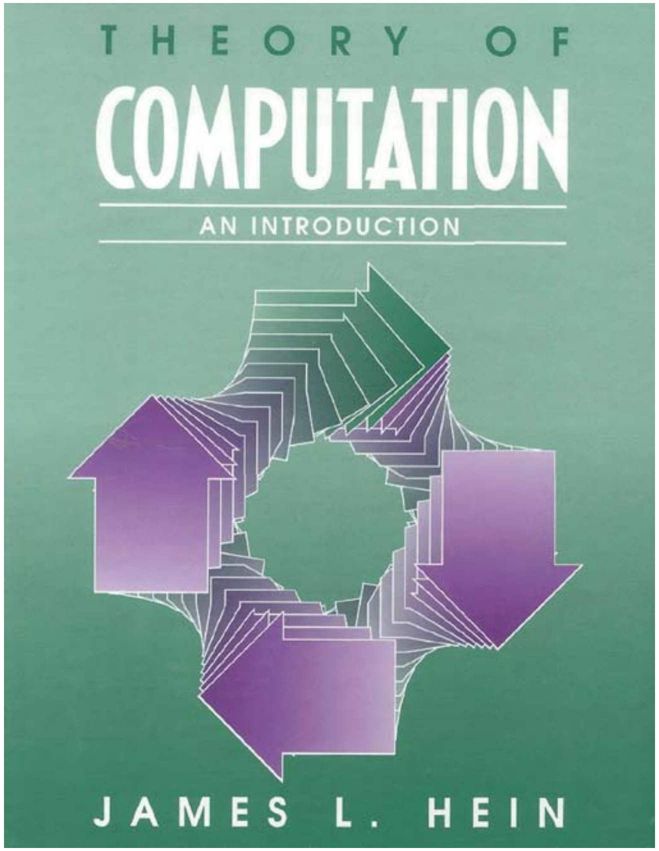
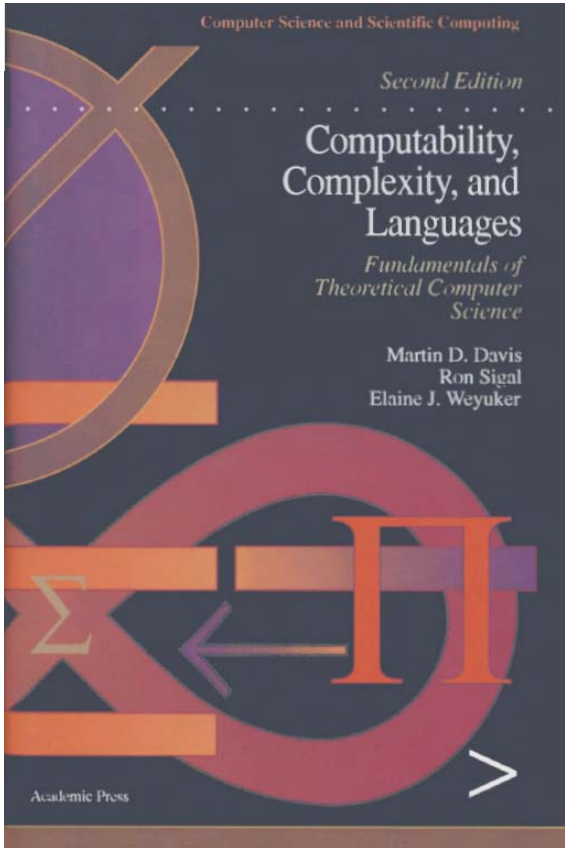
Theory of Computation

Lecture 02

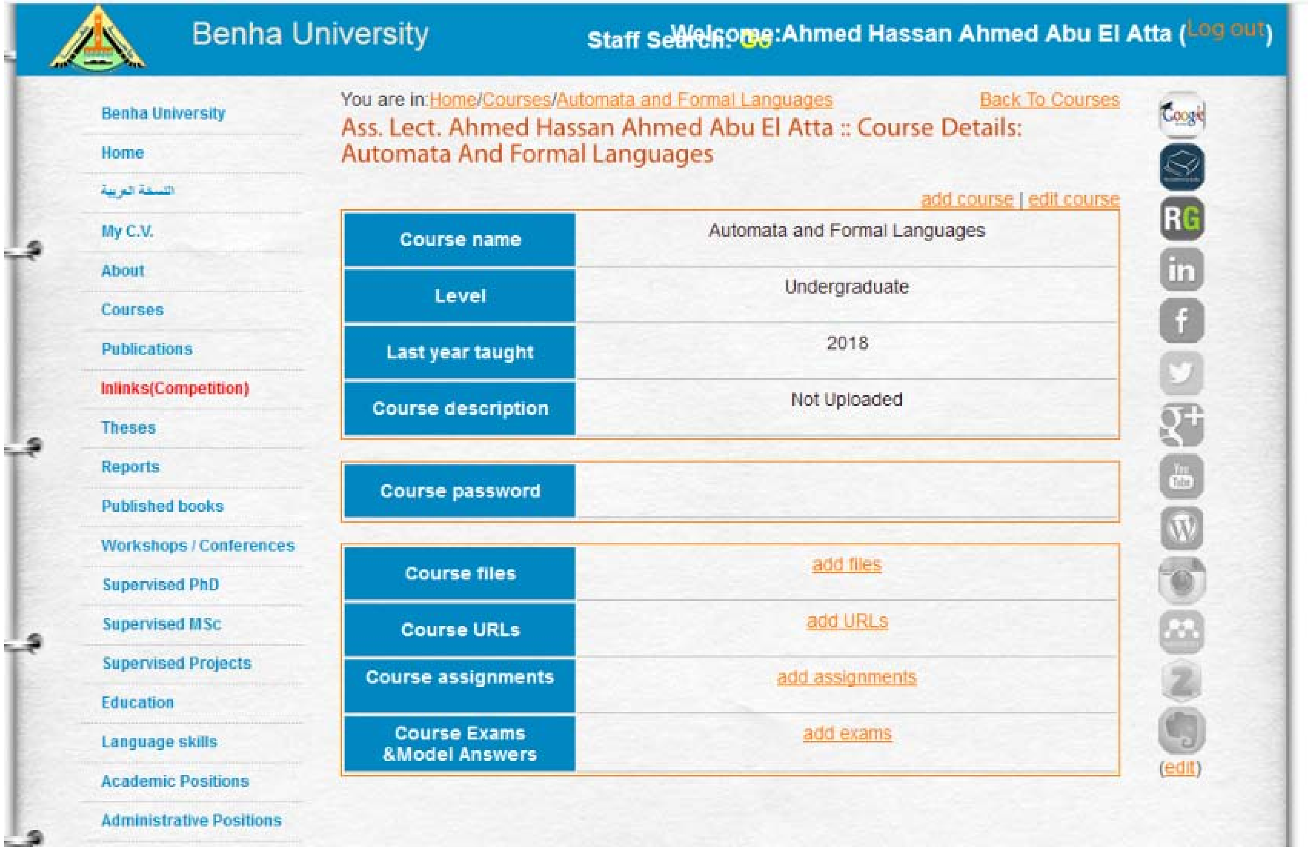
Books


THEORY OF
COMPUTATION
AN INTRODUCTION
JAMES L. HEIN


Computer Science and Scientific Computing
Second Edition
Computability, Complexity, and Languages
Fundamentals of Theoretical Computer Science
Martin D. Davis
Ron Sigal
Elaine J. Weyuker
Academic Press

PowerPoint

http://www.bu.edu.eg/staff/ahmedaboalatah14-courses/14767



Welcome:Ahmed Hassan Ahmed Abu El Atta (Log out)

You are in:Home/Courses/Automata and Formal Languages

Back To Courses

Ass. Lect. Ahmed Hassan Ahmed Abu El Atta :: Course Details:
Automata And Formal Languages

add course | edit course

Course name	Automata and Formal Languages
Level	Undergraduate
Last year taught	2018
Course description	Not Uploaded

Course password	

Course files	add files
Course URLs	add URLs
Course assignments	add assignments
Course Exams &Model Answers	add exams

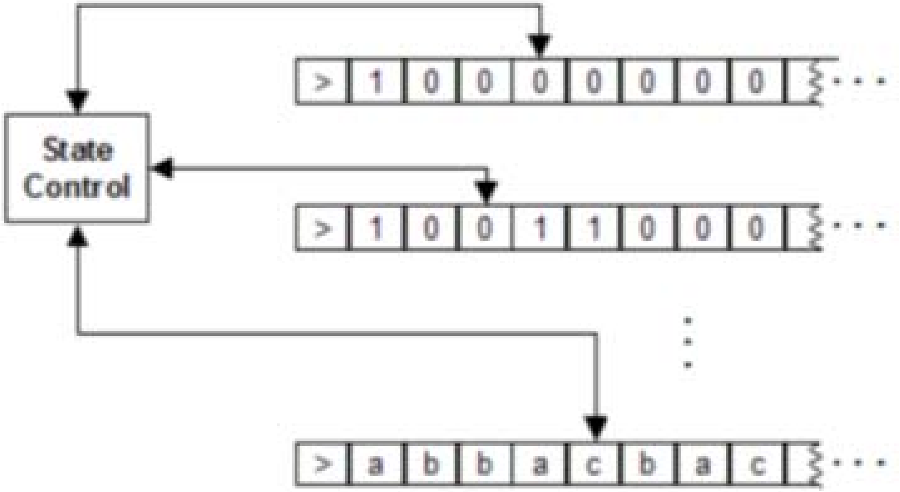
Turing Machine

Agenda

- Multi-Tape Turing machine
- EXAMPLE 1 *(Multiplying Natural Numbers)*
- Move one cell to right
- Turing machines are *equivalent* in power
- EXAMPLE 2

Multi-Tape Turing machine

- A multi-tape Turing machine has two or more tapes with corresponding tape heads.
- It's important to note that all these Turing machines are *equivalent* in power.
- In other words, any problem solved by one type of Turing machine can also be solved on any other type of Turing machine.



EXAMPLE 1 *(Multiplying Natural Numbers)*

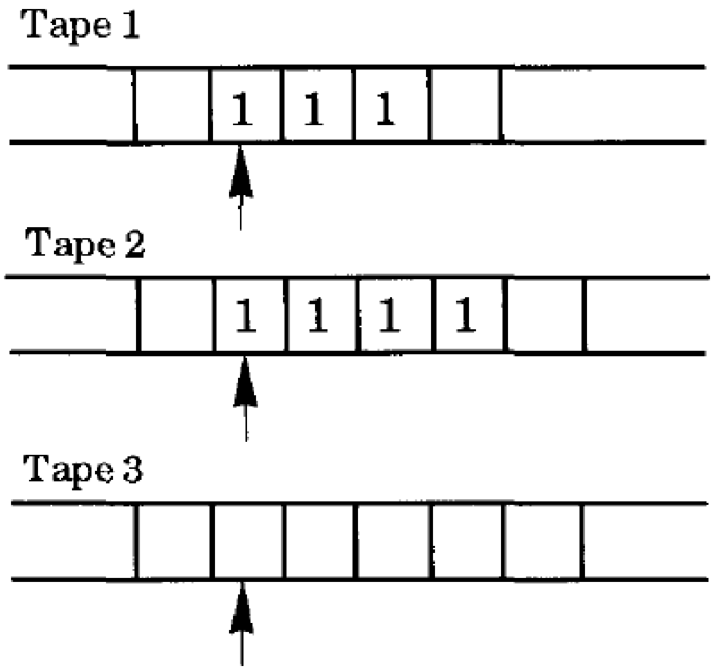
- Suppose we want to construct a Turing machine to multiply two natural numbers, each represented as a unary string of ones.

- We'll use a three-tape Turing machine, where the first two tapes hold the input numbers and the third tape will hold the answer.

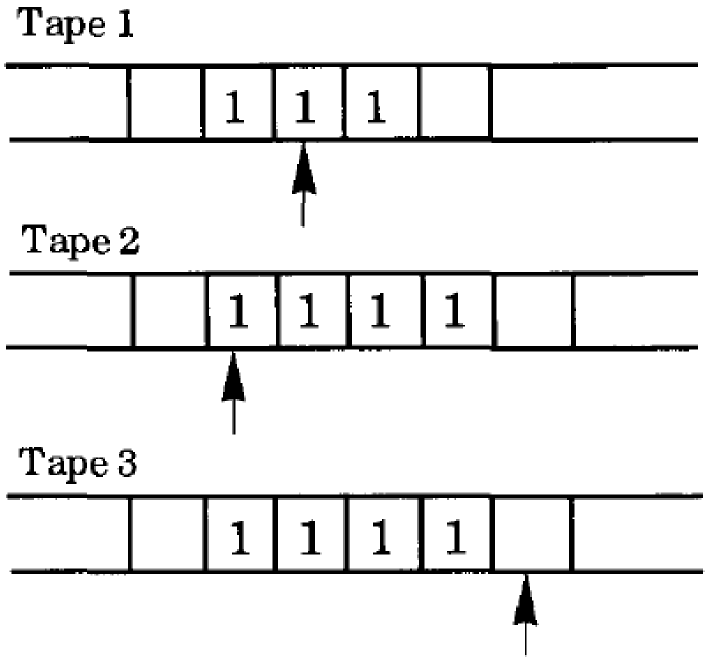
$$\langle i, \langle a, b, c\rangle, \langle x, y, z\rangle, \langle R, L, S\rangle, j\rangle.$$

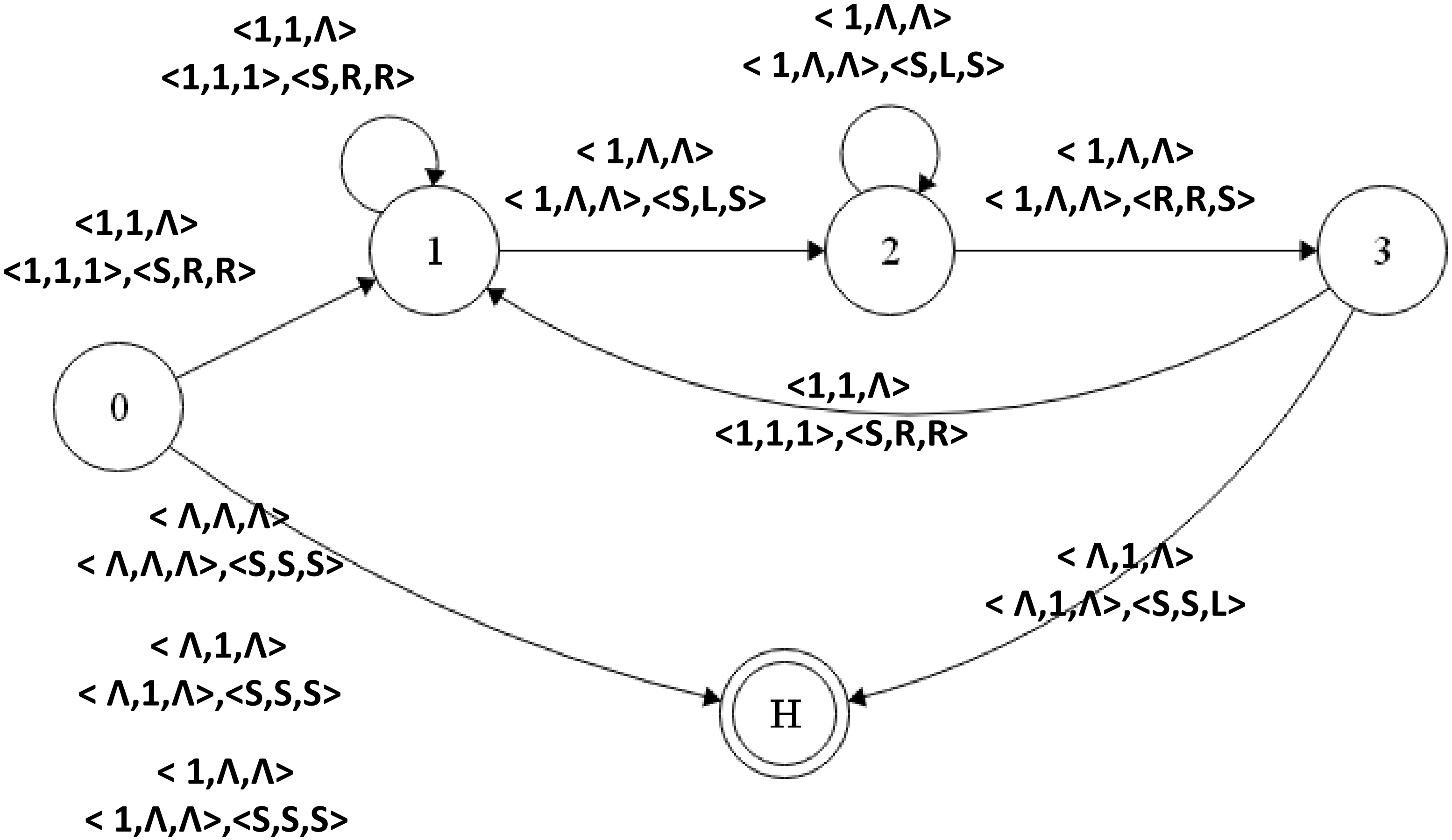
For example, the diagrams show the contents of the three tapes before the computation of 3.4 and before the start of the second of the three additions

TAPES BEFORE STARTING TO COMPUTE 3. 4



TAPES BEFORE STARTING THE SECOND ADDITION





<1,1,Λ>
<1,1,1>,<S,R,R>

< 1,Λ,Λ>
< 1,Λ,Λ>,<S,L,S>

< 1,Λ,Λ>
< 1,Λ,Λ>,<S,L,S>

< 1,Λ,Λ>
< 1,Λ,Λ>,<R,R,S>

<1,1,Λ>
<1,1,1>,<S,R,R>

<1,1,Λ>
<1,1,1>,<S,R,R>

< Λ,Λ,Λ>
< Λ,Λ,Λ>,<S,S,S>

< Λ,1,Λ>
< Λ,1,Λ>,<S,S,S>

< 1,Λ,Λ>
< 1,Λ,Λ>,<S,S,S>

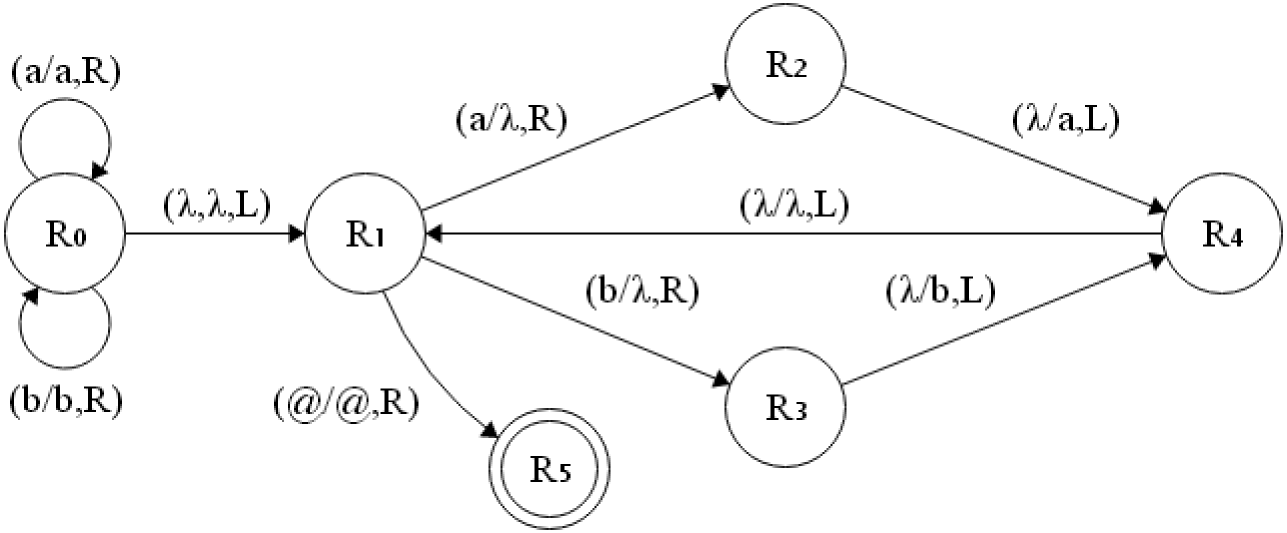
< Λ,1,Λ>
< Λ,1,Λ>,<S,S,L>

Move one cell to right

Construct a Turing machine to move an input string over (a, b} to the right one cell position. Assume that the tape head is at the left end of the input string if the string is nonempty. The rest of the tape cells are blank. The machine moves the entire string to the right one cell position, leaving all remaining tape cells blank.

		@	a	a	b				

		@	Λ	a	a	b			



Turing machines are *equivalent* in power

- Assume that we have a Turing machine **T** that has two tapes, each with a single tape head.

- We'll describe a new single-tape, single-head machine **M** that will start with its tape containing the two nonblank portions taken from the tapes of **T**, separated by a new tape symbol

@

Turing machines are *equivalent* in power

Whenever **T** executes an instruction (which is actually a pair of instructions, one for each tape), **M** simulates the action by performing two corresponding instructions, one instruction for the left side of**@** and the other instruction for the right side of**@**.

Turing machines are *equivalent* in power

Since **M** has only one tape head, it must chase back and forth across **@** to execute instructions.

So it needs to keep track of the positions of the two tape heads that it is simulating.

One way to do this is to place a position marker **.** in every other tape cell.

$$\ldots\ \Lambda \cdot \# \cdot a \cdot b\ \hat{}\ c \cdot @ \cdot x \cdot y\ \hat{}\ z \cdot w \cdot \# \cdot \Lambda\ \ldots.$$

Turing machines are *equivalent* in power

To indicate a current cell, we'll write the symbol

^

in place of**.** in the adjacent cell to the right of the current cell for the left tape and to the adjacent cell to the left of the current cell for the right tape.

For example, if the two tapes of **T** contain the strings **abc** and **xyzw**, *with tape* heads pointing at **b** and **z**, then the tape for **M** has the following form, where the symbol **#** marks the left end and the right end of the relevant portions of the tape:

... Λ · # · *a* · *b* ^ *c* · @ · *x* · *y* ^ *z* · *w* · # · Λ ....

Turing machines are *equivalent* in power

<i,<a,x>,<0,1>,<R,R>,j>

		^	x	y	z				

		^	a	b	c				

	.	#	.	a	^	b	.	c	.	@	^	x	.	y		z	.	#	.	

<i, a, 0, R, k>

<k, ^, ., R, k>

<k, ., ^, R, h>

<k, anyElse, anyElse, R, k>

<h, ^, ., R, q>

<h, AnyElse, AnyElse, R, h>

<q, x, 1, R, p>

<p, .,^,L,g>

<g, ^, ^, L, j>

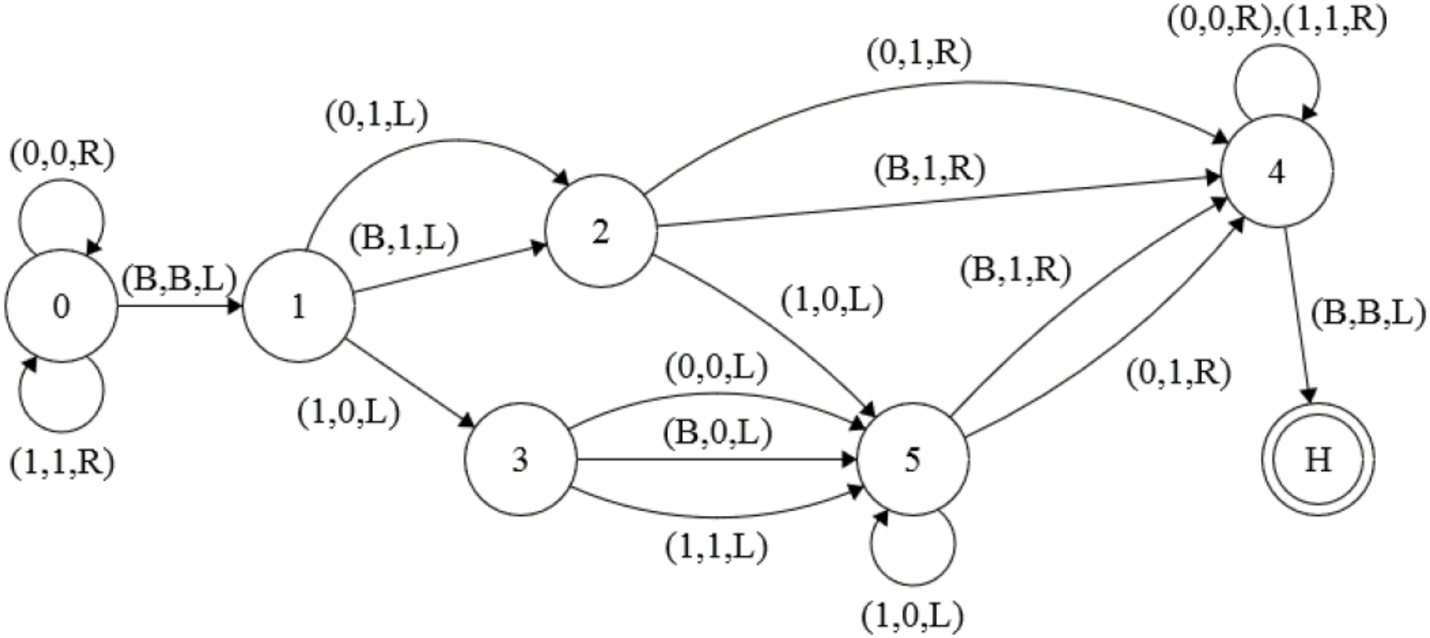
<g, AnyElse, AnyElse, L, g>

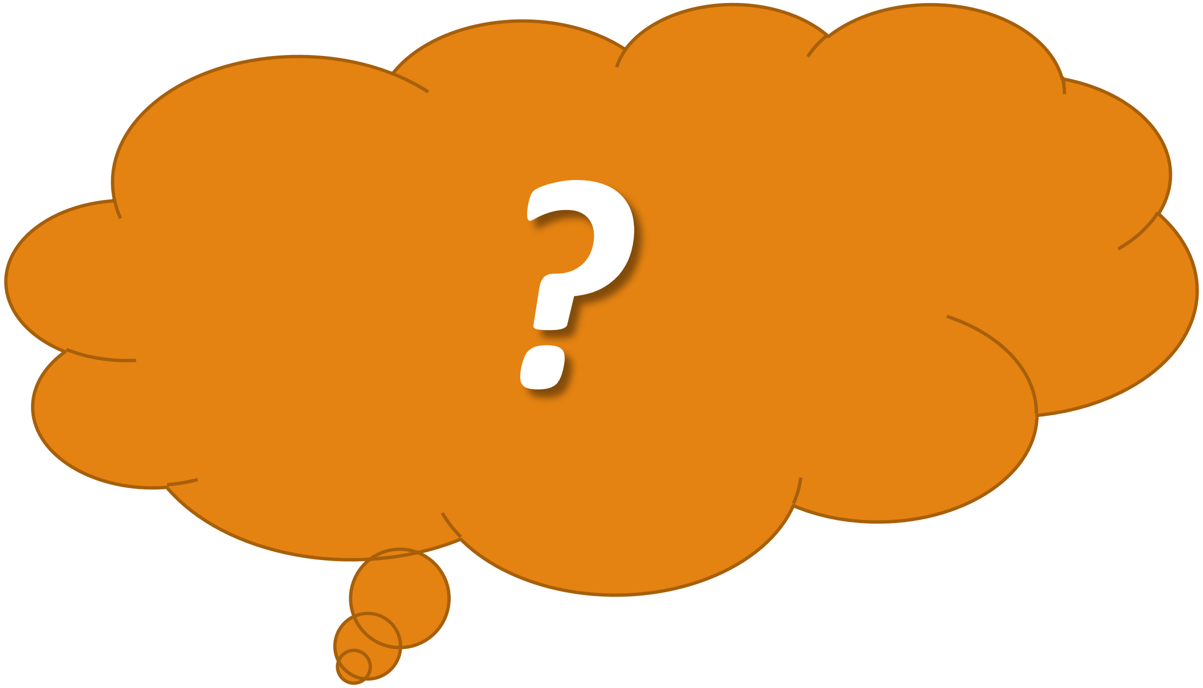
Example 2

Construct Turing machine to add 3 to a natural number represented as a binary string.

Example 2

Construct Turing machine to add 3 to a natural number represented as a binary string.





?