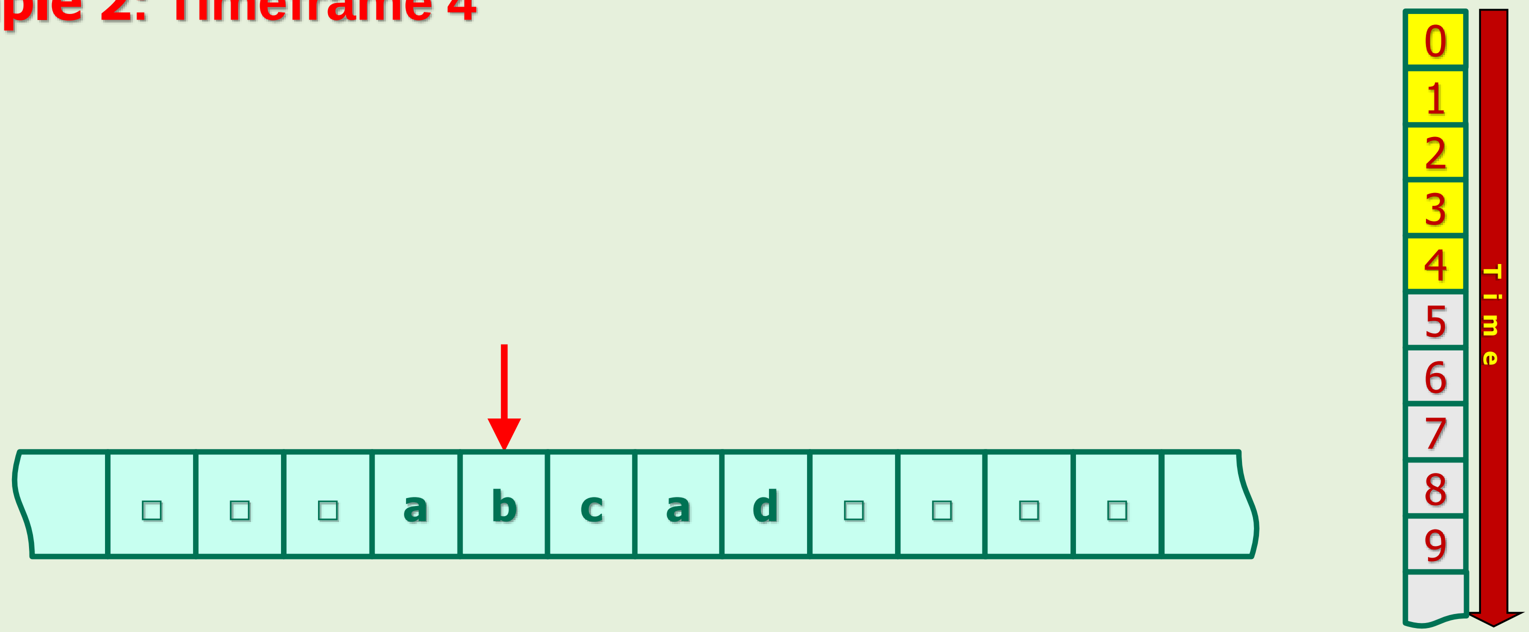
Transition

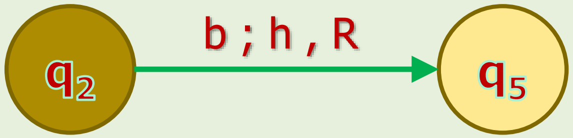
- Several tasks happen during a timeframe.
- The combination of these tasks is called a "transition".

- Before describing them in detail, let's visualize them through an example.

Transition Example

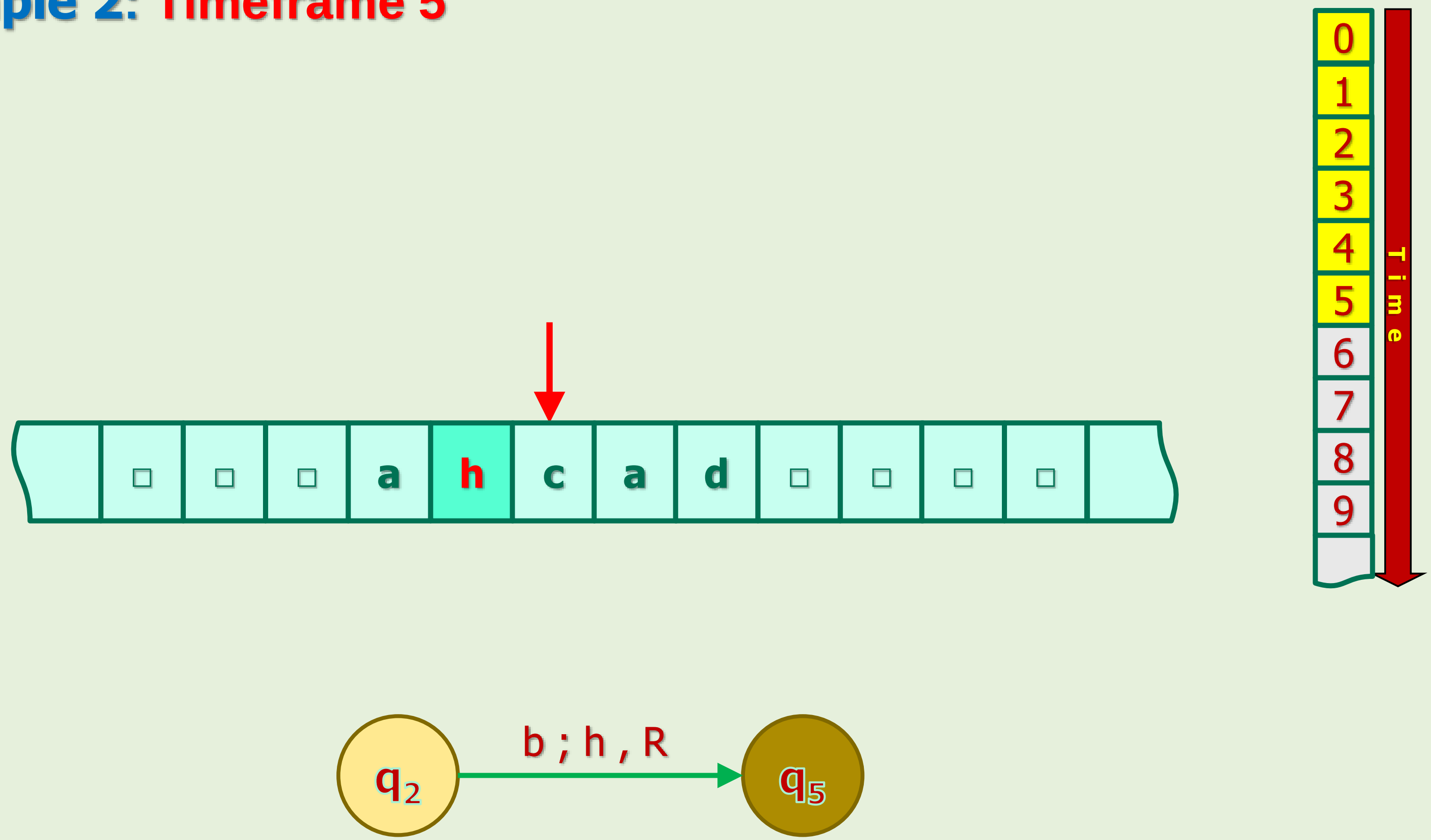
Example 2: Timeframe 4





Transition Example

Example 2: Timeframe 5

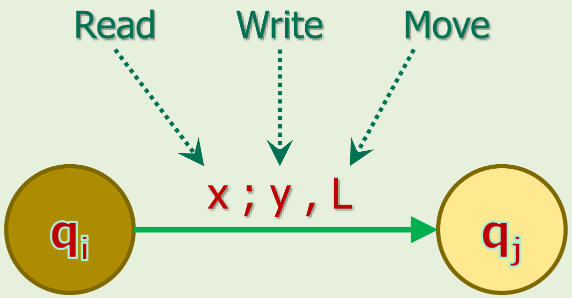


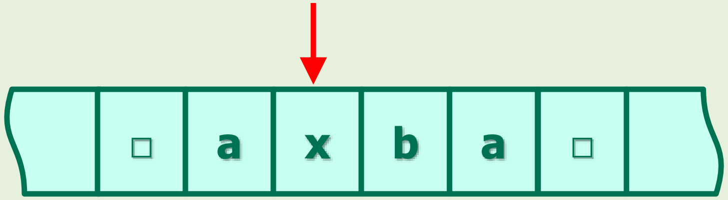
4.2. What Happens During a Timeframe

Transition

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

The Logic of Transitions





If (Condition)

– the input symbol is 'x'

How does the machine look like after this transition?

Then (Operation)

– replace 'x' with 'y' on tape

AND

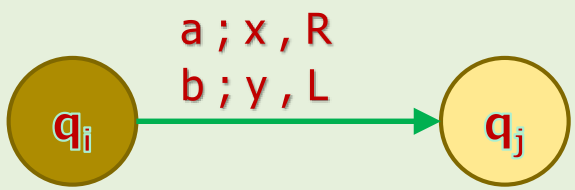
– move the head to the Left

AND

– transit to q_j

Multiple Labels

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



If	Then		If	Then
– the input symbol is 'a'	– replace it with 'x' AND – move to the right AND – transit to q_j	**OR**	– the input symbol is 'b'	– replace it with 'y' AND – move to the left AND – transit to q_j

Transition Examples

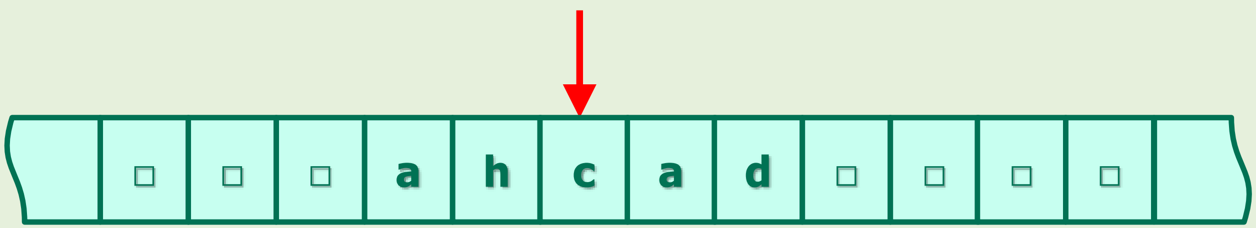
Transition Examples

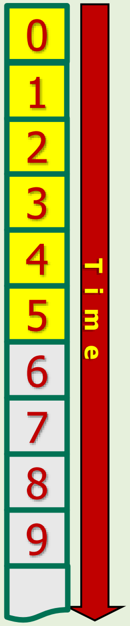
- The examples will show:
  - a partial transition graph
  - an input / output tape
  - a clock

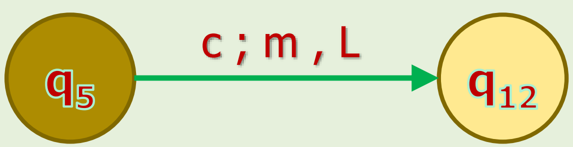
- We assume that the machine is in the middle of its operation at timeframe n.
- The question is: in what configuration would the machine be at timeframe n+1?

Transition Examples

Example 3: Timeframe 5

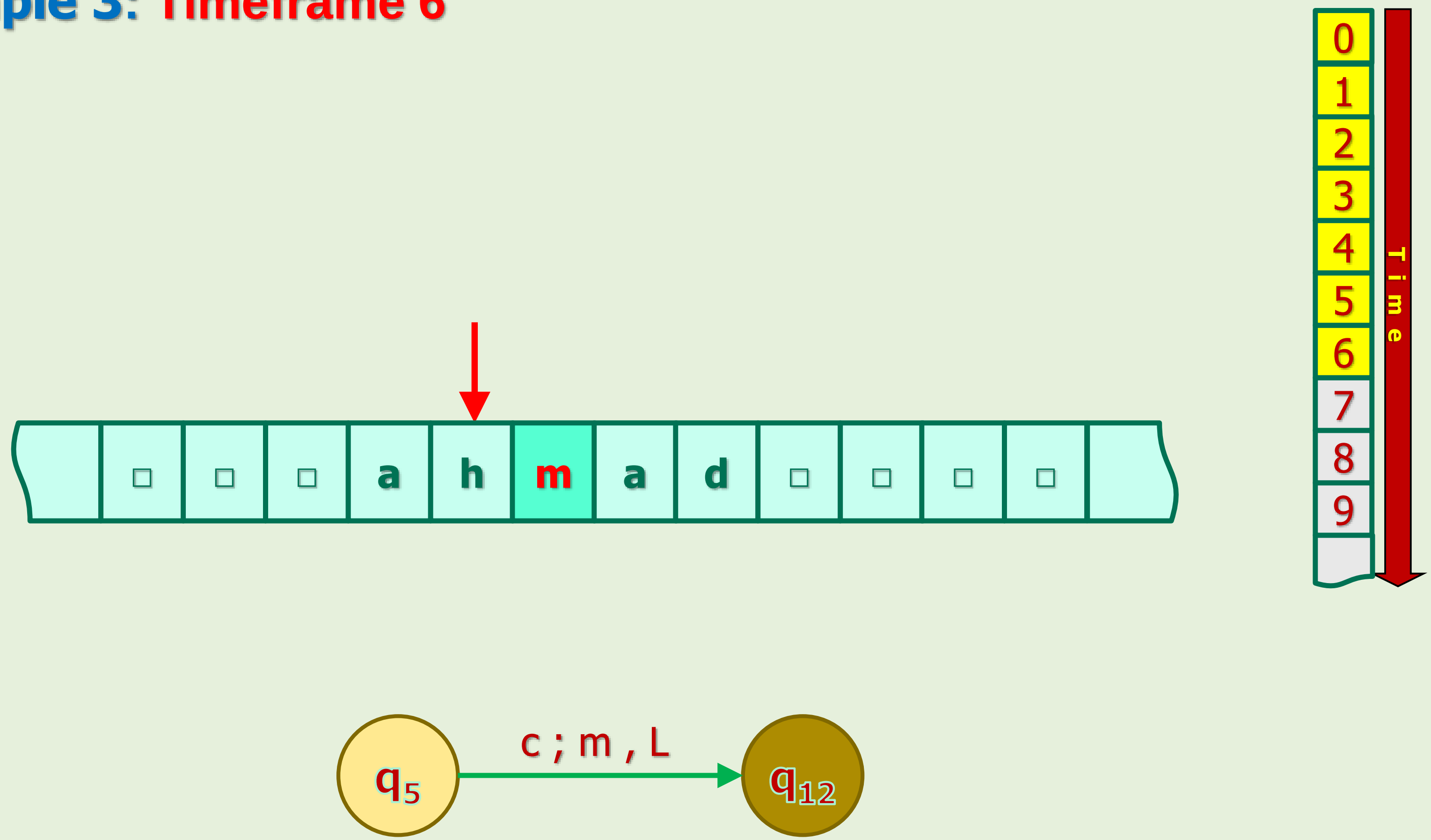






Transition Examples

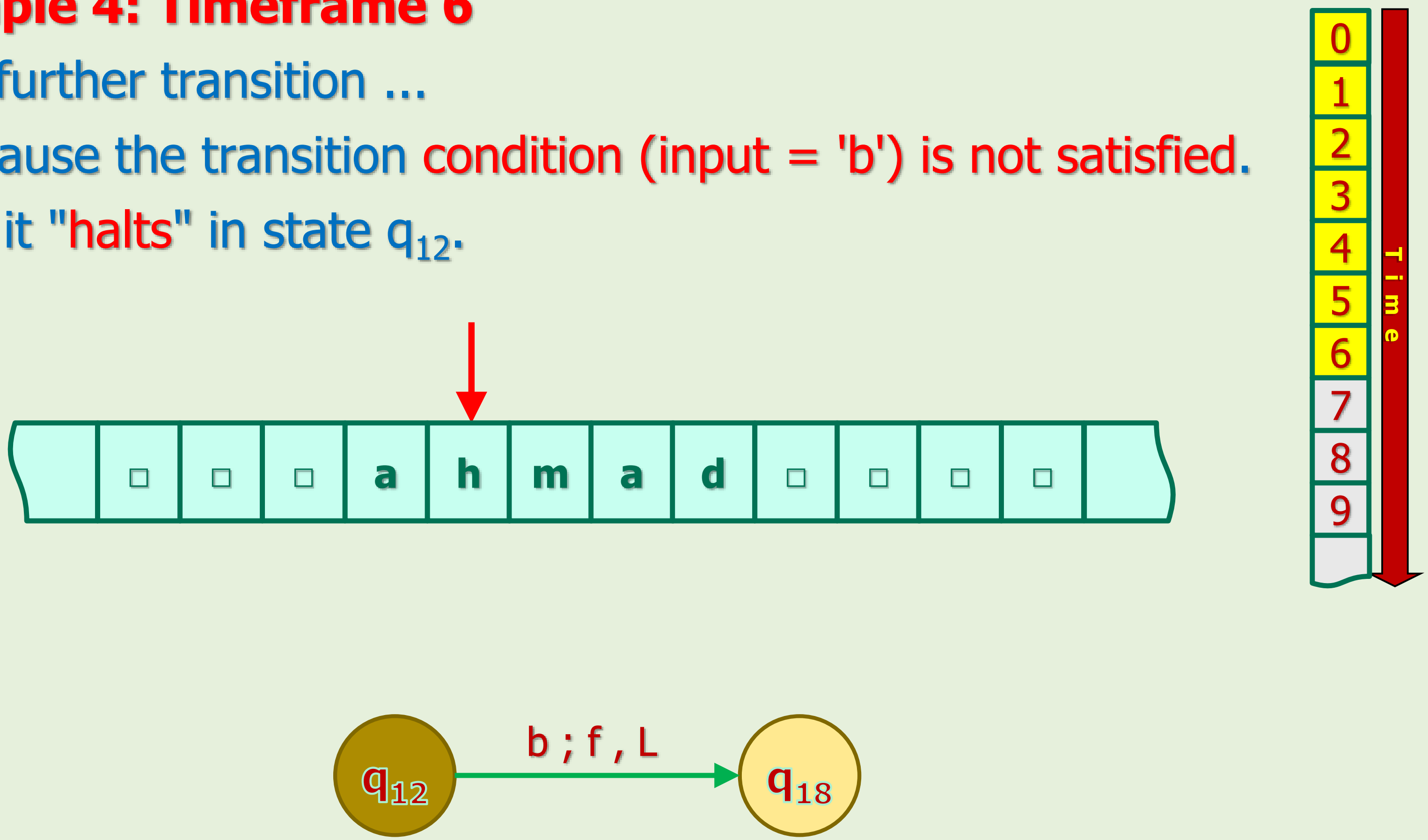
Example 3: Timeframe 6



Transition Examples

Example 4: Timeframe 6

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



4.3. When TMs Halt

- TMs halt when the next transition condition is NOT satisfied.

Halt Logical Representation

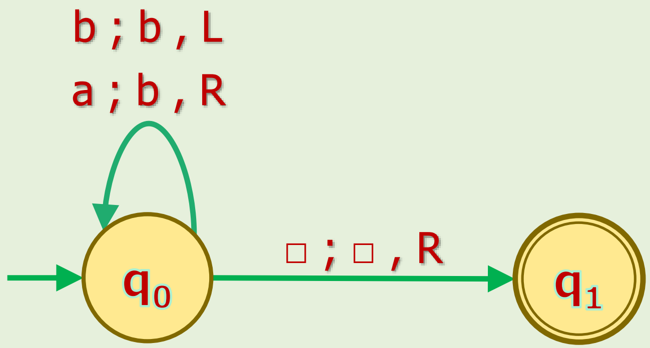
$$\left.\begin{array}{l}\text{TMs halt.} \equiv \mathbf{h} \\ \text{IFF} \\ \text{They have zero transition.} \equiv \mathbf{z}\end{array}\right\} \mathbf{z} \leftrightarrow \mathbf{h}$$

5. TMs in Action

5. TMs in Action

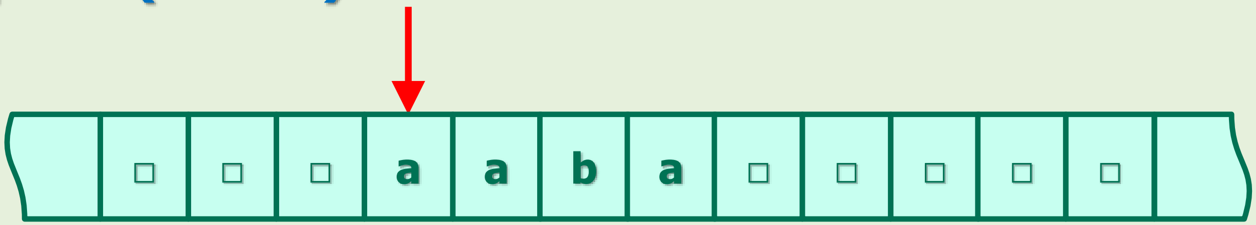
Example 5

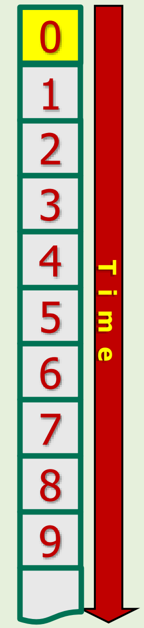
- Consider the following TM over $\Sigma = \{a, b\}$.
- Trace the machine's operations for the input "aaba".

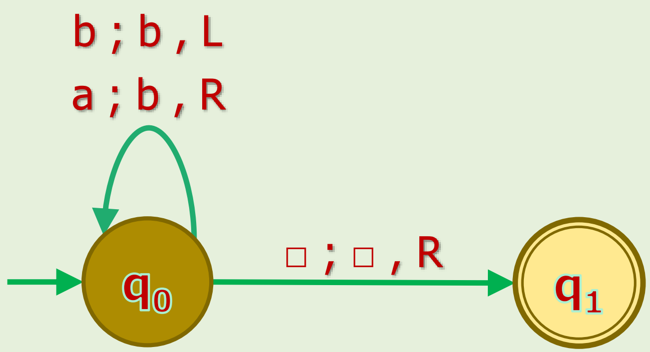


5. TMs in Action

Example 5 (cont'd)

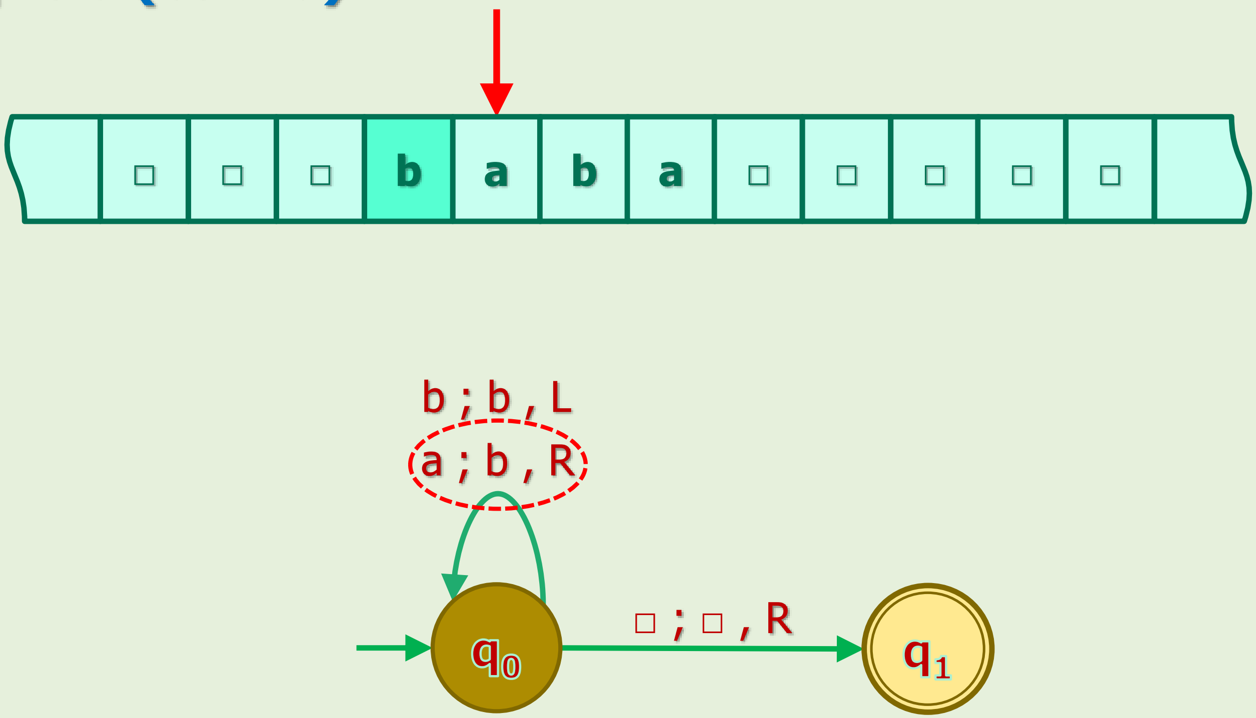






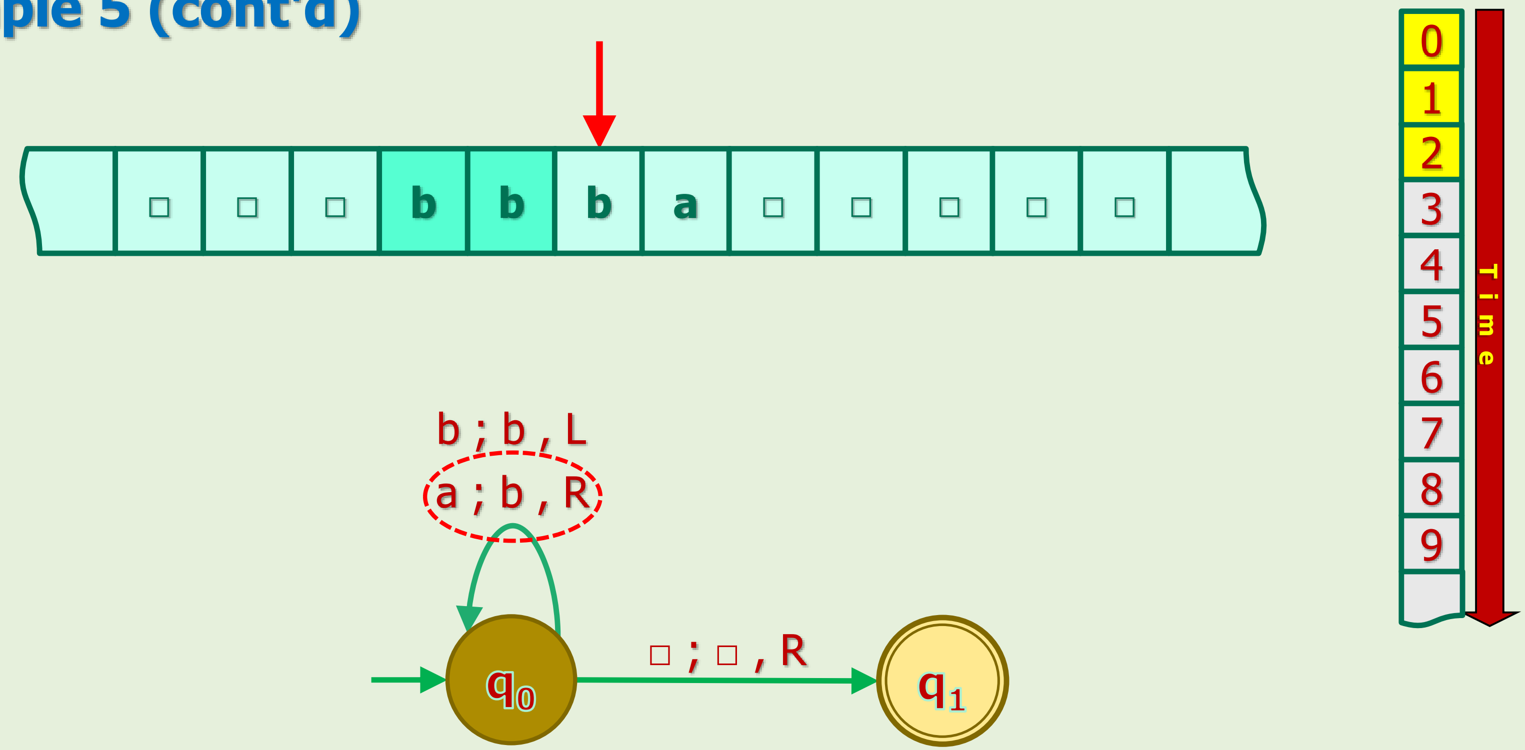
5. TMs in Action

Example 5 (cont'd)



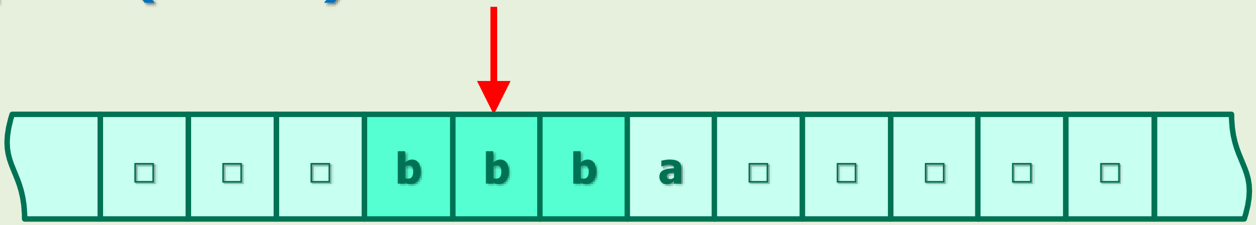
5. TMs in Action

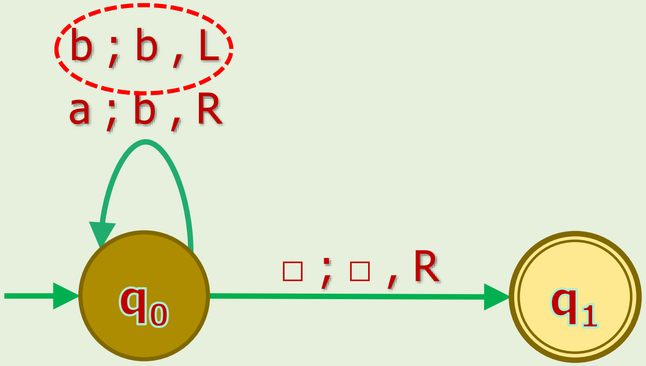
Example 5 (cont'd)



5. TMs in Action

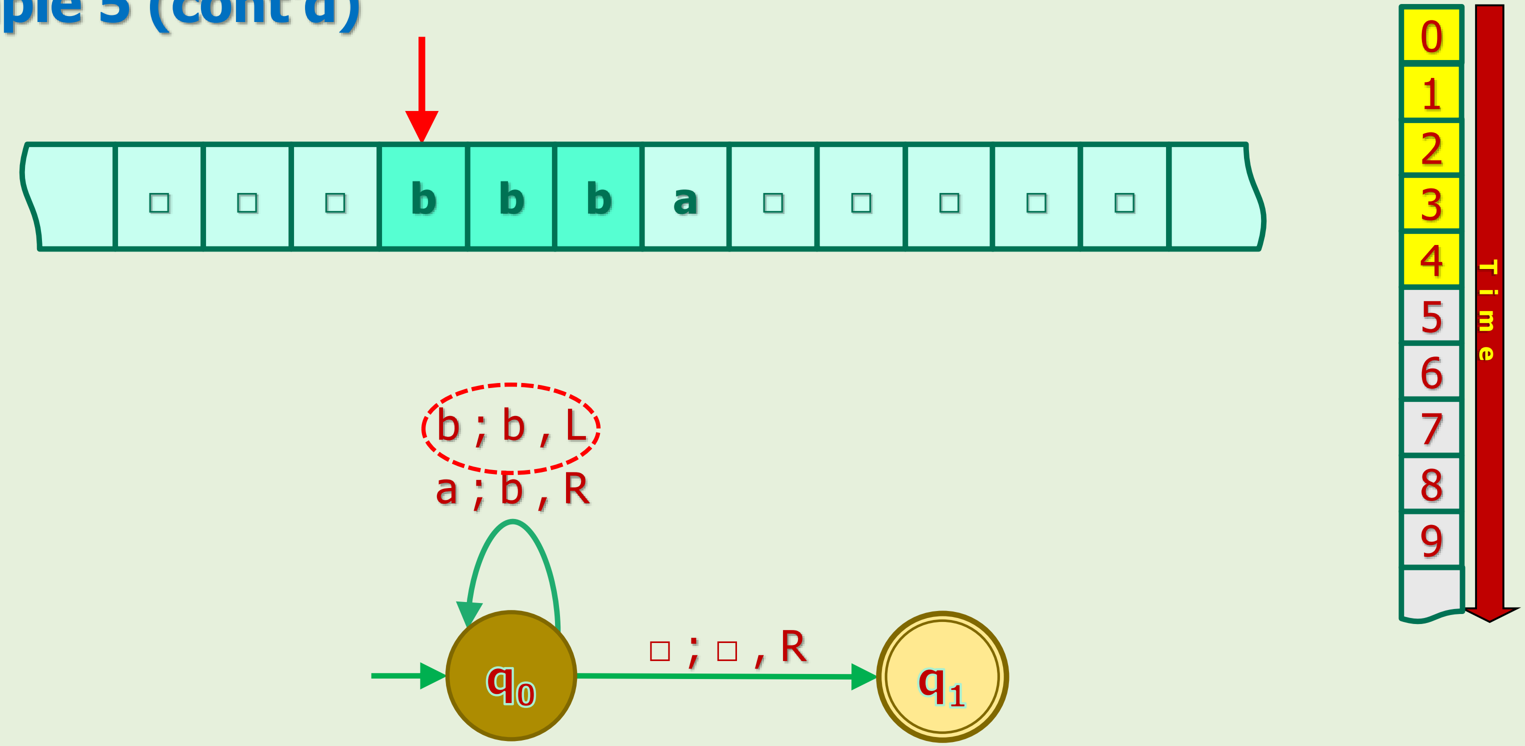
Example 5 (cont'd)





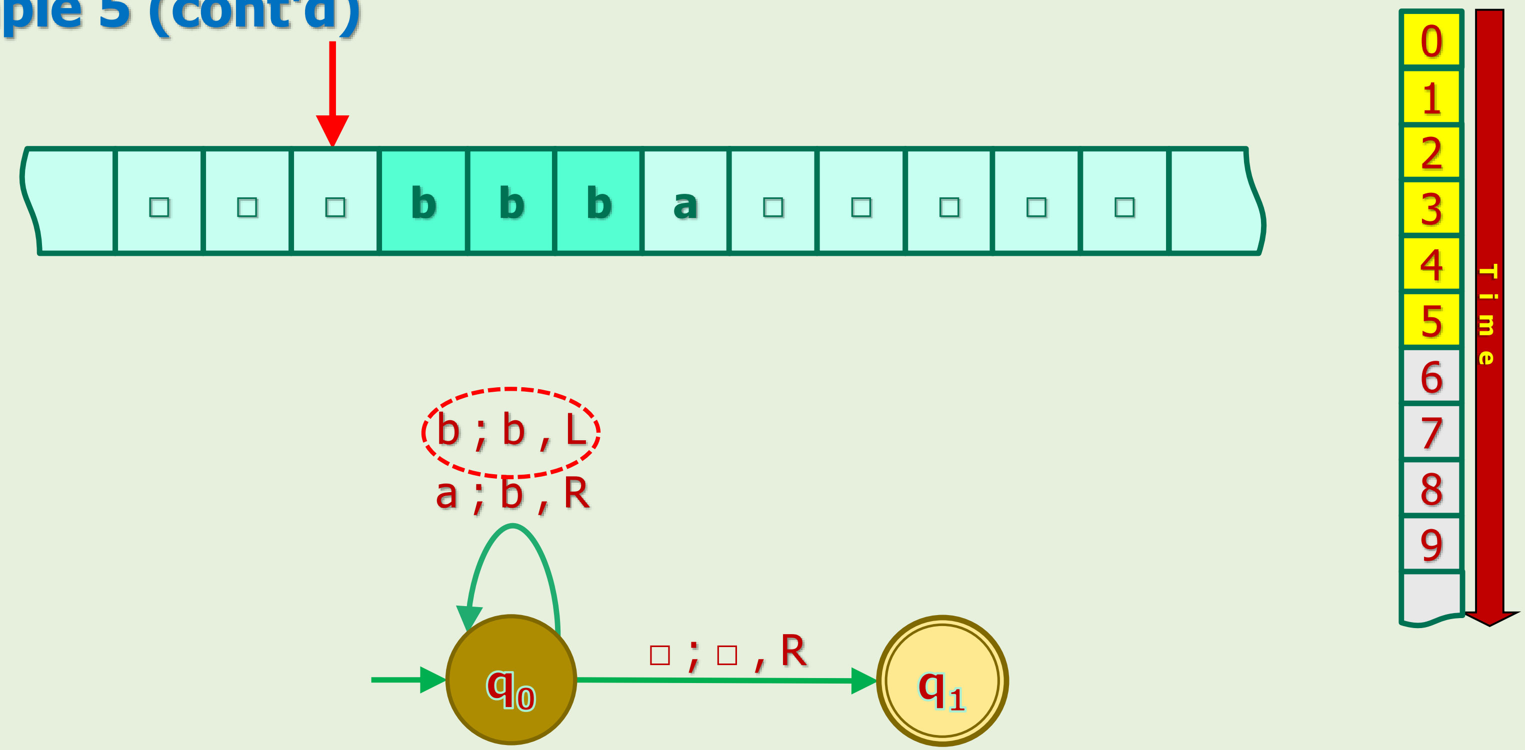
5. TMs in Action

Example 5 (cont'd)



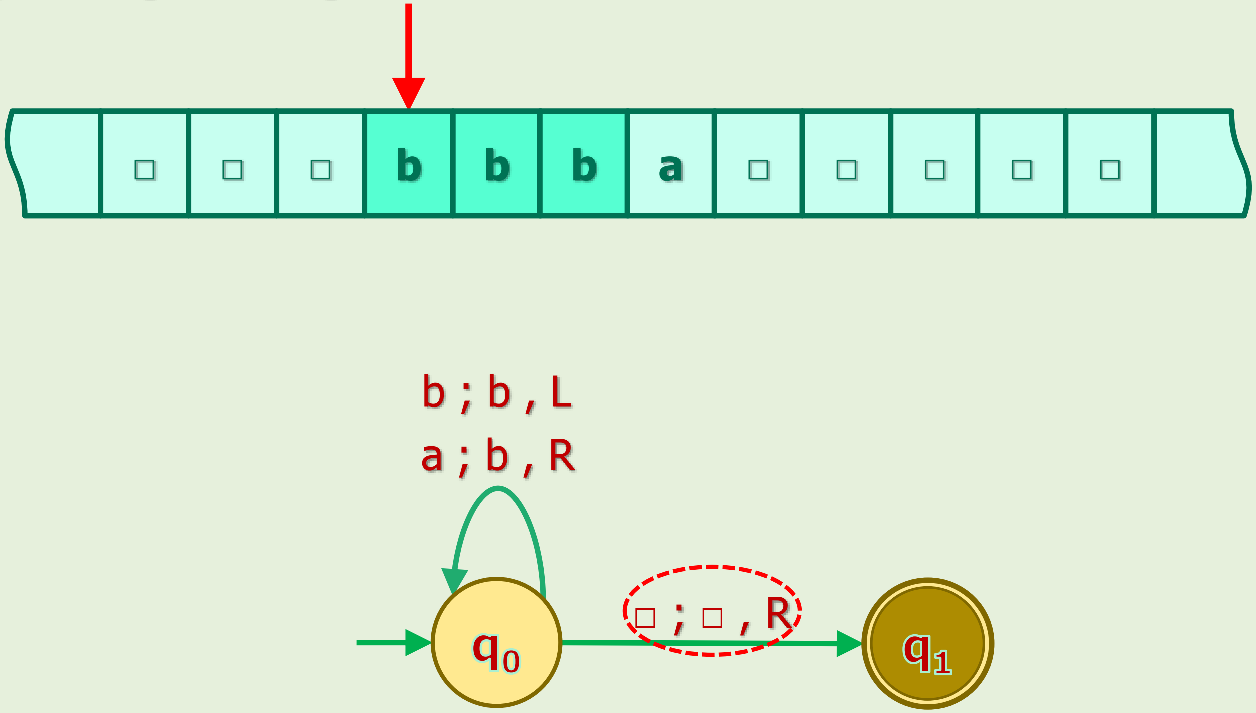
5. TMs in Action

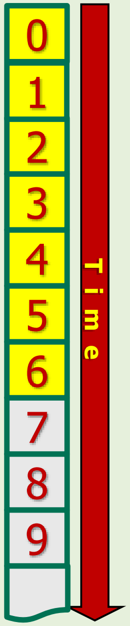
Example 5 (cont'd)



5. TMs in Action

Example 5 (cont'd)

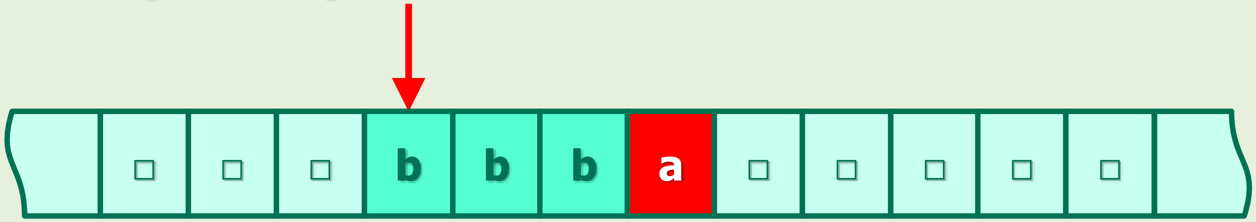




- The machine has no more transition.
- So, it halts.

Was The String Accepted?

Example 5 (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?

- 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:

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

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

all symbols were not consumed.

- But consuming the input symbols is meaningless for TMs. Why?
  - Because the head can move left as well as right.
  - So, some symbols might be scanned several times while some other never reached.

- In practice, this is the TMs' designers responsibility to make sure that the machine halts in an accepting state if all symbols are consumed.
- Otherwise, it should halts in a non-accepting state.

4.4. How TMs Accept/Reject Strings

Logical Representation of Accepting Strings

- So, if we remove **c** from the conditions, then theoretically, the criteria of accepting strings is ...

$$h \land f \leftrightarrow a$$

Translation

The string w is accepted. ≡ **a**

IFF

The TM halts. ≡ **h**

AND

The TM is in an accepting (final) state. ≡ **f**

- 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

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

Translation

The string w is rejected. ≡ $\sim$**a**

IFF

The TM does NOT halt. ≡ $\sim$**h**

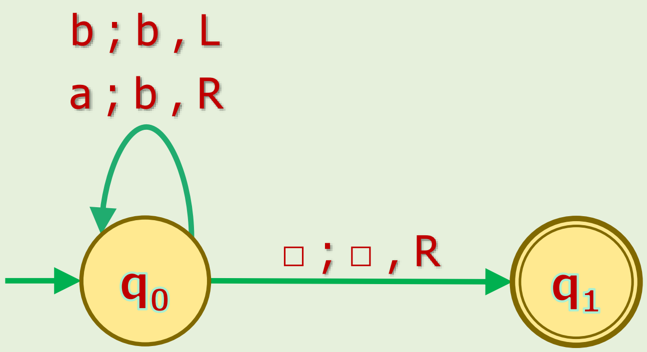
OR

The TM is NOT in an accepting (final) state. ≡ $\sim$**f**

One More Question

Example 5 (cont'd)

- The second question that might be raised (only for this example) is:
  - What was the goal of the TM in this example?

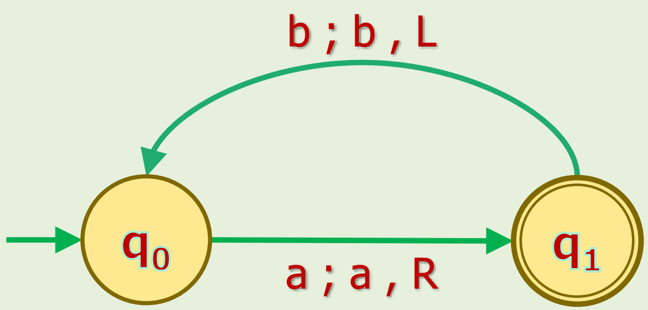


- How can you describe the operation was done by this TM?

A Special Phenomenon in TMs

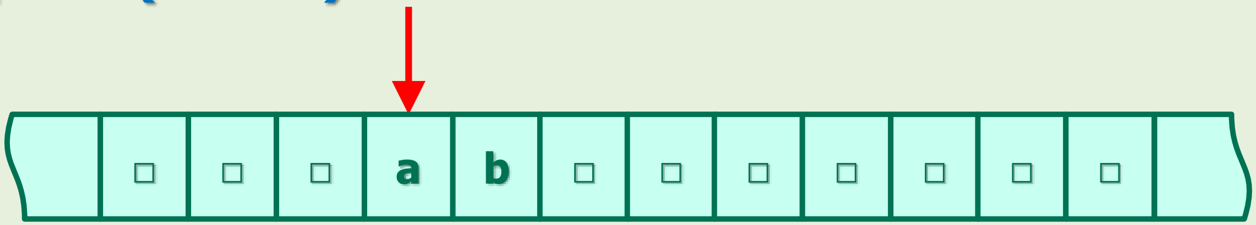
Example 6

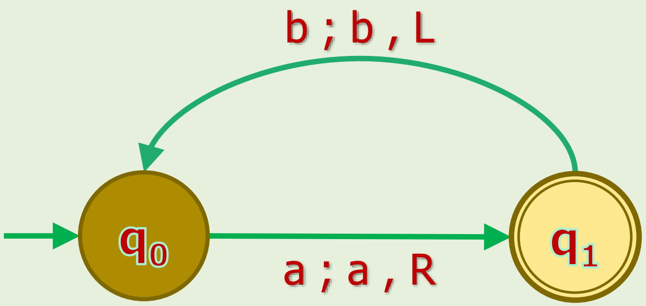
- Trace the following TM for the input "ab".



A Special Phenomenon in TMs

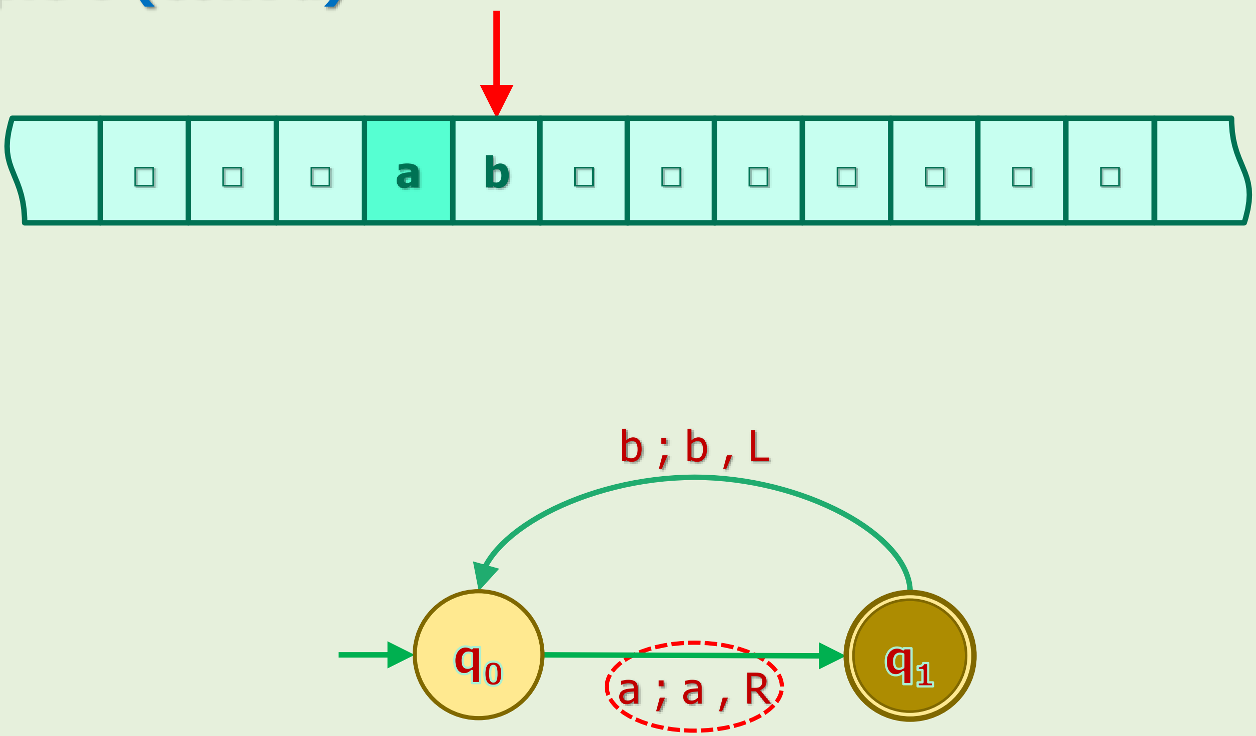
Example 6 (cont'd)





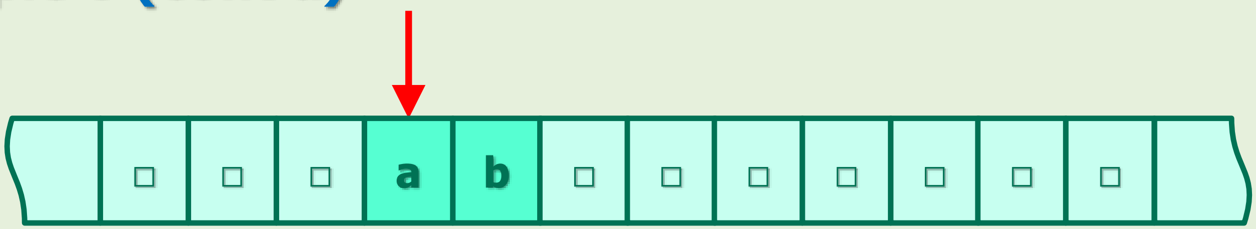
A Special Phenomenon in TMs

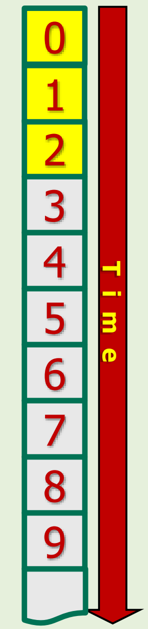
Example 6 (cont'd)

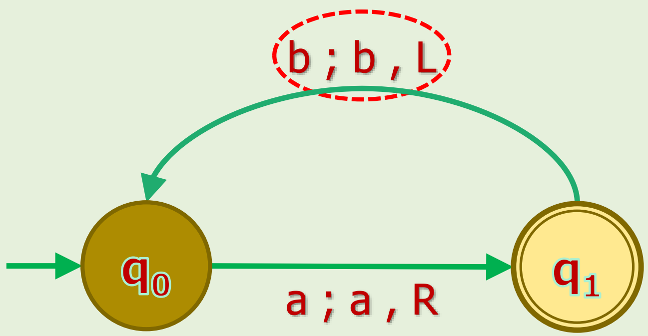


A Special Phenomenon in TMs

Example 6 (cont'd)

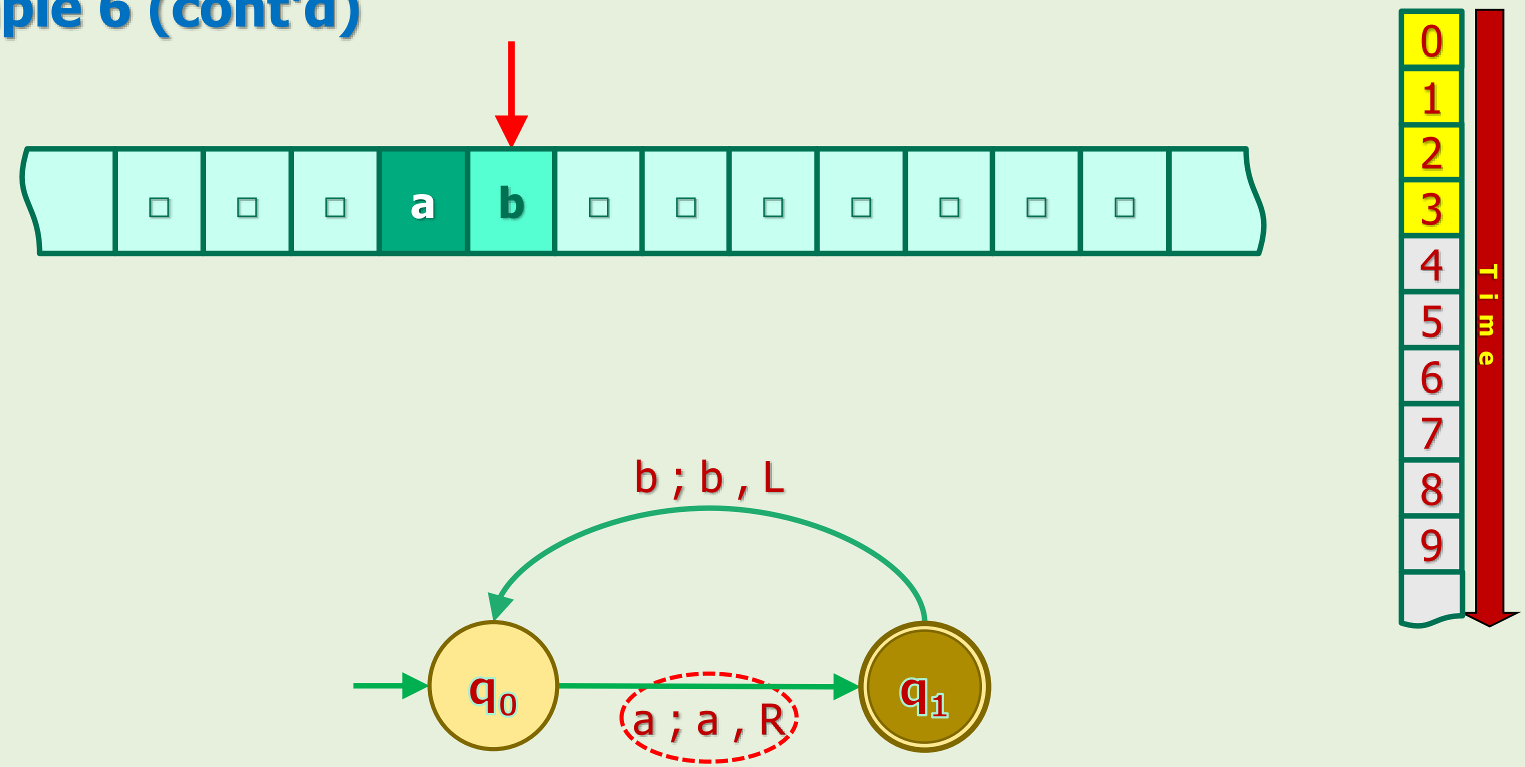






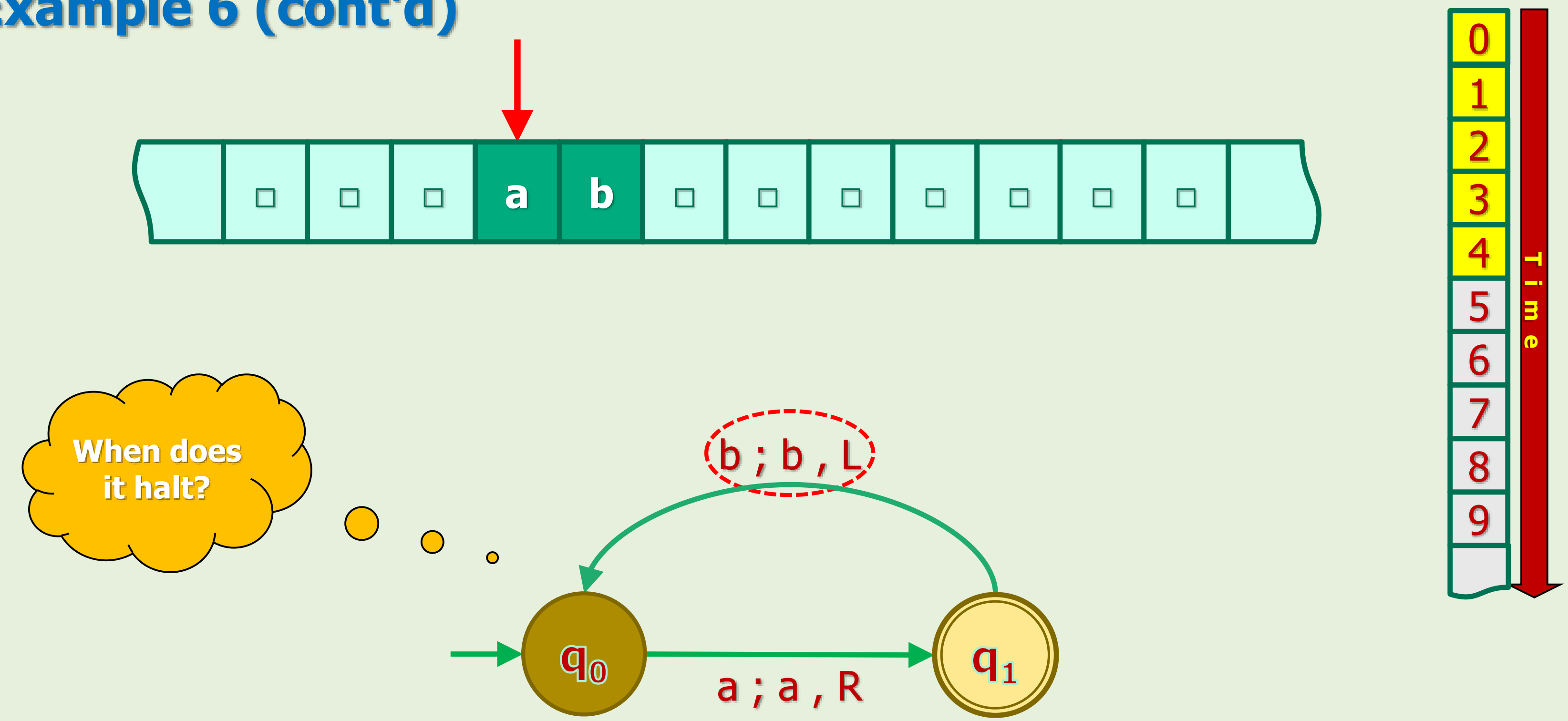
A Special Phenomenon in TMs

Example 6 (cont'd)



A Special Phenomenon in TMs

Example 6 (cont'd)



What was that phenomenon?

- The TM never halts.
- In other words, in some situations,

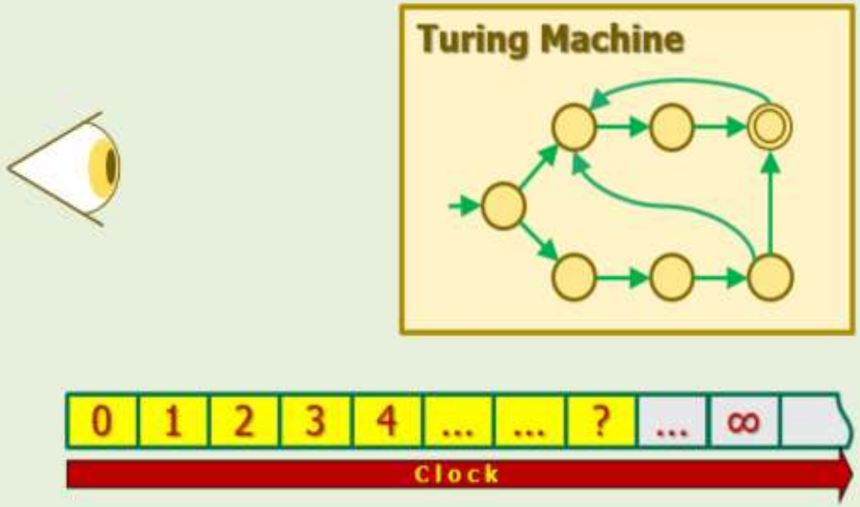
  A TM can fall into an "infinite loop".

- This phenomenon ...
- ... never happened in the previous "deterministic machines".

- What do you think is the reason?
  - This is the consequence of ...

    "having freedom of moving the read-write head to left or right".

A Vital ($1,000,000) Question



- An observer is watching a TM that is working for a long time, ...
- How can the observer figure out whether ...

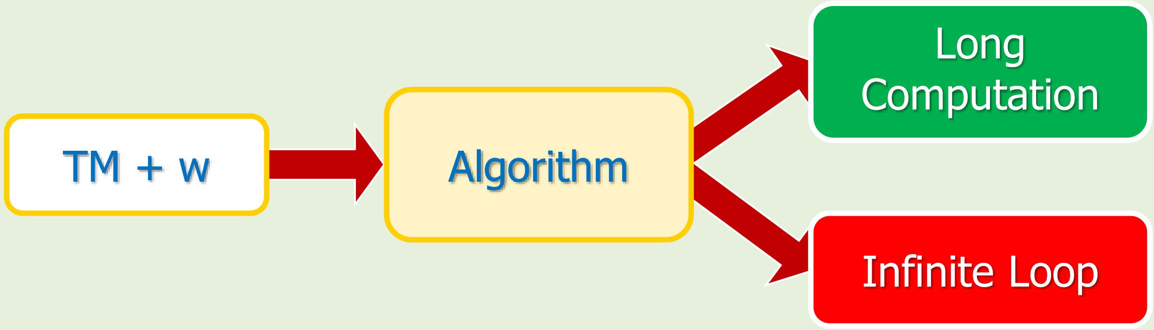
  ... it is in an infinite loop,

  OR

  ... it is in the middle of a very long computation?

A Vital ($1,000,000) Question

- In fact, we are looking for the following algorithm.



- There is no known algorithm to answer this!
  - Note that the algorithm must be able to solve the problem for any TM and any string w.
- We'll get back to this question later!

A Side Note About Rejecting String

- Note that based on the string rejection logic:

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

- If we can prove somehow that the machine falls into an infinite loop, then ...
- ... the string, that is being processed, is considered as rejected.
- ... because $\sim h \equiv$ True.

Nice Videos

1. Turing machines explained visually
   https://www.youtube.com/watch?v=-ZS_zFg4w5k

2. A Turing machine – Overview
   https://www.youtube.com/watch?v=E3keLeMwfHY

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