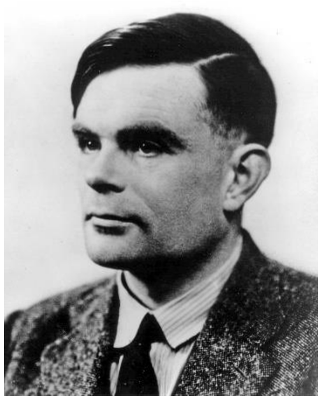
3. Introduction to computer science

3.1 Models for computation



Alan Turing
1912-1954

Turing machines

Hilbert: Can all problems in mathematics be solved by an algorithm? (*entscheidungsproblem*)

Turing & Church: What is a mathematical definition of an algorithm?

I) Invent a theoretical algortithm ”machine” – *Turing machine*

II) Postulate/conjecture/show:

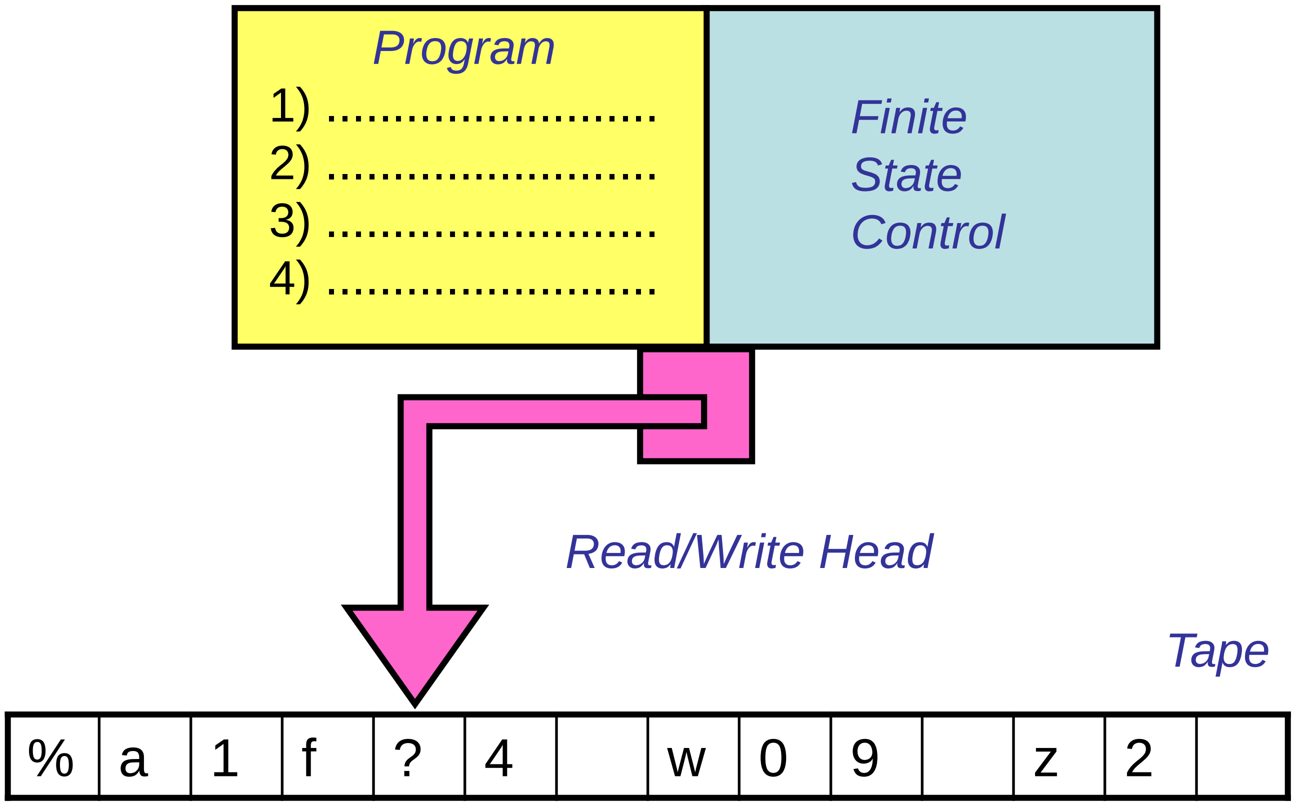
> *Church-Turing thesis: Functions computable on a Turing machine are functions that we regard as functions computable by an algorithm.*

III) Answer to Hilbert (reformulating Gödel): Not all mathematical problems are computable by an algorithm. Example: Halting problem.

IV) Laying foundation to modern theory of computer science.

Turing machine

Four elementary building blocks

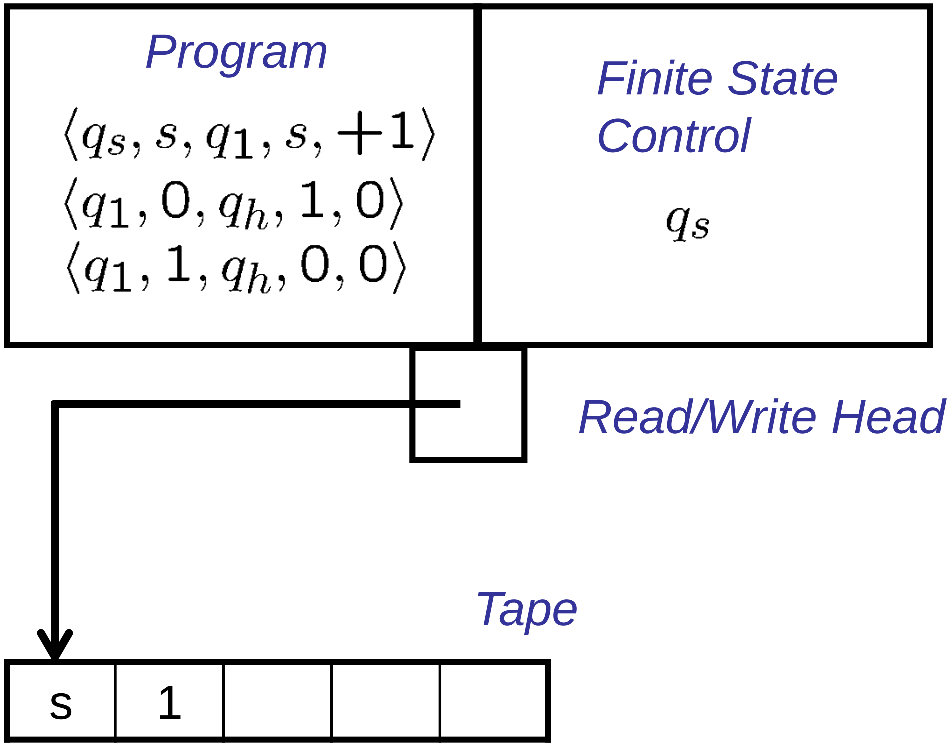


Internal states of finite state control: $q_s, q_1, \ldots, q_m, q_h$

Example: NOT operation

- Four element language: $s, 0, 1, b$ $\quad$ b is blank
- Internal states: q_s, q_1, q_h

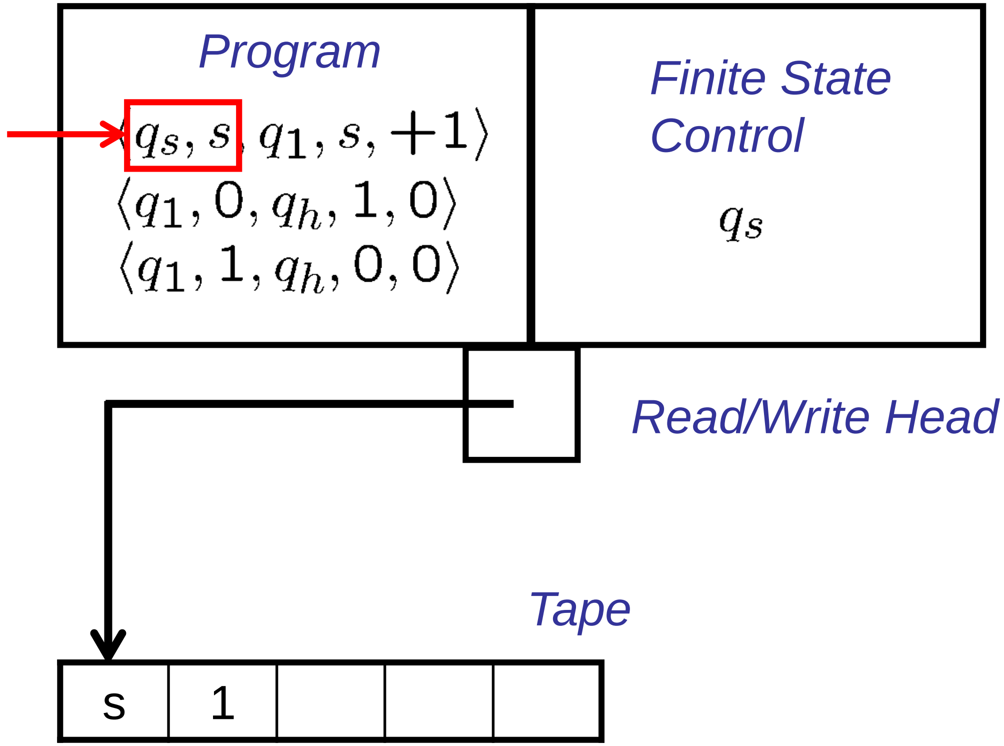
Starting with: Tape written, finite state control in q_s



Example: NOT operation

- Four element language: $s, 0, 1, b$ $\quad b$ is blank
- Internal states: q_s, q_1, q_h

Starting with: Tape written, finite state control in q_s

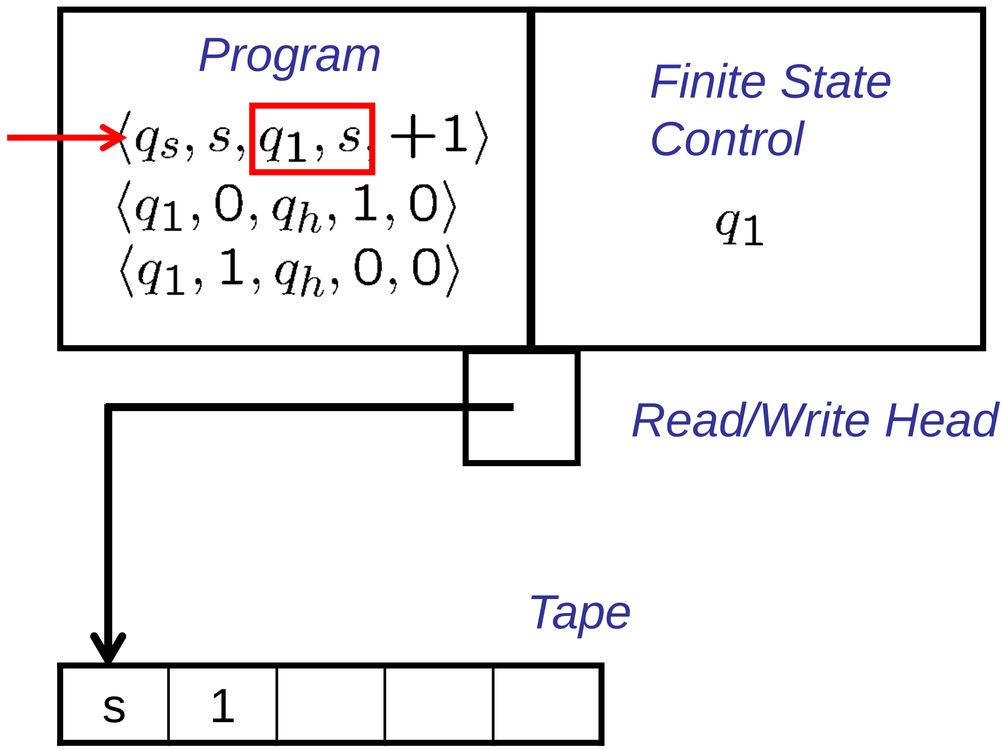


Execute program: 1) Go through lines top to bottom.
2) Find match to FSC + tape (*if not found, stop run*).

Example: NOT operation

- Four element language: $s, 0, 1, b$ $\quad\quad$ b is blank
- Internal states: q_s, q_1, q_h

Starting with: Tape written, finite state control in q_s

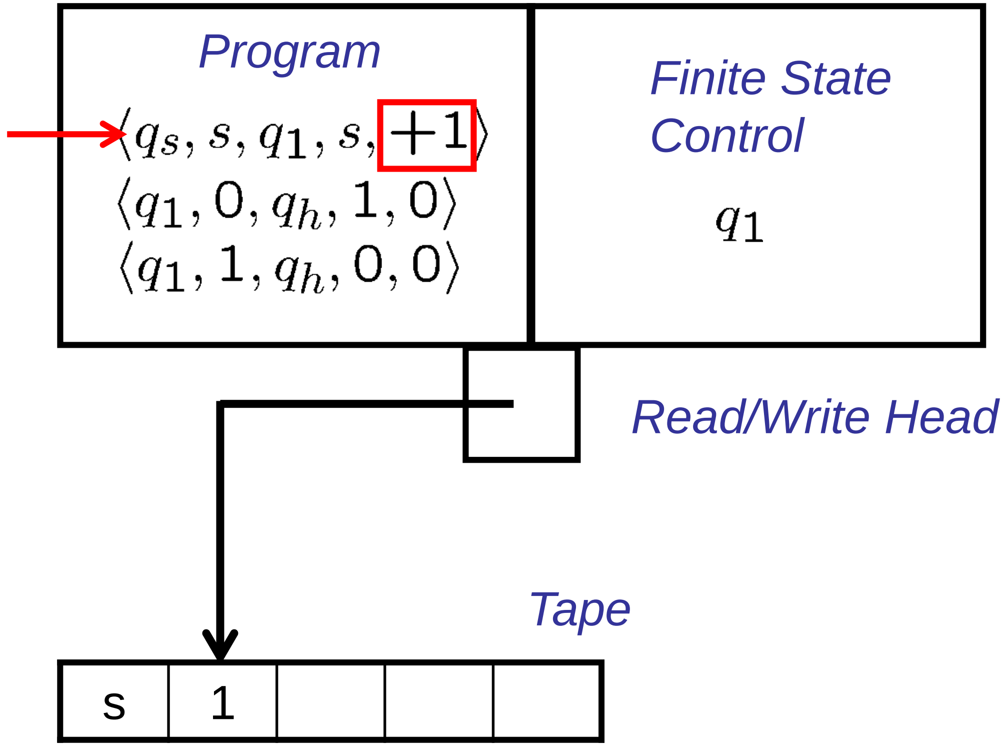


Execute program: 3) Change FSC + tape.

Example: NOT operation

- Four element language: $s, 0, 1, b$ $\quad$ b is blank
- Internal states: q_s, q_1, q_h

Starting with: Tape written, finite state control in q_s

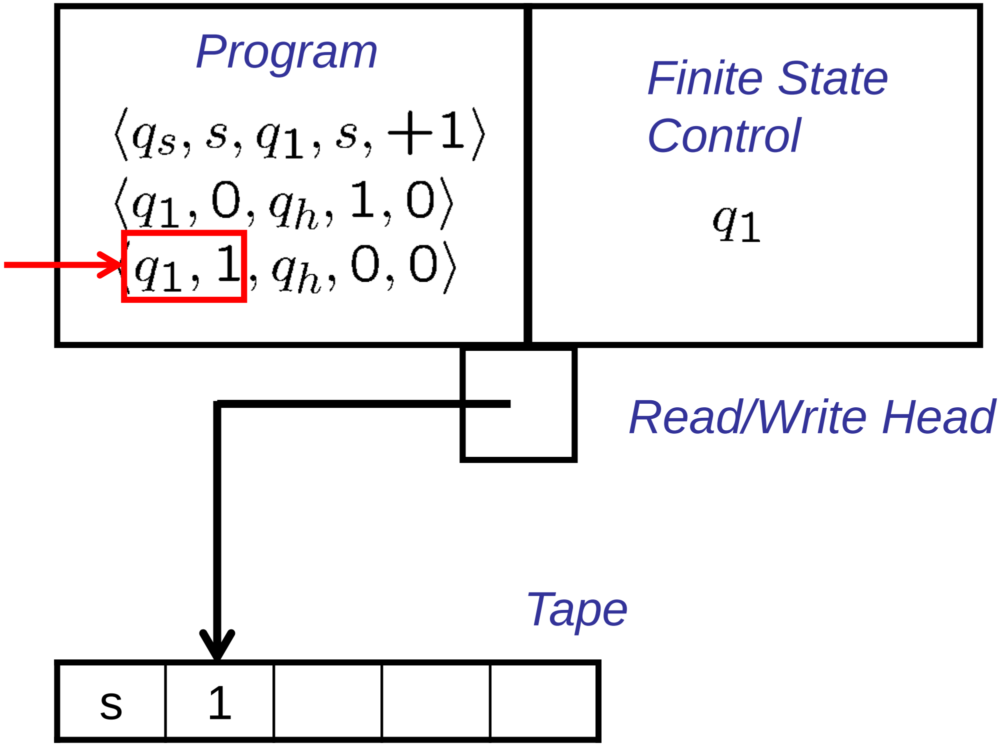


Execute program: 3) Change FSC + tape.
4) Move head.

Example: NOT operation

- Four element language: $s, 0, 1, b$ $\quad b$ is blank
- Internal states: q_s, q_1, q_h

Starting with: Tape written, finite state control in q_s

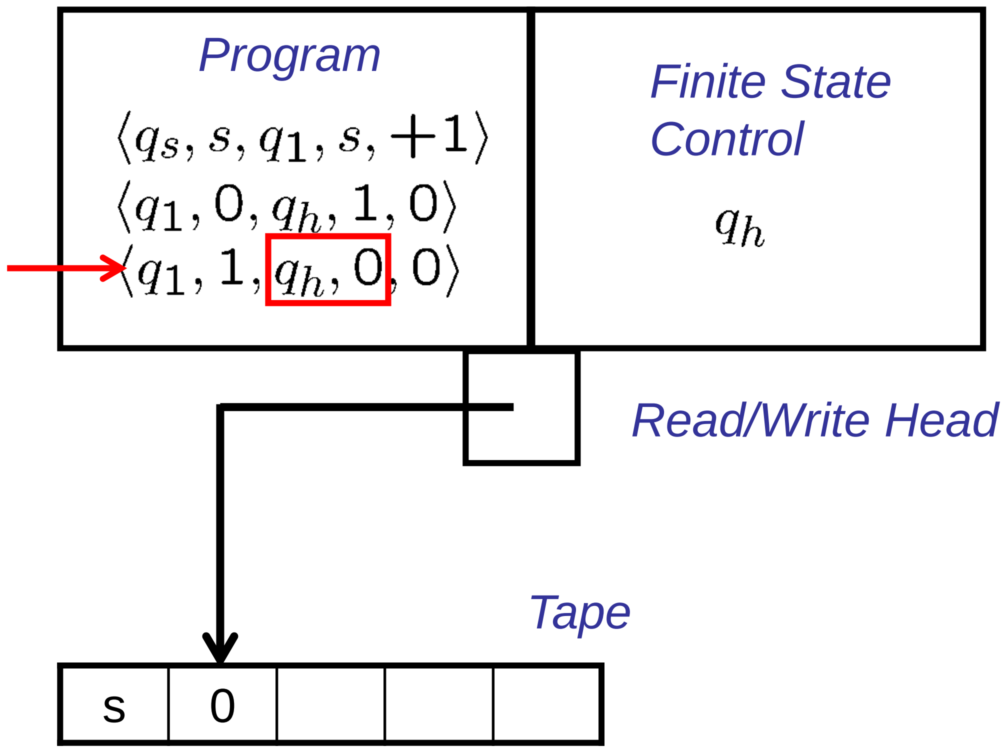


Execute program: 1) Go through lines top to bottom.
2) Find match to FSC + tape.

Example: NOT operation

- Four element language: $s, 0, 1, b$ $\qquad$ b is blank
- Internal states: q_s, q_1, q_h

Starting with: Tape written, finite state control in q_s

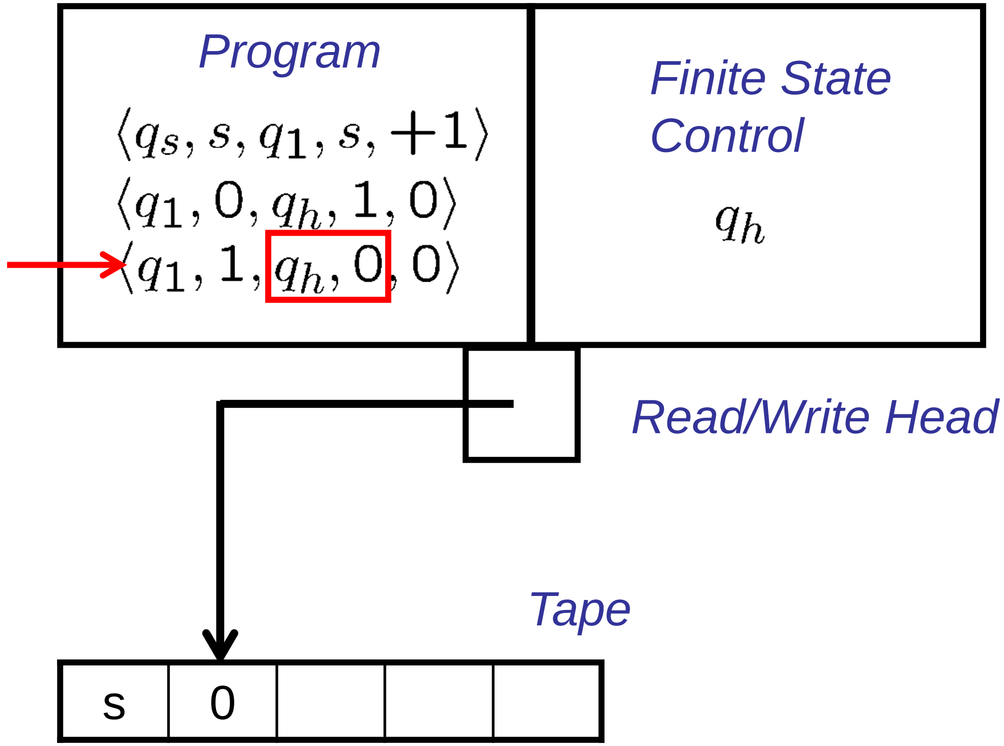


Execute program: 3) Change FSC + tape.
4) Move head.

Example: NOT operation

- Four element language: $s, 0, 1, b$ b is blank
- Internal states: q_s, q_1, q_h

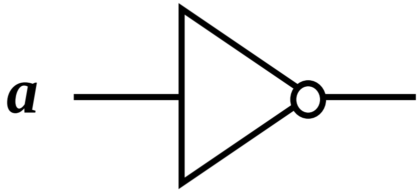
Starting with: Tape written, finite state control in q_s



FSC is in q_h : run ends. Output on tape.

<u>Basic circuit elements</u>

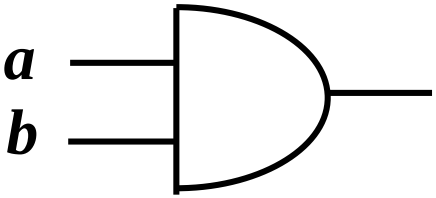
NOT



c = NOT a

a	c
0	1
1	0

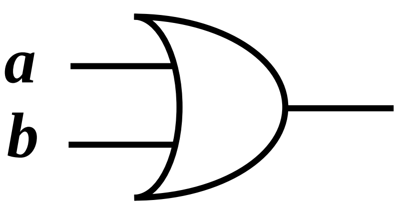
AND



$c = a$ AND b

a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

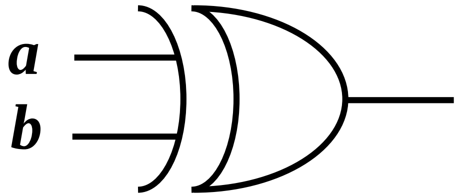
OR



$c = a$ OR b

a	b	c
0	0	0
0	1	1
1	0	1
1	1	1

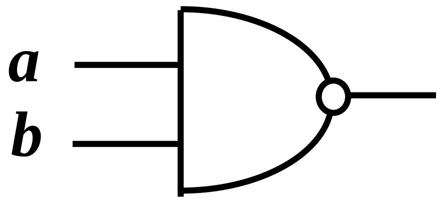
XOR



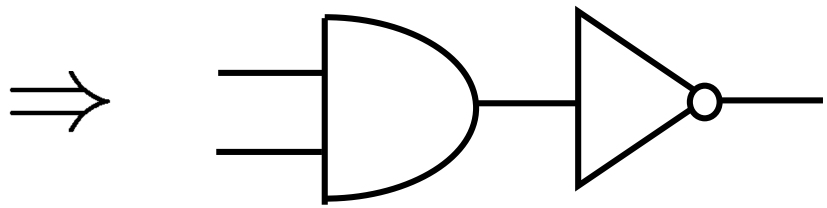
$c = a \text{ XOR } b$

a	*b*	*c*
0	0	0
0	1	1
1	0	1
1	1	0

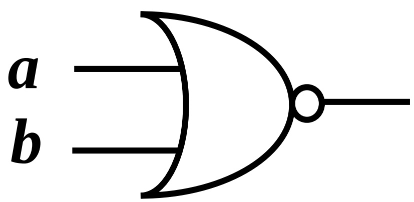
NAND



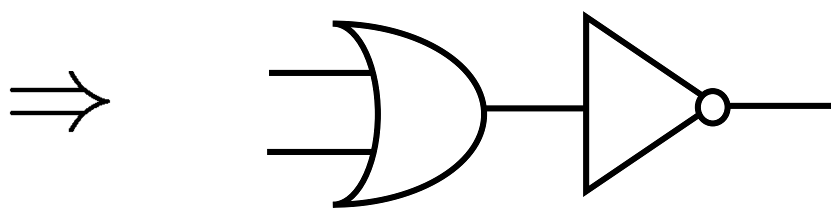
$c = a \text{ NAND } b$



NOR

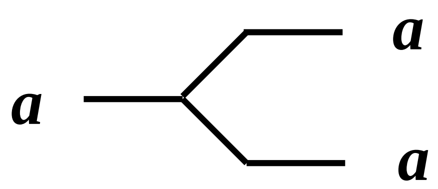


$c = a \text{ NOR } b$

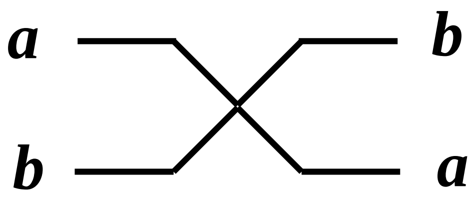


Additional operations

FANOUT



CROSSOVER



Universal computation

Any function $f : \{0,1\}^n \rightarrow \{0,1\}^m$ can be calculated using

1) Wires
2) Ancilla bits
3) FANOUT operations
4) CROSSOVER operations
5) AND, XOR and NOT gates

Exercise 3.8: Show that 1), 2), 3), 4) and NAND are also universal.