Theoretical Computer Science

Zdeněk Sawa

Department of Computer Science, FEI,
Technical University of Ostrava
17. listopadu 2172/15, Ostrava-Poruba 708 00
Czech republic

September 25, 2024

Lecturer

Name: doc. Ing. Zdeněk Sawa, Ph.D.

E-mail: zdenek.sawa@vsb.cz

Room: EA413

Web: http://www.cs.vsb.cz/sawa/tcs

On these pages you will find:

- Information about the course
- Slides from lectures
- Exercises for tutorials
- Recent news for the course
- A link to a page with animations

In MS Teams (team "*Theoretical Computer Science 2024/25*"):

- some other texts
- materials for presentations

Requirements

- **Credit** (35 points):
  - **Presentation** (15 points) — it is necessary to obtain at least 5 points

    a correction is possible for at most 10 points, for the correction, at least 1 point must be obtained
  - **Written test** (20 points) — it is necessary to obtain at least 10 points

    there will be a possible correction test — for at most 17 points, required minimum is still 10 points

  It is necessary to obtain from the presentation and the written test in total at least 15 points.

- **Exam** (65 points):
  - written form
  - it is necessary to obtain at least 25 points

Theoretical Computer Science

Theoretical computer science — a scientific field on the border between computer science and mathematics

- investigation of general questions concerning algorithms and computations
- study of different kinds of formalisms for description of algorithms
- study of different approaches for description of syntax and semantics of formal languages (mainly programming languages)
- a mathematical approach to analysis and solution of problems (proofs of general mathematical propositions concerning algorithms)

Theoretical Computer Science

Examples of some typical questions studied in theoretical computer science:

- Is it possible to solve the given problem using some algorithm?
- If the given problem can be solved by an algorithm, what is the computational complexity of this algorithm?
- Is there an efficient algorithm solving the given problem?
- How to check that a given algorithm is really a correct solution of the given problem?
- What kinds instructions are sufficient for a given machine to perform a given algorithm?

Algorithms and Problems

Algorithm — mechanical procedure that computes something (it can be executed by a computer)

Algorithms are used for solving **problems**.

An example of an algorithmic problem:

Input: Natural numbers a and b.

Output: Natural number c such that $c = a + b$.

Algorithms and Problems

Algorithm — mechanical procedure that computes something (it can be executed by a computer)

Algorithms are used for solving **problems**.

An example of an algorithmic problem:

Input: Natural numbers a and b.

Output: Natural number c such that $c = a + b$.

A particular input of a problem is called an **instance** of the problem.

Example: An example of an instance of the problem given above is a pair of numbers 728 and 34.

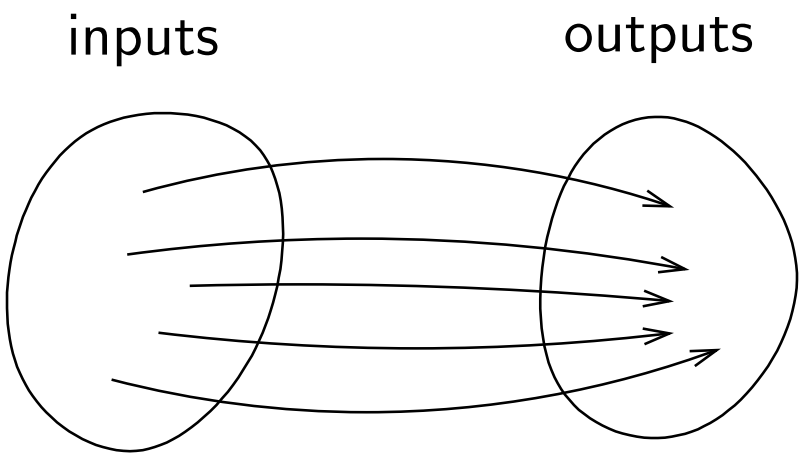
The corresponding output for this instance is number 762.

Problems

Problem

When specifying a **problem** we must determine:

- what is the set of possible inputs
- what is the set of possible outputs
- what is the relationship between inputs and outputs



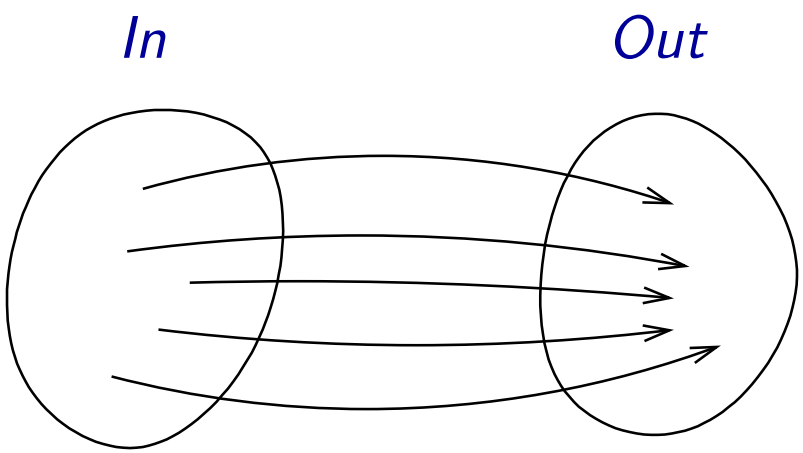
Problems

Problem

So formally, a **problem** can be defined as a tuple (In, Out, R), where:

- In is the set of possible inputs
- Out is the set of possible outputs
- $R \subseteq In \times Out$ is a relation assigning corresponding outputs to each input. This relation must satisfy

$$\forall x \in In : \exists y \in Out : R(x, y).$$



Examples of Problems

"Sorting" Problem

Input: A sequence of elements $a_1, a_2, \ldots, a_n$.

Output: Elements of the sequence $a_1, a_2, \ldots, a_n$ ordered from the least to the greatest.

An example of an instance of the problem and its corresponding output:

- Input: $8, 13, 3, 10, 1, 4$
- Output: $1, 3, 4, 8, 10, 13$

Remark: For one problem there can be many diffent algorithms that correctly solve the problem, for example for the "sorting" problem:

- insertion sort, selection sort, bubble sort, merge sort, quicksort, heapsort, ...

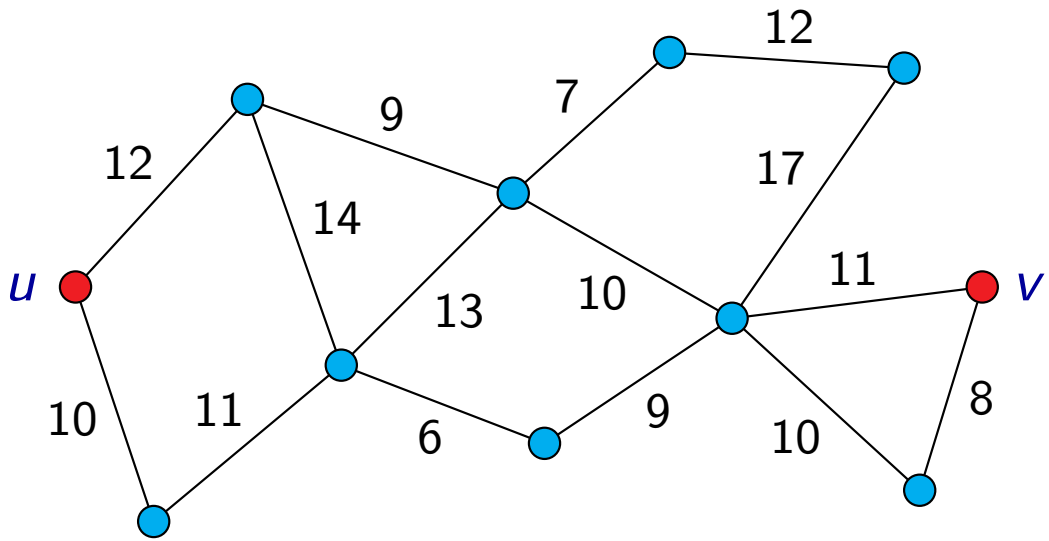
An example of an algorithmic problem

Problem "Finding the shortest path in an (undirected) graph"

Input: An undirected graph $G = (V, E)$ with edges labelled with numbers, and a pair of nodes $u, v \in V$.

Output: The shortest path from node u to node v.
(Or information that there is no such path.)

Example:



Algorithms and Problems

An algorithm **solves** a given problem if:

- For each input, the computation of the algorithm halts after a finite number of steps.
- For each input, the algorithm produces a correct output.

Correctness of an algorithm — verifying that the algorithm really solves the given problem

Computational complexity of an algorithm:

- **time complexity** — how the running time of the algorithm depends on the size of input data
- **space complexity** — how the amount of memory used by the algorithm depends on the size of input data

Formal Languages

An area of theoretical computer science dealing with questions concerning **syntax**.

- **Alphabet** — a set of **symbols** (or **letters**)
- **Word** — a sequences of symbols from some alphabet
- **Language** — a set of words

Words and languages appear in computer science on many levels:

- Representation of input and output data
- Representation of programs
- Manipulation with character strings or files
- ...

Formal Languages

Formálně:

- **Alphabet** — a nonempty finite set of **symbols**

  Example: $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$

- **Word** — a finite sequence of symbols from the given alphabet

  Example: $\mathtt{cabcbba}$

The set of all words of alphabet Σ is denoted with Σ^*.

Remark: A special case of a word is the empty word (e.g., the word of length 0).
The empty word is denoted ε.

Formal Languages

Definition

A **(formal) language** L over an alphabet Σ is a subset of Σ^*, i.e., $L \subseteq \Sigma^*$.

Example: Let us assume that $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$:

- Language $L_1 = \{\, \mathtt{aab},\ \mathtt{bcca},\ \mathtt{aaaaa} \,\}$
- Language
  $L_2 = \{\, w \in \Sigma^* \mid \text{the number of occurrences of } \mathtt{b} \text{ in } w \text{ is even} \,\}$

Encoding of Input and Output

Inputs and outputs of an algorithm could be encoded as words over some alphabet Σ, i.e., $In = Out = \Sigma^*$.

Some other object (numbers, sequences of numbers, graphs, . . . ,) then can be written (encoded) as words over this alphabet.

Example: For example, for problem "Sorting" we can take alphabet $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ,\}$.

An example of input data (as a word over alphabet Σ):

`826,13,3901,128,562`

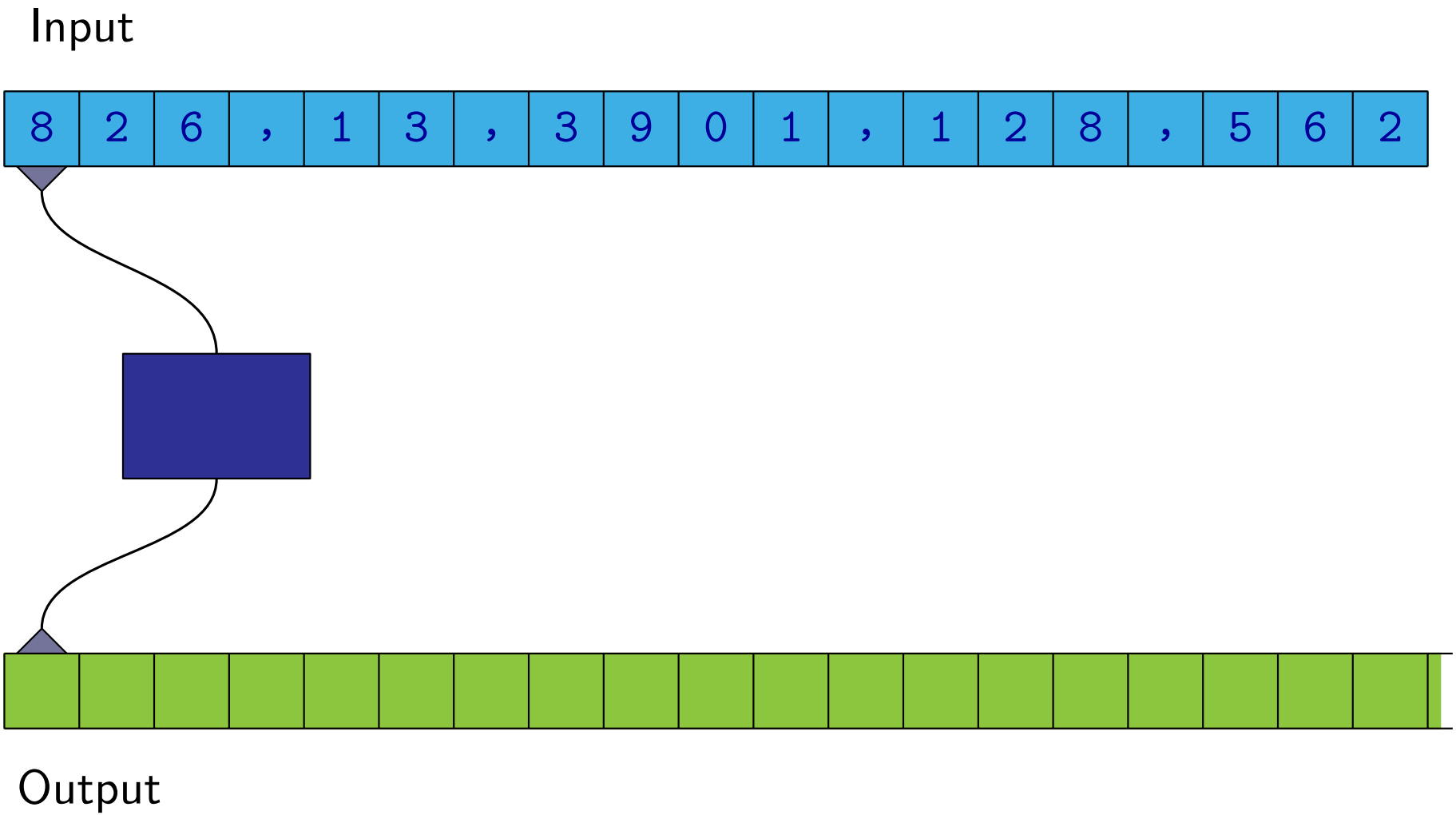
and the corresponding output data (as a word over alphabet Σ)

`13,128,562,826,3901`

Remark: It is often the case that only some words over the given alphabet represent valid input or output.

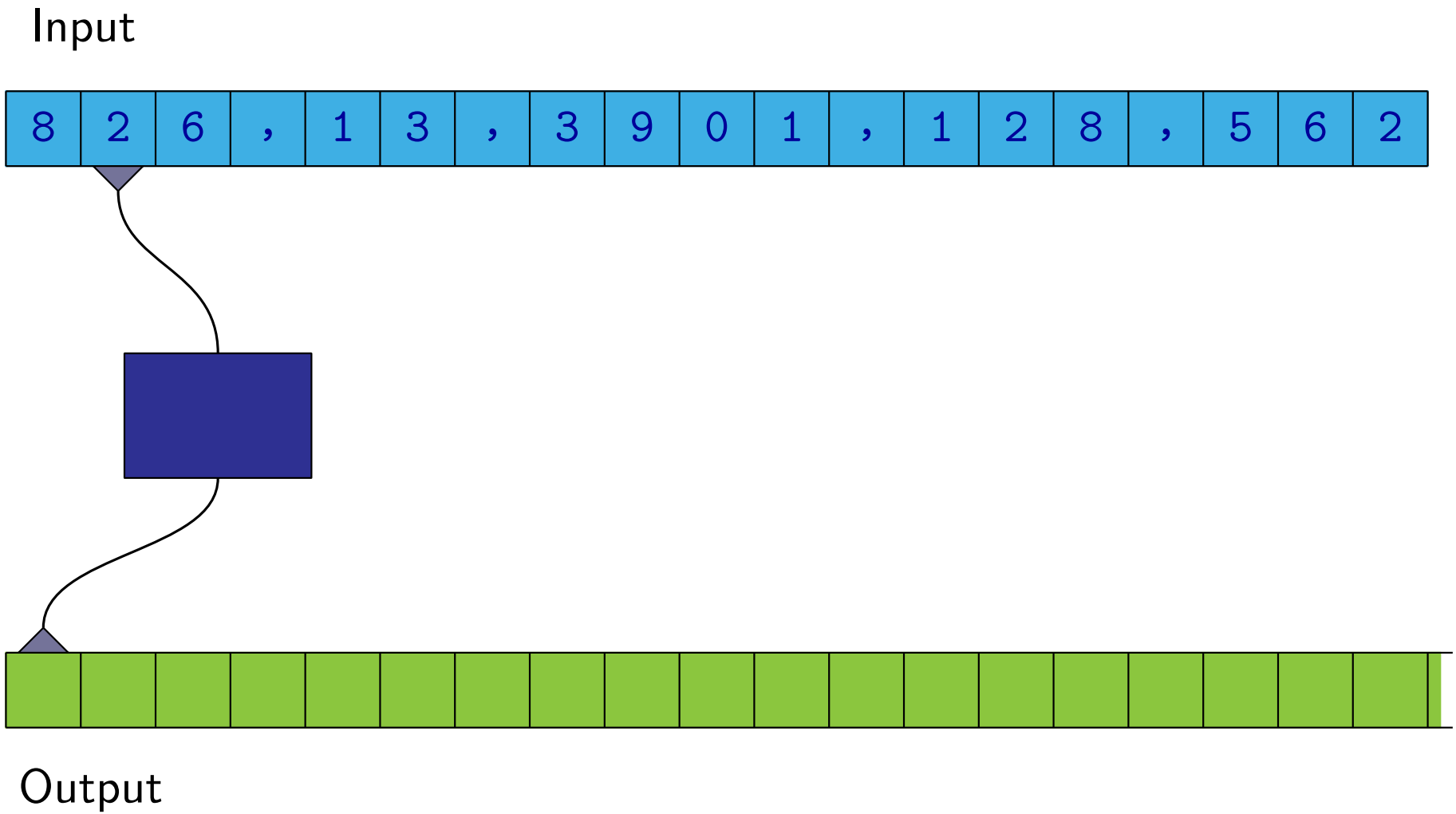
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



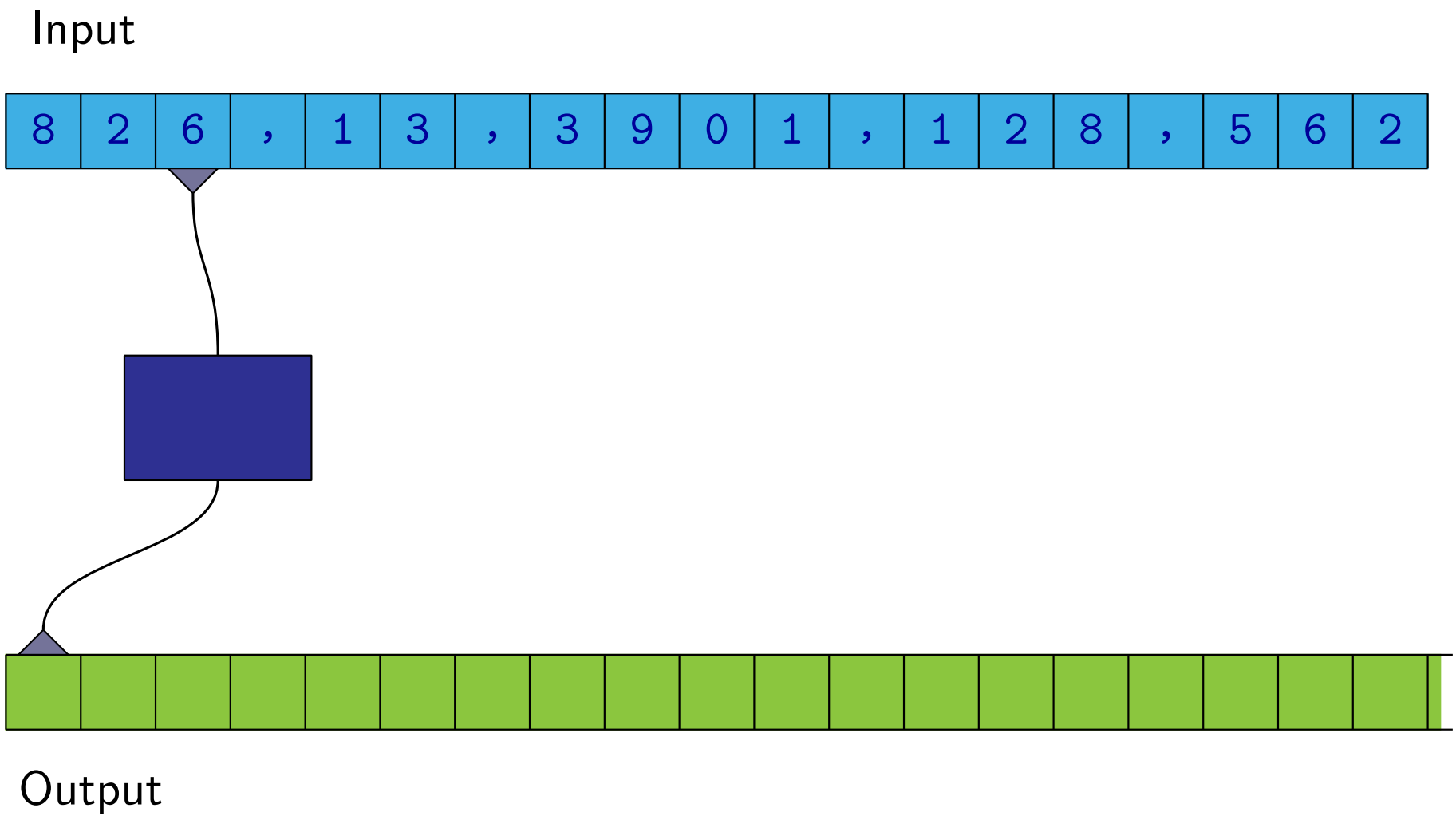
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



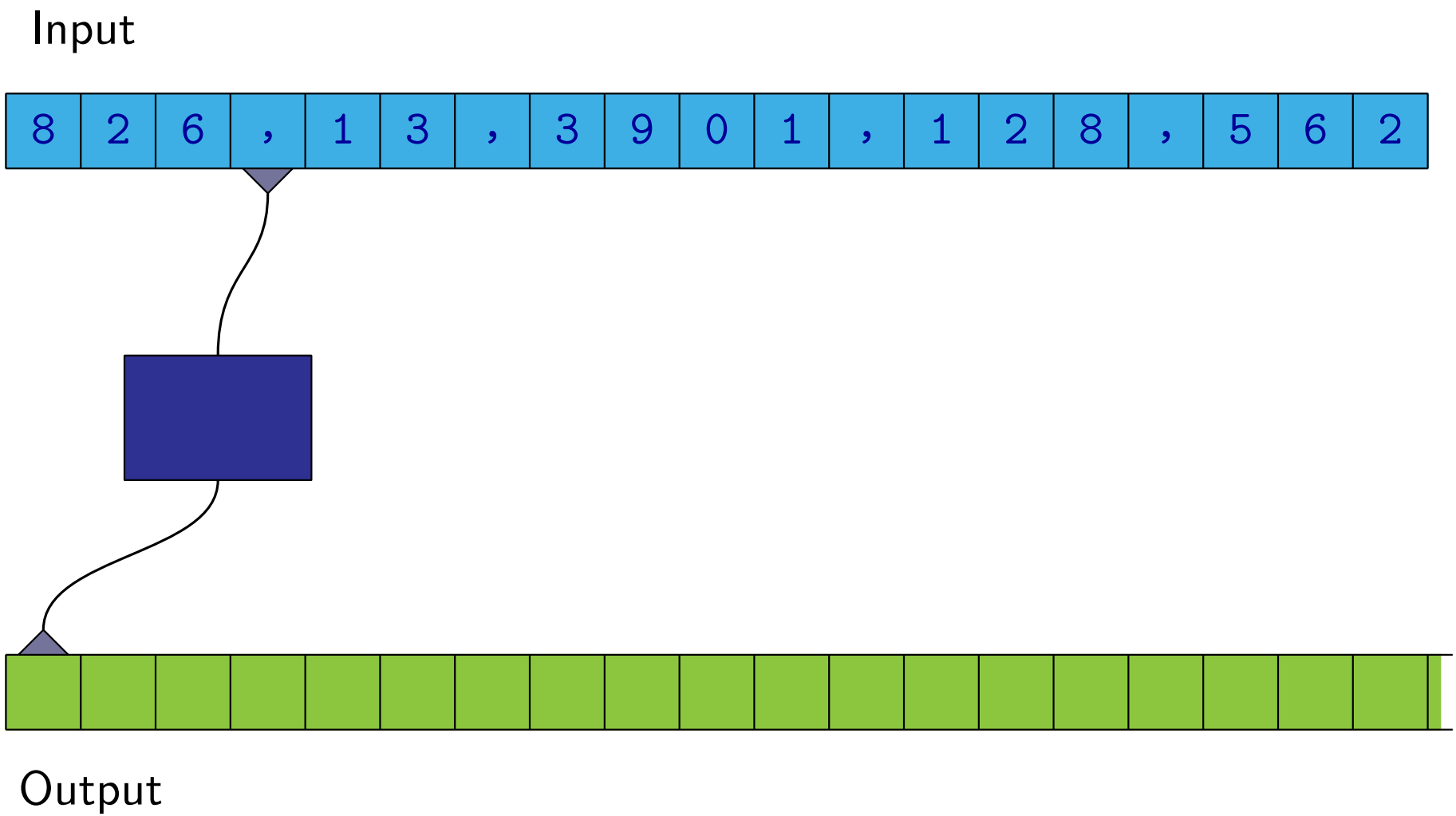
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



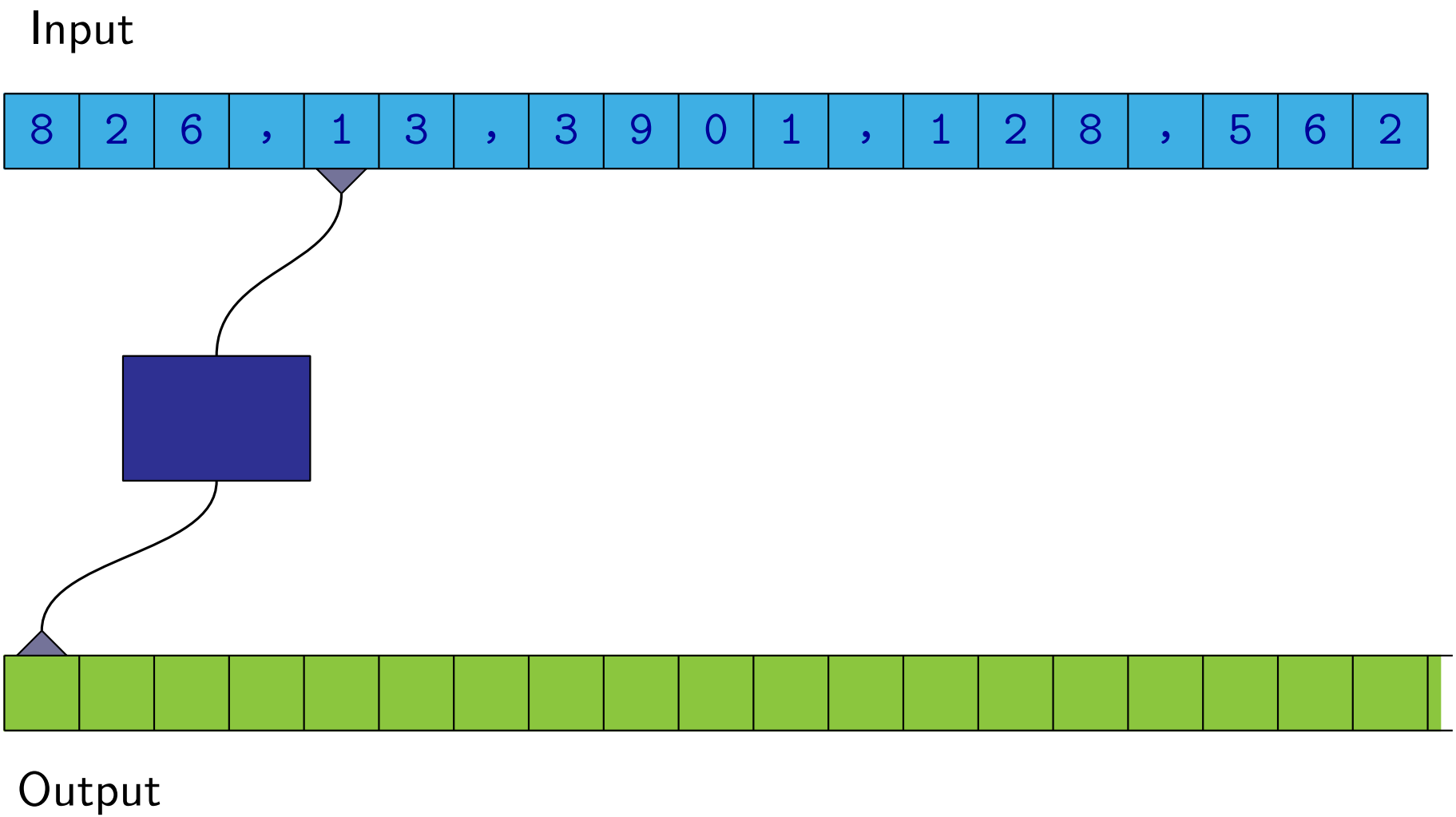
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



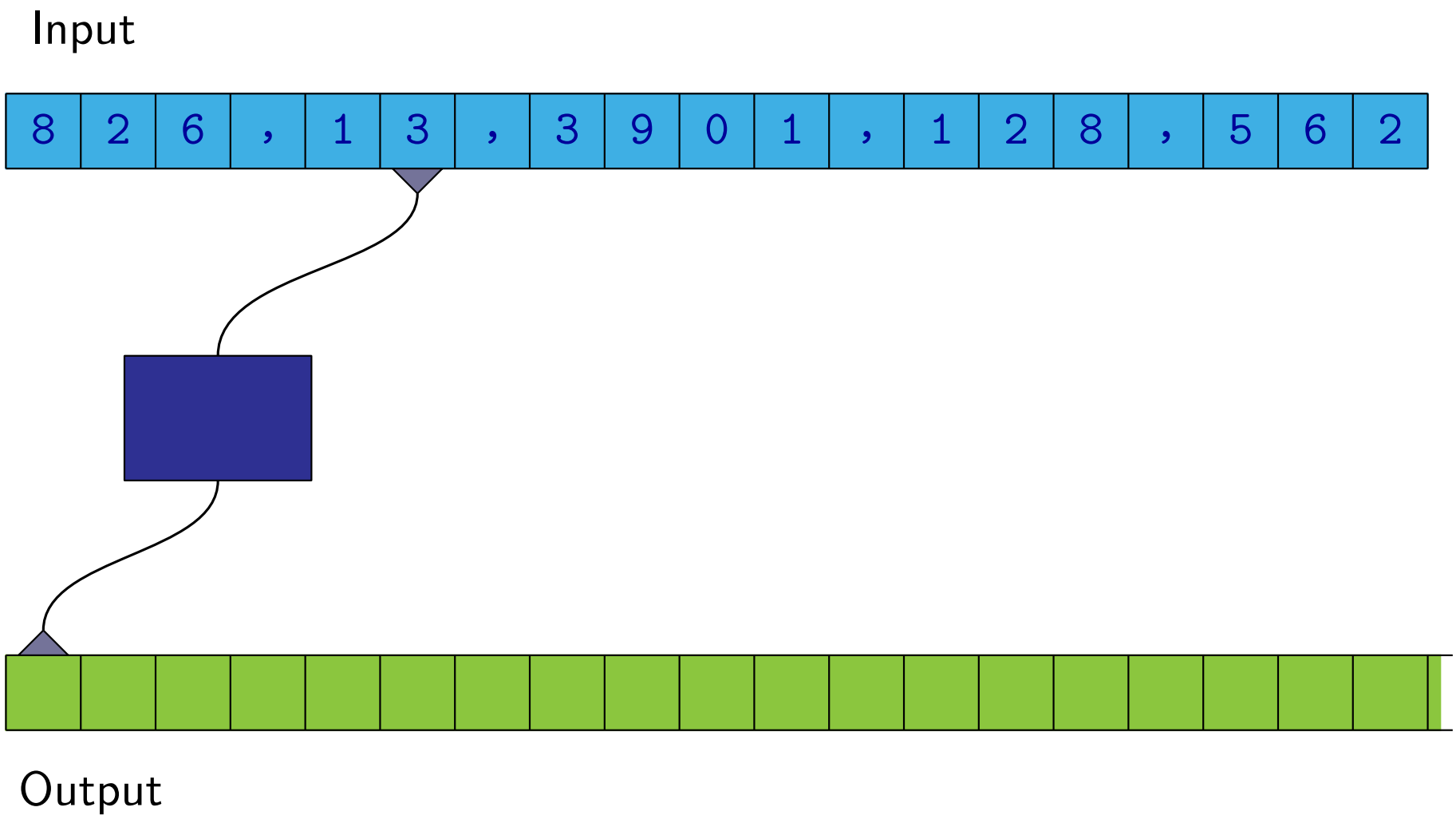
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



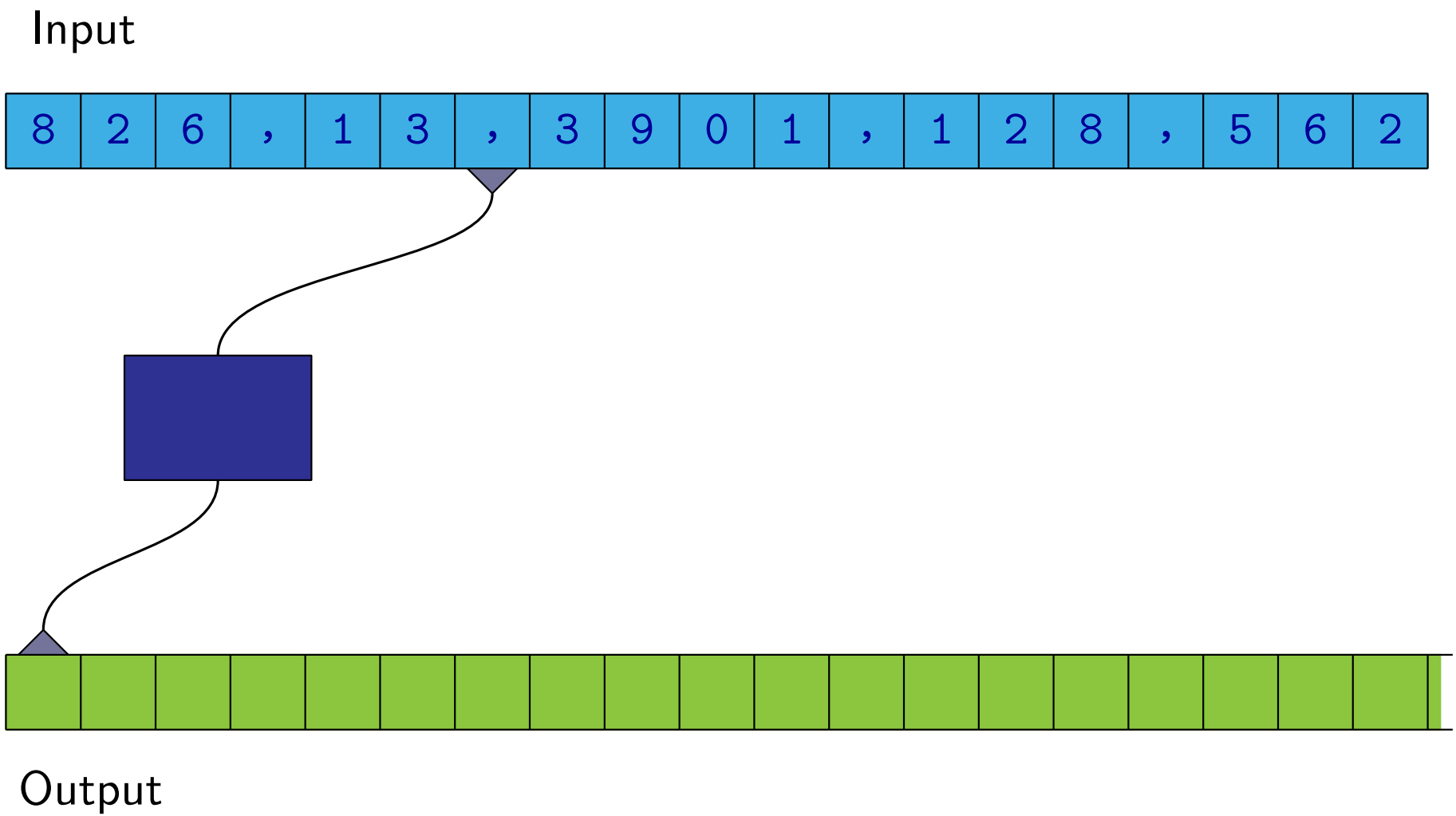
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



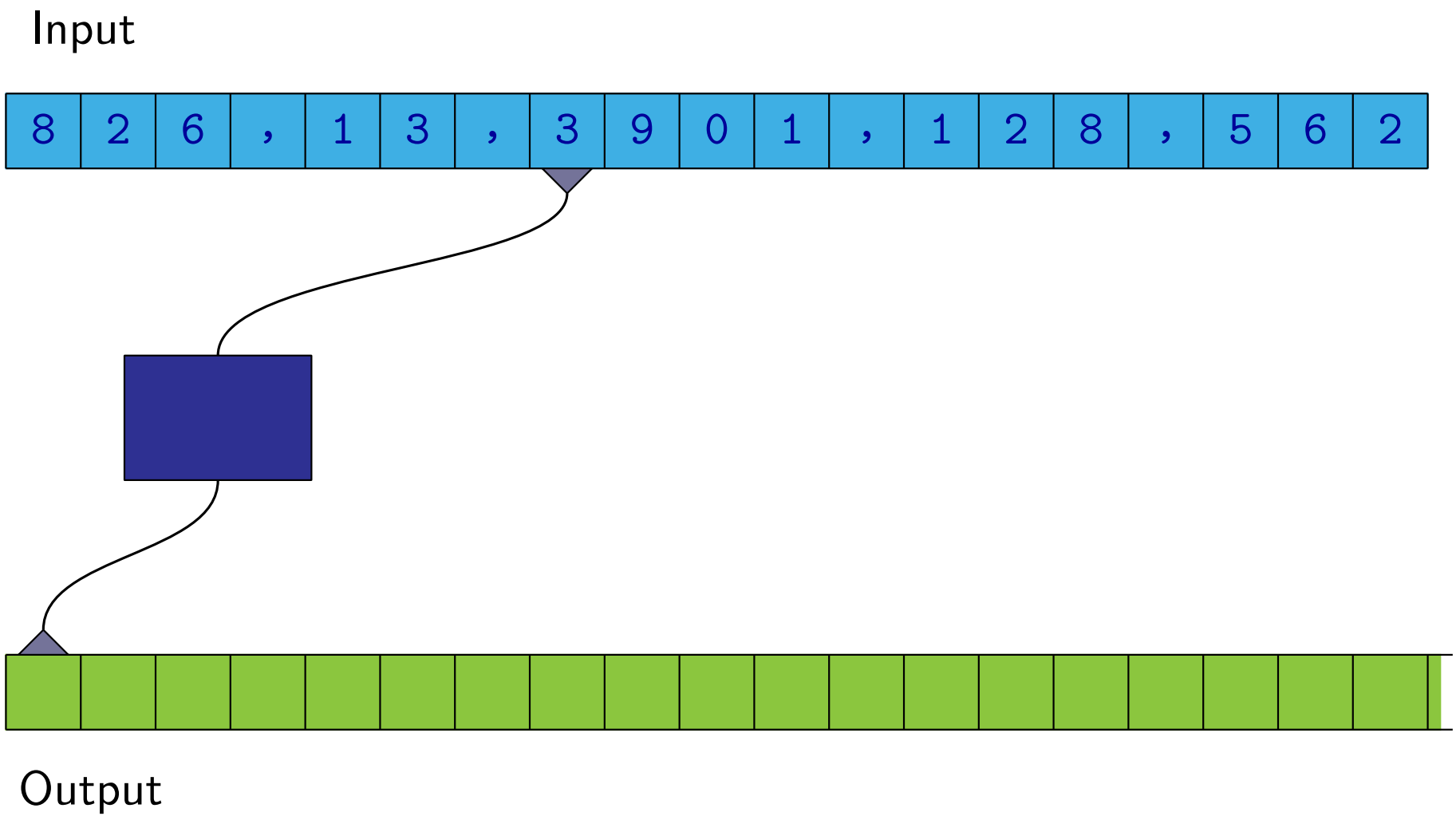
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



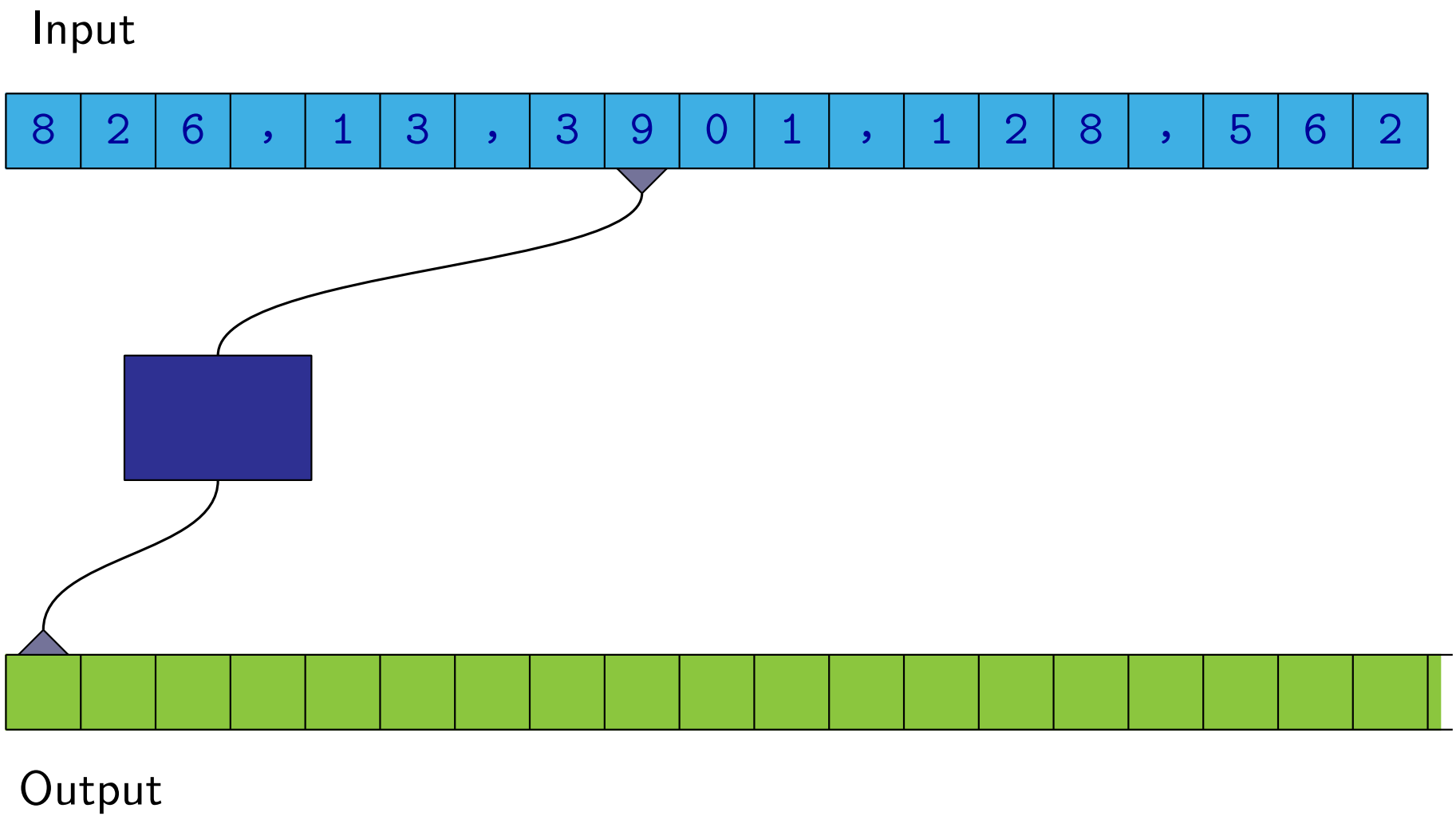
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



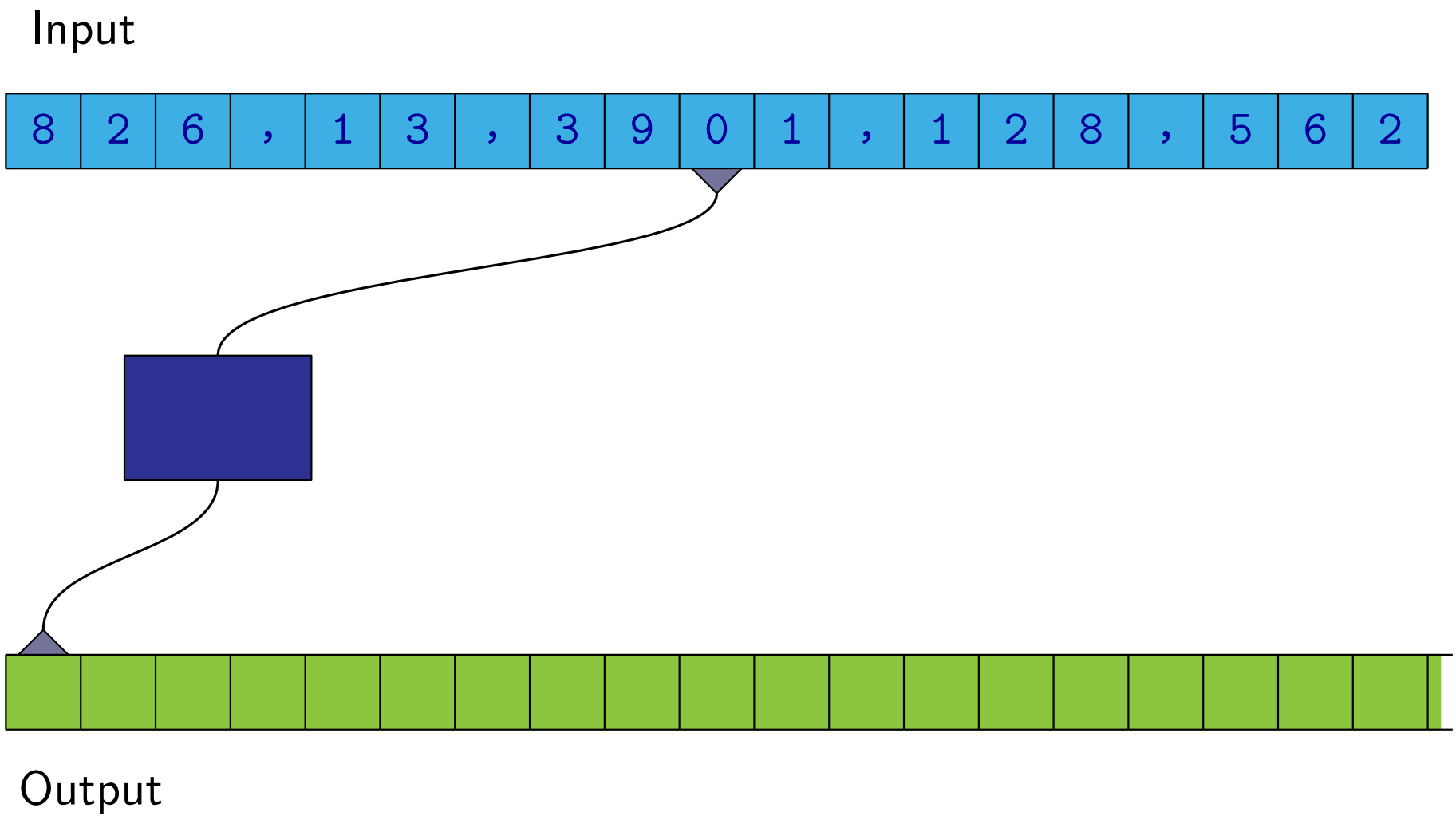
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



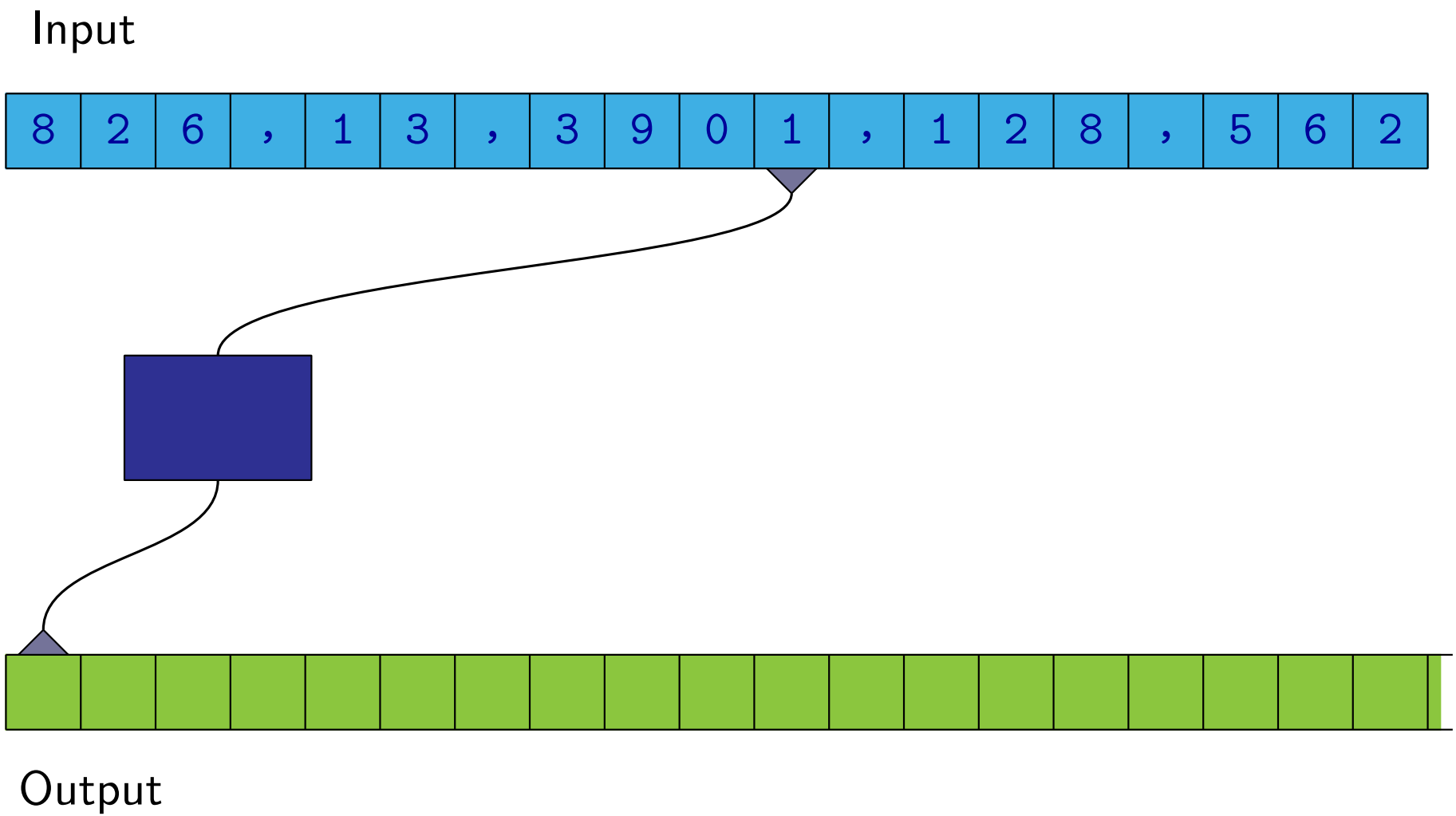
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



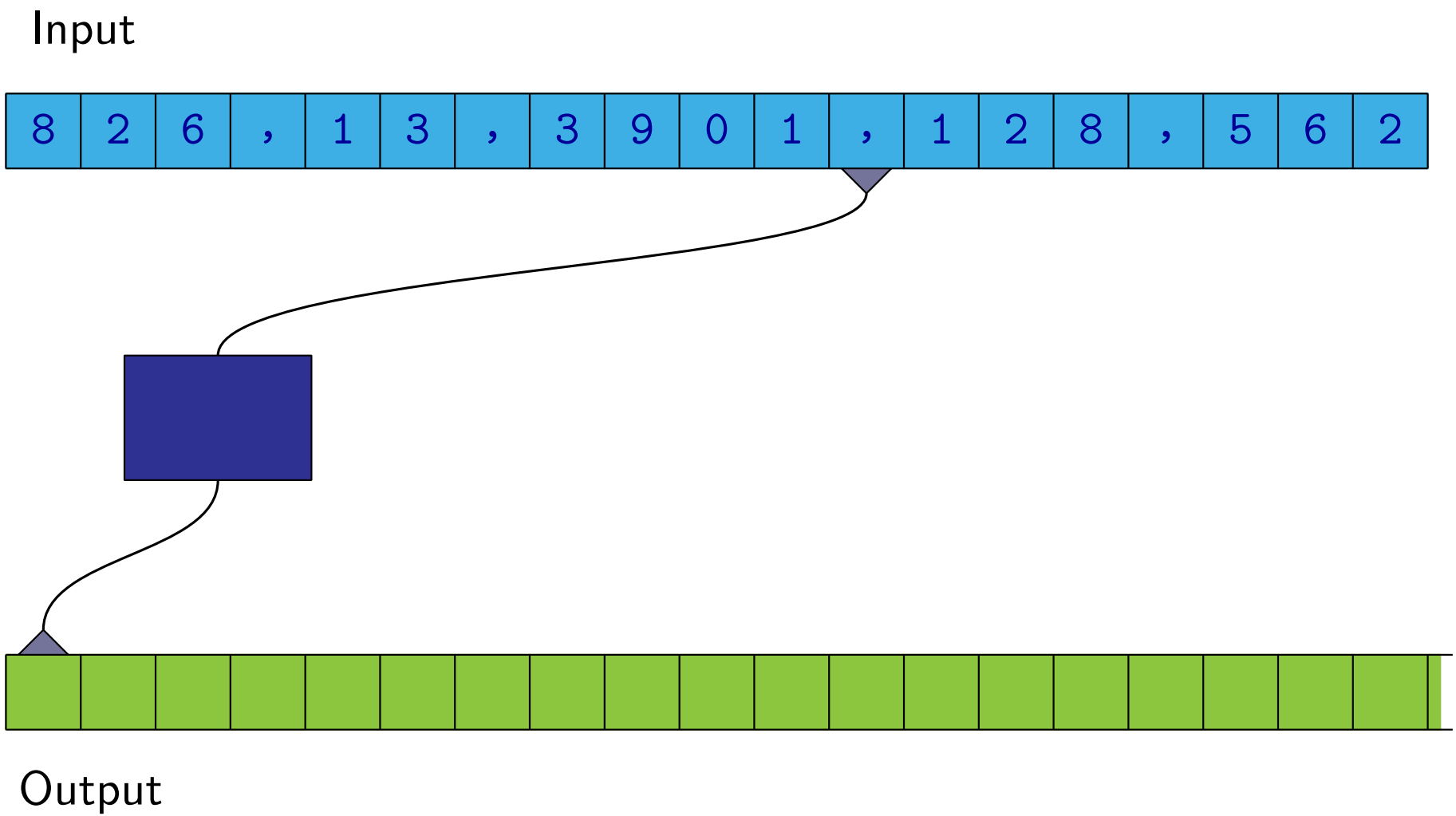
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



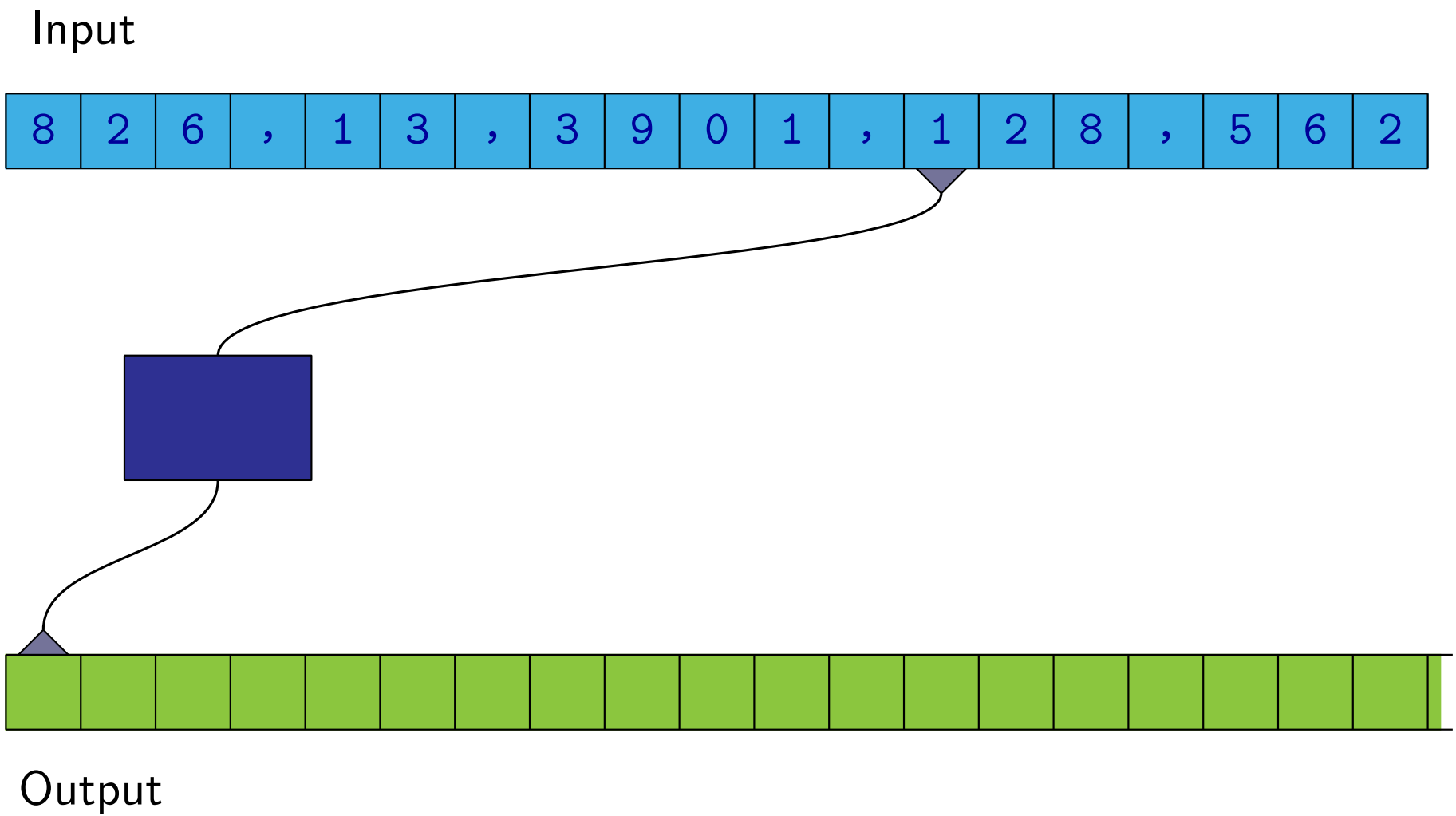
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



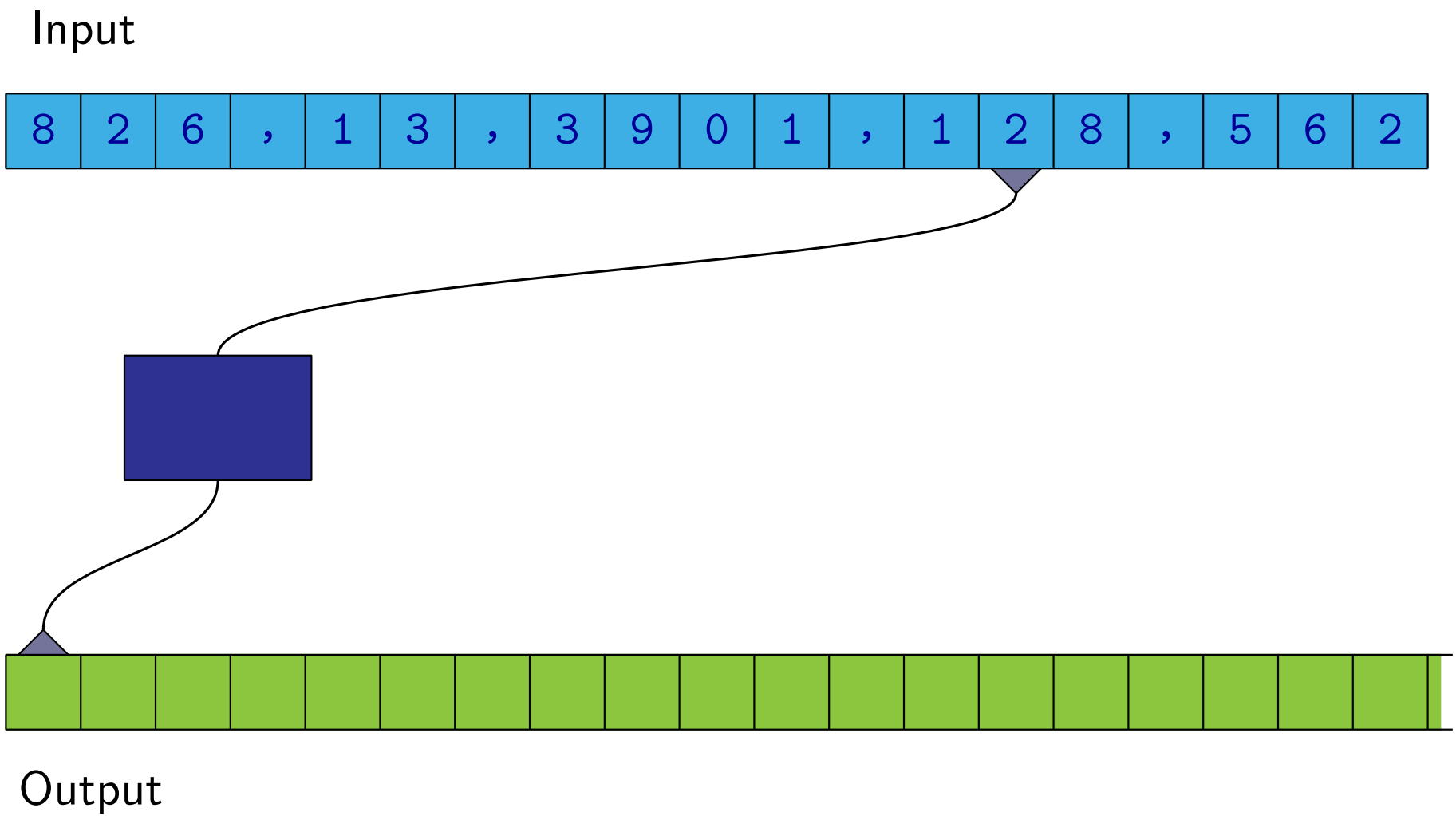
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



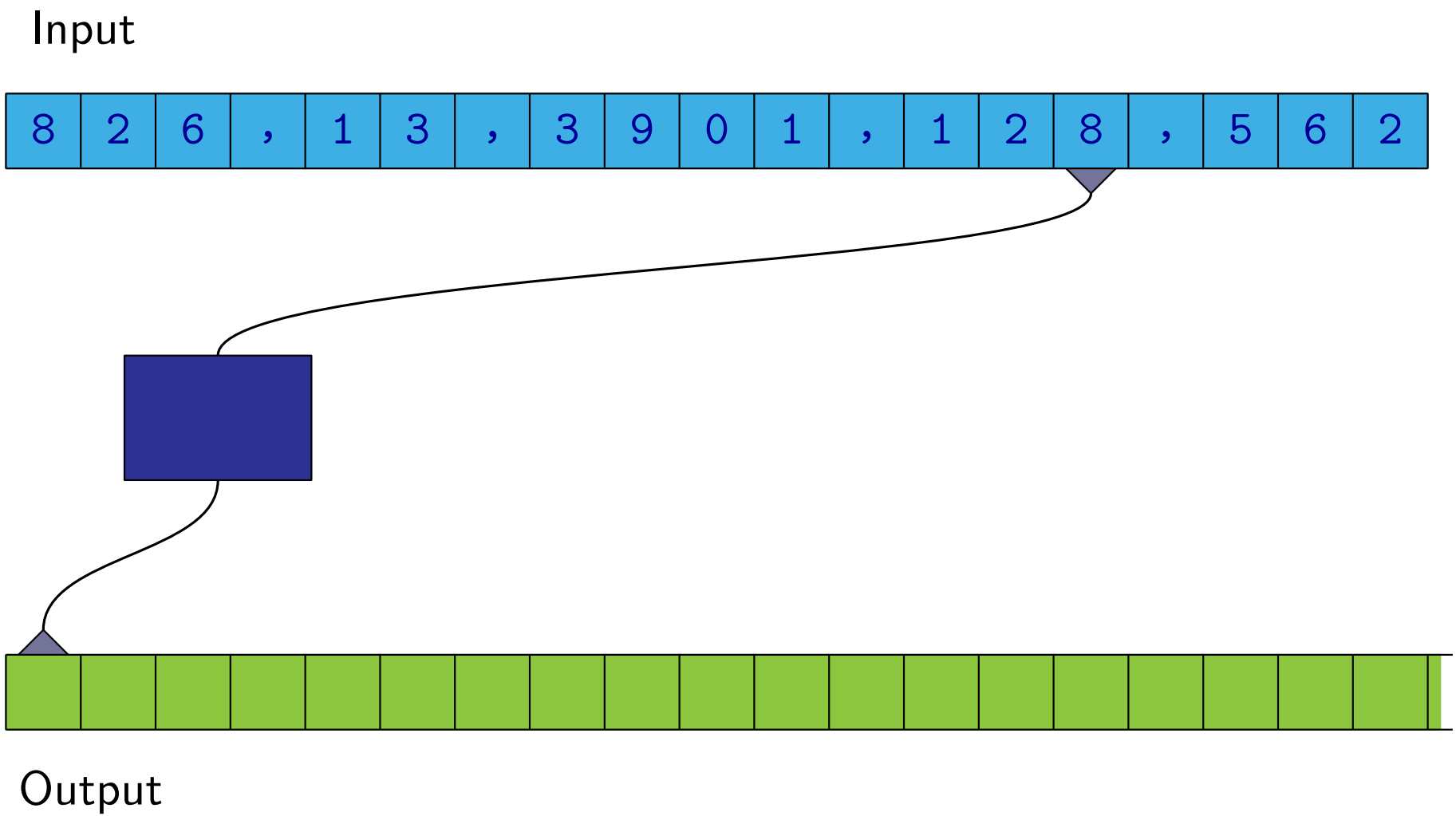
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



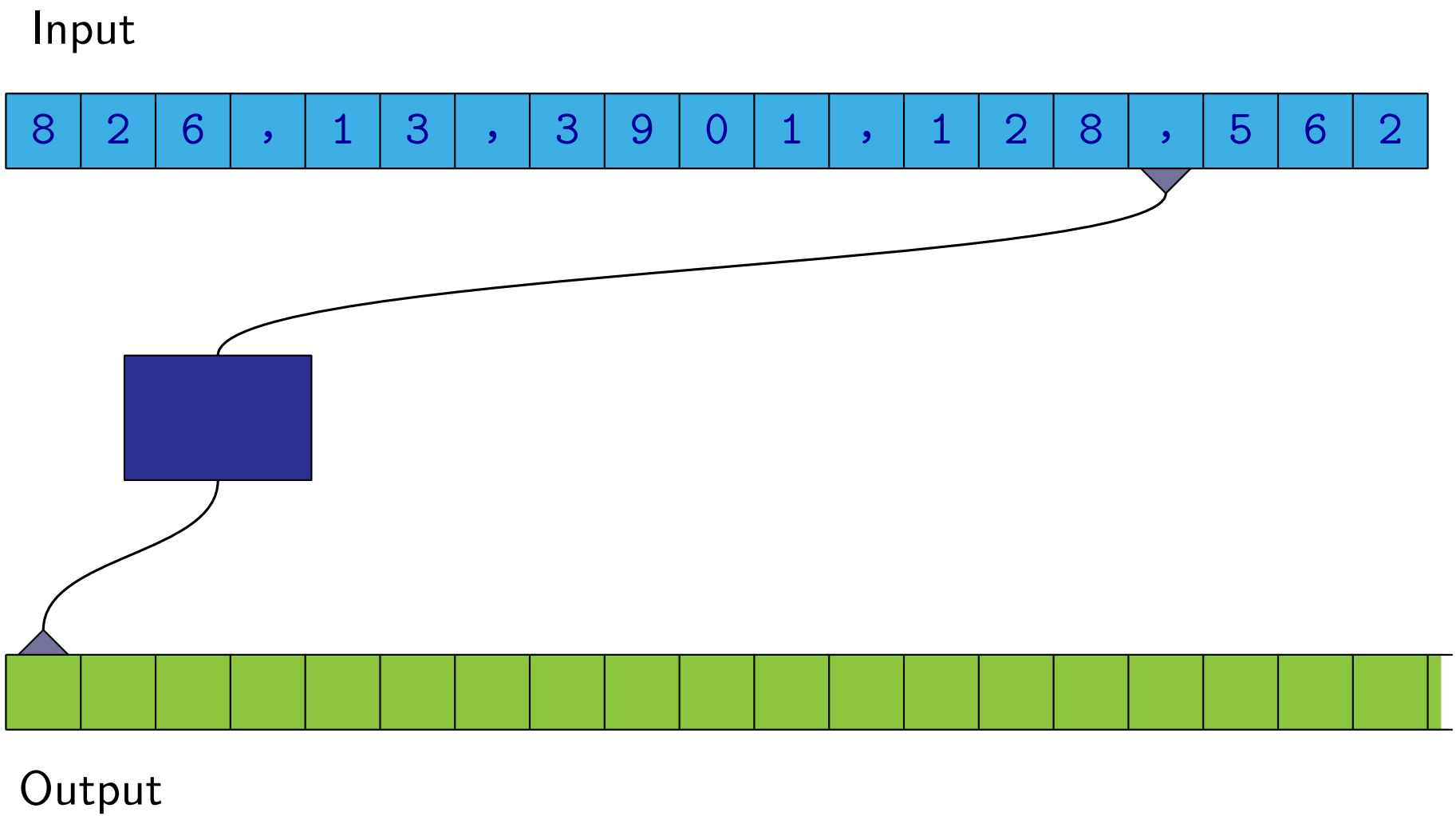
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



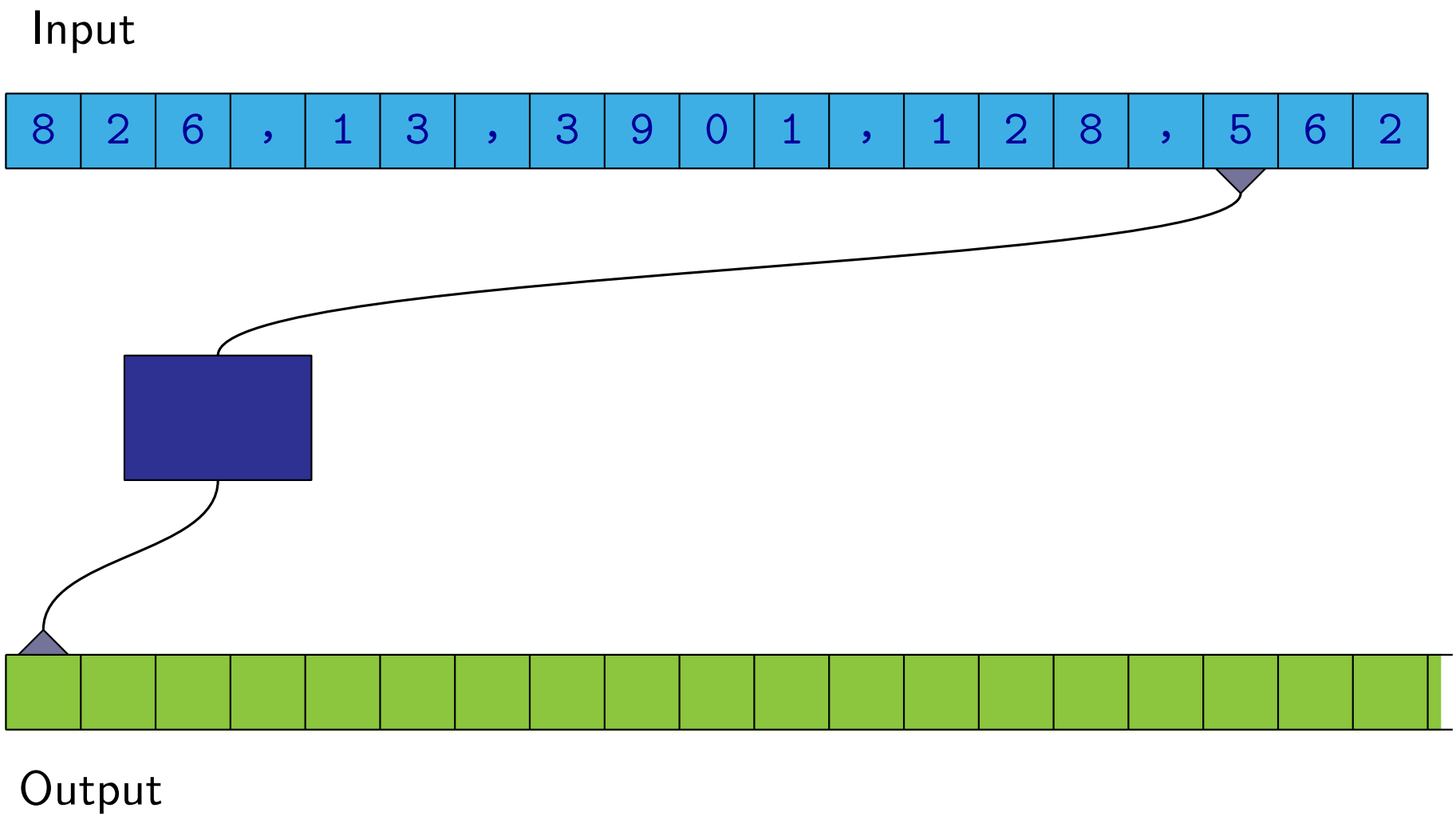
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



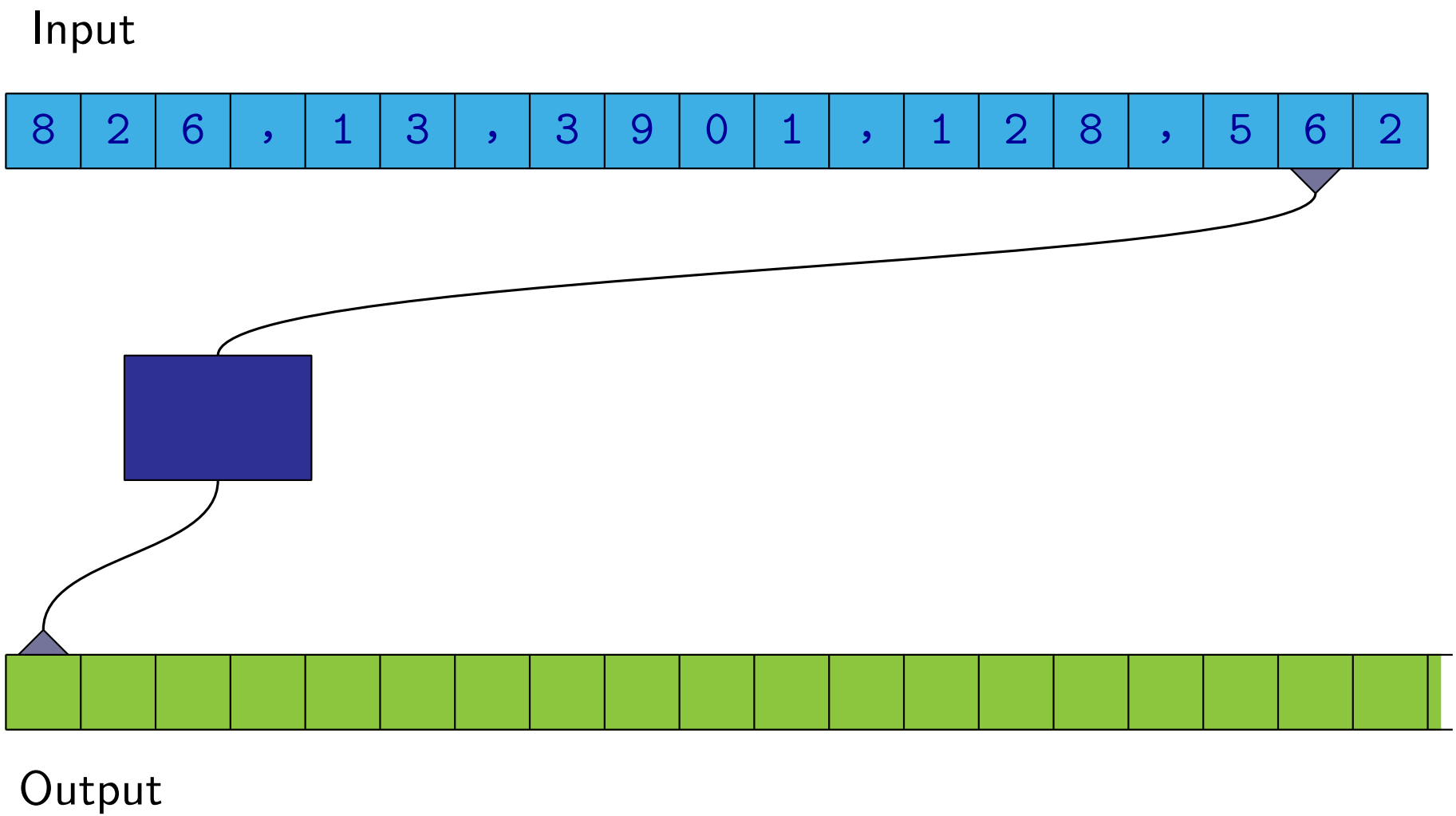
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



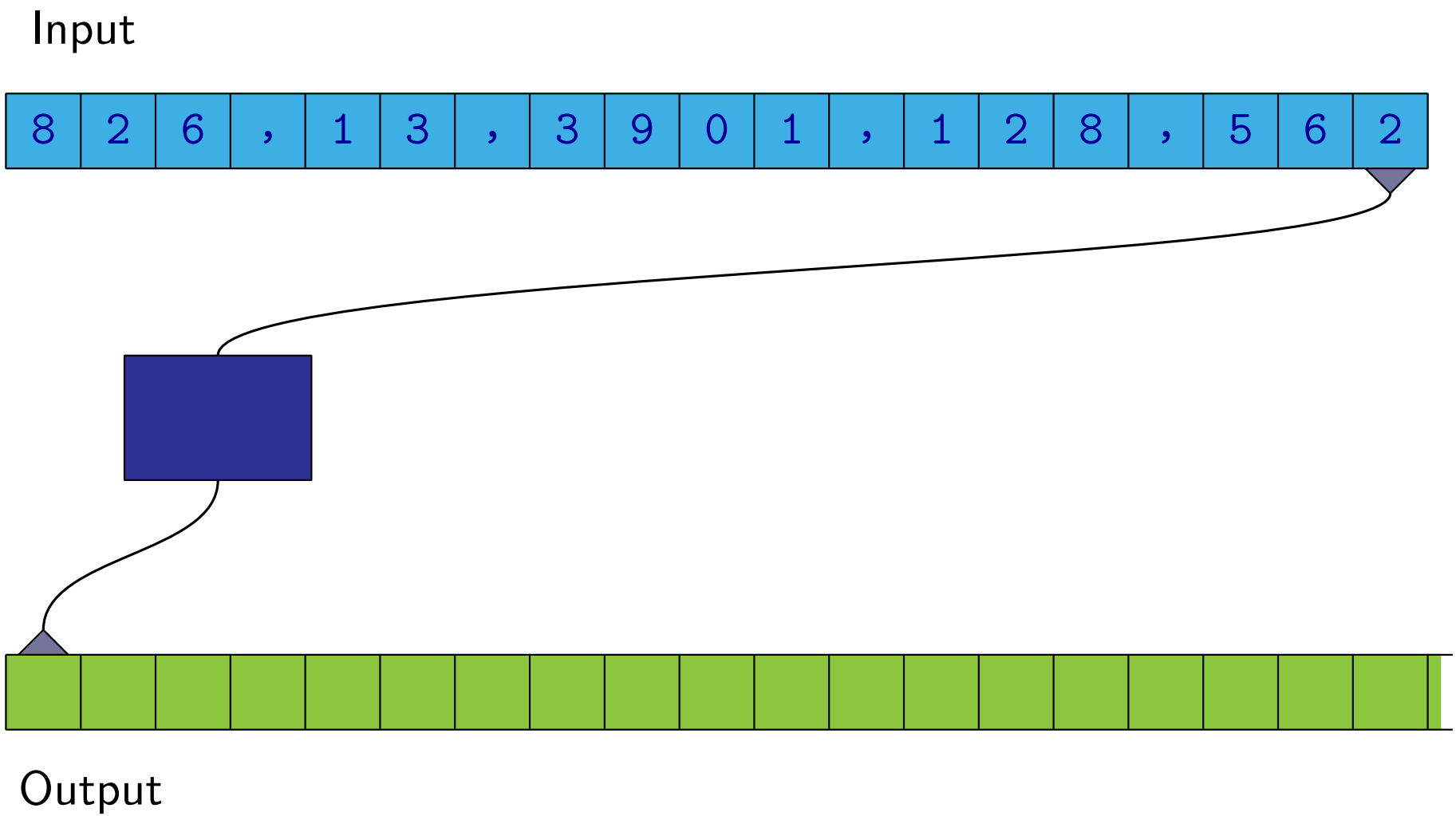
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



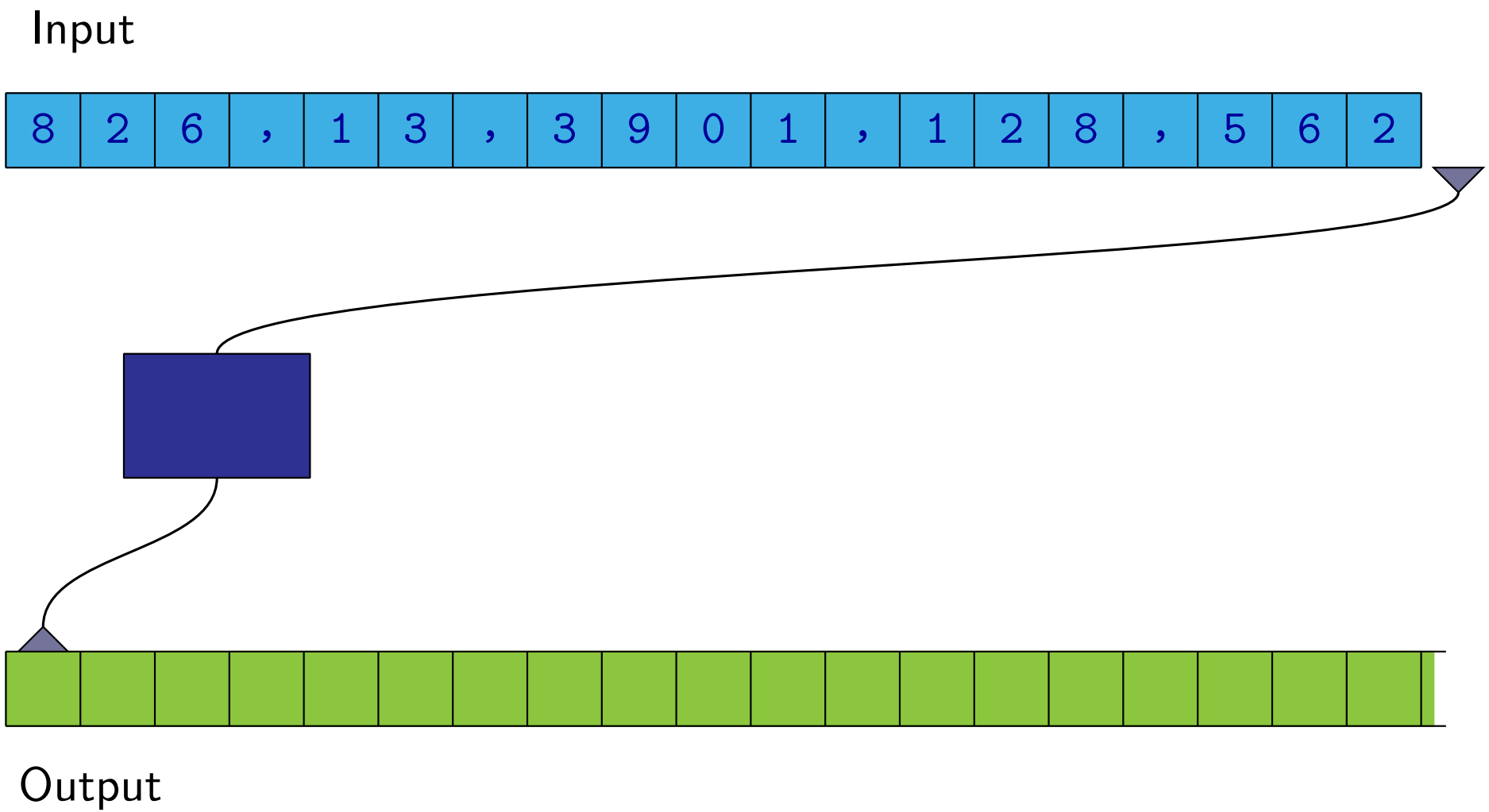
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



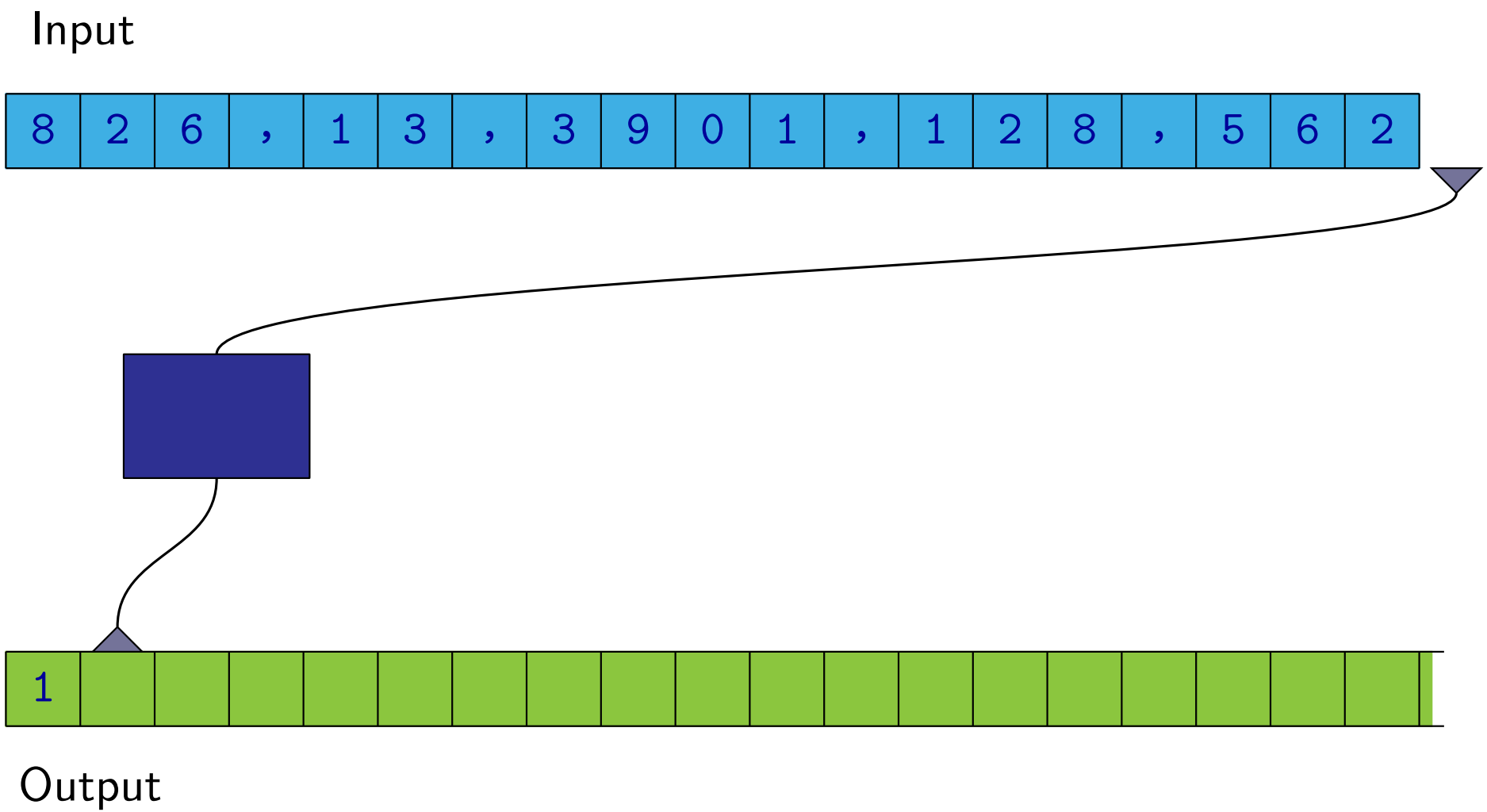
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



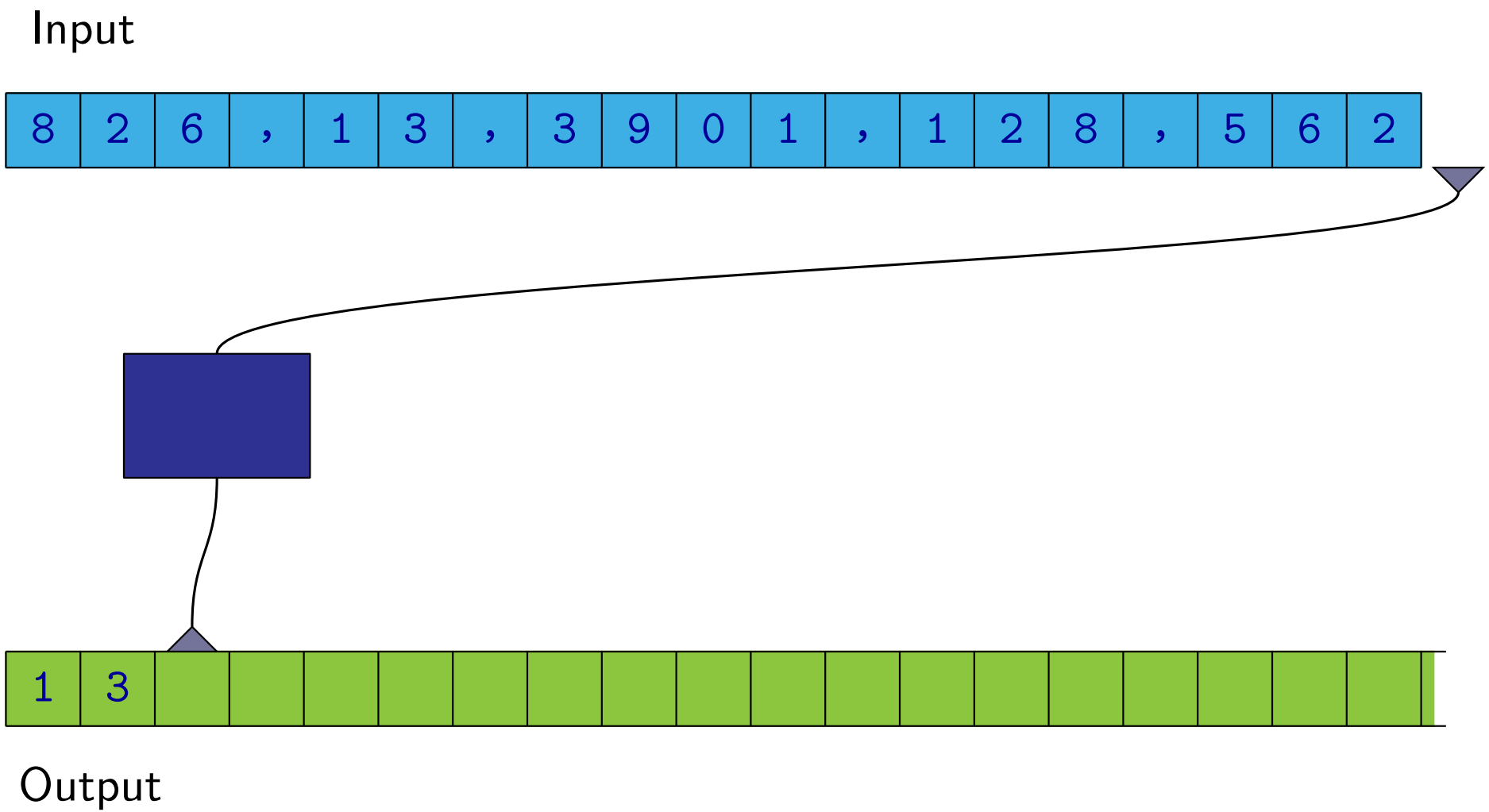
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



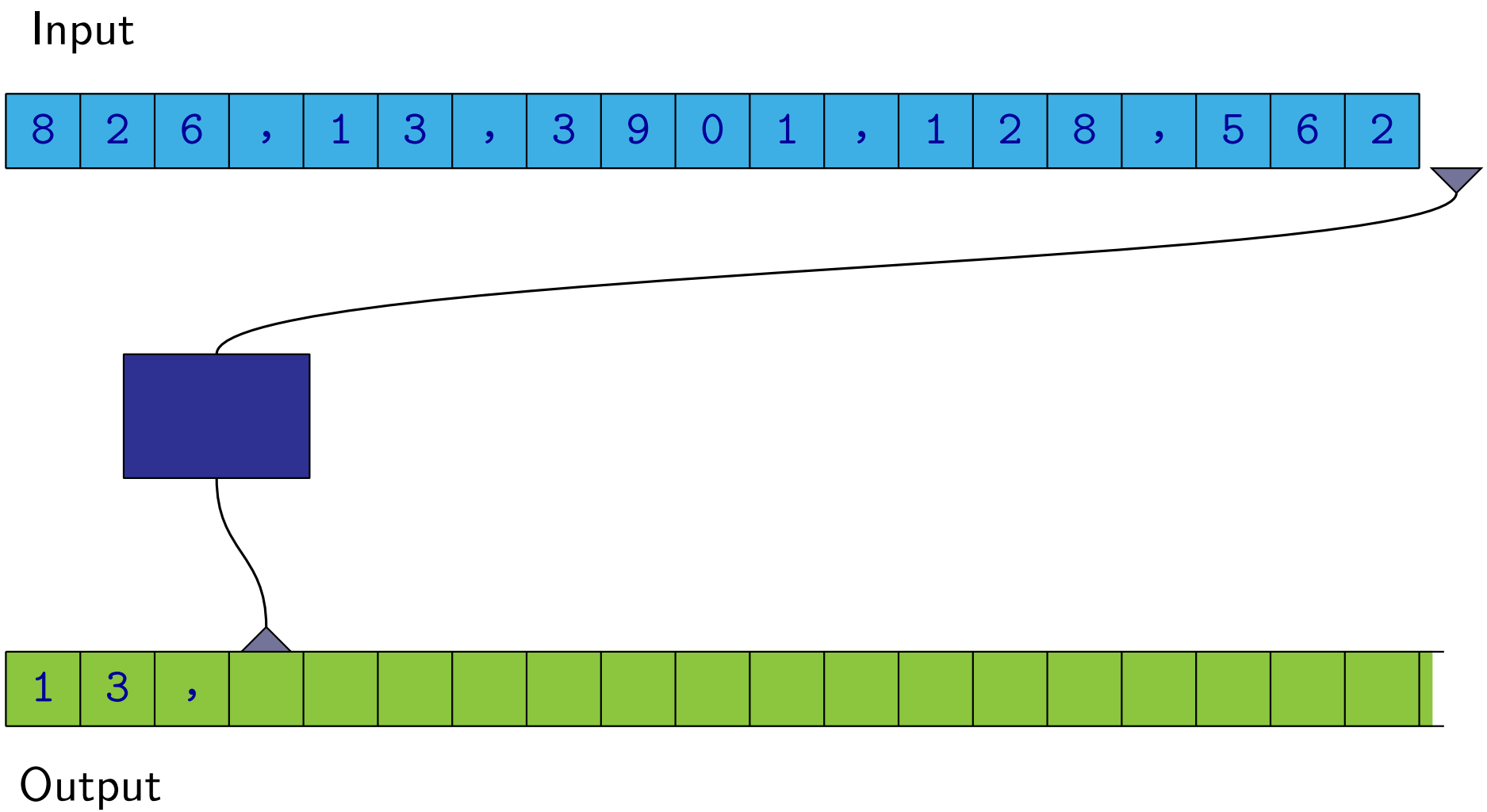
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



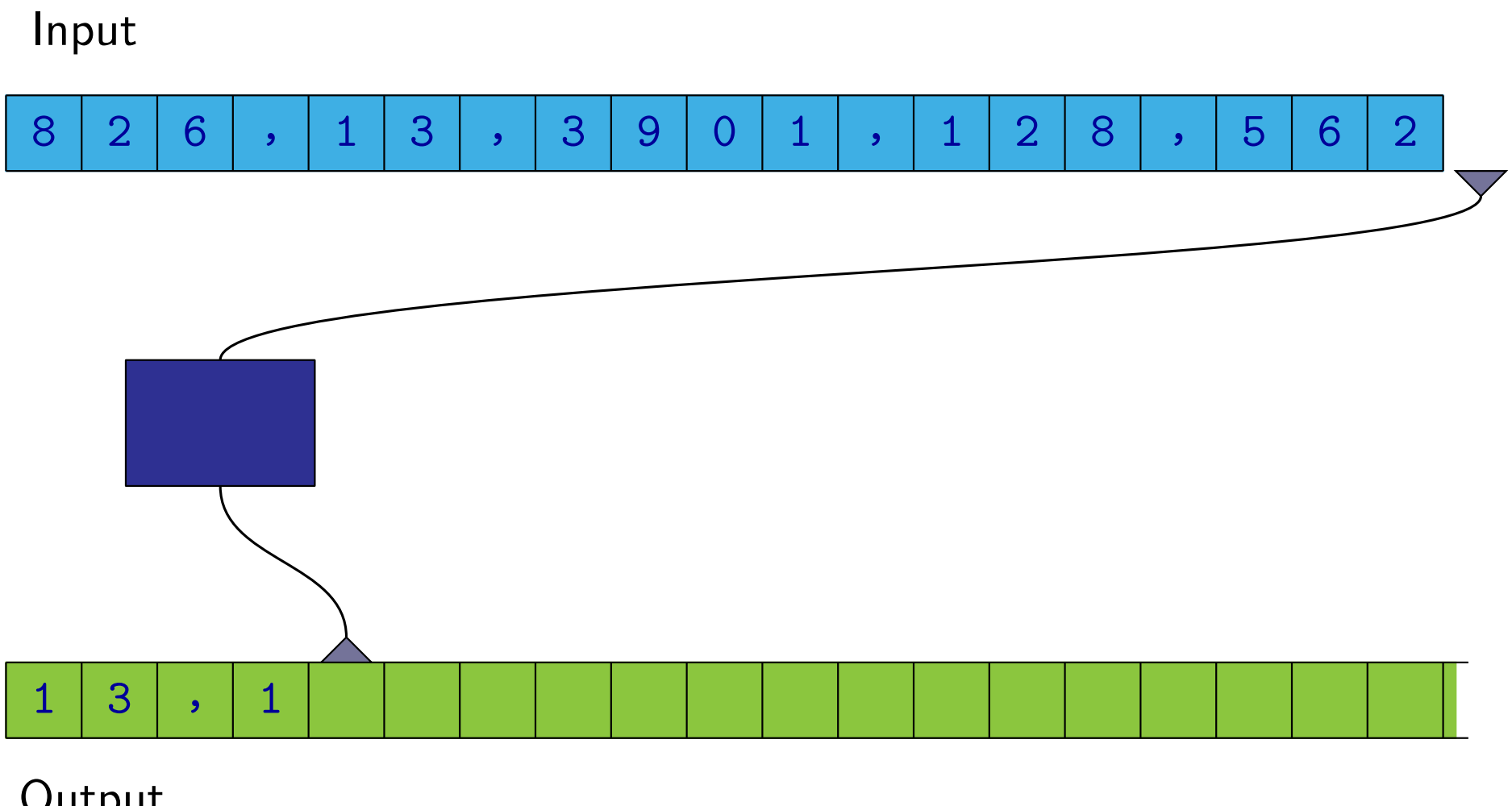
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



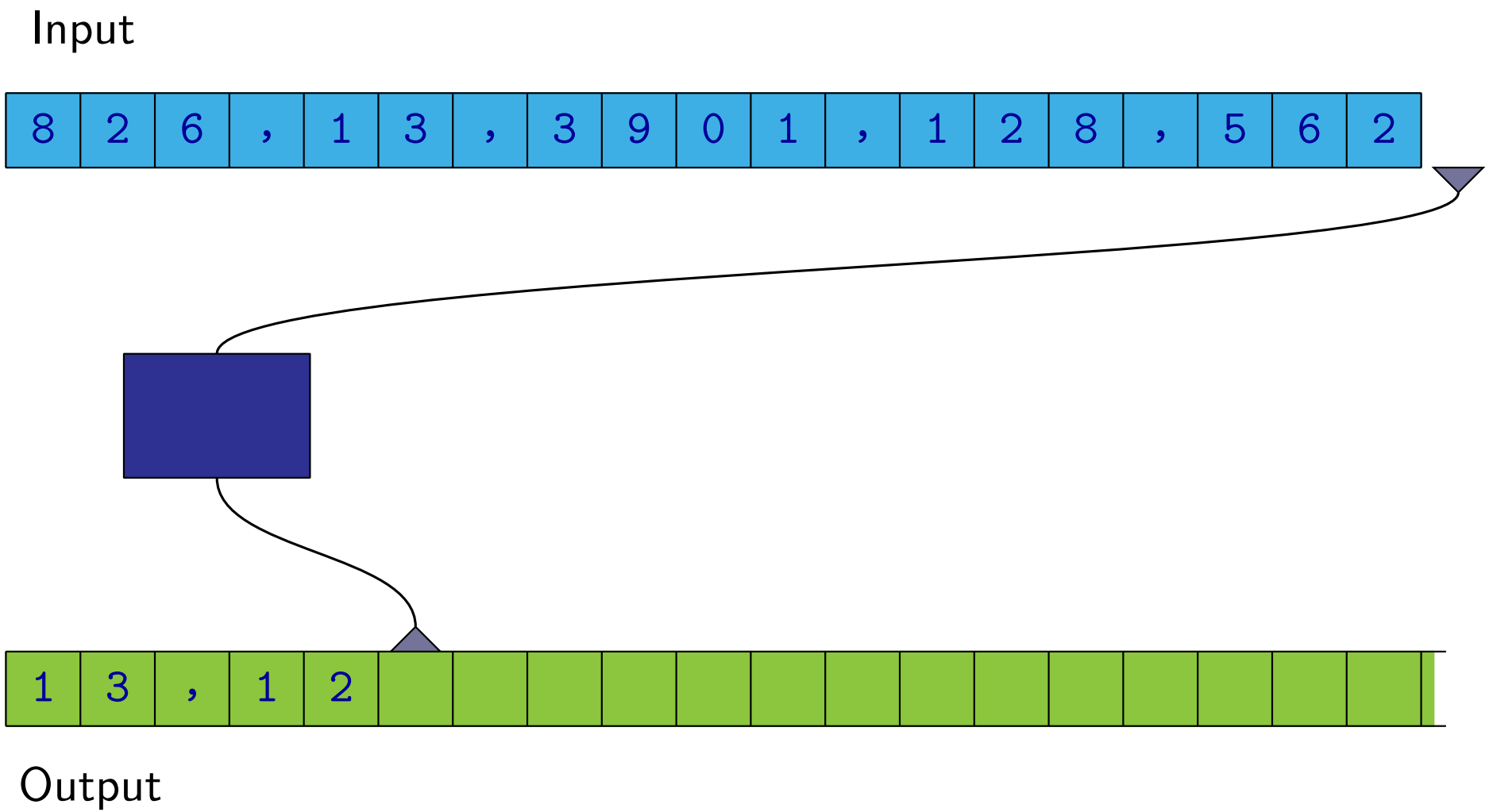
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



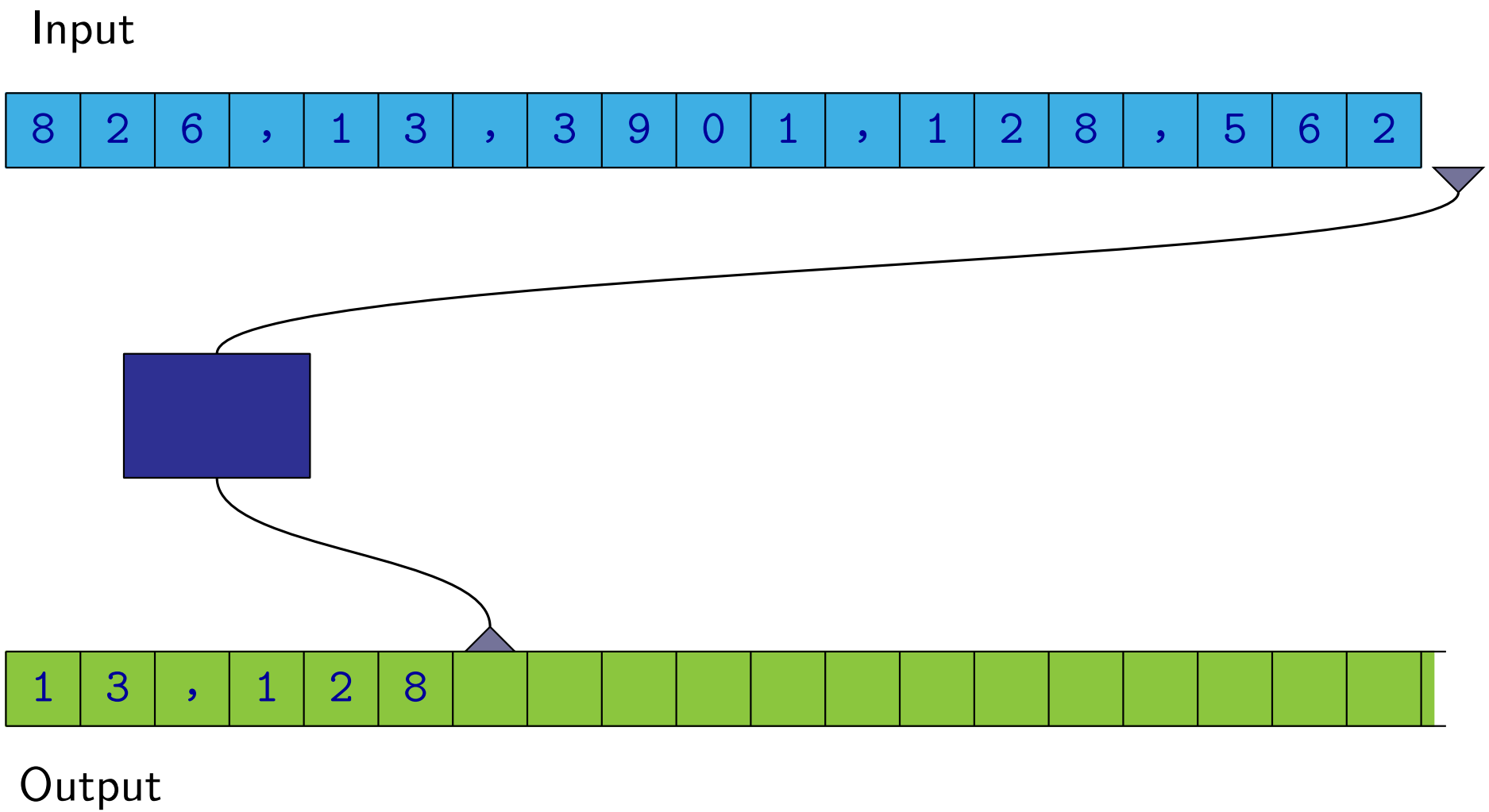
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



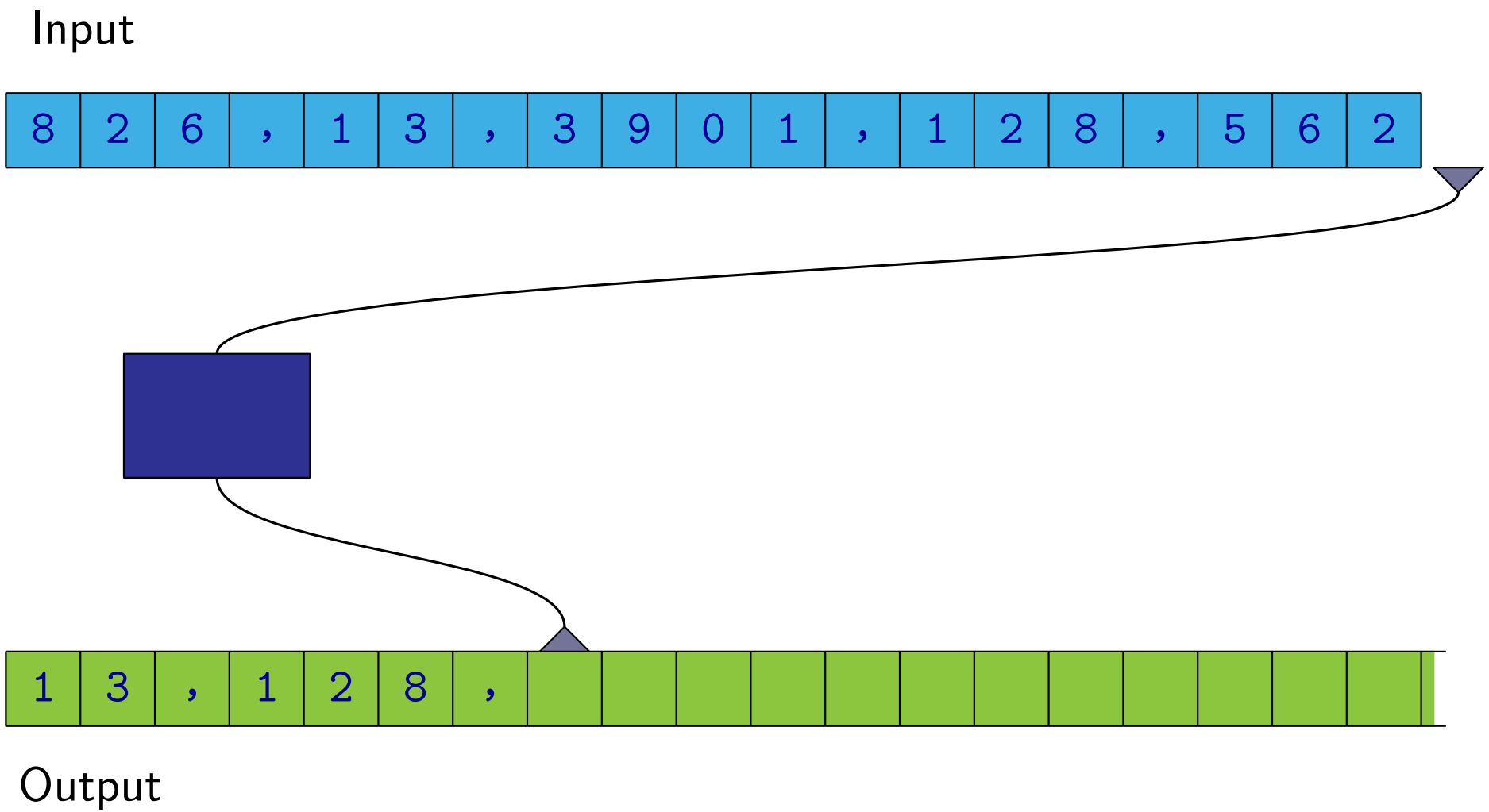
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



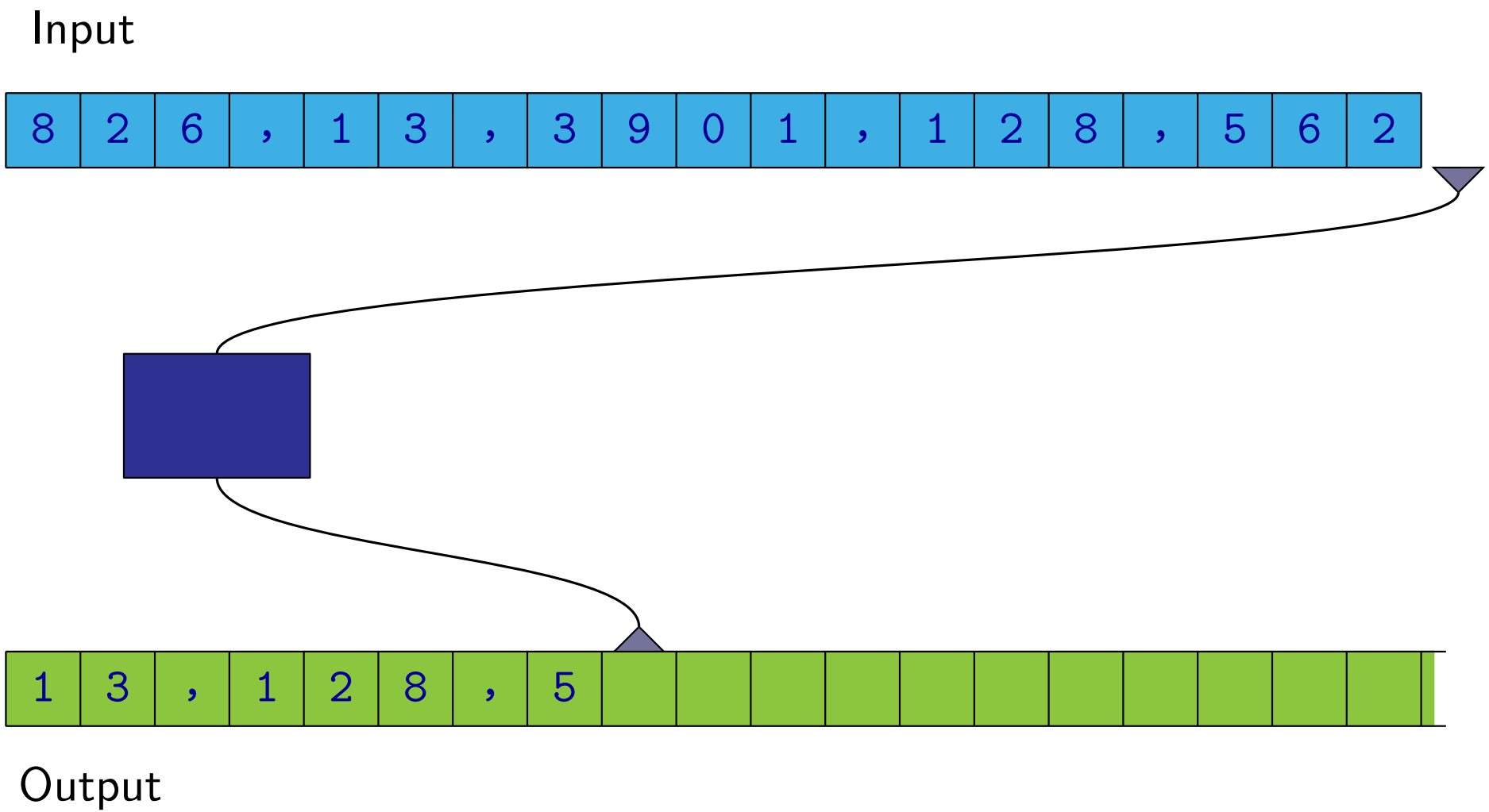
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



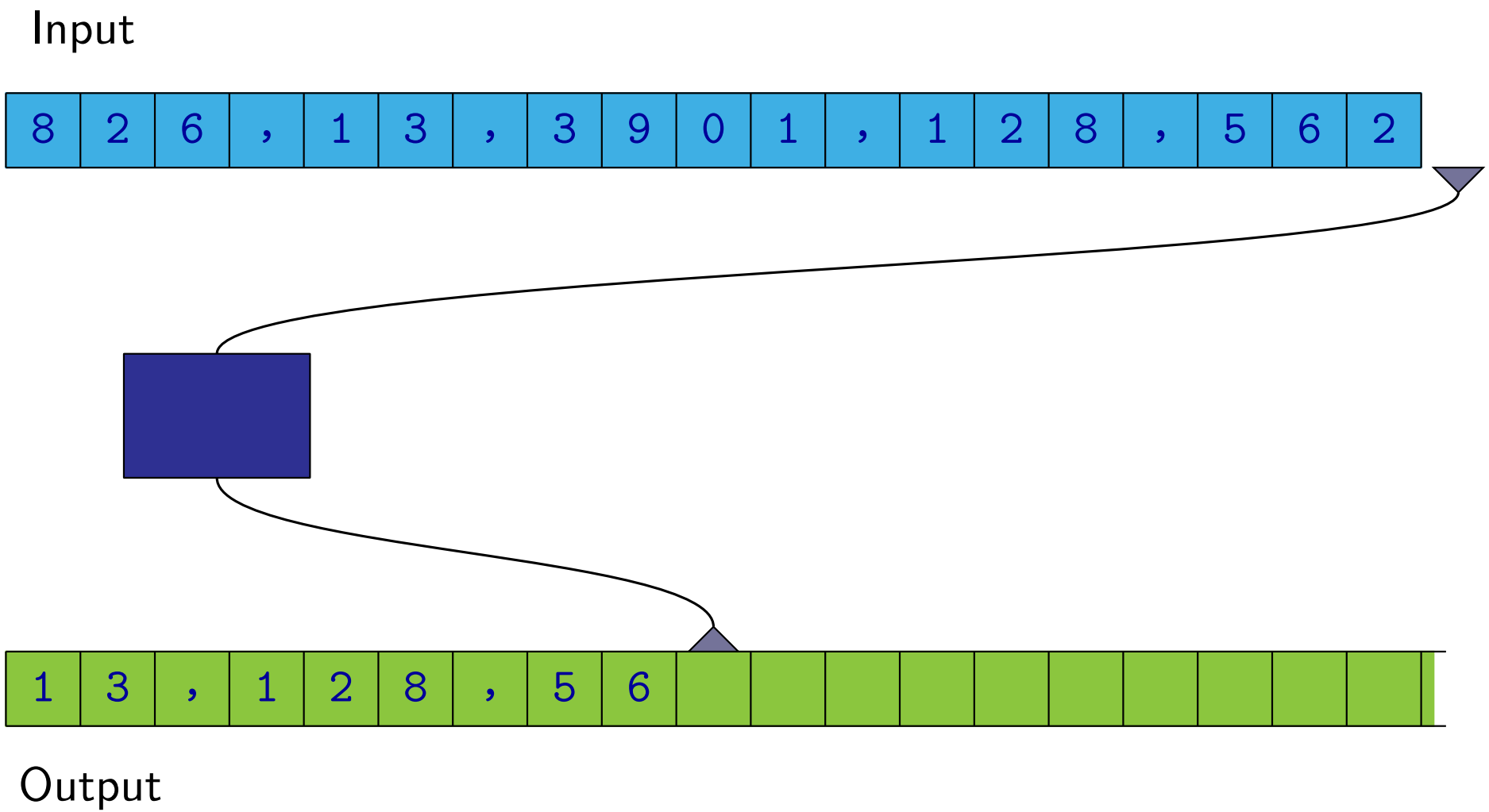
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



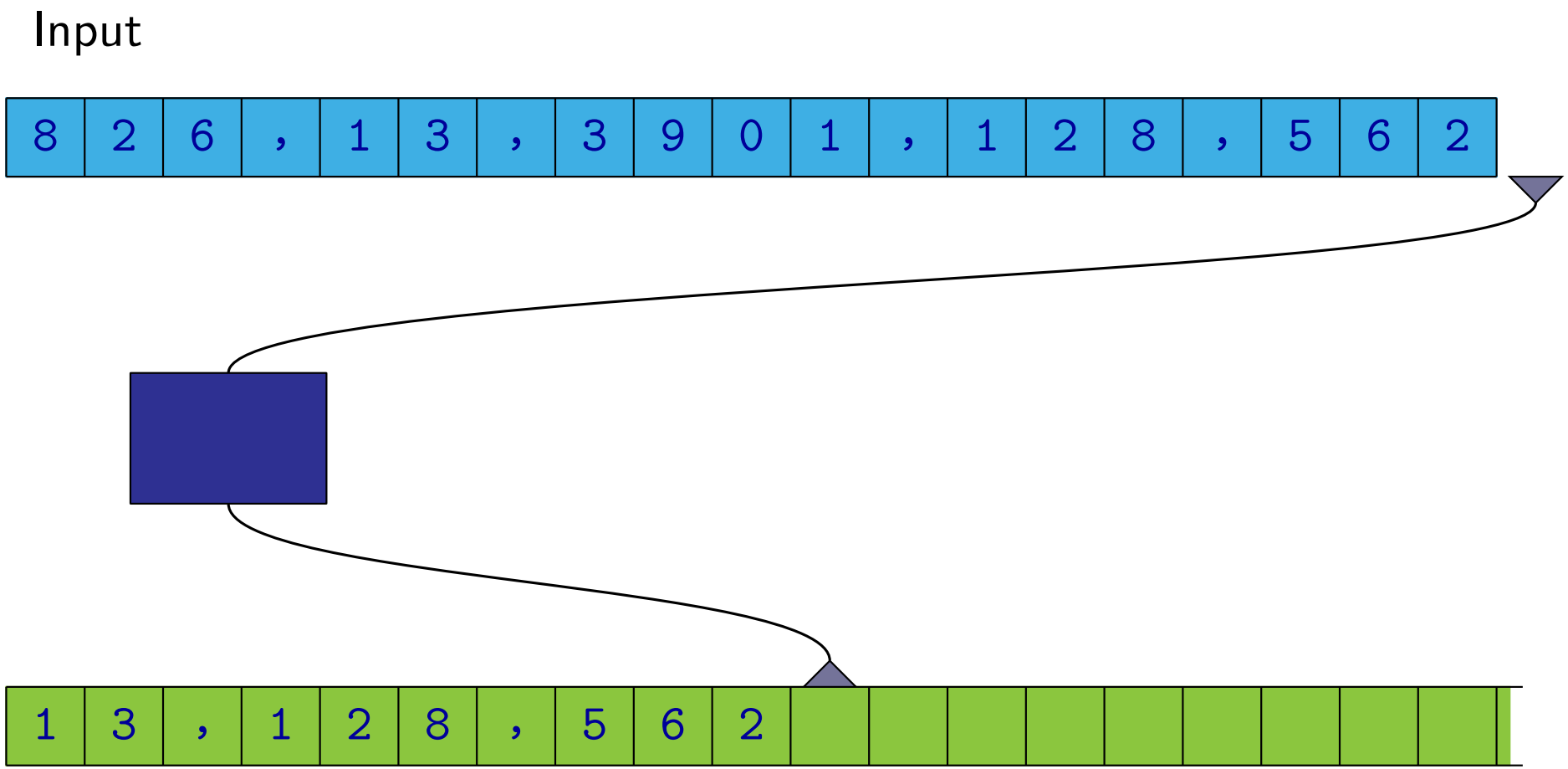
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



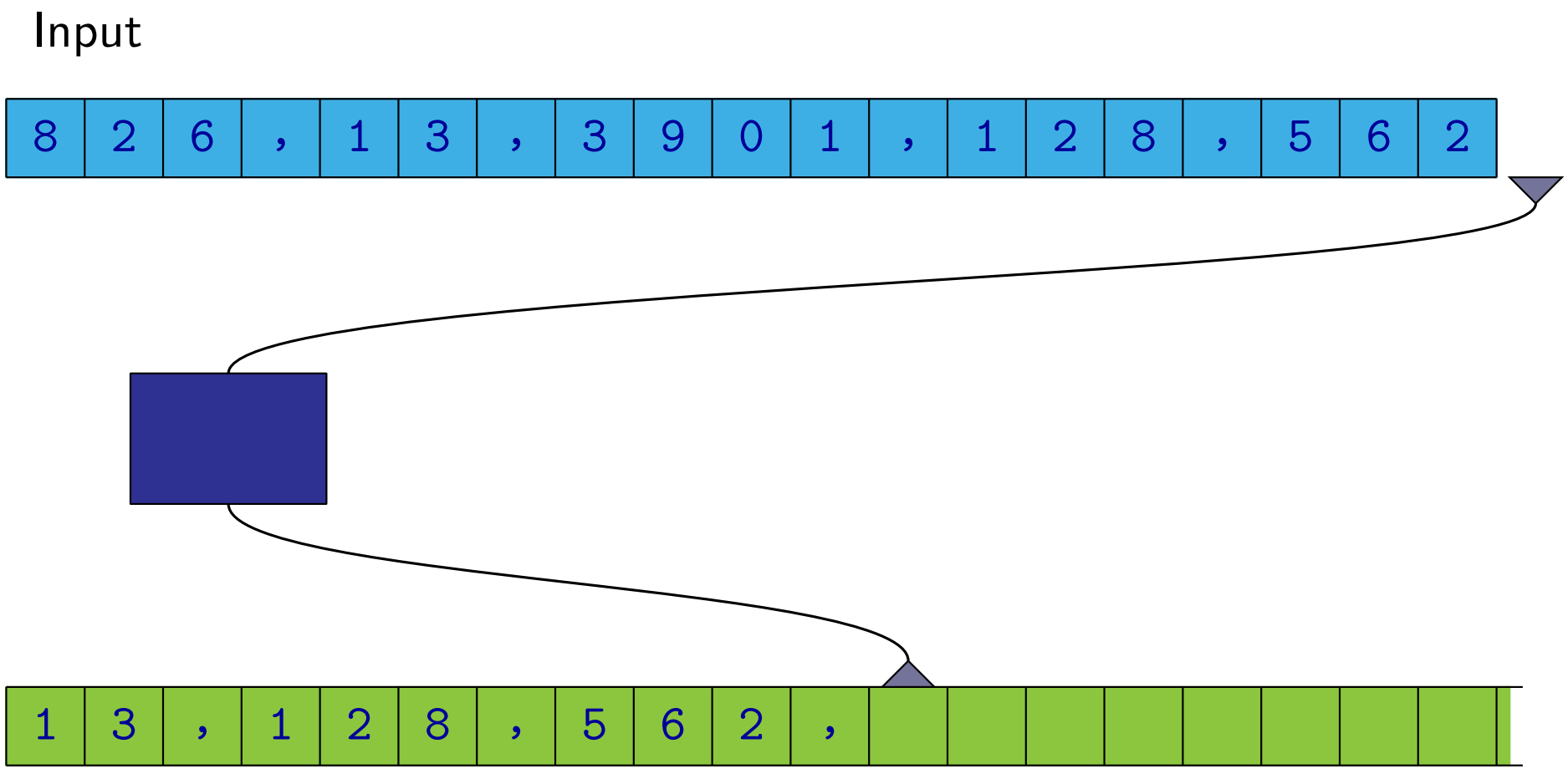
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



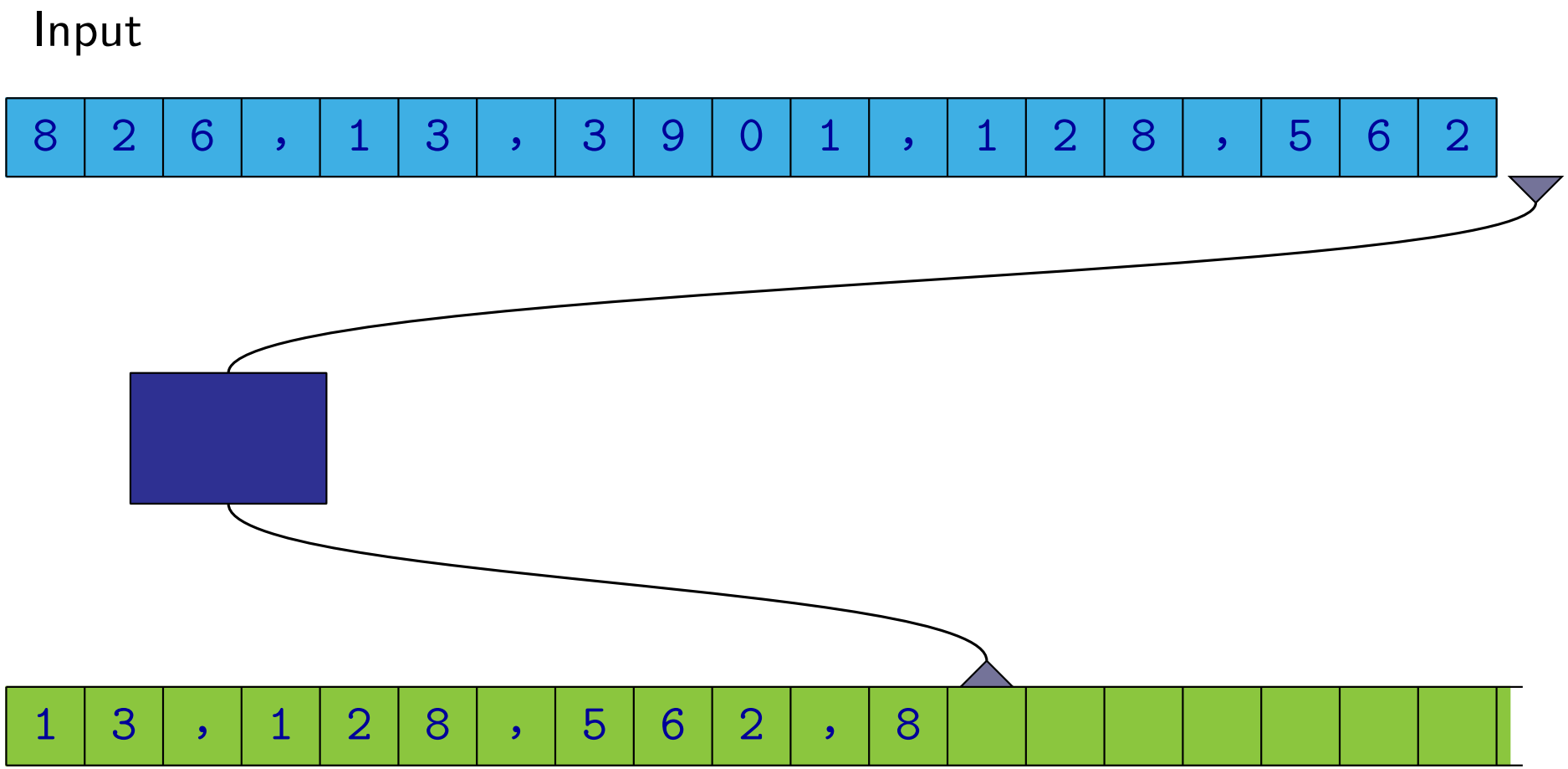
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



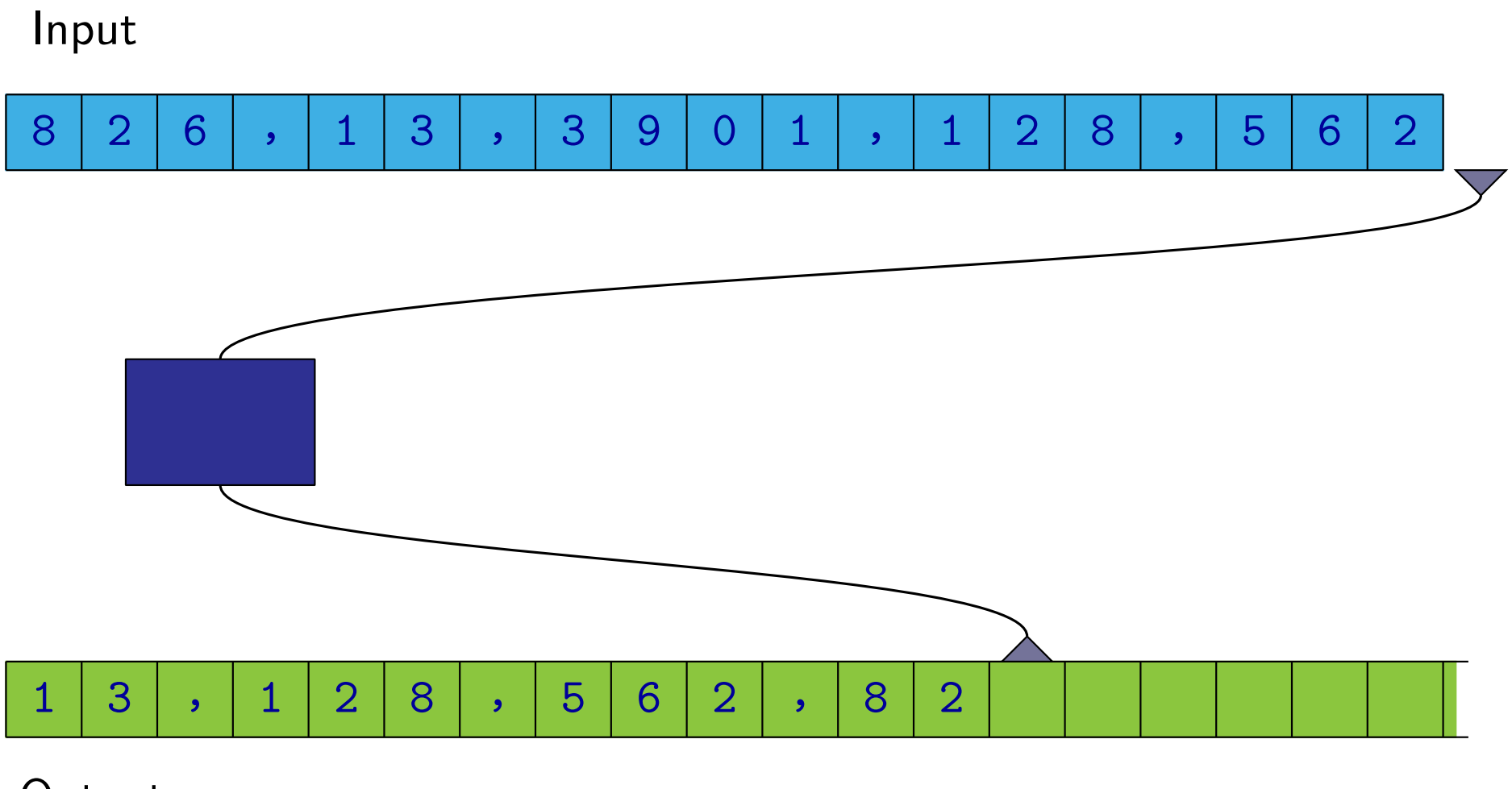
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



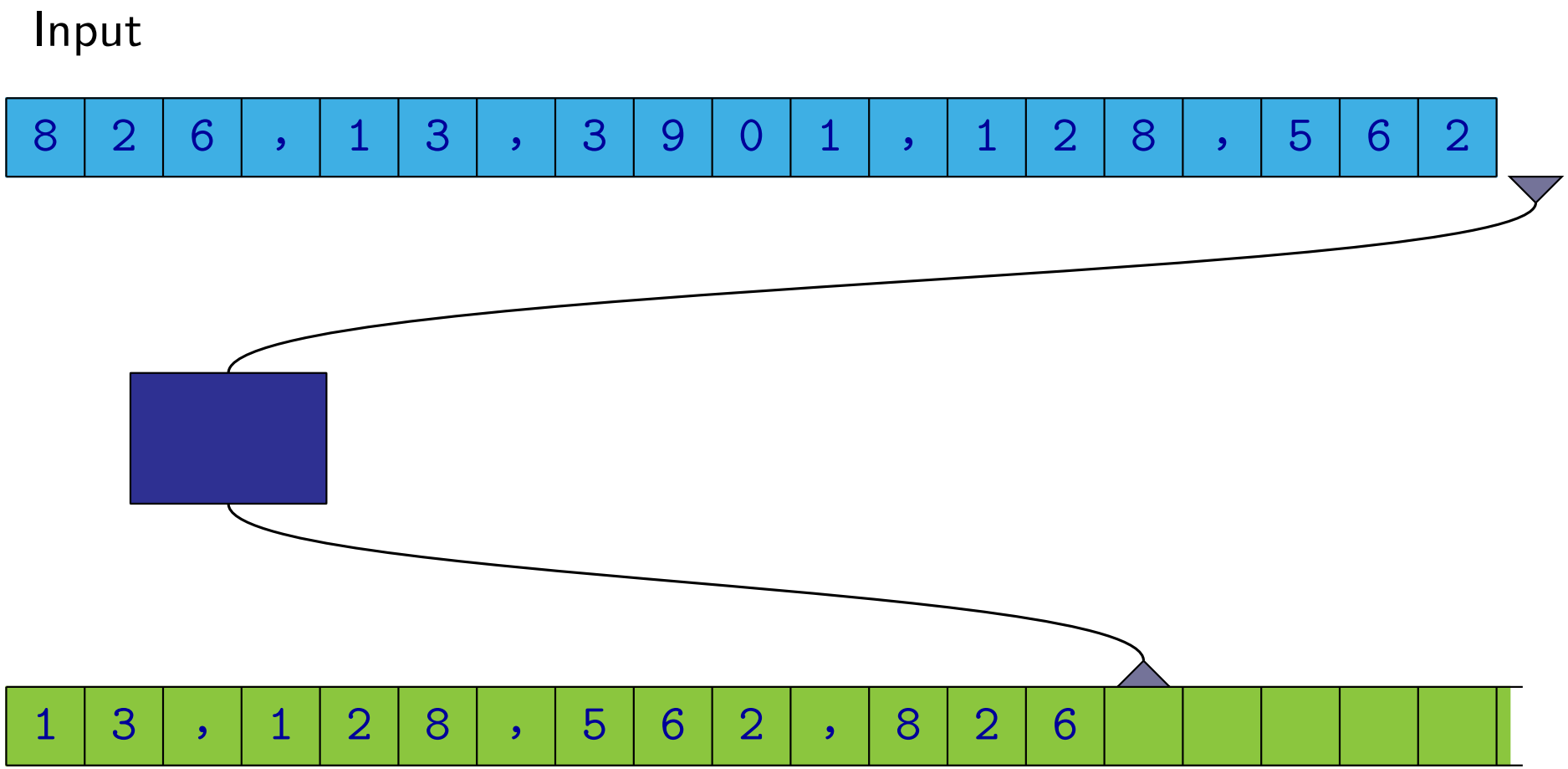
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



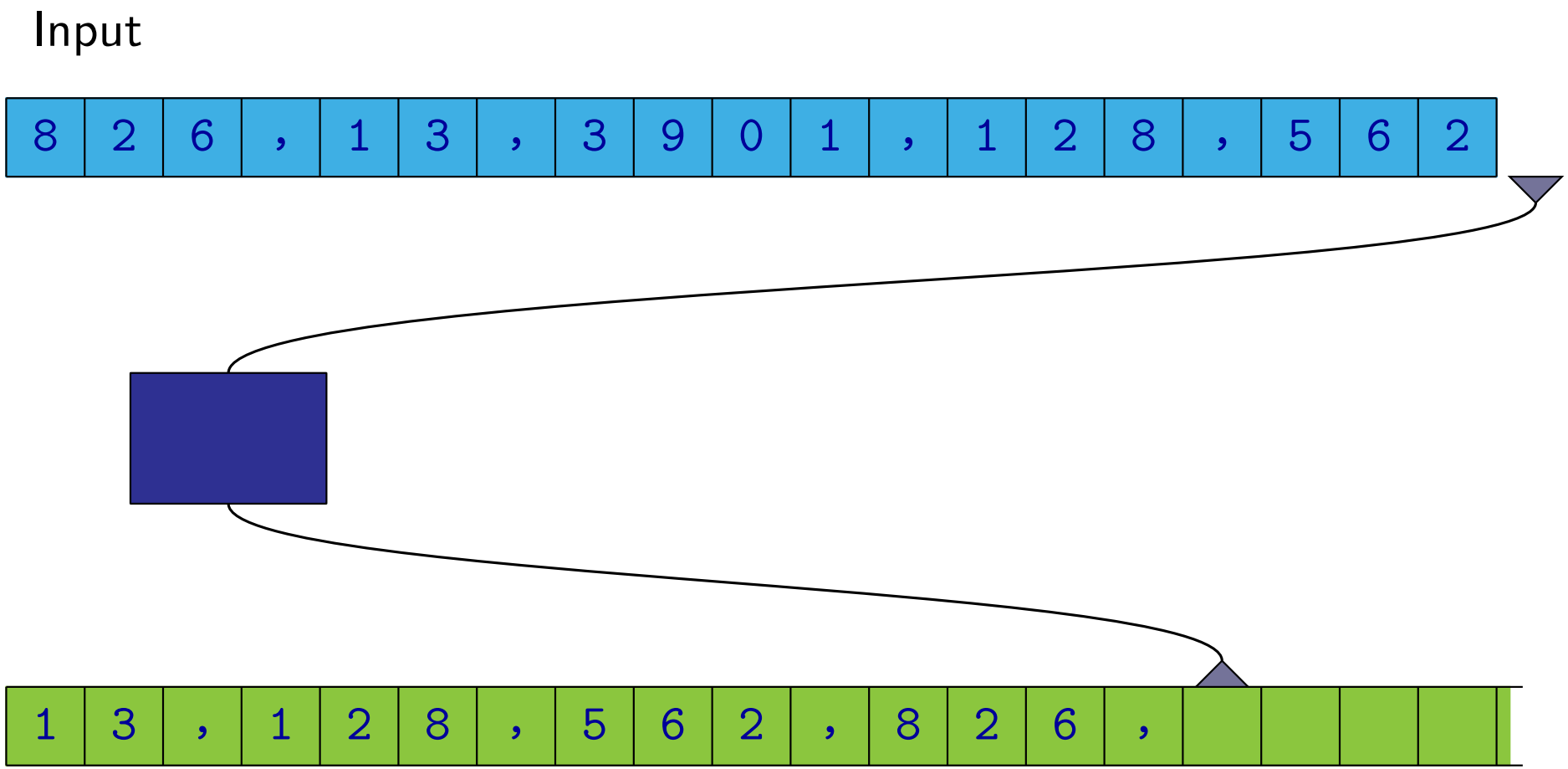
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



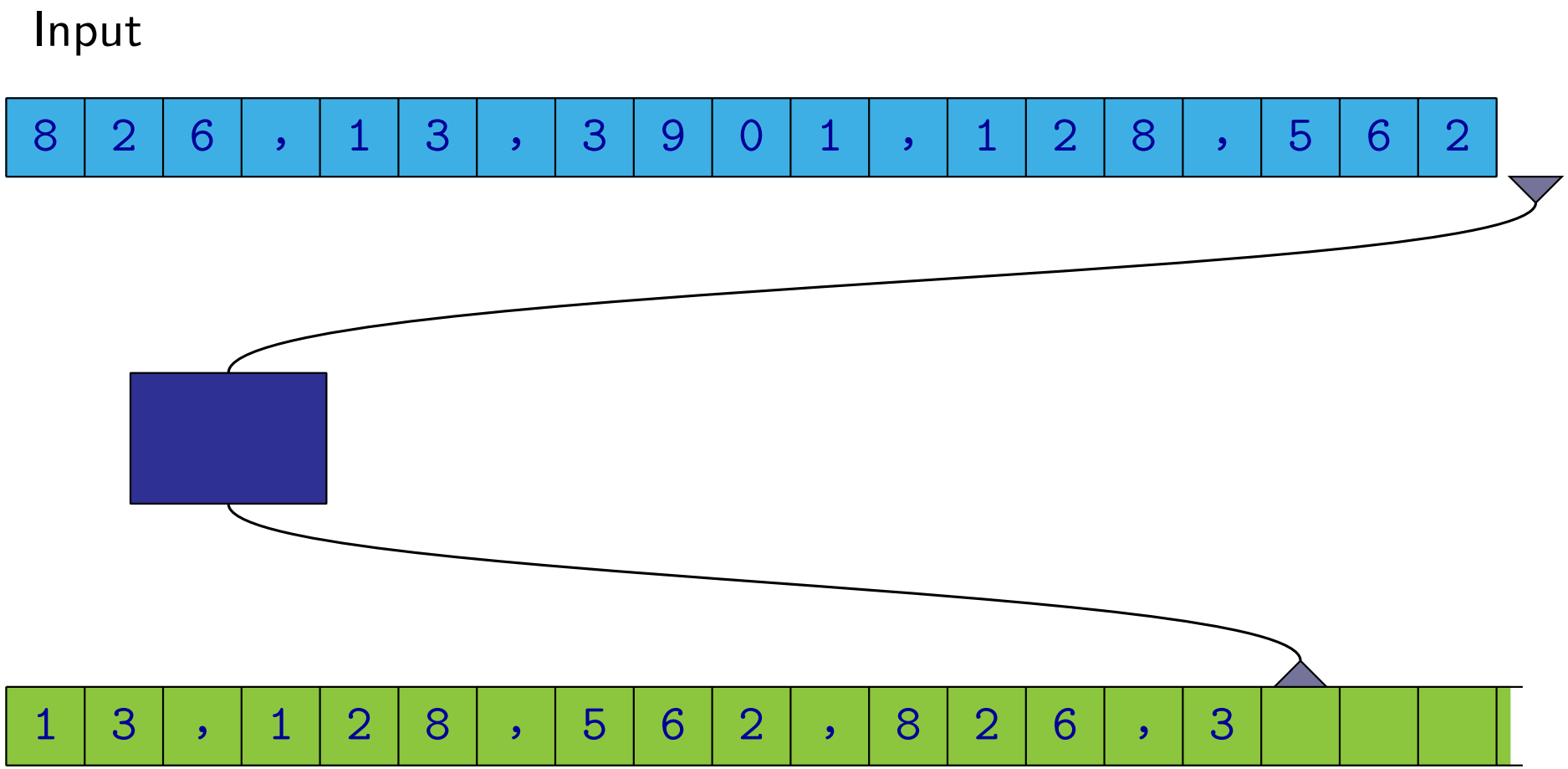
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



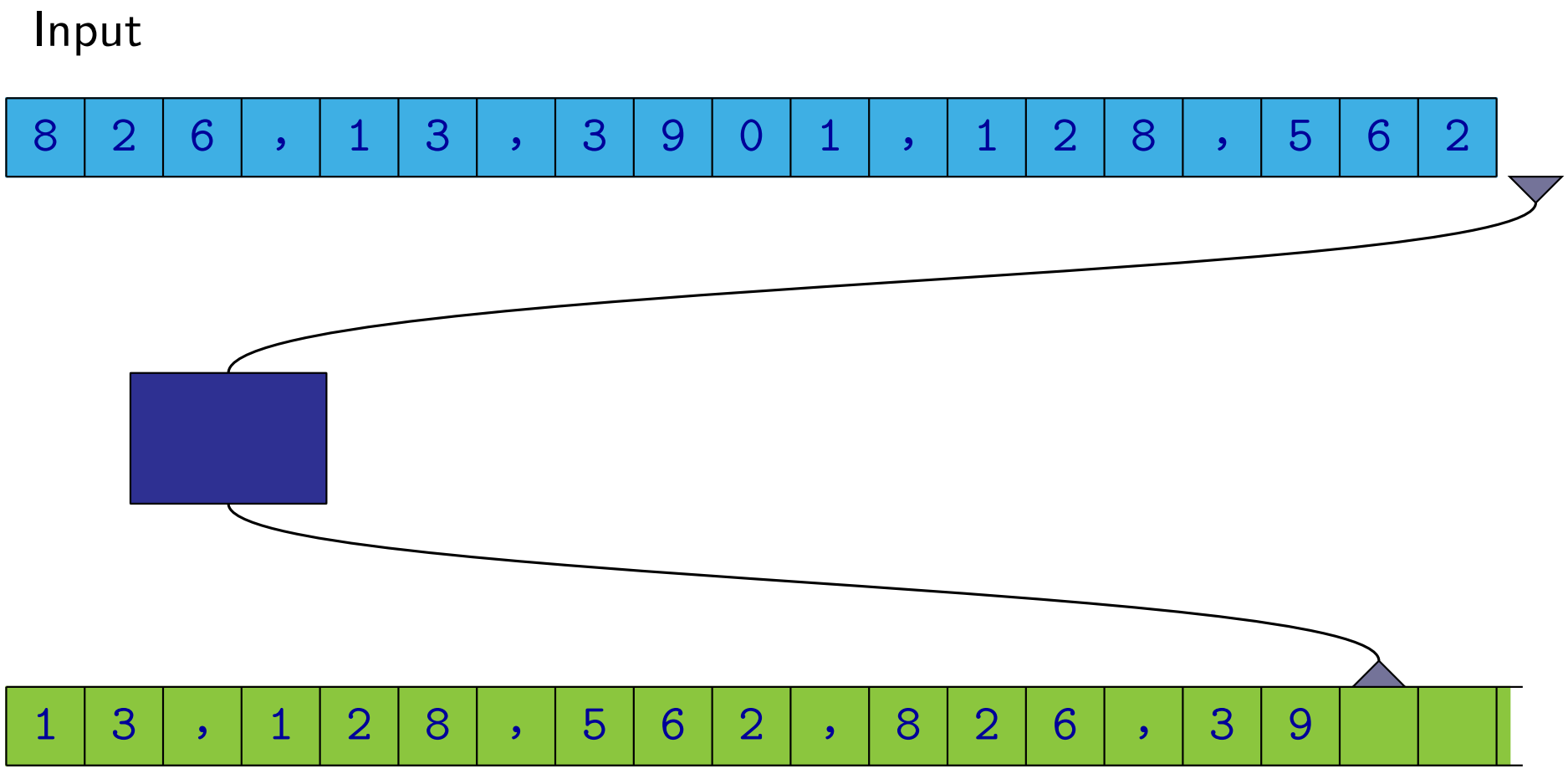
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



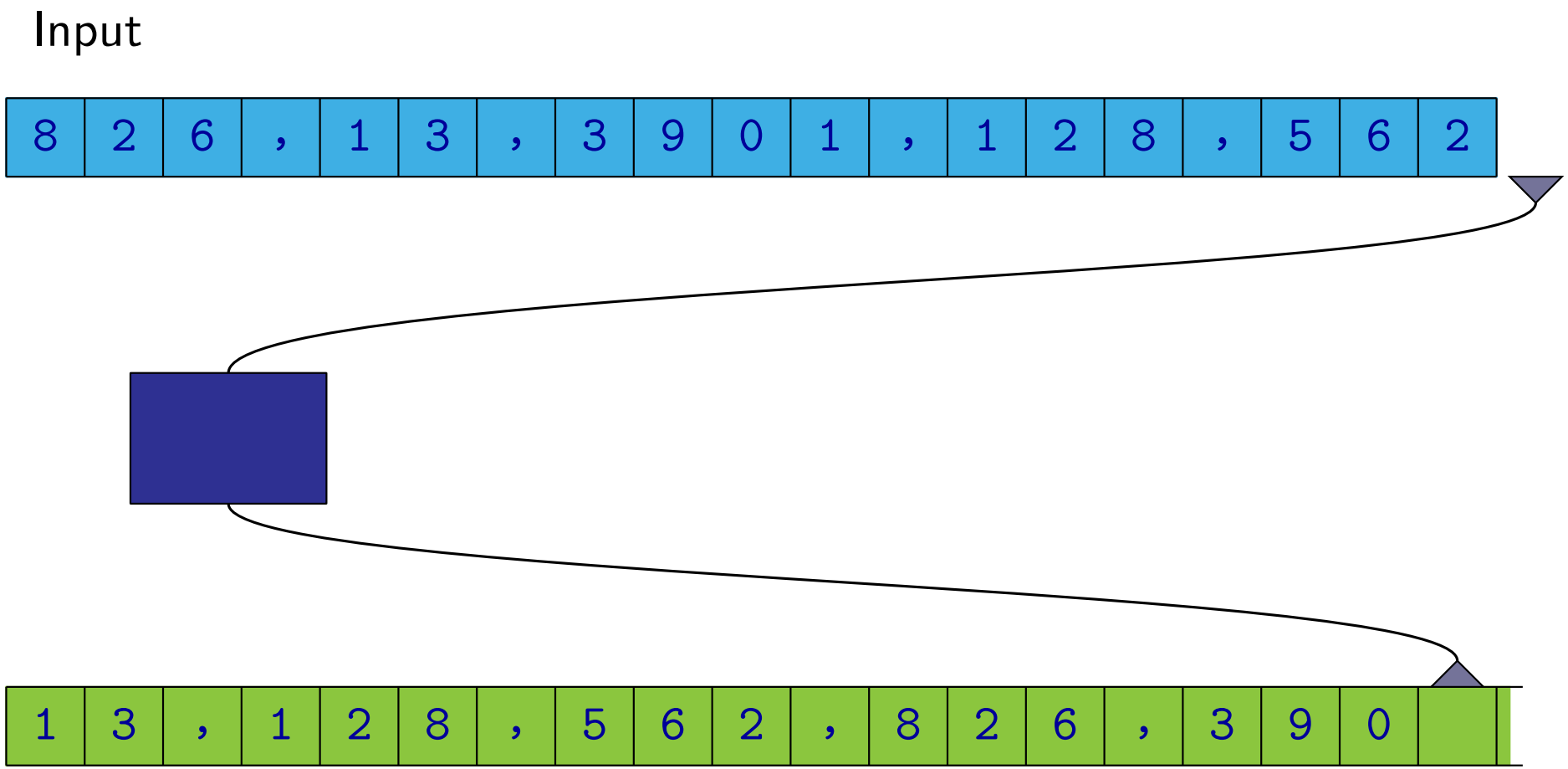
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



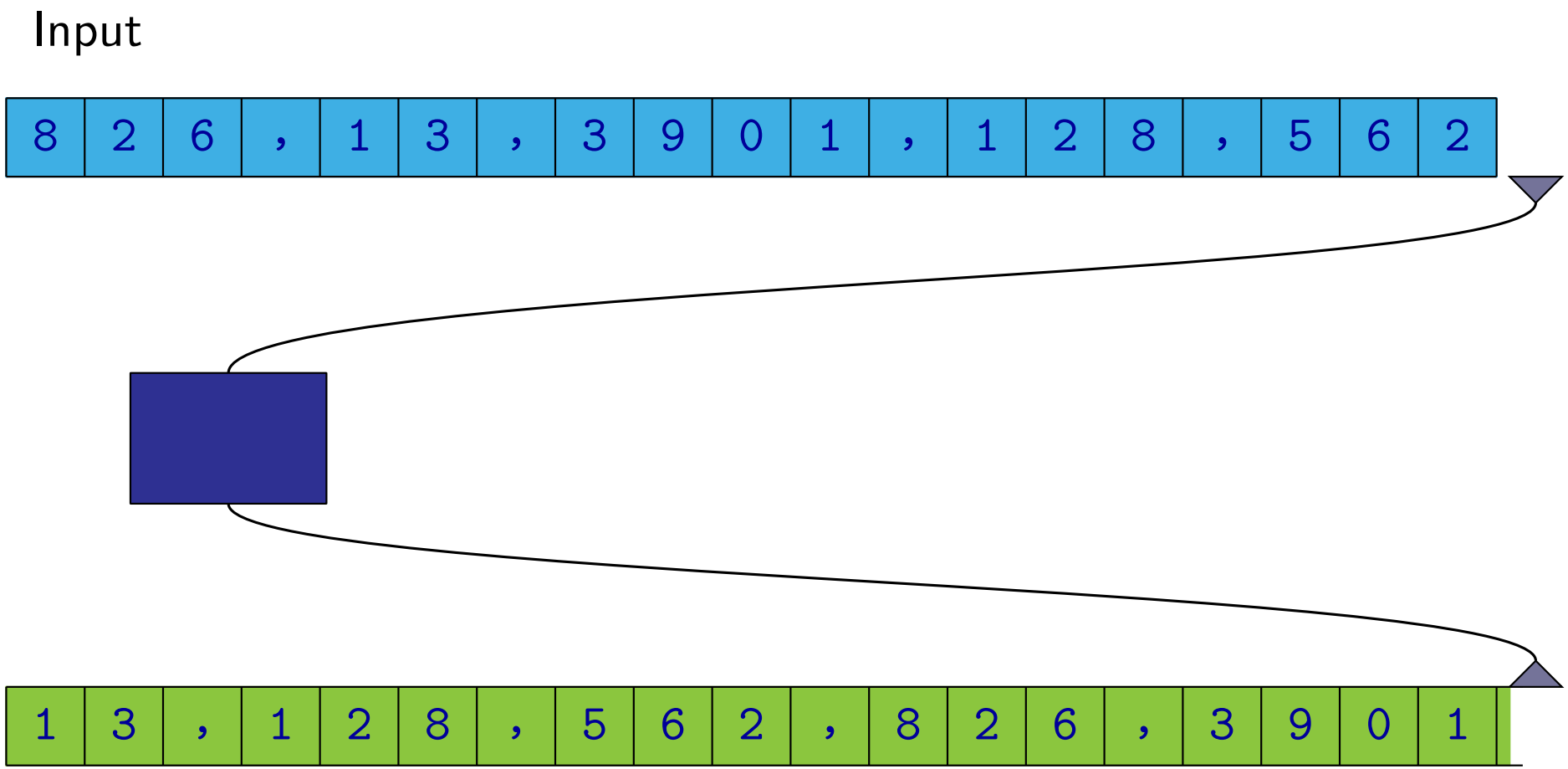
Input/Output Behaviour of an Algorithm

We can assume that the algorithm is executed on a certain type of machine.



Input/Output Behaviour of an Algorithm

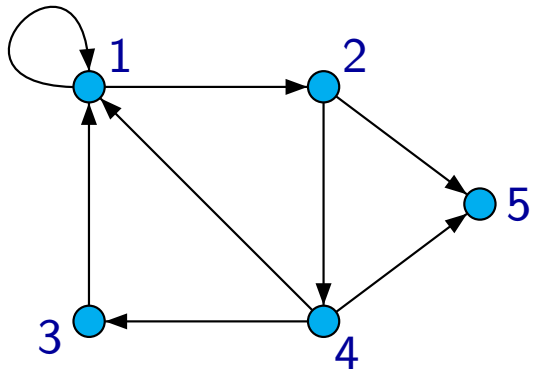
We can assume that the algorithm is executed on a certain type of machine.



Encoding of Input and Output

Example: If an input for a given problem is a graph, it could be represented as a pair of two lists — a list of nodes and a list of edges:

For example, the following graph



could be represented as a word

`(1,2,3,4,5),((1,2),(2,4),(4,3),(3,1),(1,1),(2,5),(4,5),(4,1))`

over alphabet $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ,, (,)\}$.

Encoding of Input and Output

We can restrict our attention to the case where both inputs and outputs are encoded as words over alphabet $\{0, 1\}$ (i.e., as sequences of bits).

Symbols of any other alphabet can be represented as sequences of bits.

Example: Alphabet $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}, \mathtt{g}\}$

$$
\begin{aligned}
\mathtt{a} &\leftrightarrow 001 \\
\mathtt{b} &\leftrightarrow 010 \\
\mathtt{c} &\leftrightarrow 011 \\
\mathtt{d} &\leftrightarrow 100 \\
\mathtt{e} &\leftrightarrow 101 \\
\mathtt{f} &\leftrightarrow 110 \\
\mathtt{g} &\leftrightarrow 111
\end{aligned}
$$

Word '$\mathtt{defb}$' can be represented as '100101110010'.

Encoding of Input and Output

Words over alphabet $\Sigma = \{0,1\}^*$ (i.e., sequences of bits) can be viewed as representations of numbers written in binary.

This means that one number can encode an arbitrarily long sequence of bits.

So the whole input, resp. output, can be represented by a single (usually very big) natural number.

So in this variant, we can assume that

$$In = Out = \mathbb{N},$$

where $\mathbb{N} = \{0,1,2,3,\ldots\}$ is the set of natural numbers.

Encoding of Input and Output

Other alternative is the variant where inputs and outputs are not encoded as words in some alpabet but rather as sequences of numbers.

It is sufficient to choose some encoding assigning number codes to symbols of the given alphabet.

Example: Alphabet $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}, \mathtt{g}\}$

$$
\begin{aligned}
\mathtt{a} &\leftrightarrow 1 \\
\mathtt{b} &\leftrightarrow 2 \\
\mathtt{c} &\leftrightarrow 3 \\
\mathtt{d} &\leftrightarrow 4 \\
\mathtt{e} &\leftrightarrow 5 \\
\mathtt{f} &\leftrightarrow 6 \\
\mathtt{g} &\leftrightarrow 7
\end{aligned}
$$

Word '$\mathtt{defb}$' can be represented as $[4, 5, 6, 2]$.

Other Examples of Problems

Problem "Primality"

Input: A natural number n.

Output: YES if n is a prime, NO otherwise.

Remark: A natural number n is a **prime** if it is greater than 1 and is divisible only by numbers 1 and n.

Few of the first primes: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, ...

Decision Problems

The problems, where the set of outputs is $\{\text{YES}, \text{NO}\}$ are called **decision problems**.

Decision problems are usually specified in such a way that instead of describing what the output is, a question is formulated.

Example:

Problem "Primality"

Input: A natural number n.

Question: Is n a prime?

Decision Problems

Decision problem

One possibility, how the notion of a **decision problem** can be defined formally, is to define it as a pair (In, T), where:

- In is the set of all inputs,
- $T \subseteq In$ is the set of all inputs, for which the answer is $\textsc{Yes}$.

Decision Problems and Languages

If we restrict to the cases where inputs are words over some alphabet Σ, then decision problems can be viewed as languages.

A language corresponding to a given decision problem is the set of those words from Σ^* that represent inputs for which the answer is YES.

I.e., the given language contains words:

- that represent instances where the answer is YES,

and does not contain words:

- that represent instance where answer is NO, or
- do not represent instances of the given problem.

Example: A language consisting of those words from $\{0,1\}^*$ that are binary representations of primes.

For example, $101 \in L$ but $110 \notin L$.

Another Example of a Decision Problem

SAT problem (boolean satisfiability problem)

Input: Boolean formula φ.

Question: Is φ satisfiable?

Remark: Formula φ is **satisfiable** if there exists a truth assignment ν, for which the formula is true, i.e., it holds $\nu(\varphi) = 1$.

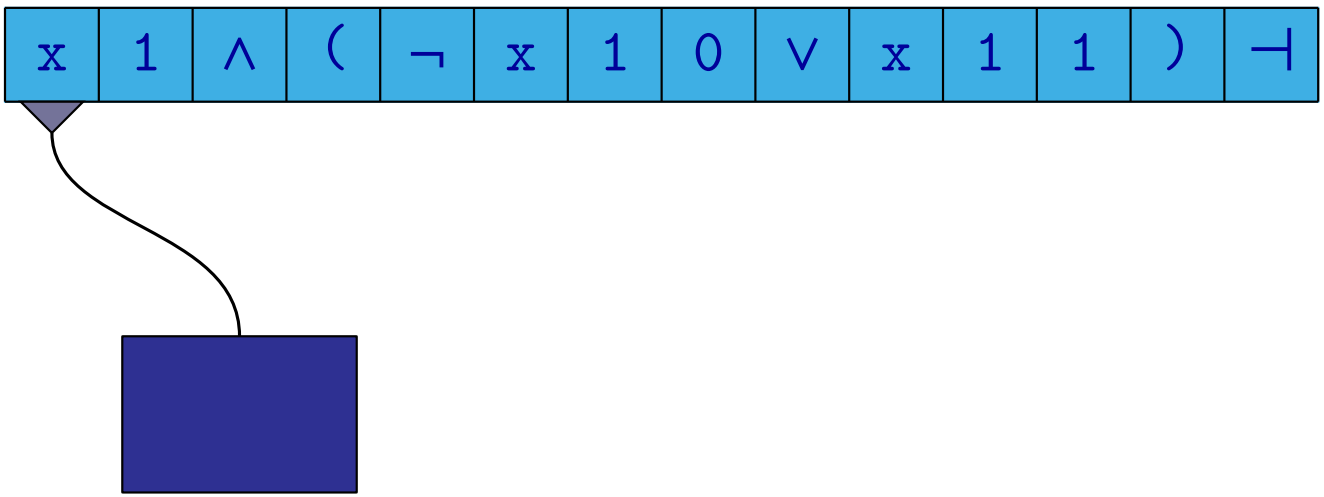
Example:

- Formula $\varphi_1 = x_1 \wedge (\neg x_2 \vee x_3)$ is satisfiable
  — e.g., for valuation ν where we have $\nu(\varphi_1) = 1$:
  $$x_1 \mapsto 1$$
  $$x_2 \mapsto 0$$
  $$x_3 \mapsto 1$$
- Formula $\varphi_2 = (x_1 \wedge \neg x_1) \vee (\neg x_2 \wedge x_3 \wedge x_2)$ is not satisfiable
  — for every valuation ν it holds that $\nu(\varphi_2) = 0$.

Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

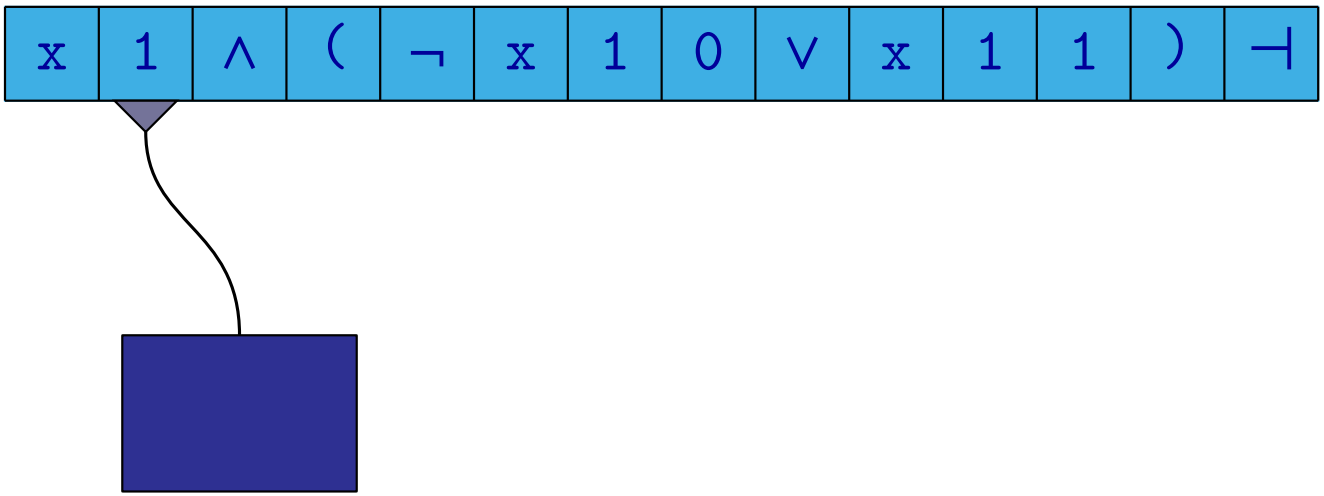
- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

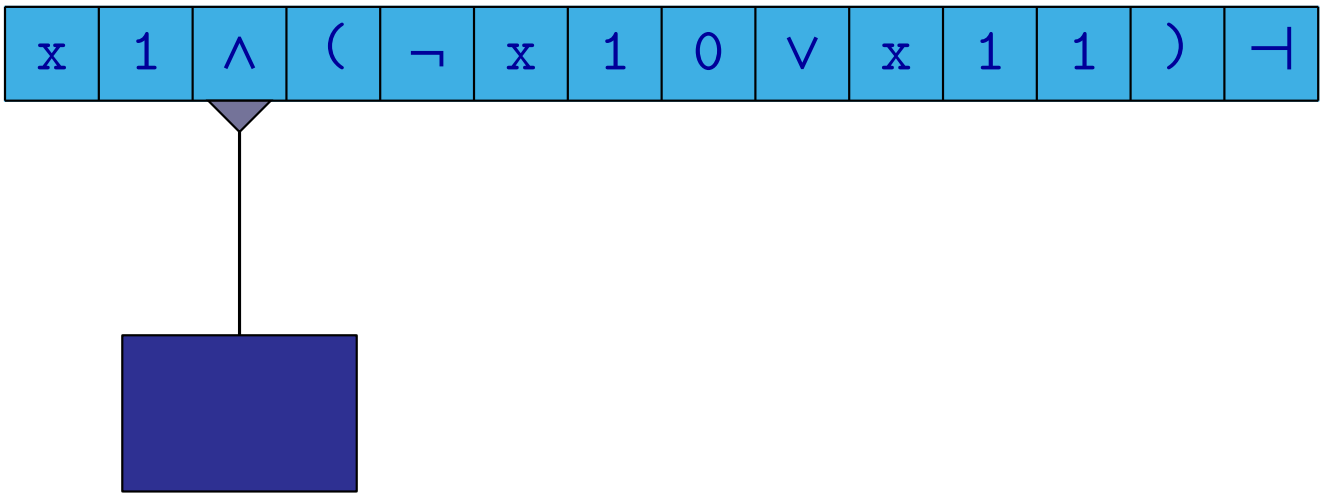
- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

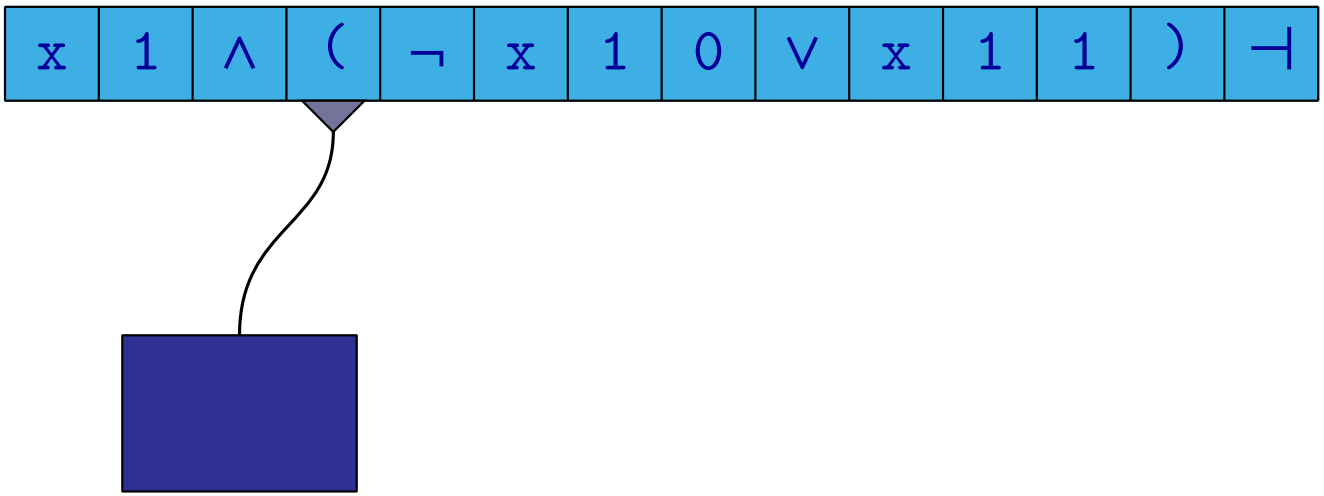
- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

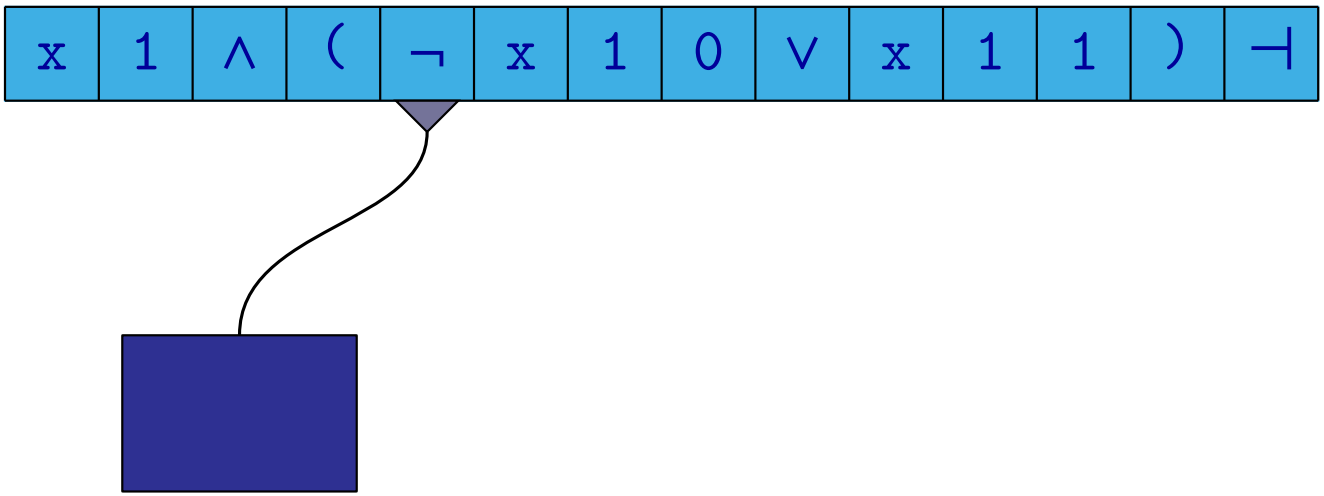
- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

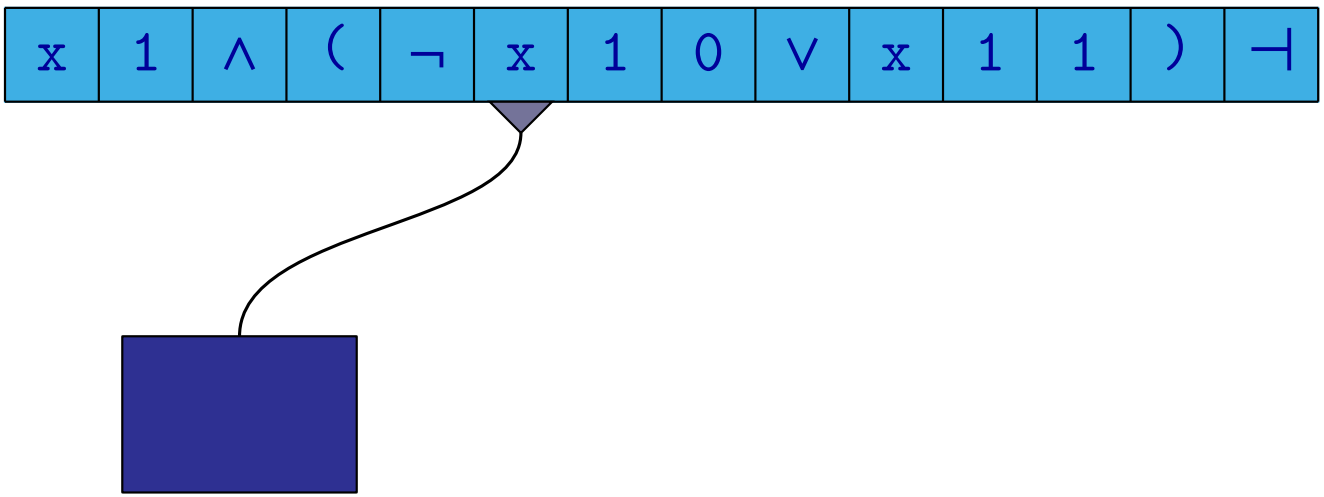
- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

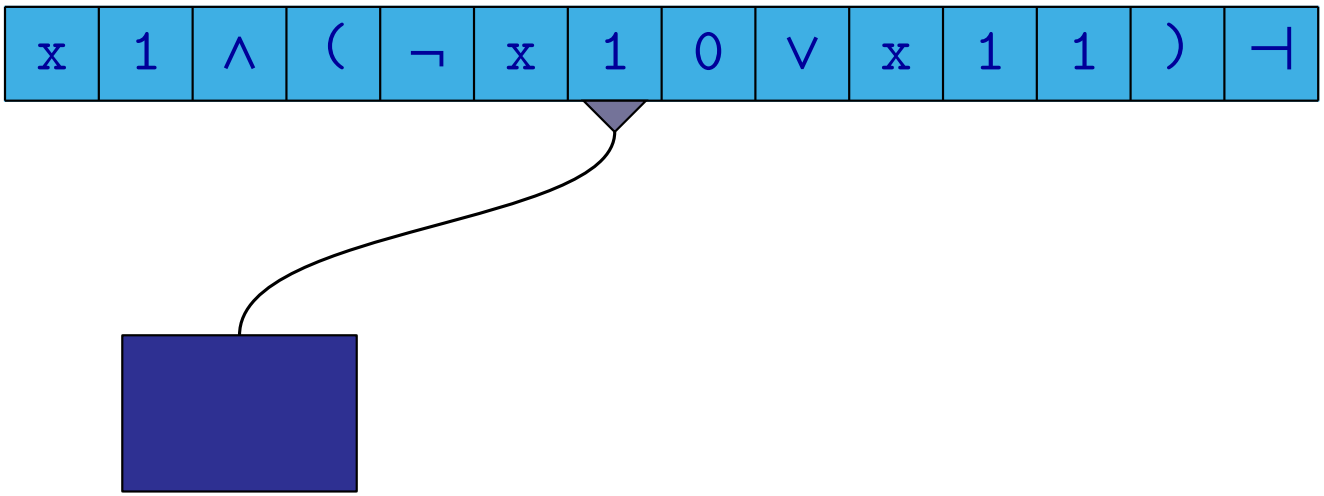
- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

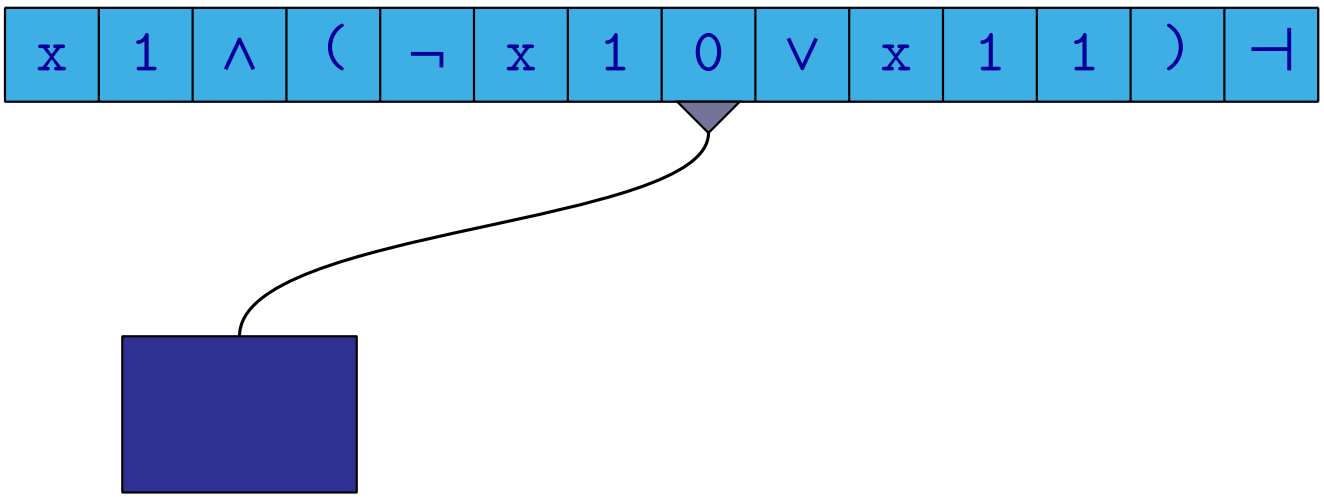
- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

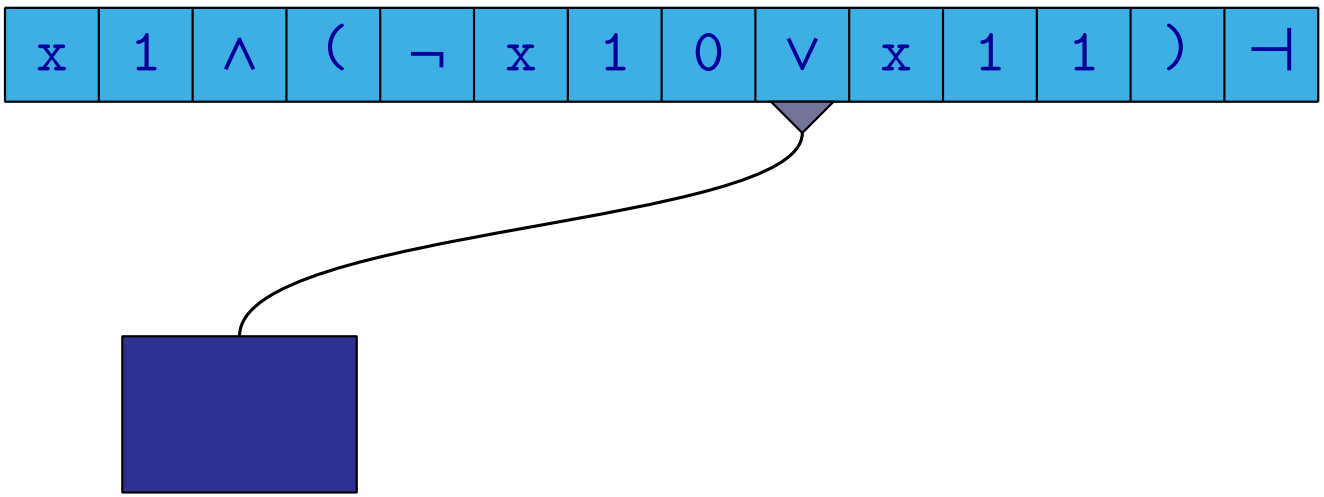
- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

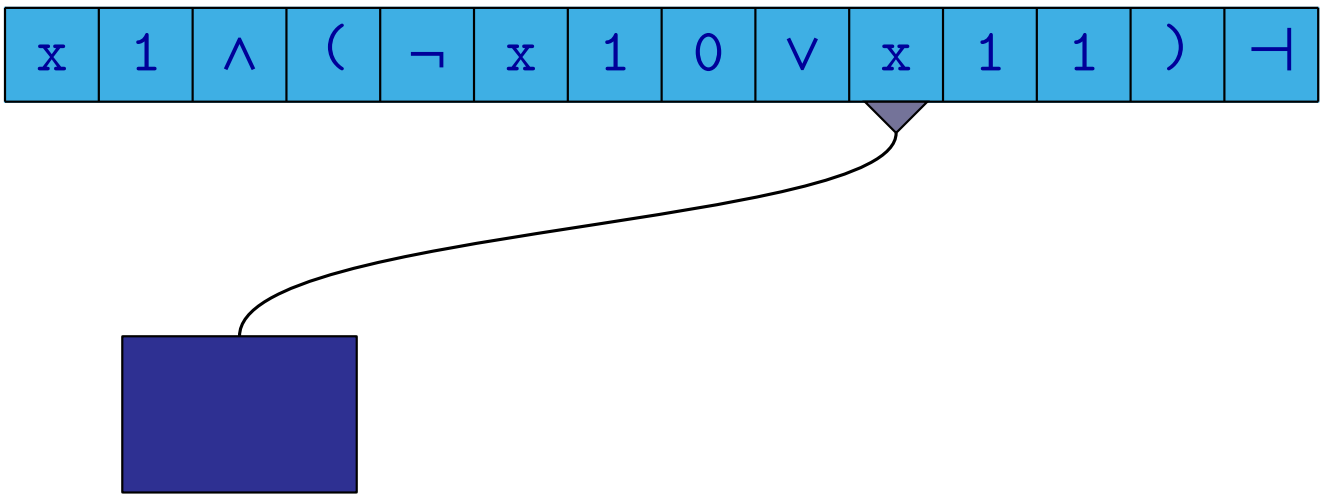
- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

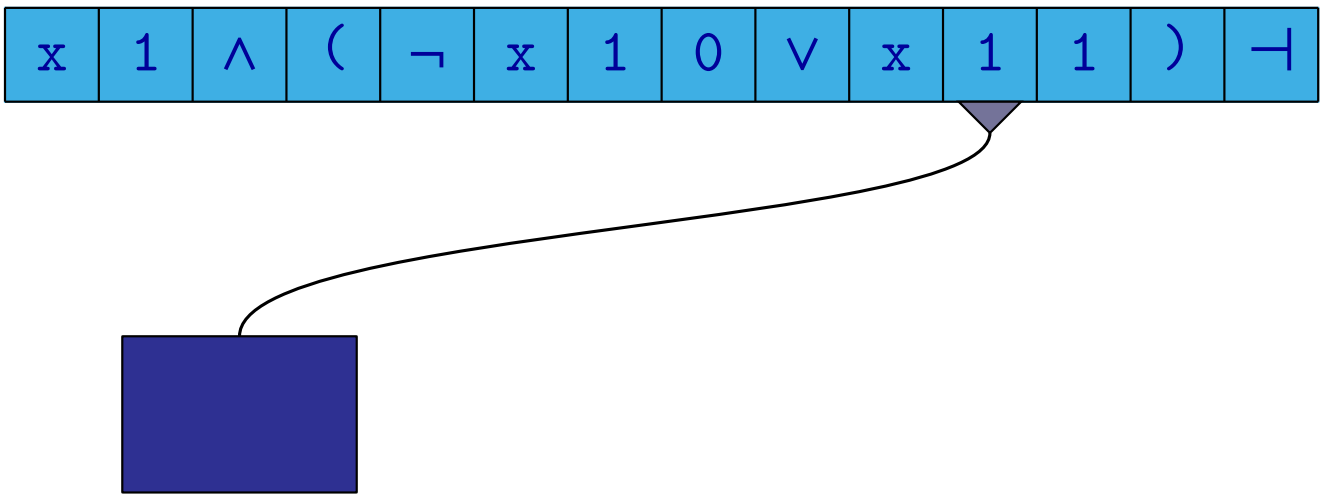
- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

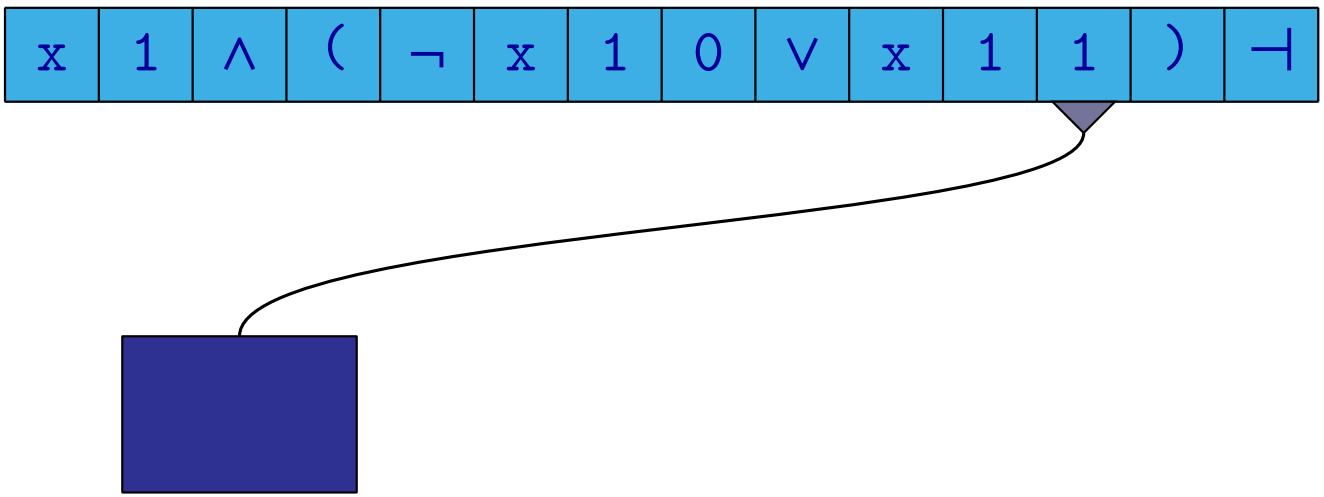
- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

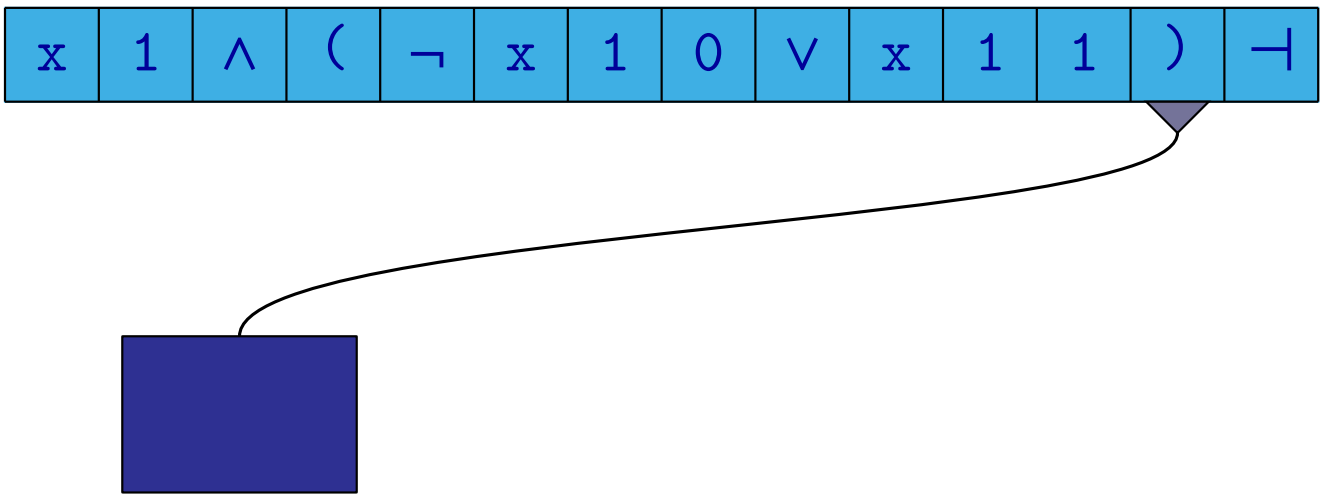
- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

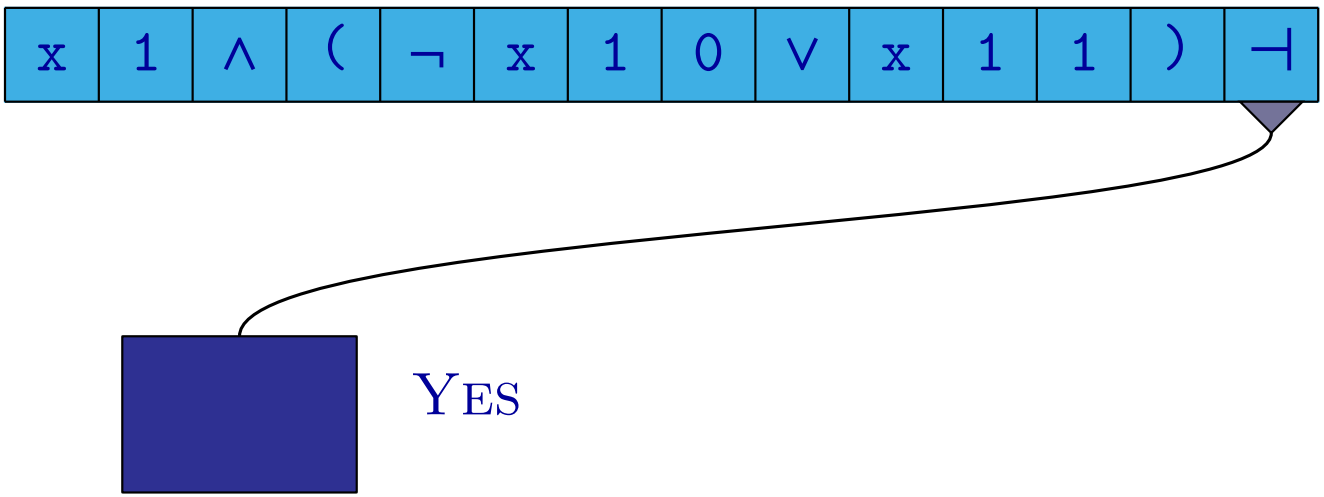
- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Input and Output of an Algorithm for a Decision Problem

Input and **output** of an algorithm solving **SAT problem**:

- boolean formulas are encoded using some reasonable encoding as words over some alphabet
- for words that represent satisfiable formulas it gives answer YES
- for words that represent unsatisfiable formulas it gives answer NO
- for words that do not represent well-formed formulas it gives answer NO



Algorithms

What Is an Algorithm?

Algorithm

An **algorithm** is a mechanical procedure consisting of some simple elementary steps that for any given **input** produces an **output**.

An algorithm can be described:

- in plain English
- by a pseudocode
- as a computer program in a programming language
- as a hardware circuit
- ...

Algorithms

Example: An algorithm described by **pseudocode**:

Algorithm: An algorithm for finding the maximal element in an array

```
FIND-MAX (A, n):
    k := 0
    for i := 1 to n − 1 do
        if A[i] > A[k] then
            k := i
    return A[k]
```

Algorithms

Algorithm

- processes an **input**
- generates an **output**

From the point of view of an analysis how a given algorithm works, it usually makes only a little difference if the algorithm:

- reads input data from some input device (e.g., from a file, from a keyboard, etc.)
- writes data to some output device (e.g., to a file, on a screen, etc.)

or

- reads input data from a memory (e.g., they are given to it as parameters)
- writes data somewhere to memory (e.g., it returns them as a return value)

Computation of an Algorithm

An algorithm is executed by a **machine** — it can be for example:

- real computer — executes instructions of a machine code
- virtual machine — executes instructions of a bytecode
- some idealized mathematical model of a computer
- ...

It depends on the type of the machine:

- what is the type of data, with which the machine works
- how this data are organized in its memory
- what operations can be performed on these data

The machine can be:

- specialized — executes only one algorithm
- universal — can execute arbitrary algorithm, given in a form of a **program**

Computation of an Algorithm

The machine performs **steps**.

It can store information to some kind of **working memory**.

During a computation, it goes through a sequence of **configurations**.

A **configuration** is a description of a global state of the machine in some particular moment during the computation.

It typically includes information about:

- the current instruction
- the content of its working memory

A computation:

- starts in an **initial configuration** — it depends on an input
- halts in a **final configuration** — an output can be extracted from it

Solutions of Problems

Solving a problem

An algorithm **solves** a given problem if every input of the given problem (i.e., for every input instance):

1. It halts after some finite number of steps.
2. It produces an output from the set of possible outputs that satisfies the conditions specified in the problem statement.

For one problem there can be many diffent algorithms that correctly solve the problem.

Remark: **correctness of an algorithm** — the algorithm solves the given problem

Algorithmically Solvable Problems

Let us assume we have a problem P.

If there is an algorithm solving the problem P then we say that the problem P is **algorithmically solvable**.

If P is a decision problem and there is an algorithm solving the problem P then we say that the problem P is **decidable (by an algorithm)**.

If we want to show that a problem P is algorithmically solvable, it is sufficient to show some algorithm solving it (and possibly show that the algorithm really solves the problem P).

Algorithmically Solvable Problems

For many problems it is immediately obvious that they are algorithmically solvable, as for example:

- Sorting
- Finding a shortest path in a graph
- Primality

where it is sufficient to test all possible solutions (in all these examples there are only finitely many possibilities that must be tested), although such trivial algorithm based on **brute force** solution is usually not very efficient.

On the other hand, there are many problems where it is not so clear.

- It can be a nontrivial task to find an algorithm solving the given problem and to show that it really solves it.
- It is possible that there is no algorithm solving the given problem.

Necessity of Formalization of the Notion of "Algorithm"

The above mentioned characterization of the notion of an algorithm was a little bit vague.

If we would like to prove for a problem that there is no algorithm solving the given problem, it would probably not be possible with such vague definition of the notion of an algorithm.

Intuitively, we understand what properties an algorithm should have:

- It should consist of simple steps that can be performed "mechanically" without understanding the problem.
- Objects, with which the algorithm manipulates, and also the performed operations should be finite.

A more precise and concrete definition of a notion of algorithm is also necessary when we want to talk about a computational complexity of an algorithm:

- For example, when we want to analyze the number of operations performed by an algorithm during a computation, we need a more precise definition of what exactly is considered as a single operation.

Models of Computation

We can consider different types of machines that are able to perform an algorithm.

There can be many different kinds of differences between these types of machines:

- what types of instructions they can execute
- what types of dates they can store in their memory and this memory is organised
- ...

Different kinds of such machines are called **models of computation**.

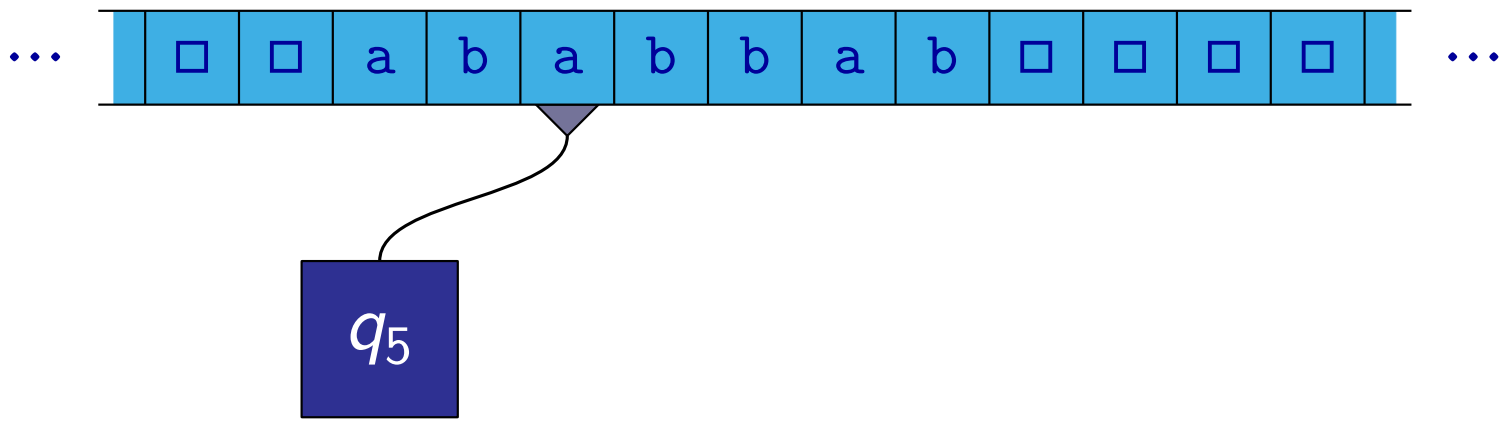
Examples of models of computation:

- Turing machines
- Random Access Machines (RAMs)

Turing Machine

Turing machine — a device similar to a finite automaton with the following differences:

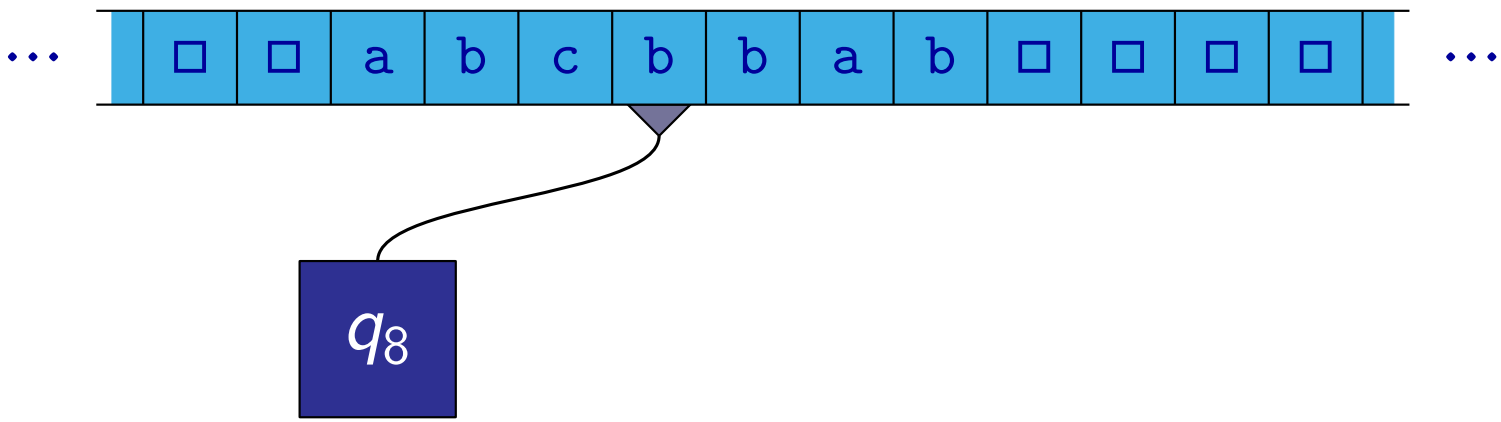
- the head can move in both directions
- it is possible to write on a current position of the head
- the tape is infinite



Turing Machine

Turing machine — a device similar to a finite automaton with the following differences:

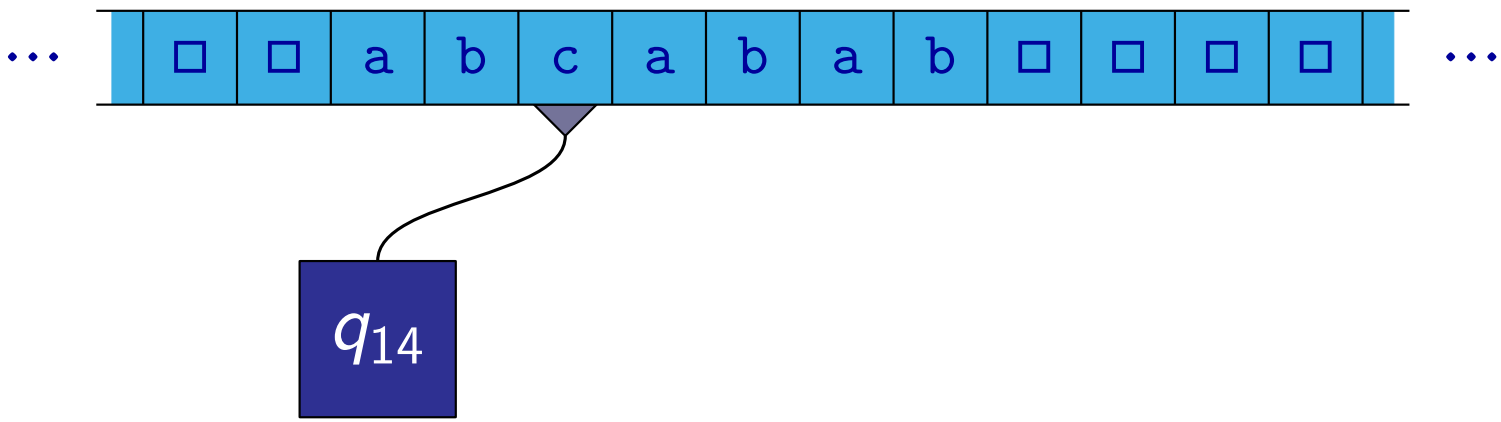
- the head can move in both directions
- it is possible to write on a current position of the head
- the tape is infinite



Turing Machine

Turing machine — a device similar to a finite automaton with the following differences:

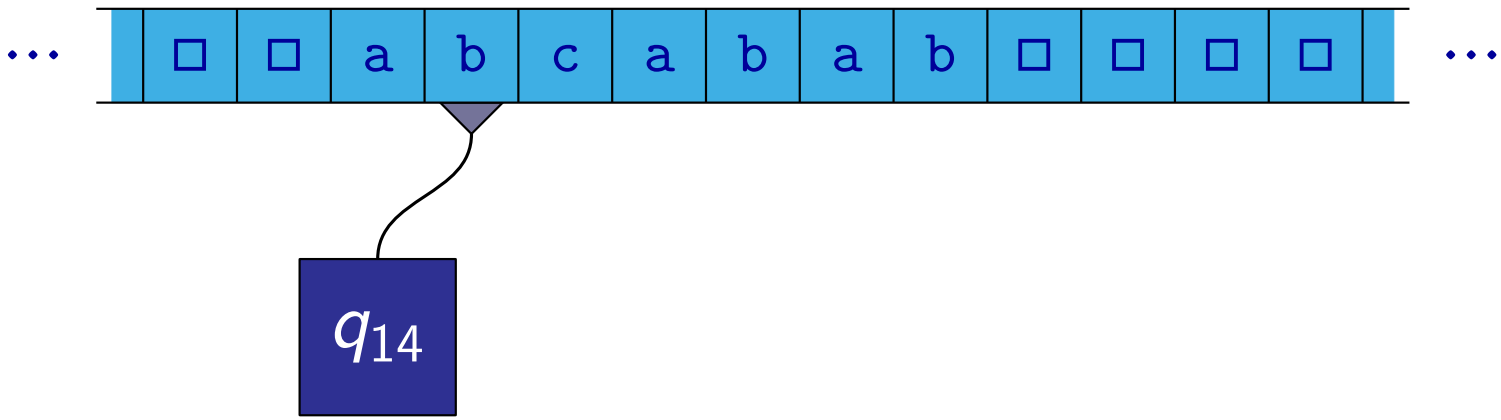
- the head can move in both directions
- it is possible to write on a current position of the head
- the tape is infinite



Turing Machine

Turing machine — a device similar to a finite automaton with the following differences:

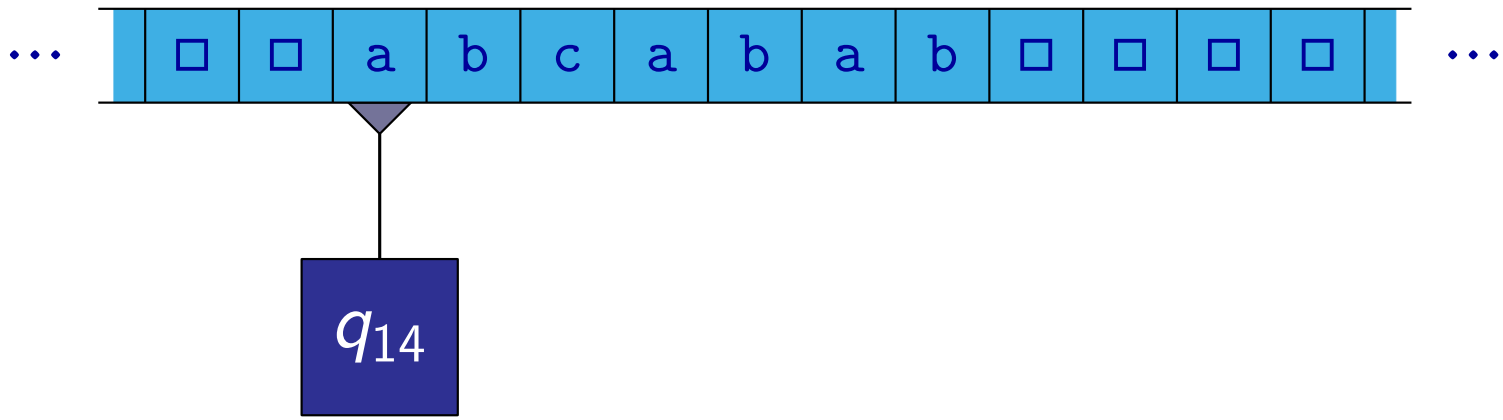
- the head can move in both directions
- it is possible to write on a current position of the head
- the tape is infinite



Turing Machine

Turing machine — a device similar to a finite automaton with the following differences:

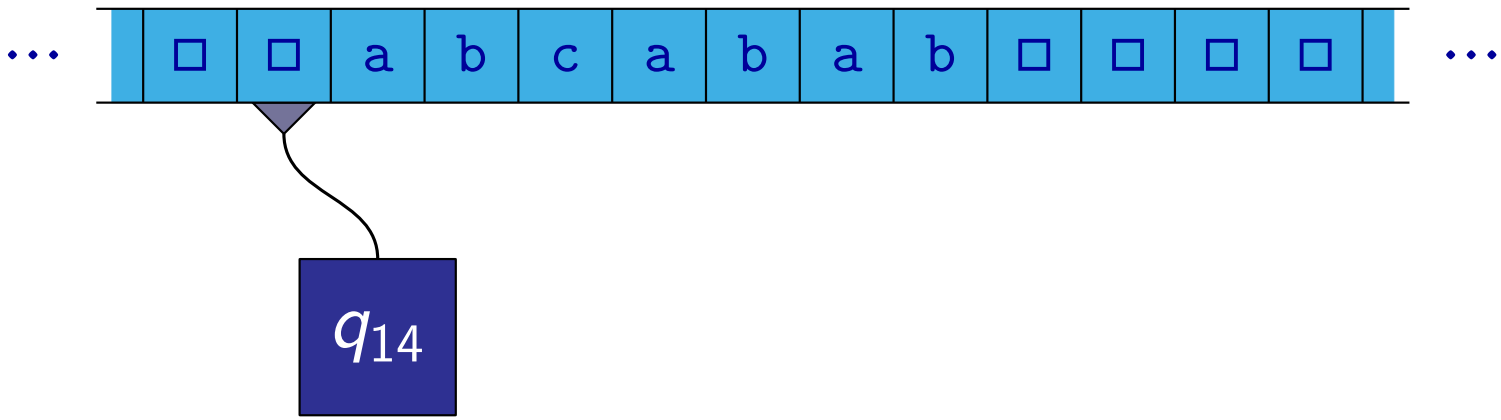
- the head can move in both directions
- it is possible to write on a current position of the head
- the tape is infinite



Turing Machine

Turing machine — a device similar to a finite automaton with the following differences:

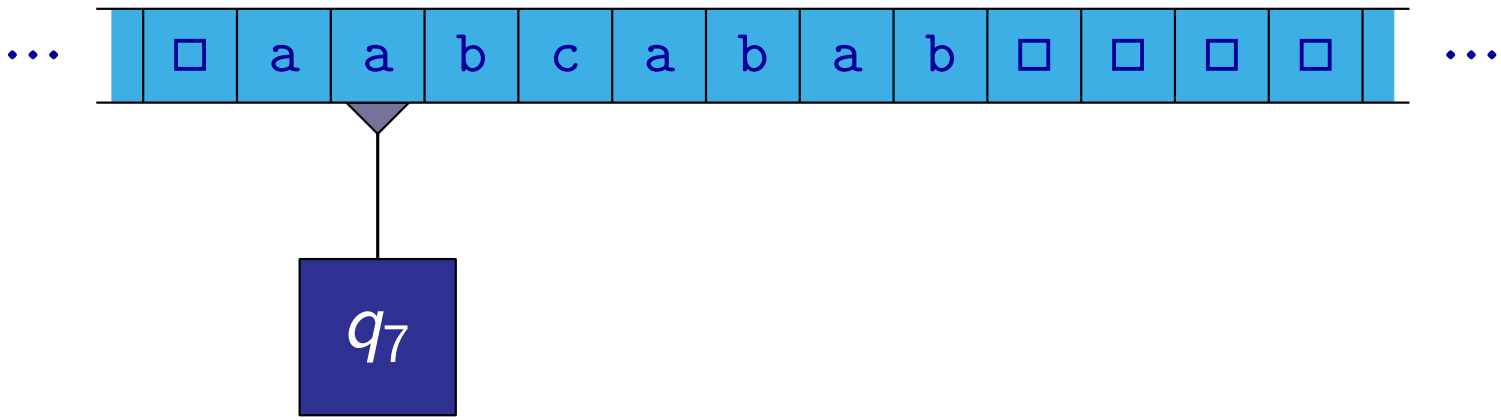
- the head can move in both directions
- it is possible to write on a current position of the head
- the tape is infinite



Turing Machine

Turing machine — a device similar to a finite automaton with the following differences:

- the head can move in both directions
- it is possible to write on a current position of the head
- the tape is infinite



Turing Machine

Alan M. Turing, "On Computable Numbers, with an application to the Entscheidungsproblem", *Proceedings of the London Mathematical Society*, 42 (1936), pp. 230–265, Erratum: Ibid., 43 (1937), pp. 544–546.

This article introduced several fundamental new ideas:

- A Turing machines proposed as a formalization of an intuitive notion of *algorithm*.
- the idea of a **universal Turing machine** — i.e., a particular single machine that is able to simulate all other Turing machines (i.e., a single hardware that can execute arbitrary algorithm represented as a program)
- a proof that there exist well defined algorithmic problems, for which it can be shown that there can not exist any algorithm solving them

Church-Turing Thesis

Church-Turing thesis

Every algorithm can be implemented as a Turing machine.

It is not a theorem that can be proved in a mathematical sense – it is not formally defined what an algorithm is.

It is one of possible approaches how an informal notion of **algorithm** can be represented formally.

The thesis was formulated in 1930s independently by Alan Turing and Alonzo Church.

Remark: Alonzo Church used **lambda calculus** in his formulation instead of Turing machines.

Church-Turing Thesis

Examples of other mathematical formalisms modelling the notion of an algorithm:

- Random Access Machines
- Minsky machines
- Lambda calculus
- Recursive functions
- ...

We can also mention:

- An arbitrary (general purpose) programming language (for example C, Java, Python, Lisp, Haskell, Prolog, etc.).

All these models are equivalent with respect to algorithms that can be implemented by them.

Turing-complete Models of Computation

Such languages (resp. machines), which are general enough, so that programs written in any other programming language can be translated to them, are called **Turing-complete**.

It may not be immediately obvious that Church-Turing theses really holds, i.e., that **every** algorithm can be implemented as a Turing machine (or some other Turing-complete model).

It will be demonstrated that an algorithm represented as a pseudocode (or written in some high-level programming language) can be in several steps translated to a form represented by some other models of computations:

- control-flow graph
- RAM
- multitape Turing machine
- single tape Turing machine

Control-flow graphs

Algorithm Described Using Pseudocode

An example of an algorithm described by a **pseudocode**:

Algorithm: An algorithm for finding the maximal element in an array

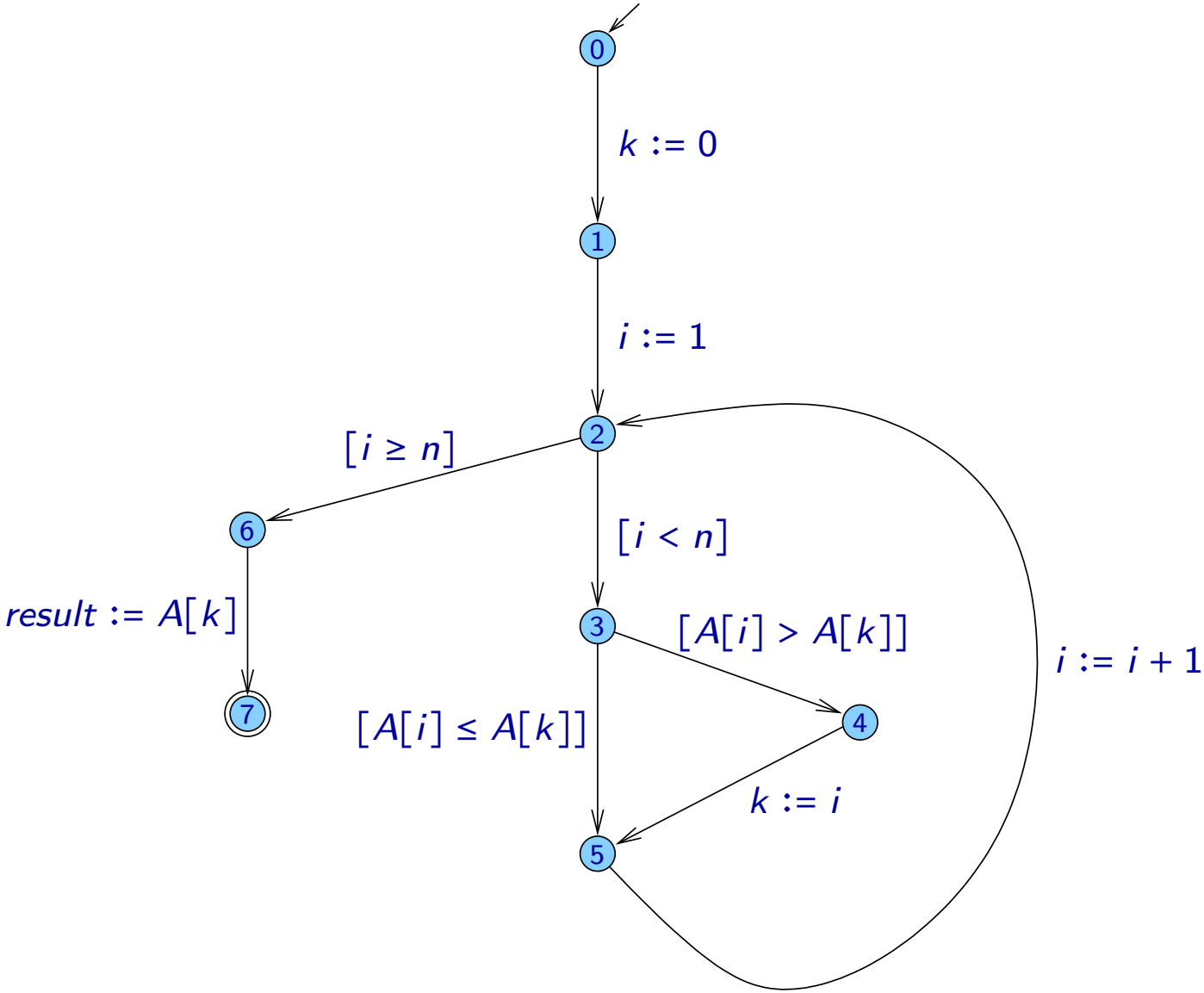
```
FIND-MAX (A, n):
    k := 0
    for i := 1 to n − 1 do
        if A[i] > A[k] then
            k := i
    return A[k]
```

Control Flow

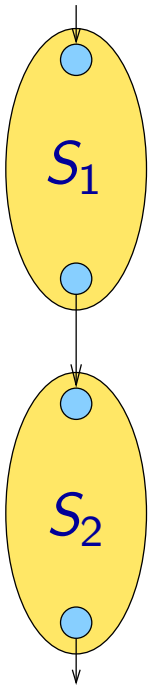
Instructions can be roughly devided into two groups:

- instructions working directly with data:
  - assignment
  - evaluation of values of expressions in conditions
  - reading input, writing output
  - ...
- instructions affecting the **control flow** — they determine, which instructions will be executed, in what order, etc.:
  - branching (if, switch, ...)
  - cycles (while, do .. while, for, ...)
  - organisation of instructions into blocks
  - returns from subprograms (return, ...)
  - ...

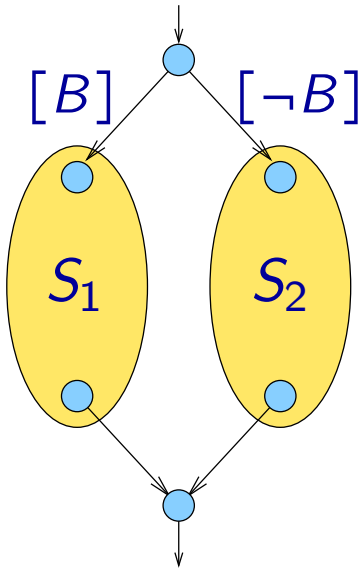
Control Flow Graph



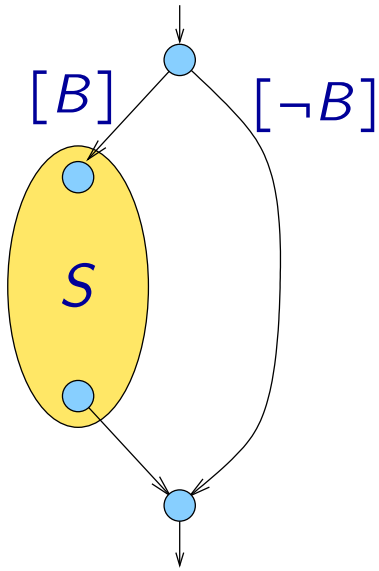
Some Basic Constructions of Structured Programming



$S_1; S_2$

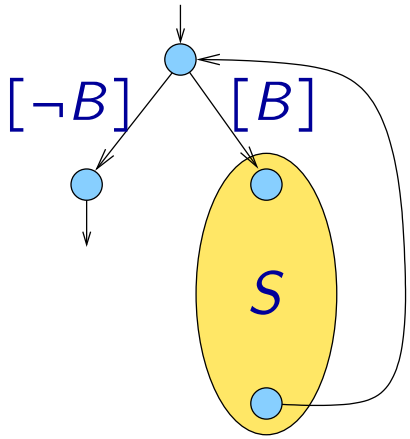


if B **then** S_1 **else** S_2

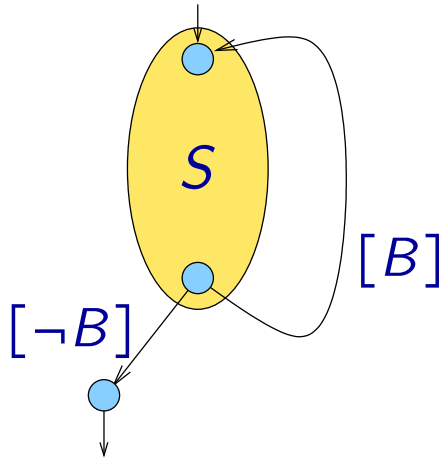


if B **then** S

Some Basic Constructions of Structured Programming

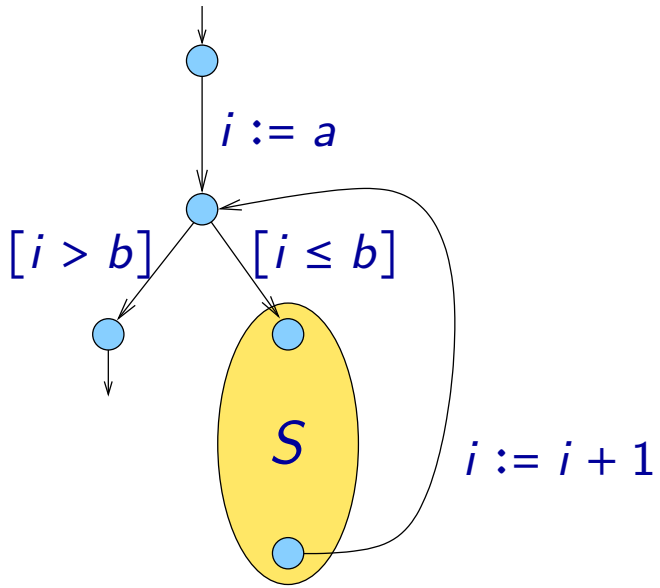


while B **do** S



do S **while** B

Some Basic Constructions of Structured Programming



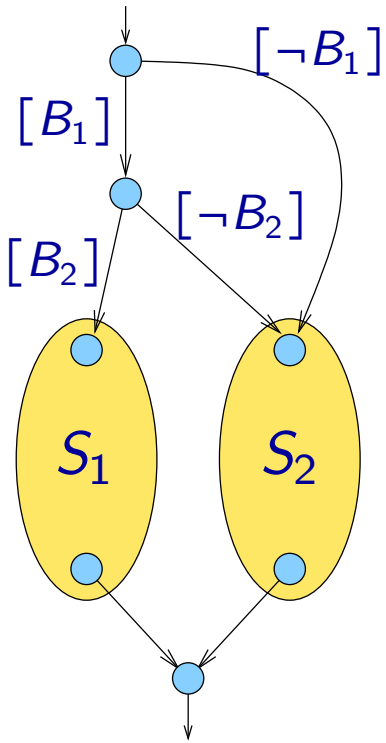
$i := a$
while $i \leq b$ **do**
  S
  $i := i + 1$

for $i := a$ **to** b **do** S

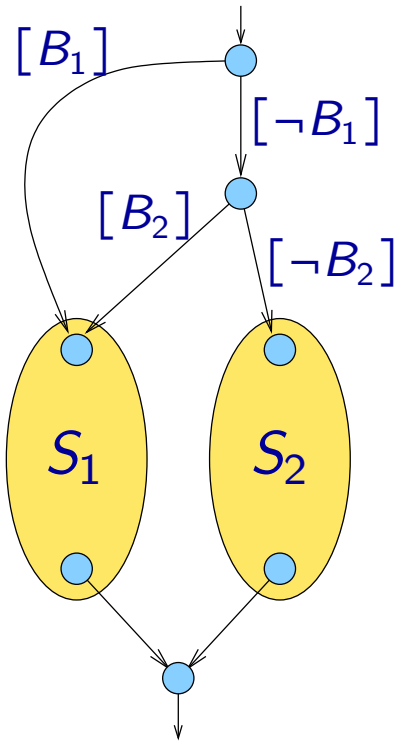
Some Basic Constructions of Structured Programming

Short-circuit evaluation of compound conditions, e.g.:

while $i < n$ **and** $A[i] > x$ **do** ...



if B_1 **and** B_2 **then** S_1 **else** S_2



if B_1 **or** B_2 **then** S_1 **else** S_2

Control-flow Realized by GOTO

- **goto** ℓ — **unconditional jump**
- **if** B **then goto** ℓ — **conditional jump**

Example:

0: $k := 0$
1: $i := 1$
2: **goto** *6*
3: **if** $A[i] \leq A[k]$ **then goto** *5*
4: $k := i$
5: $i := i + 1$
6: **if** $i < n$ **then goto** *3*
7: **return** $A[k]$

Control-flow Realized by GOTO

- **goto** ℓ — **unconditional jump**
- **if** B **then goto** ℓ — **conditional jump**

Example:

$$
\begin{array}{ll}
\mathit{start}: & k := 0 \\
& i := 1 \\
& \textbf{goto } \mathit{L3} \\
\mathit{L1}: & \textbf{if } A[i] \leq A[k] \textbf{ then goto } \mathit{L2} \\
& k := i \\
\mathit{L2}: & i := i + 1 \\
\mathit{L3}: & \textbf{if } i < n \textbf{ then goto } \mathit{L1} \\
& \textbf{return } A[k]
\end{array}
$$

Evaluation of Complicated Expressions

Evaluation of a complicated expression such as

$$A[i+s] := (B[3*j+1]+x)*y+8$$

can be replaced by a sequence of simpler instructions on the lower level, such as

$$\begin{aligned}
t_1 &:= i+s \\
t_2 &:= 3*j \\
t_2 &:= t_2+1 \\
t_3 &:= B[t_2] \\
t_3 &:= t_3+x \\
t_3 &:= t_3*y \\
t_3 &:= t_3+8 \\
A[t_1] &:= t_3
\end{aligned}$$

Computation of an Algorithm

Configuration — the description of the global state of the machine in some particular step during a computation

Example: A configuration of the form

$$(q, mem)$$

where

- q — the current control state
- mem — the current content of memory of the machine — the values assigned currently to variables.

An example of a content of memory mem:

$$\langle A\colon [3,8,1,3,6],\quad n\colon 5,\quad i\colon 1,\quad k\colon 0,\quad result\colon ?\rangle$$

Computation of an Algorithm

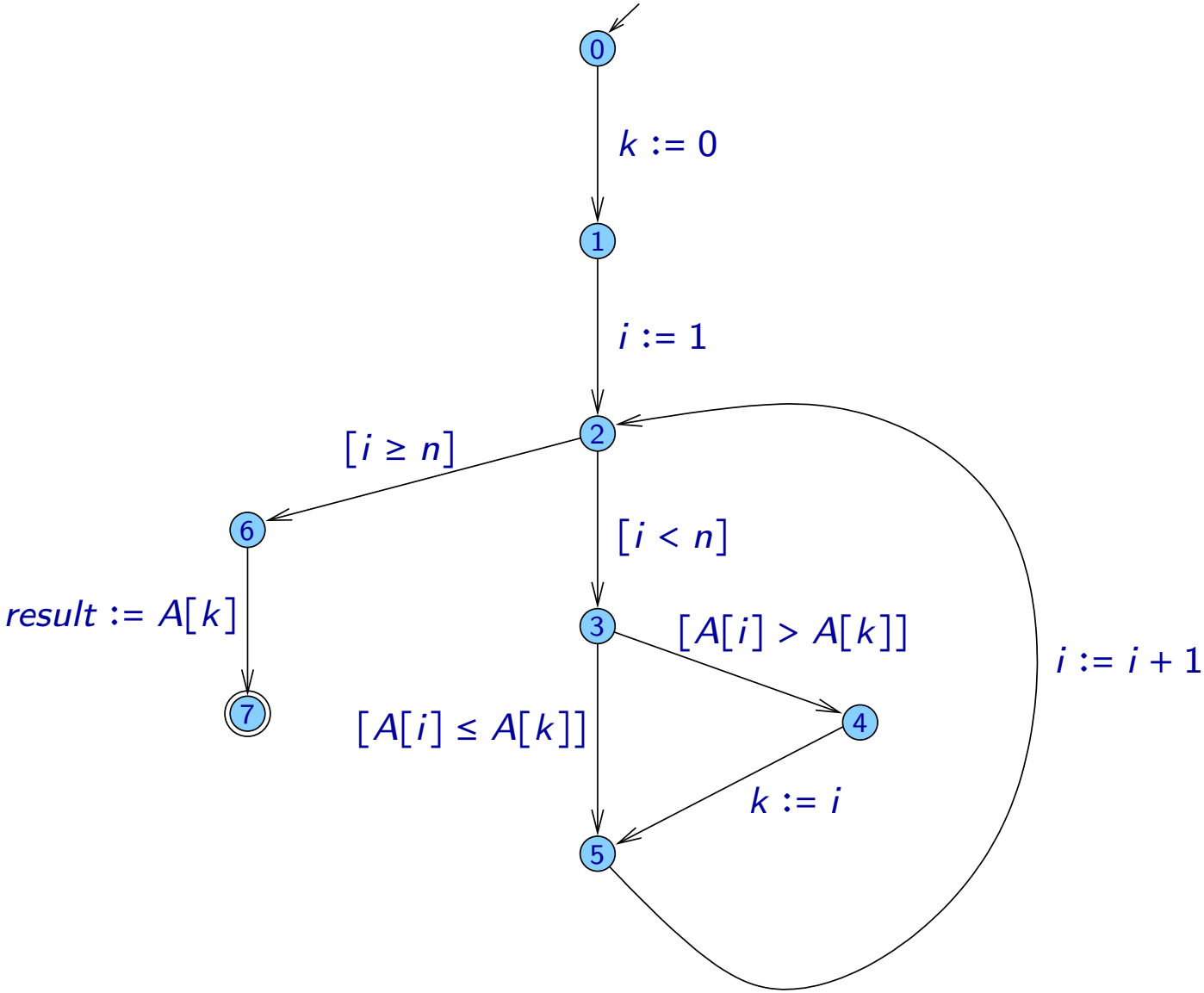
An example of a configuration:

$$(2,\ \langle A\colon [3,8,1,3,6],\ \ n\colon 5,\ \ i\colon 1,\ \ k\colon 0,\ \ \mathit{result}\colon ?\rangle)$$

A **computation** of a machine $\mathcal{M}$ executing an algorithm Alg, where it processes an input w, in a sequence of configurations.

- It starts in an **initial configuration**.
- In every step, the machine goes from one configuration to another.
- The computation ends in a **final configuration**.

Computation of an Algorithm



Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3, 8, 1, 3, 6], \quad n\text{: } 5, \quad i\text{: } ?, \quad k\text{: } ?, \quad \mathit{result}\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6], \quad n\text{: } 5, \quad i\text{: } ?, \quad k\text{: } ?, \quad \mathit{result}\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6], \quad n\text{: } 5, \quad i\text{: } ?, \quad k\text{: } 0, \quad \mathit{result}\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6], \quad n\text{: } 5, \quad i\text{: } ?, \quad k\text{: } ?, \quad \mathit{result}\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6], \quad n\text{: } 5, \quad i\text{: } ?, \quad k\text{: } 0, \quad \mathit{result}\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6], \quad n\text{: } 5, \quad i\text{: } 1, \quad k\text{: } 0, \quad \mathit{result}\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } ?, \ k\text{: } ?, \ result\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } ?, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } ?, \ k\text{: } ?, \ \mathit{result}\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } ?, \ k\text{: } 0, \ \mathit{result}\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ \mathit{result}\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ \mathit{result}\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ \mathit{result}\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } ?, \ k\text{: } ?, \ result\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } ?, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_5: $(5, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 1, \ result\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } ?,\ \mathit{result}\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } 0,\ \mathit{result}\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ \mathit{result}\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ \mathit{result}\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ \mathit{result}\text{: } ?\rangle)$
α_5: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 1,\ \mathit{result}\text{: } ?\rangle)$
α_6: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ \mathit{result}\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } ?,\ \mathit{result}\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } 0,\ \mathit{result}\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ \mathit{result}\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ \mathit{result}\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ \mathit{result}\text{: } ?\rangle)$
α_5: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 1,\ \mathit{result}\text{: } ?\rangle)$
α_6: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ \mathit{result}\text{: } ?\rangle)$
α_7: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ \mathit{result}\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } ?,\ result\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_5: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_6: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_7: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_8: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } ?,\ result\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_5: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_6: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_7: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_8: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_9: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 3,\ k\text{: } 1,\ result\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6], \; n\text{: } 5, \; i\text{: } ?, \; k\text{: } ?, \; result\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6], \; n\text{: } 5, \; i\text{: } ?, \; k\text{: } 0, \; result\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6], \; n\text{: } 5, \; i\text{: } 1, \; k\text{: } 0, \; result\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6], \; n\text{: } 5, \; i\text{: } 1, \; k\text{: } 0, \; result\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6], \; n\text{: } 5, \; i\text{: } 1, \; k\text{: } 0, \; result\text{: } ?\rangle)$
α_5: $(5, \langle A\text{: } [3,8,1,3,6], \; n\text{: } 5, \; i\text{: } 1, \; k\text{: } 1, \; result\text{: } ?\rangle)$
α_6: $(2, \langle A\text{: } [3,8,1,3,6], \; n\text{: } 5, \; i\text{: } 2, \; k\text{: } 1, \; result\text{: } ?\rangle)$
α_7: $(3, \langle A\text{: } [3,8,1,3,6], \; n\text{: } 5, \; i\text{: } 2, \; k\text{: } 1, \; result\text{: } ?\rangle)$
α_8: $(5, \langle A\text{: } [3,8,1,3,6], \; n\text{: } 5, \; i\text{: } 2, \; k\text{: } 1, \; result\text{: } ?\rangle)$
α_9: $(2, \langle A\text{: } [3,8,1,3,6], \; n\text{: } 5, \; i\text{: } 3, \; k\text{: } 1, \; result\text{: } ?\rangle)$
α_{10}: $(3, \langle A\text{: } [3,8,1,3,6], \; n\text{: } 5, \; i\text{: } 3, \; k\text{: } 1, \; result\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } ?,\ result\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_5: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_6: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_7: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_8: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_9: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 3,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{10}: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 3,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{11}: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 3,\ k\text{: } 1,\ result\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } ?,\ result\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_5: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_6: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_7: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_8: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_9: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 3,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{10}: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 3,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{11}: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 3,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{12}: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 4,\ k\text{: } 1,\ result\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } ?,\ result\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_5: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_6: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_7: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_8: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_9: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 3,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{10}: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 3,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{11}: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 3,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{12}: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 4,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{13}: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 4,\ k\text{: } 1,\ result\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } ?,\ result\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } ?,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 0,\ result\text{: } ?\rangle)$
α_5: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 1,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_6: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_7: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_8: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 2,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_9: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 3,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{10}: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 3,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{11}: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 3,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{12}: $(2, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 4,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{13}: $(3, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 4,\ k\text{: } 1,\ result\text{: } ?\rangle)$
α_{14}: $(5, \langle A\text{: } [3,8,1,3,6],\ n\text{: } 5,\ i\text{: } 4,\ k\text{: } 1,\ result\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } ?, \ k\text{: } ?, \ result\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } ?, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_5: $(5, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_6: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 2, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_7: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 2, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_8: $(5, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 2, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_9: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 3, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{10}: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 3, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{11}: $(5, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 3, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{12}: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 4, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{13}: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 4, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{14}: $(5, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 4, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{15}: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 5, \ k\text{: } 1, \ result\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } ?, \ k\text{: } ?, \ result\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } ?, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_5: $(5, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_6: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 2, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_7: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 2, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_8: $(5, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 2, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_9: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 3, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{10}: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 3, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{11}: $(5, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 3, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{12}: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 4, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{13}: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 4, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{14}: $(5, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 4, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{15}: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 5, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{16}: $(6, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 5, \ k\text{: } 1, \ result\text{: } ?\rangle)$

Computation of an Algorithm

Example: A computation, where algorithm FIND-MAX processes an input where $A = [3, 8, 1, 3, 6]$ and $n = 5$.

α_0: $(0, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } ?, \ k\text{: } ?, \ result\text{: } ?\rangle)$
α_1: $(1, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } ?, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_2: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_3: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_4: $(4, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 0, \ result\text{: } ?\rangle)$
α_5: $(5, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 1, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_6: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 2, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_7: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 2, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_8: $(5, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 2, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_9: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 3, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{10}: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 3, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{11}: $(5, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 3, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{12}: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 4, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{13}: $(3, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 4, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{14}: $(5, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 4, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{15}: $(2, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 5, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{16}: $(6, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 5, \ k\text{: } 1, \ result\text{: } ?\rangle)$
α_{17}: $(7, \langle A\text{: } [3,8,1,3,6], \ n\text{: } 5, \ i\text{: } 5, \ k\text{: } 1, \ result\text{: } 8\rangle)$

Computation of an Algorithm

By executing an instruction I, the machine goes from configuration α to configuration α':

$$\alpha \xrightarrow{I} \alpha'$$

A computation can be:

- **Finite**:

$$\alpha_0 \xrightarrow{I_0} \alpha_1 \xrightarrow{I_1} \alpha_2 \xrightarrow{I_2} \alpha_3 \xrightarrow{I_3} \alpha_4 \xrightarrow{I_4} \cdots \xrightarrow{I_{t-2}} \alpha_{t-1} \xrightarrow{I_{t-1}} \alpha_t$$

  where α_t is either a final configuration or a configuration where an error occurred and it is not possible to continue in the computation

- **Infinite**:

$$\alpha_0 \xrightarrow{I_0} \alpha_1 \xrightarrow{I_1} \alpha_2 \xrightarrow{I_2} \alpha_3 \xrightarrow{I_3} \alpha_4 \xrightarrow{I_4} \cdots$$

Computation of an Algorithm

A computation can be described in two different ways:

- as a sequence of configurations $\alpha_0, \alpha_1, \alpha_2, \ldots$
- as a sequence of executed instructions $I_0, I_1, I_2, \ldots$

Random Access Machines

Random Access Machine

A **Random Access Machine** (**RAM**) is an idealized model of a computer.

It consists of the following parts:

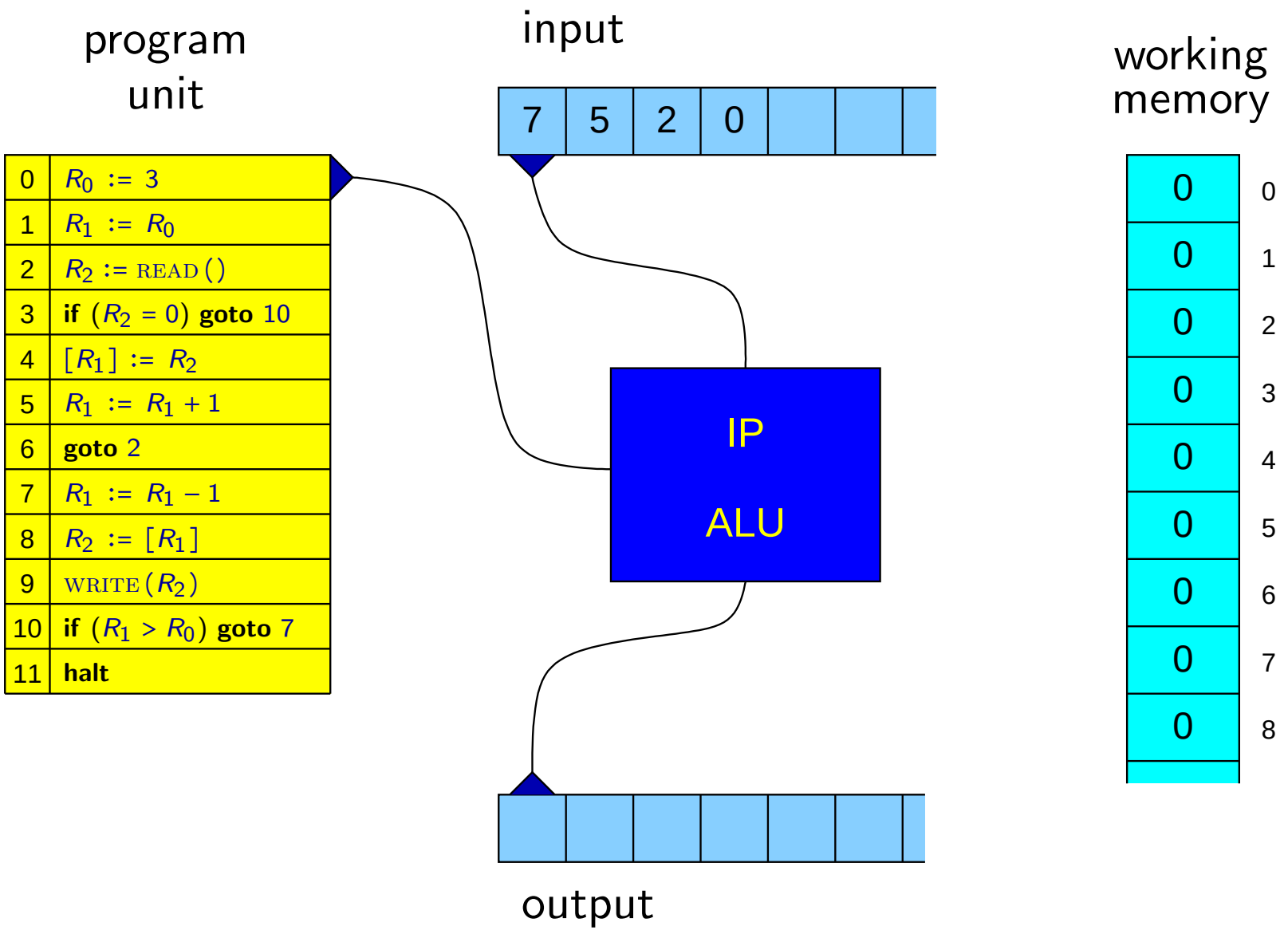
- **Program unit** – contains a program for the RAM and a pointer to the currently executed instruction
- **Working memory** consists of cells numbered $0, 1, 2, \ldots$

  These cells will be denoted $R_0, R_1, R_2, \ldots$

  The content of the cells can be read and written to.
- **Input tape** – read-only
- **Output tape** – write-only

The cells of memory, as well as the cells of input and output tapes contain integers (i.e., elements of set $\mathbb{Z}$) as their values.

Random Access Machine



Random Access Machine

Overview of instructions:

$R_i := c$	– assignment of a constant
$R_i := R_j$	– assignment
$R_i := [R_j]$	– load (reading from memory)
$[R_i] := R_j$	– store (writing to memory)
$R_i := R_j \ op \ R_k$	– arithmetic instructions, $op \in \{+, -, *, /\}$
or $R_i := R_j \ op \ c$	
if $(R_i \ rel \ R_j)$ **goto** ℓ	– conditional jump, $rel \in \{=, \neq, \leq, \geq, <, >\}$
or **if** $(R_i \ rel \ c)$ **goto** ℓ	
goto ℓ	– unconditional jump
$R_i :=$ READ $()$	– reading from input
WRITE (R_i)	– writing to output
halt	– program termination

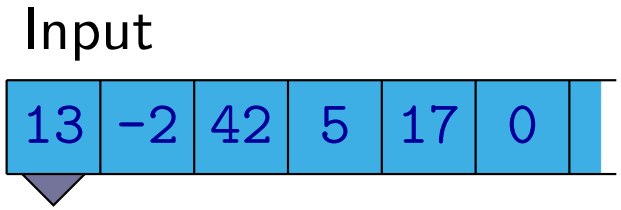
Random Access Machine

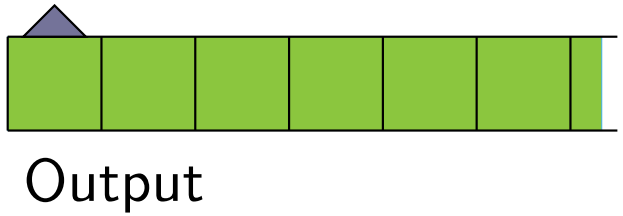
Examples of instructions:

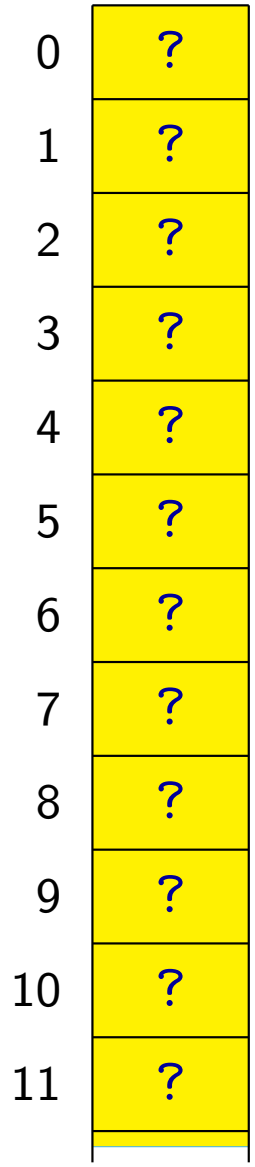
$R_5 := 42$	– assignment of a constant
$R_{12} := R_3$	– assignment
$R_8 := [R_2]$	– load (reading from memory)
$[R_{15}] := R_9$	– store (writing to memory)
$R_7 := R_3 + R_6$	– arithmetic instruction
$R_{18} := R_{18} - 1$	– arithmetic instruction
if $(R_4 \geq R_1)$ **goto** 2801	– conditional jump
if $(R_2 \neq 0)$ **goto** 3581	– conditional jump
goto 537	– unconditional jump
$R_{23} :=$ READ$()$	– reading from input
WRITE(R_{17})	– writing to output
halt	– program termination

Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt







Random Access Machine

$R_0 := 3$

$R_1 := R_0$

$L_1 : R_2 := \text{READ}()$

if $(R_2 = 0)$ **goto** L_3

$[R_1] := R_2$

$R_1 := R_1 + 1$

goto L_1

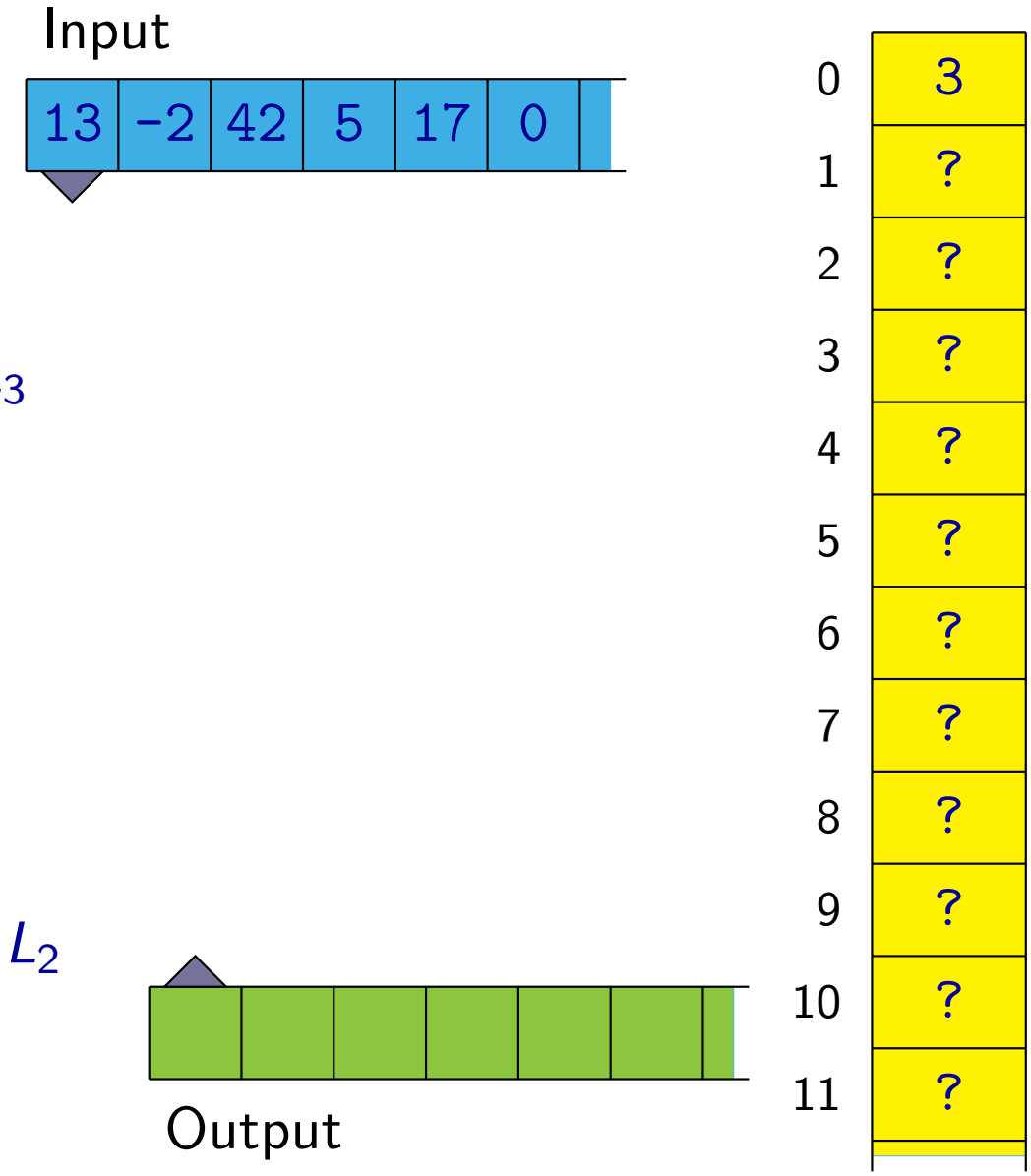
$L_2 : R_1 := R_1 - 1$

$R_2 := [R_1]$

$\text{WRITE}(R_2)$

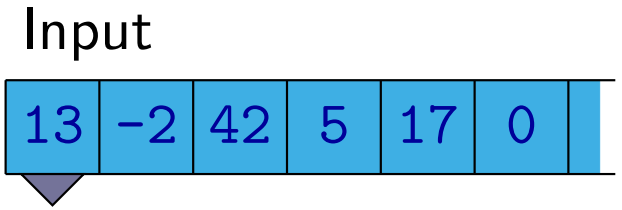
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2

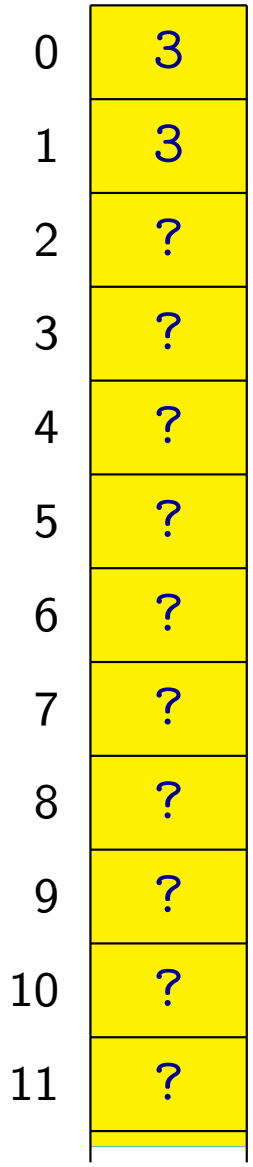
halt

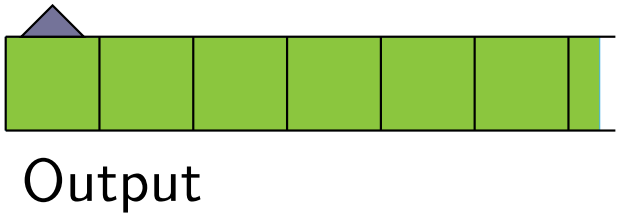


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

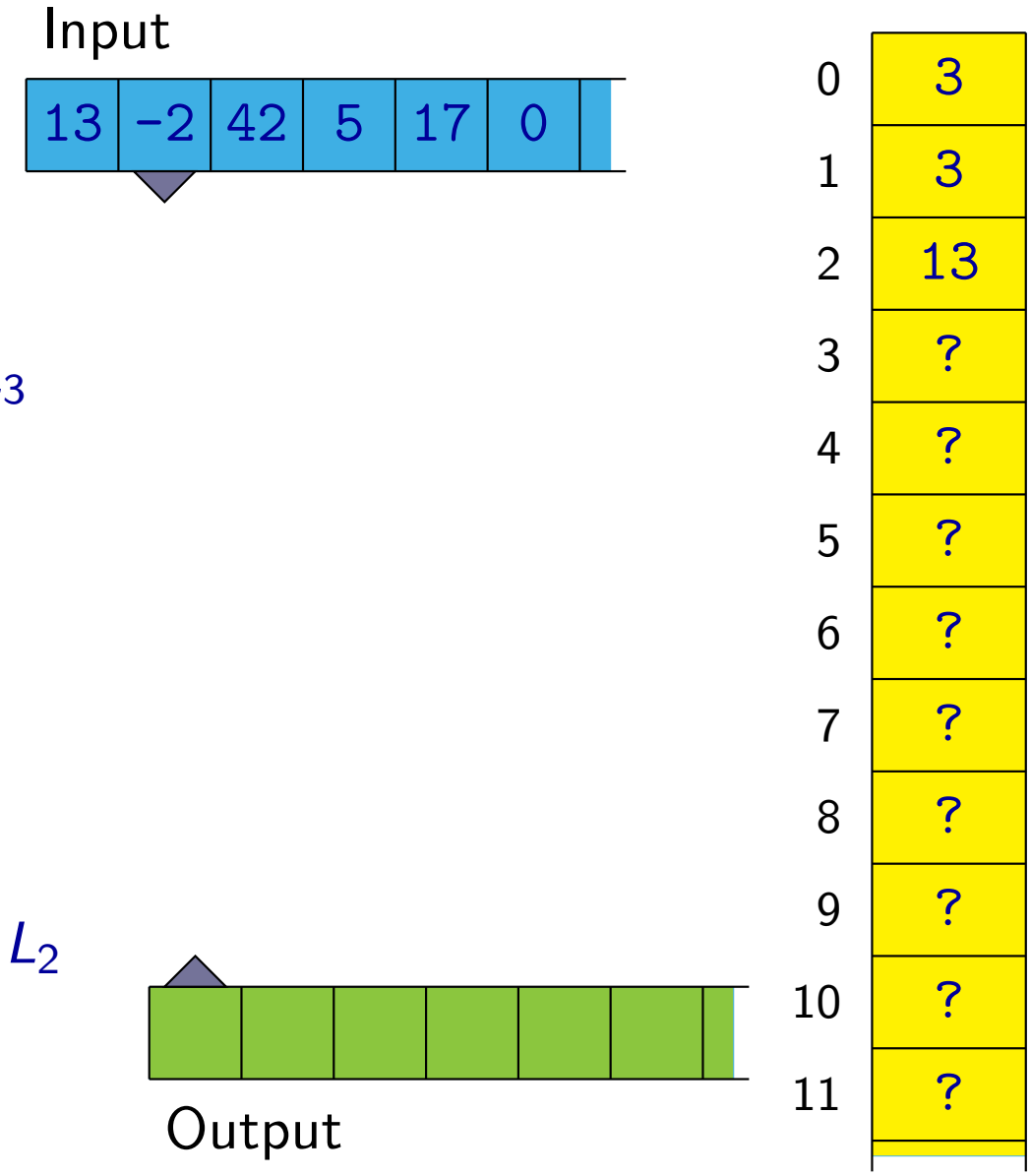






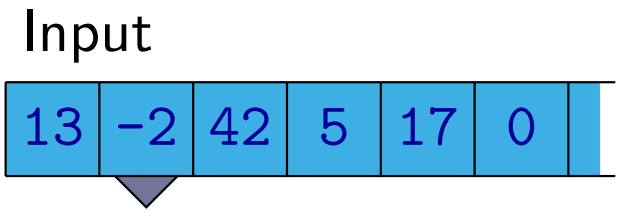
Random Access Machine

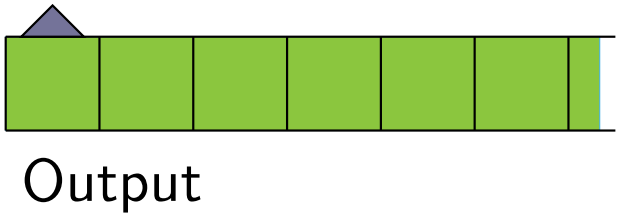
$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

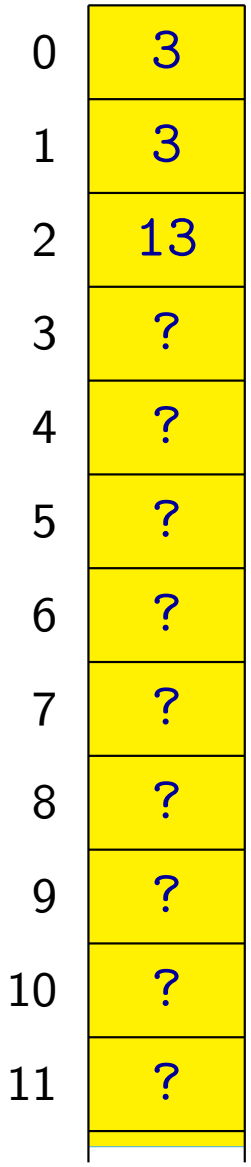


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

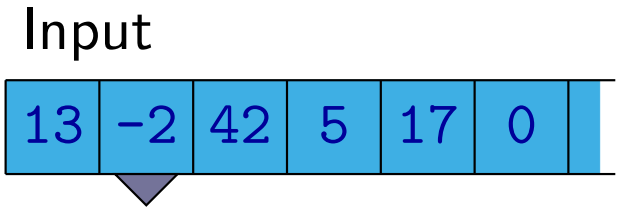


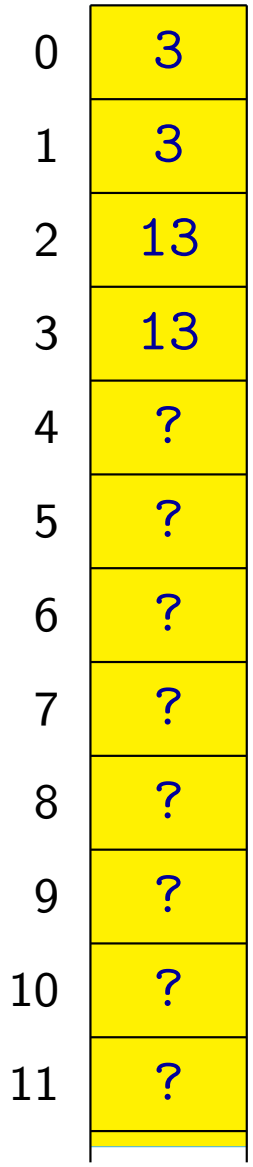


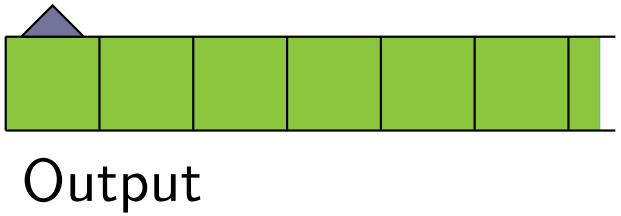


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

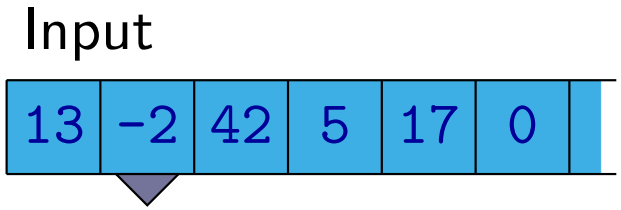


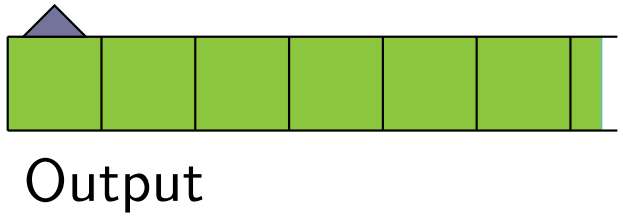


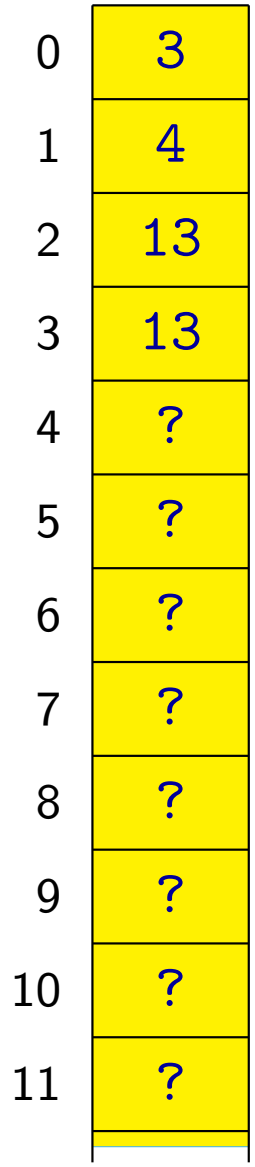


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

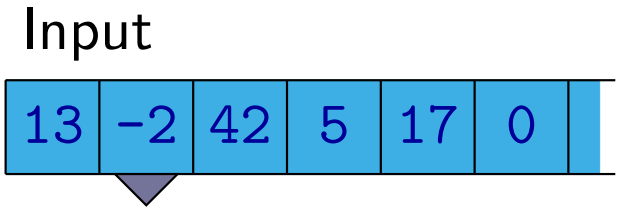


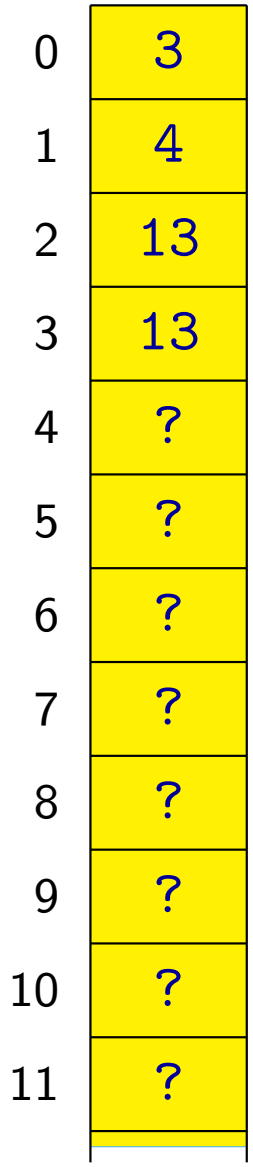


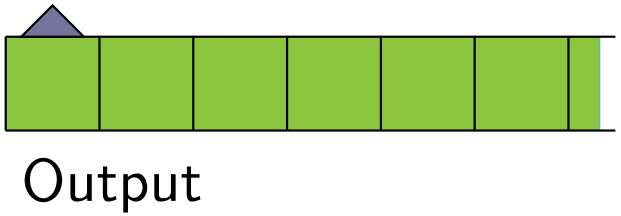


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

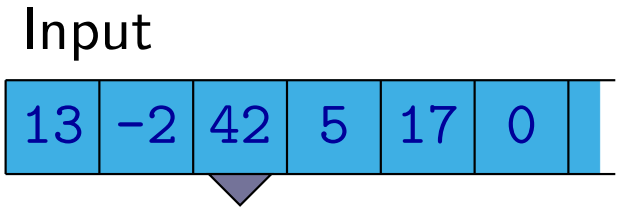


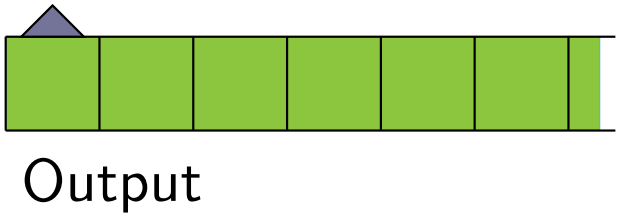


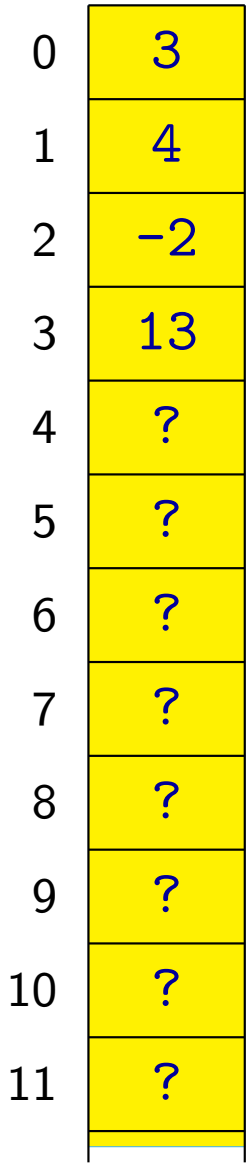


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

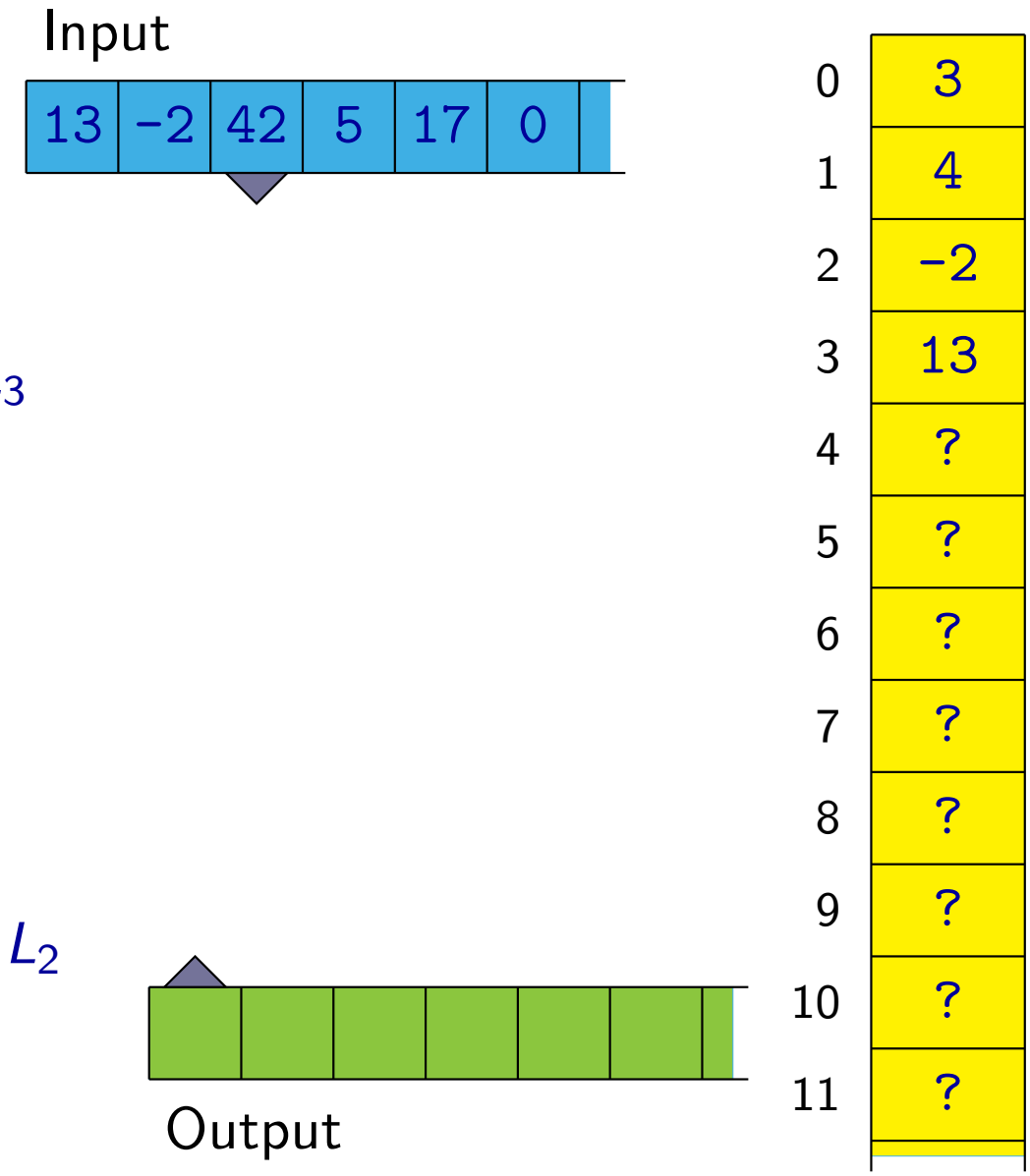






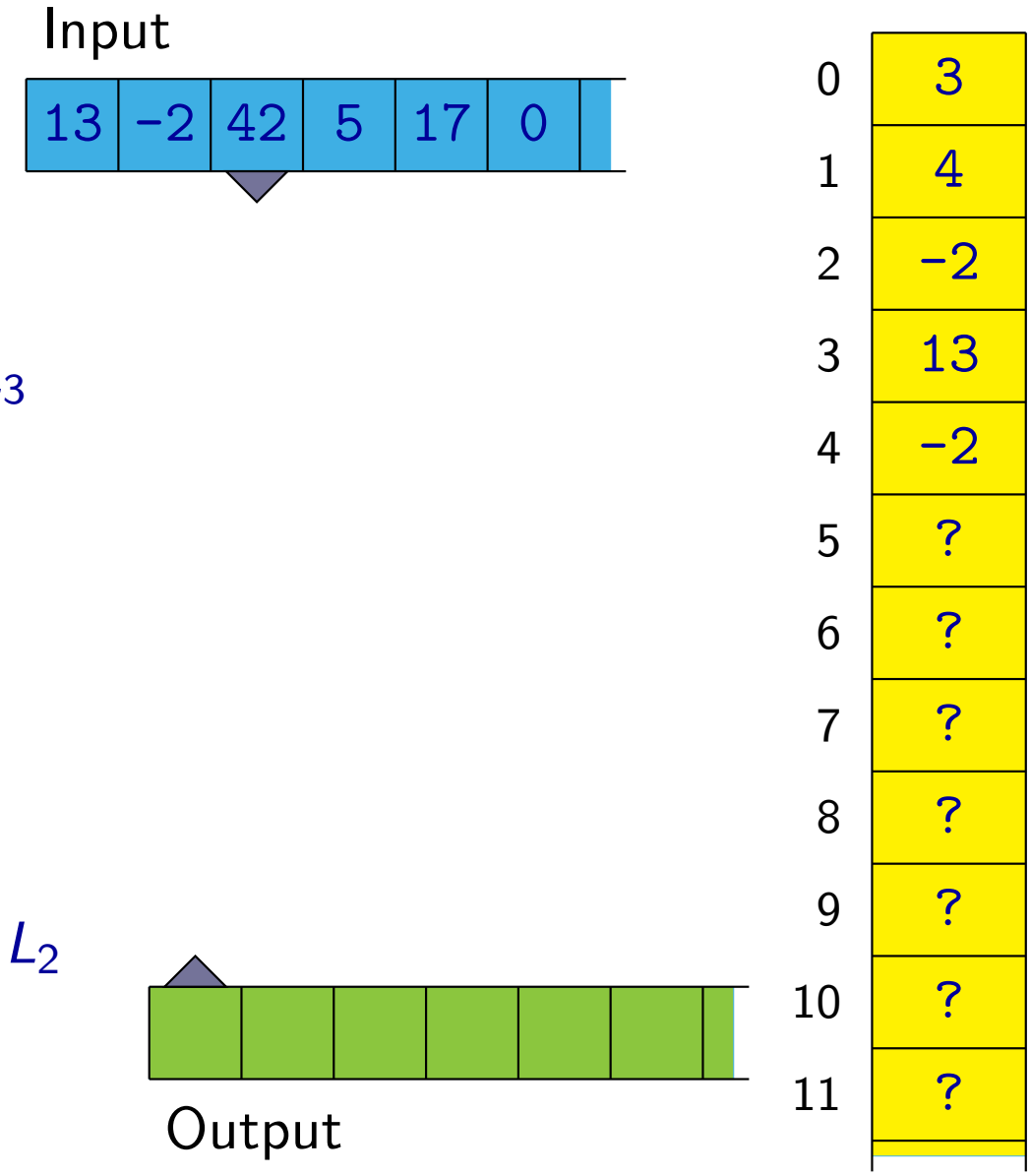
Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt



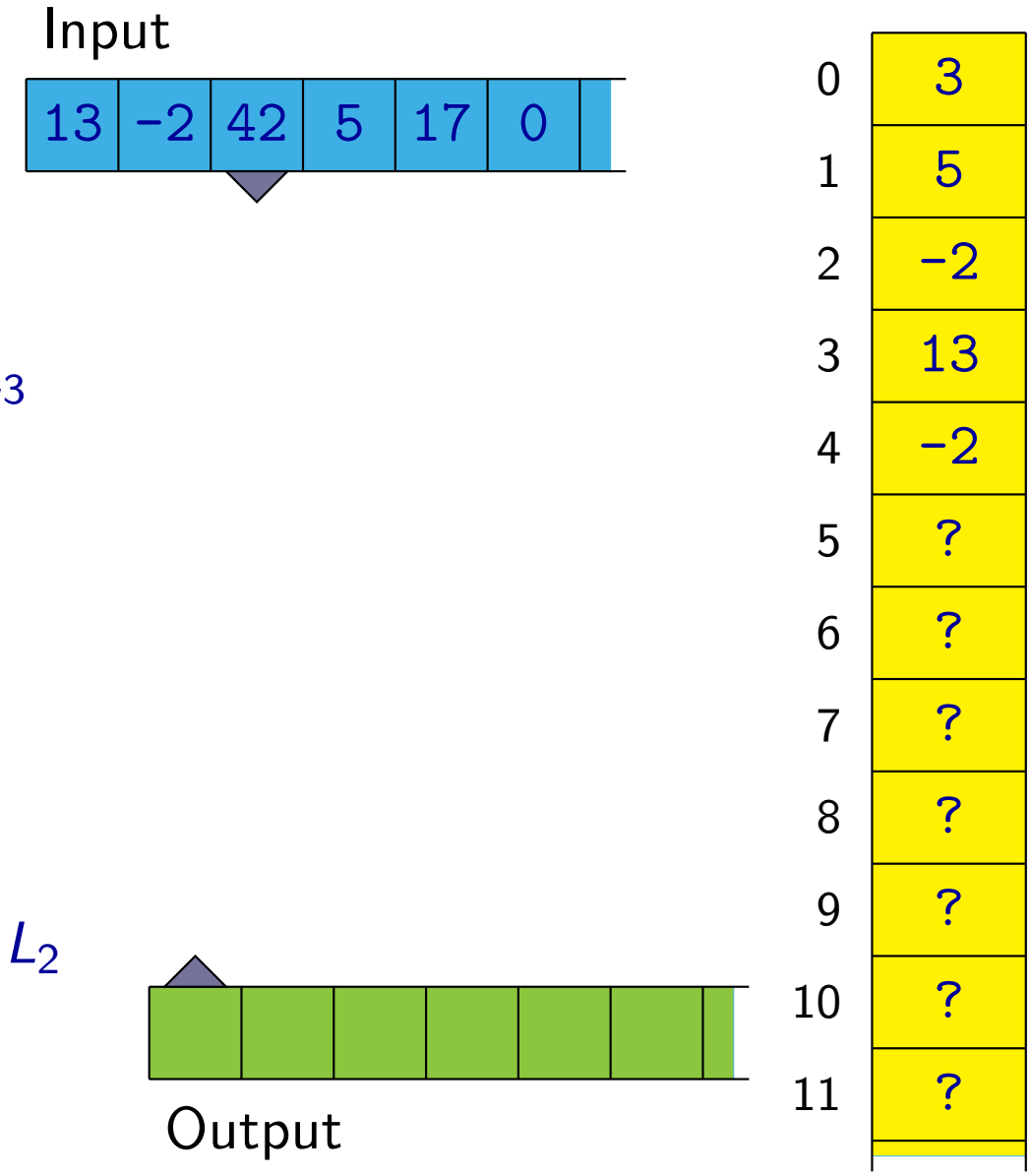
Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt



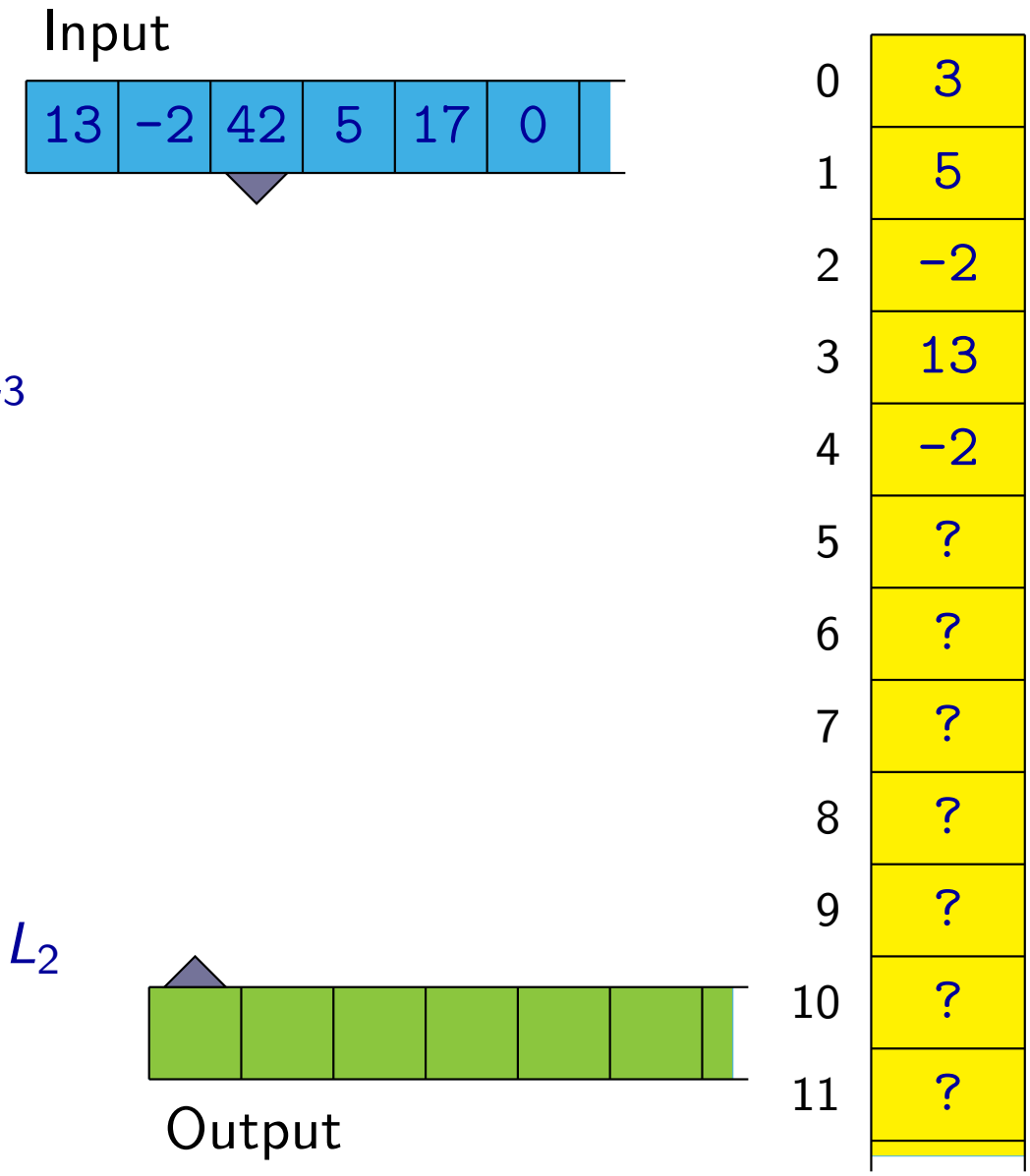
Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt



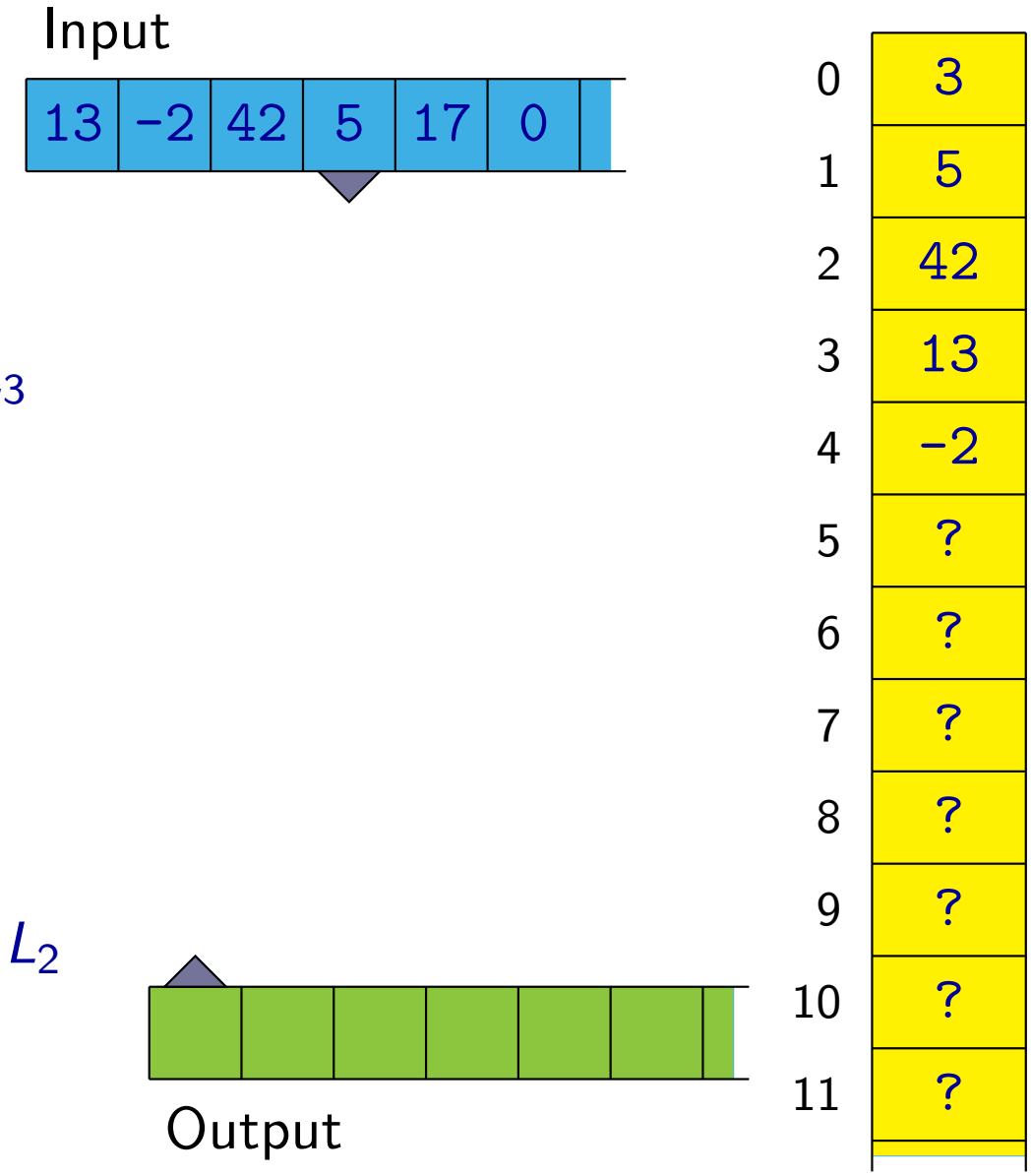
Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt



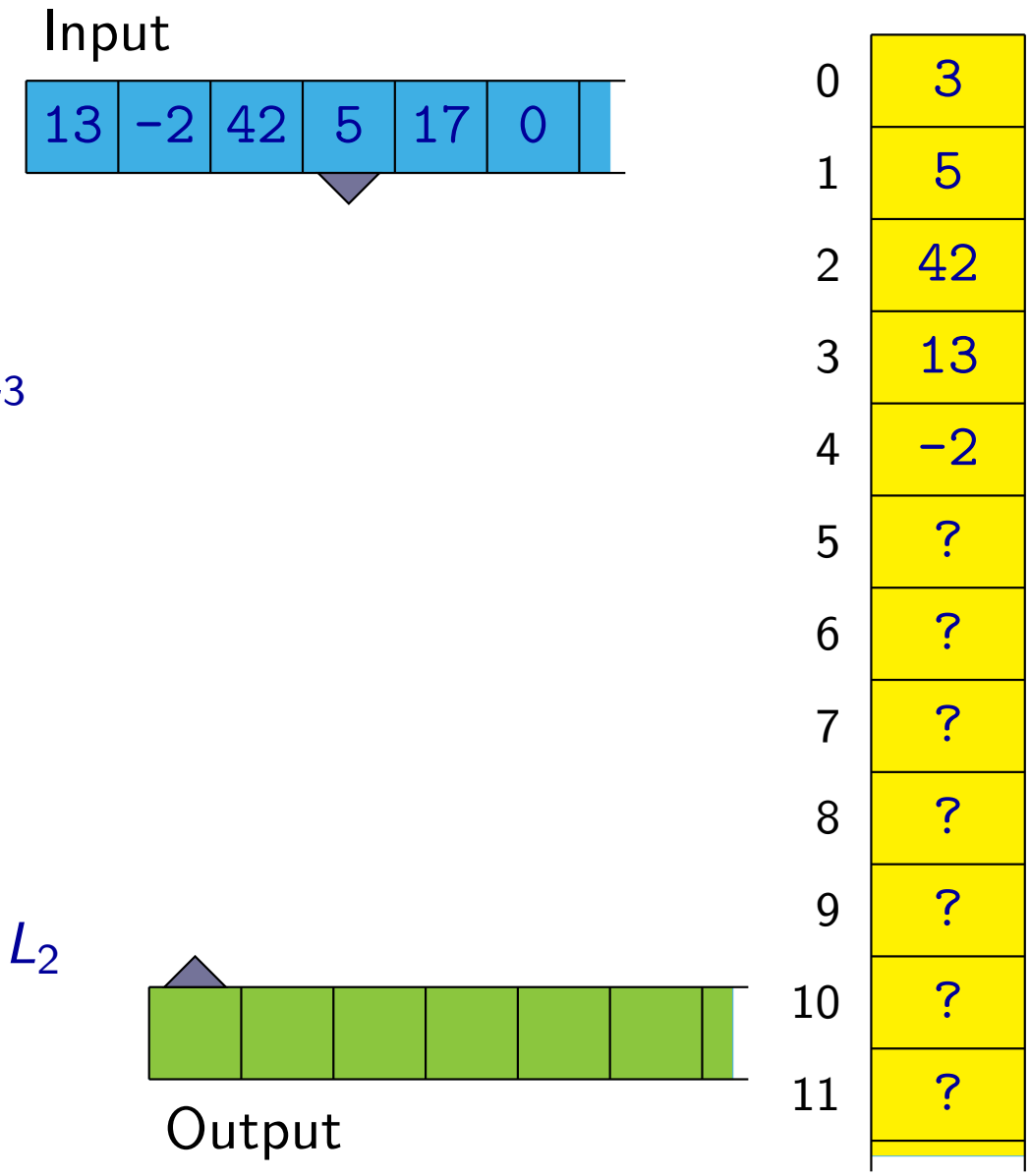
Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt



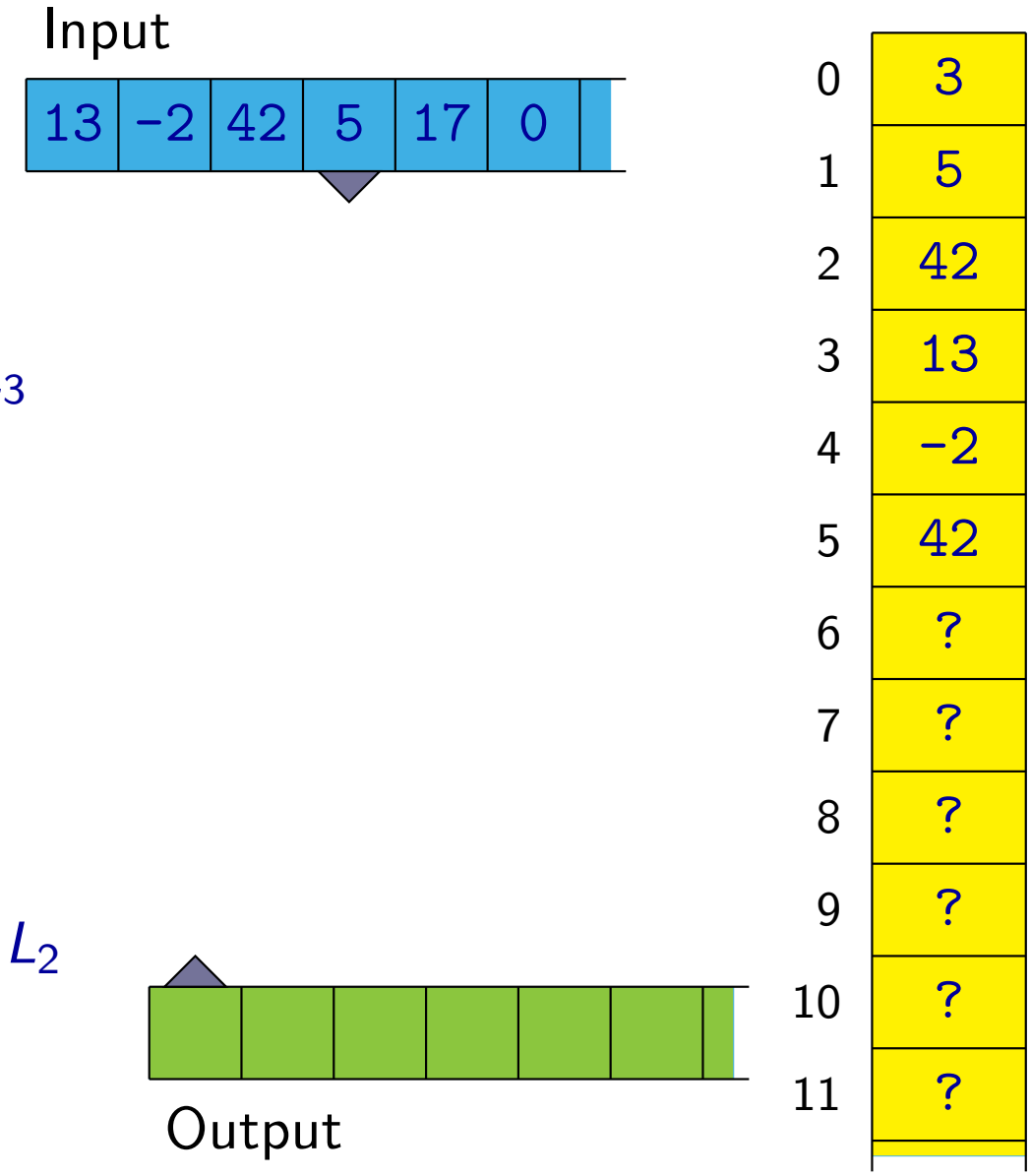
Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt



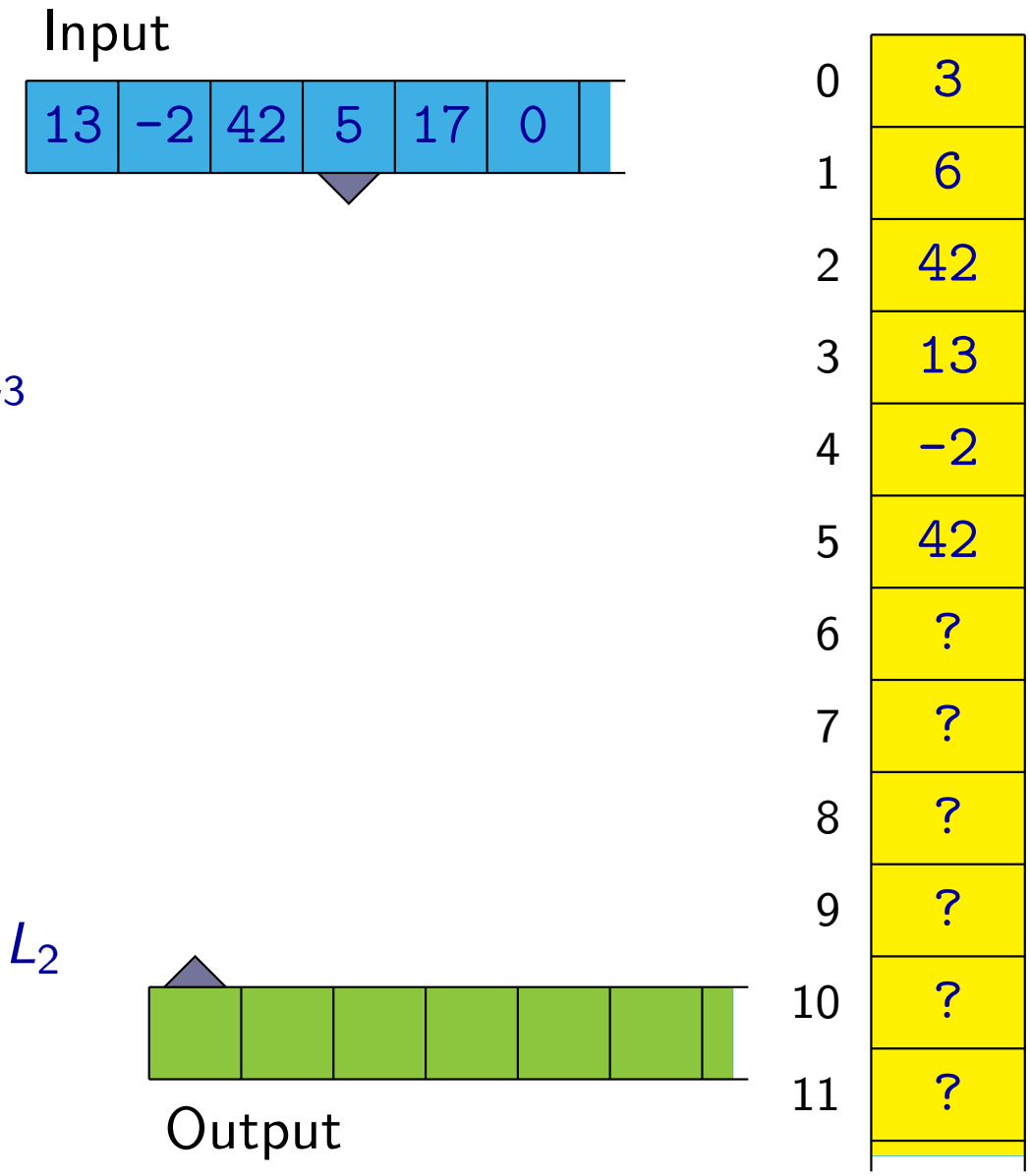
Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt



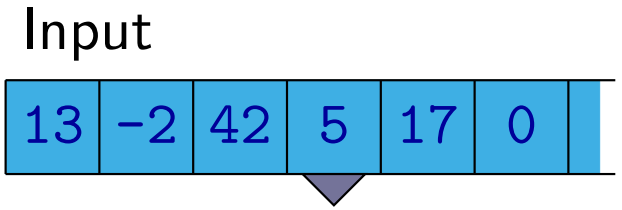
Random Access Machine

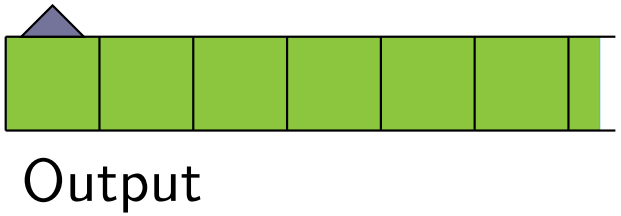
$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

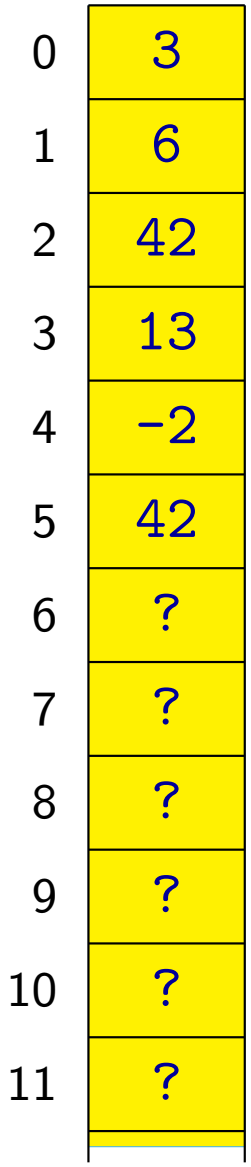


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

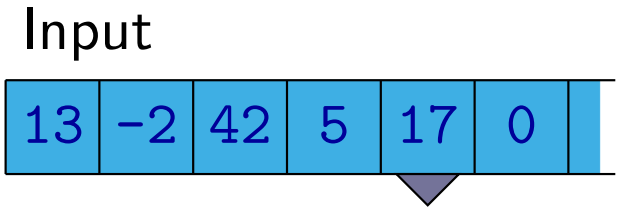


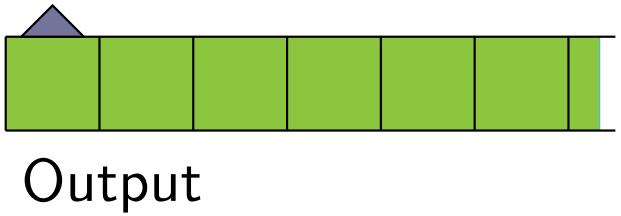


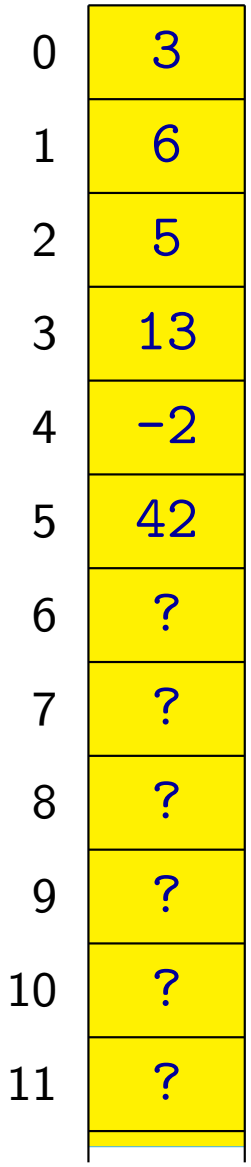


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

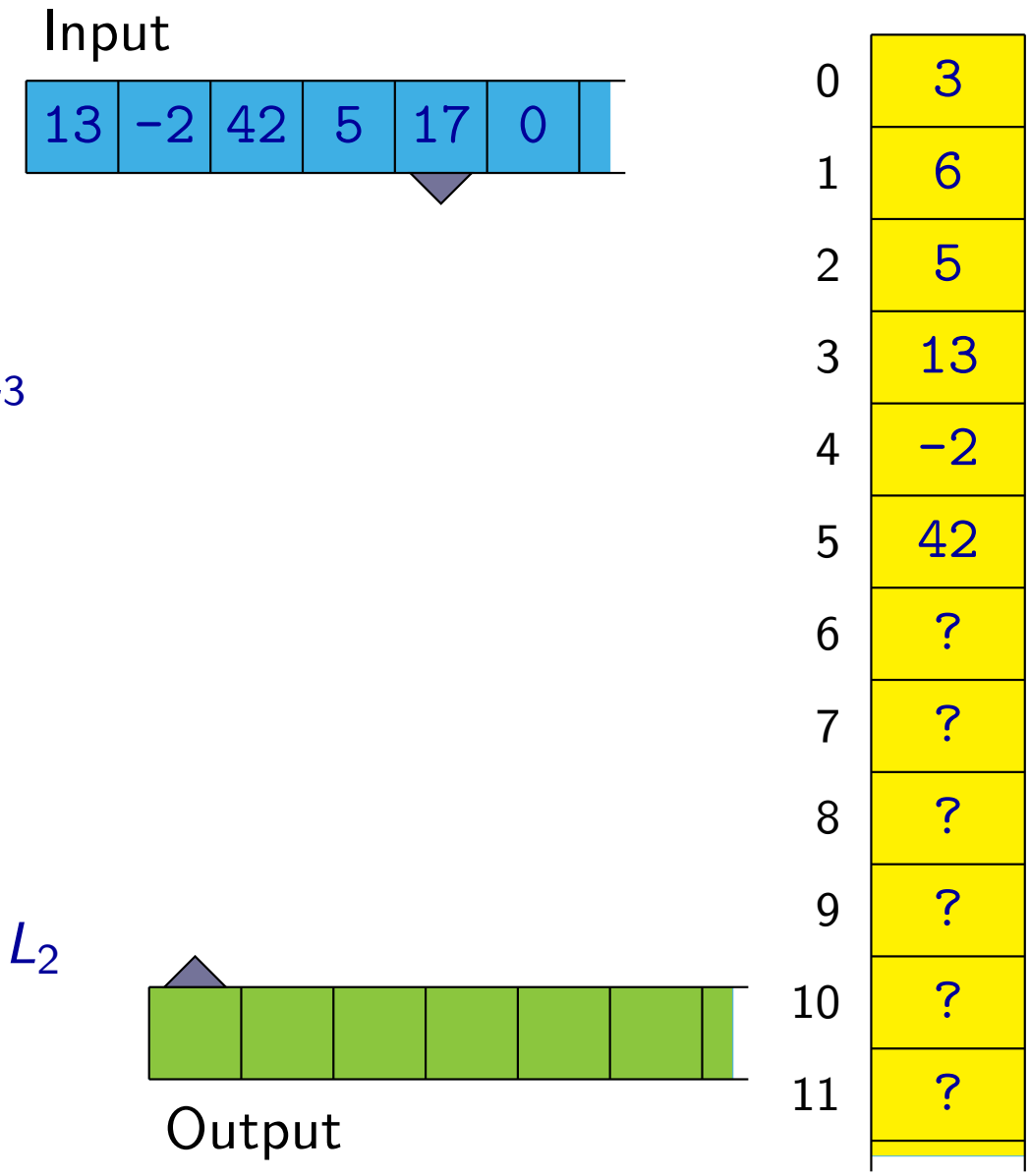






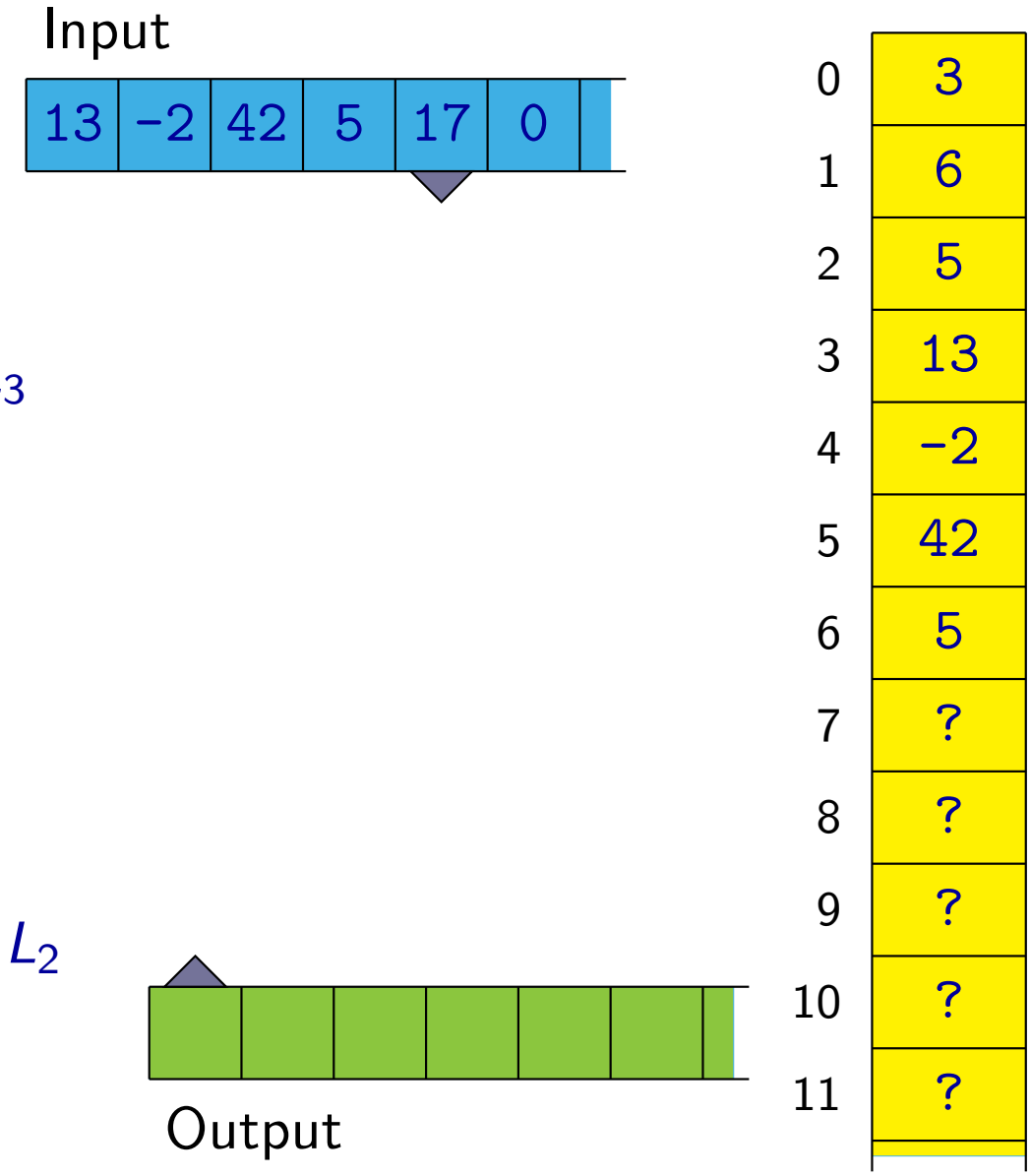
Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt



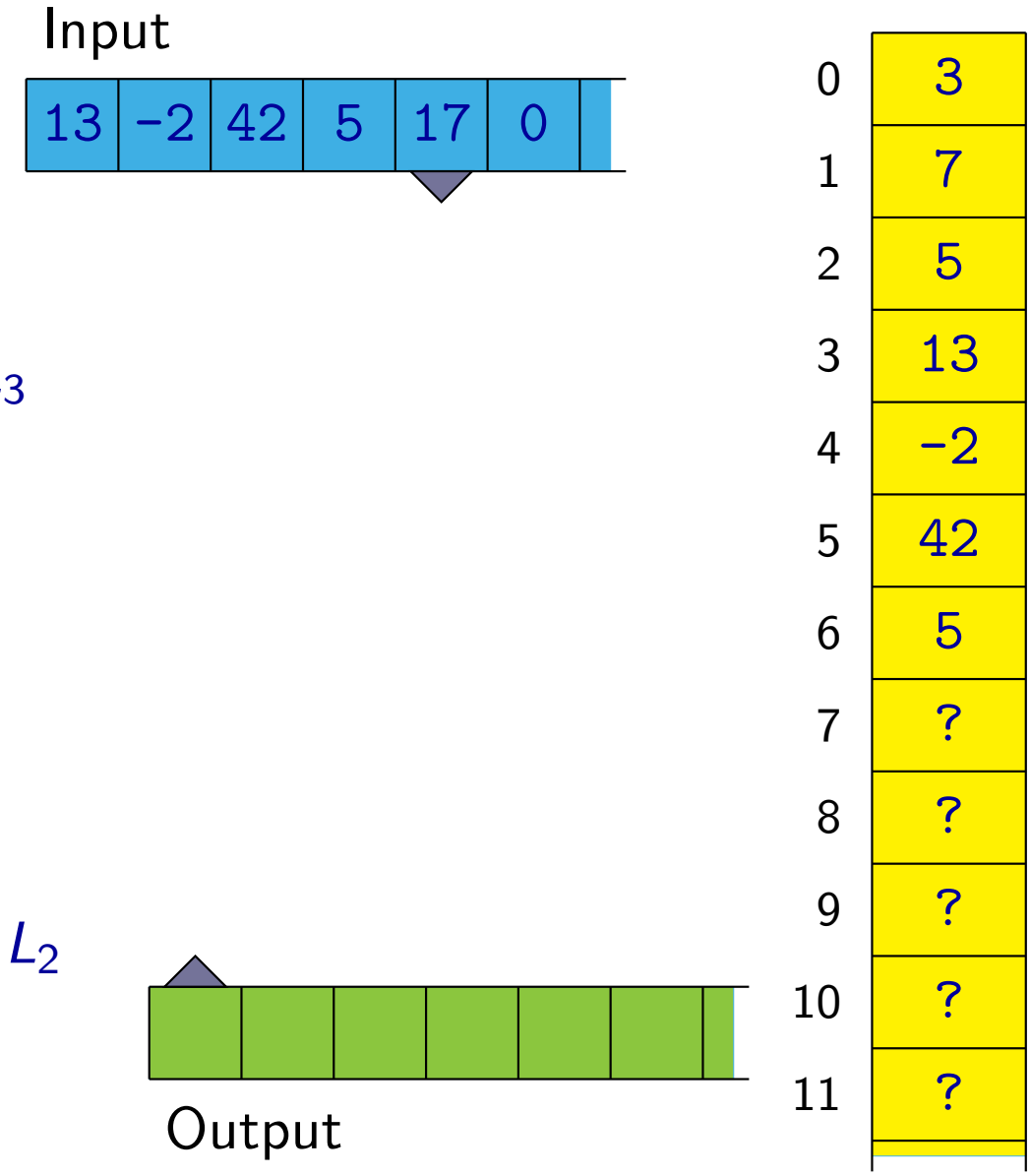
Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt



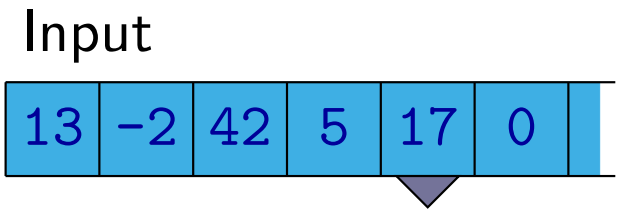
Random Access Machine

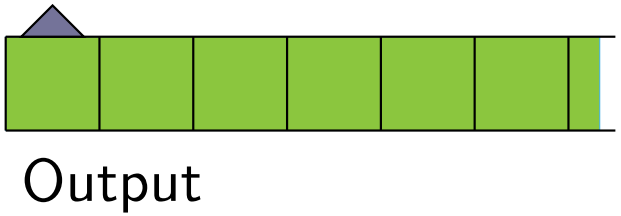
$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

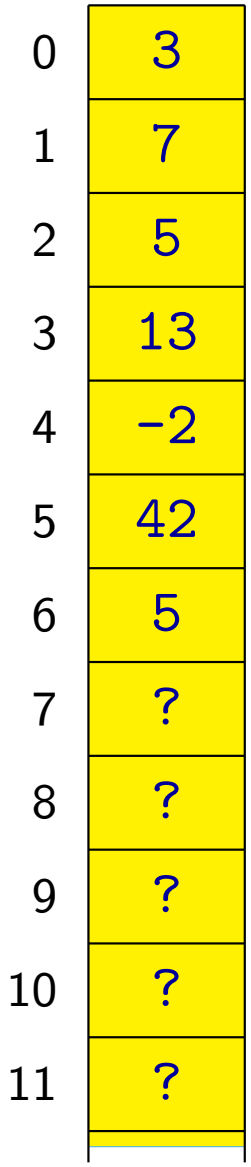


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

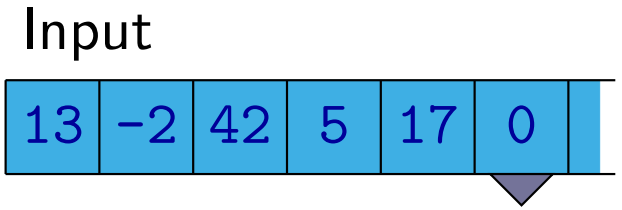


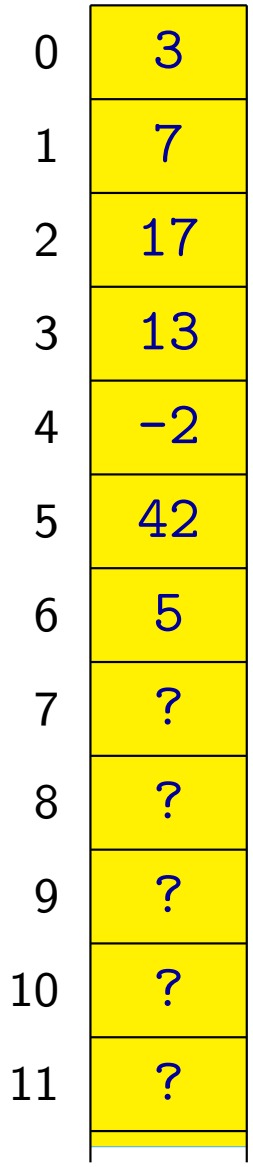


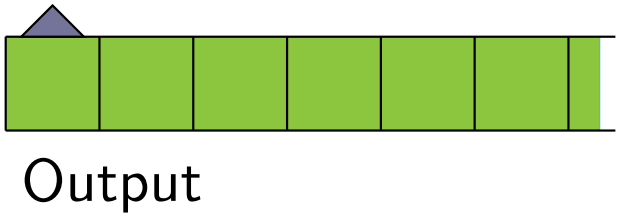


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

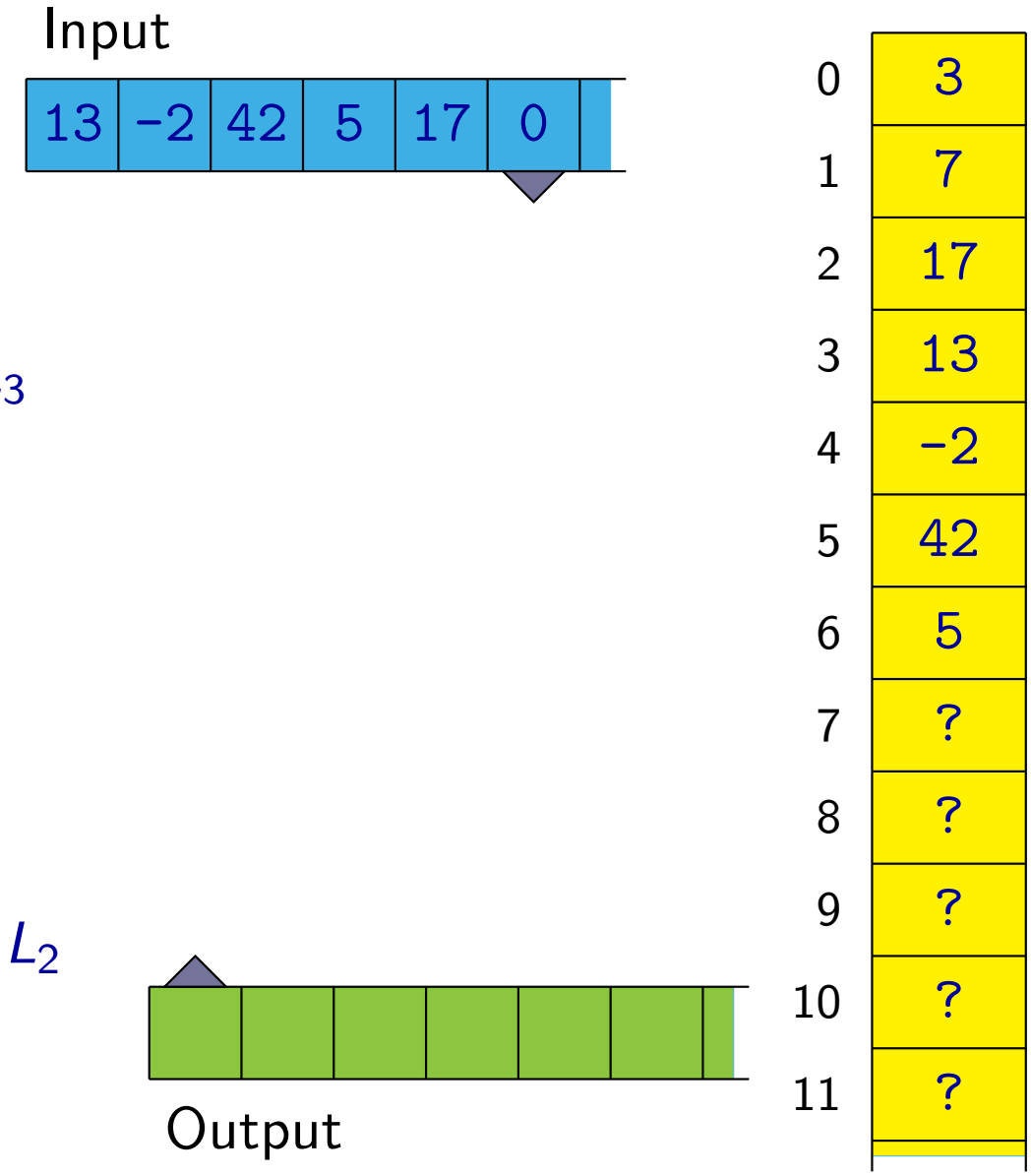






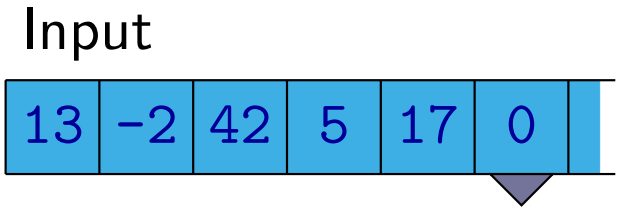
Random Access Machine

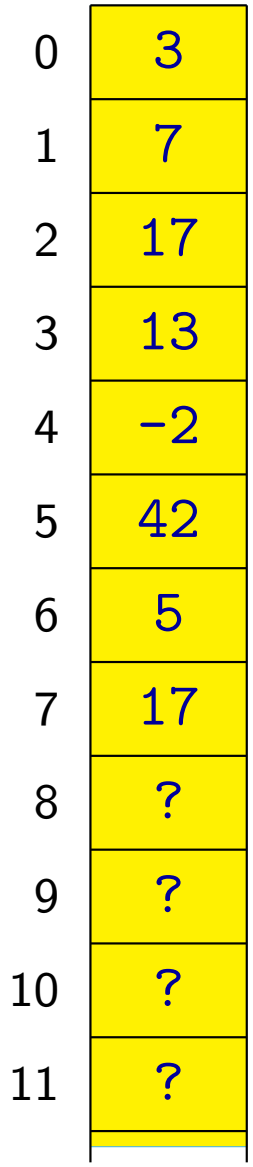
$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

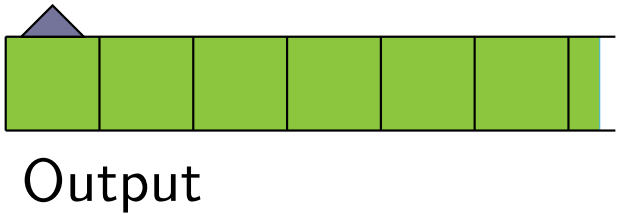


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

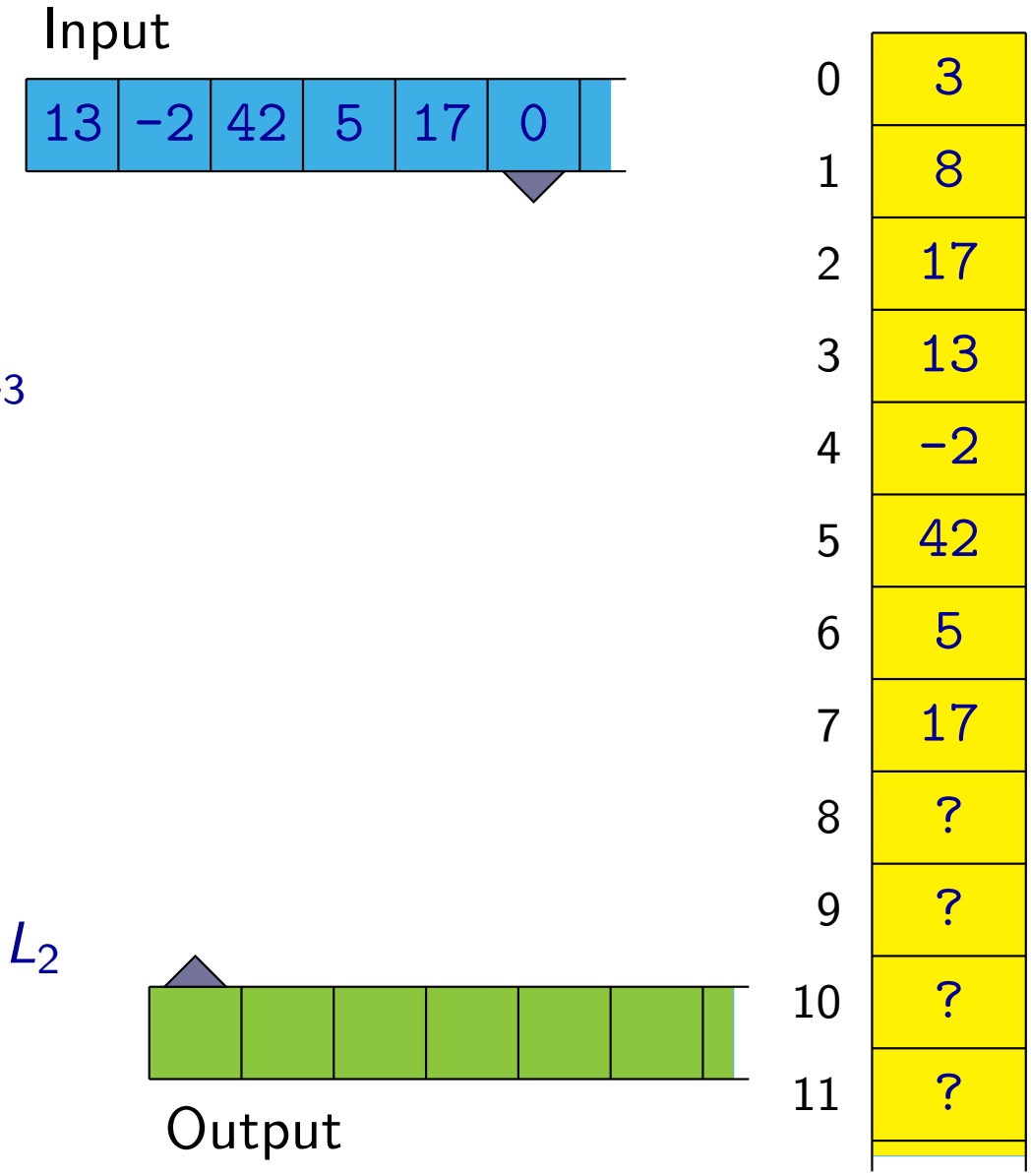






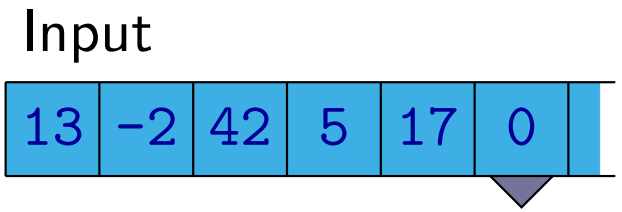
Random Access Machine

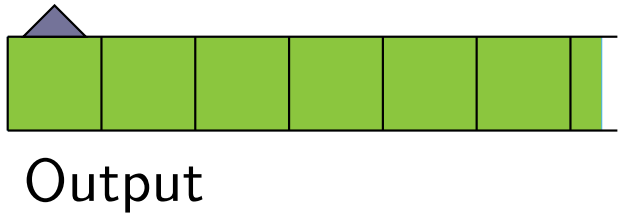
$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

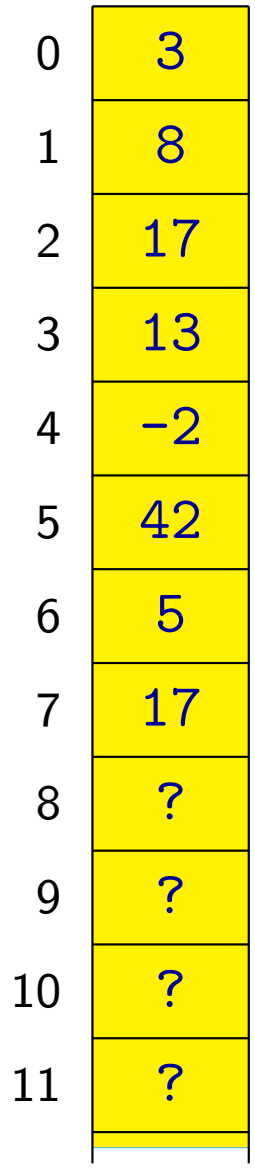


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

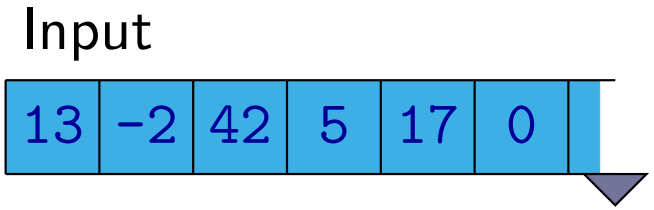


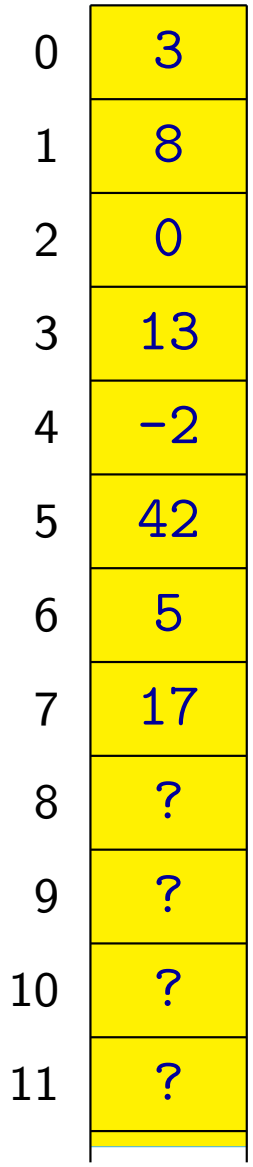


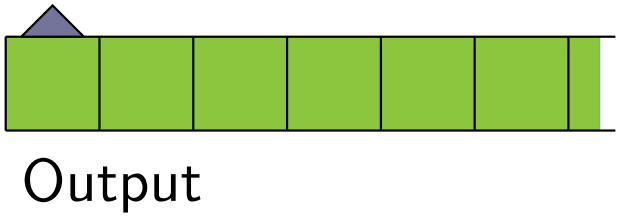


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

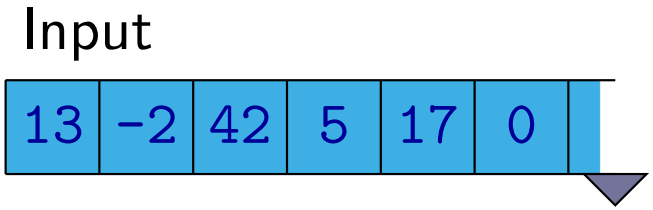


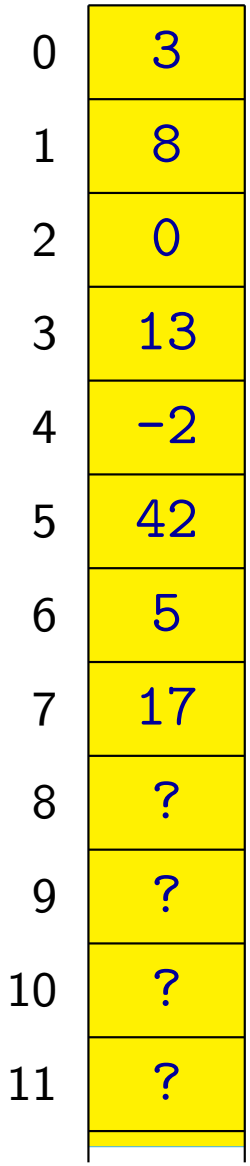


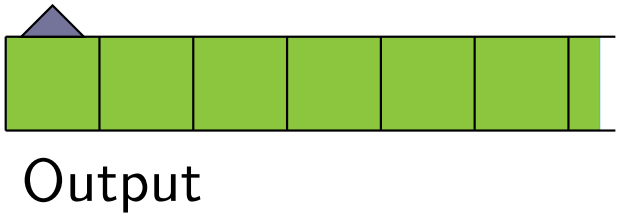


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

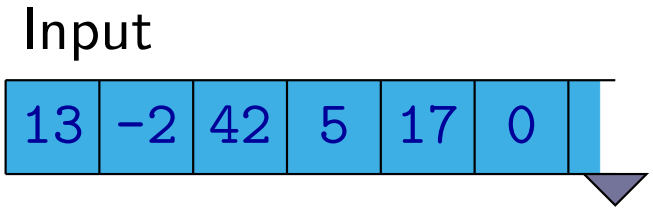


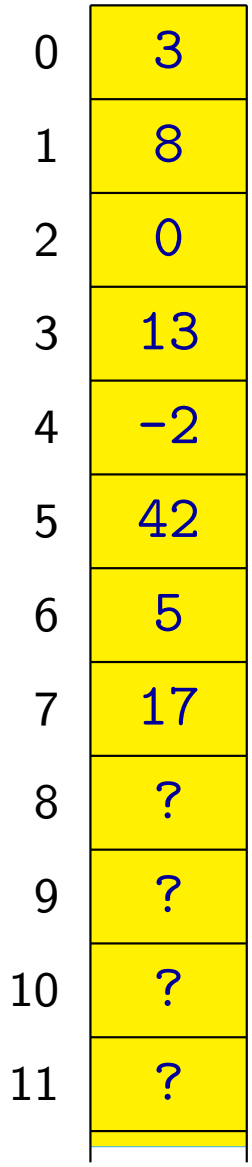


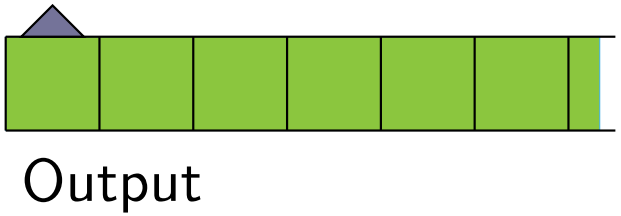


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

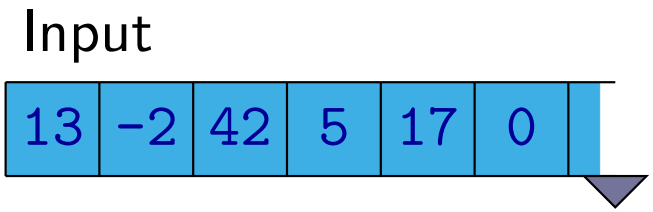


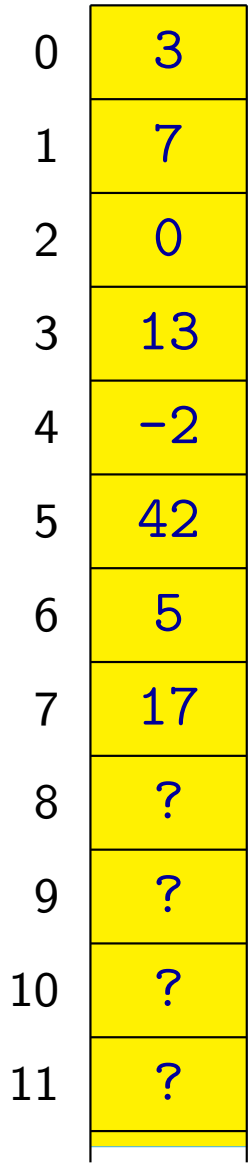


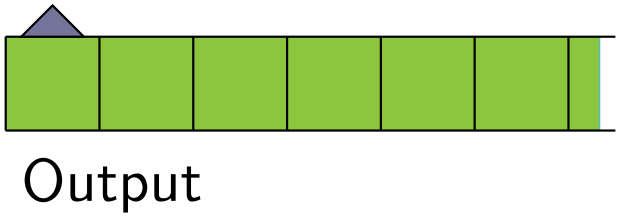


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

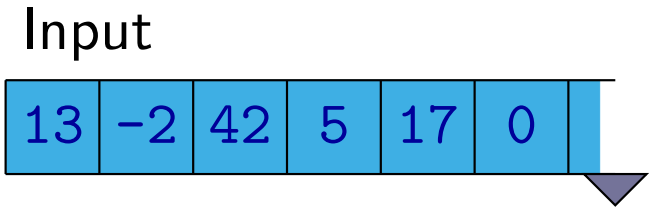


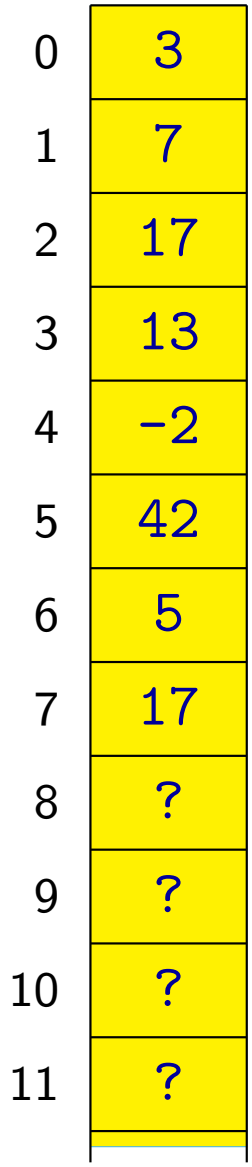


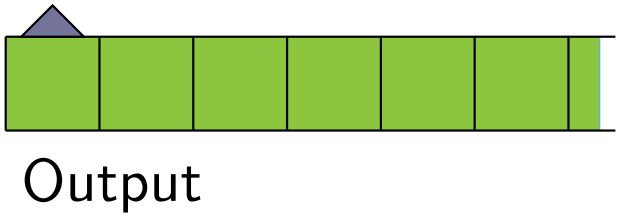


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

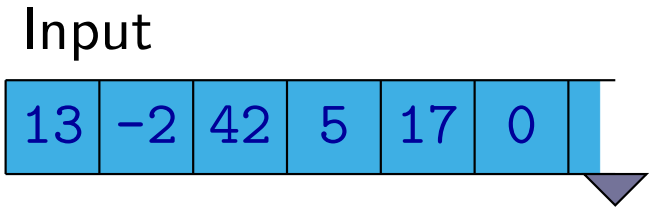


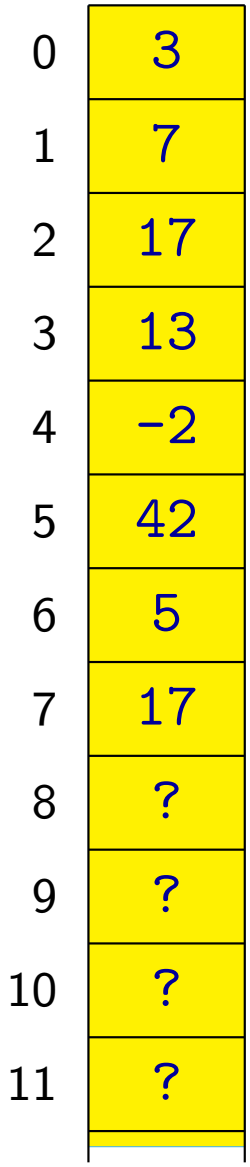


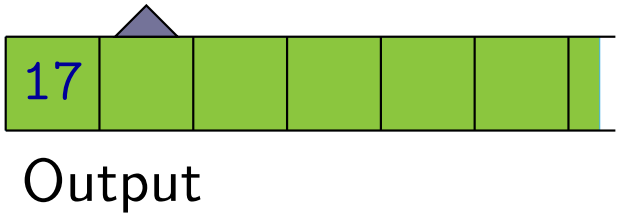


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

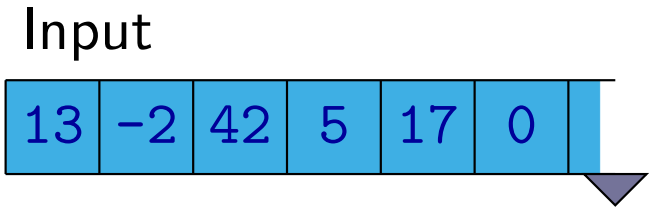


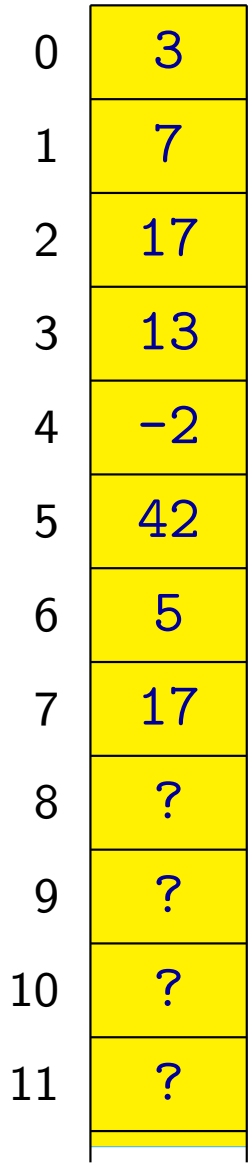


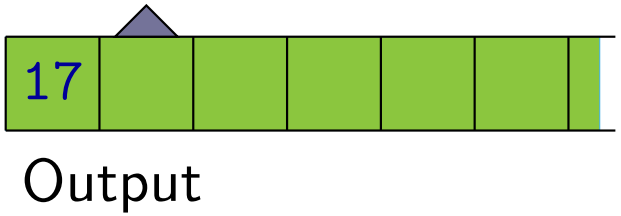


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

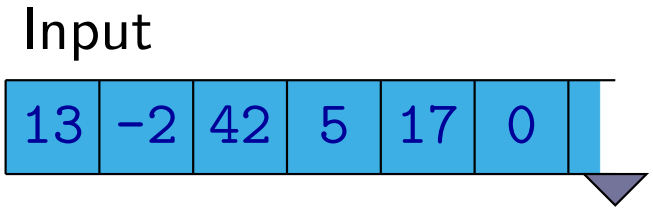


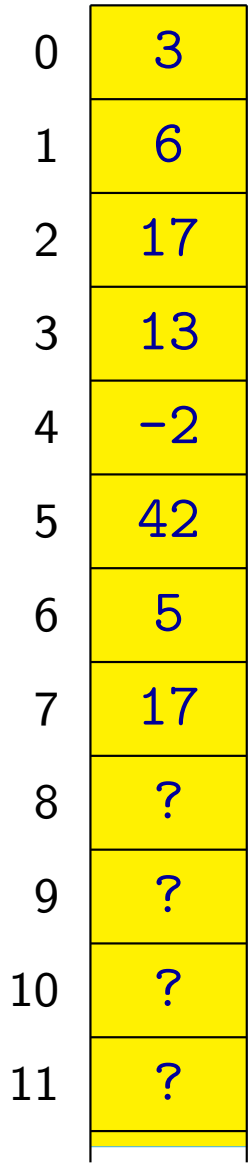


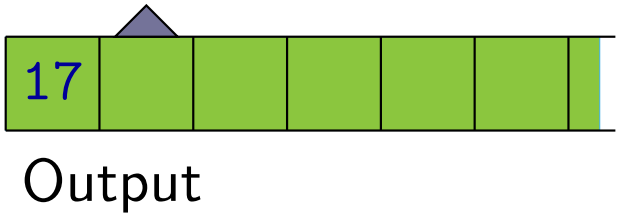


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

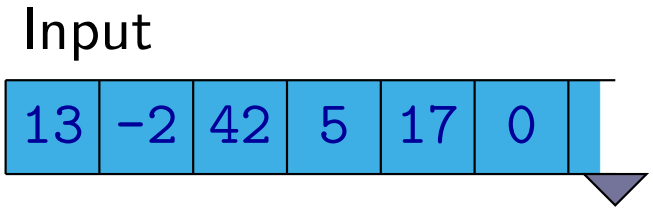


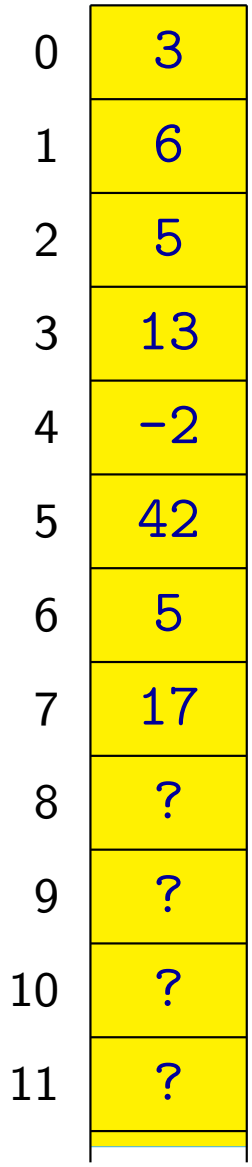


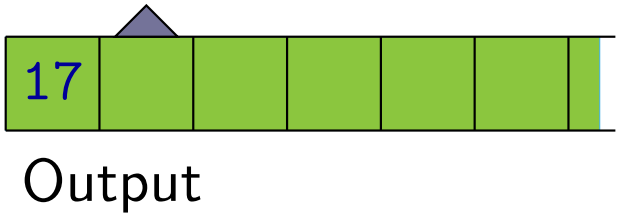


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

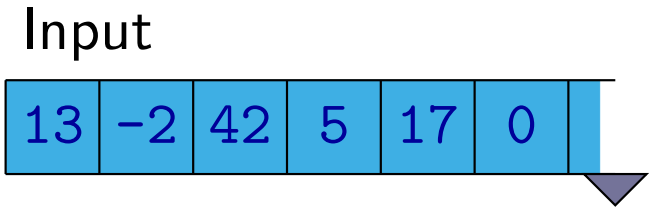


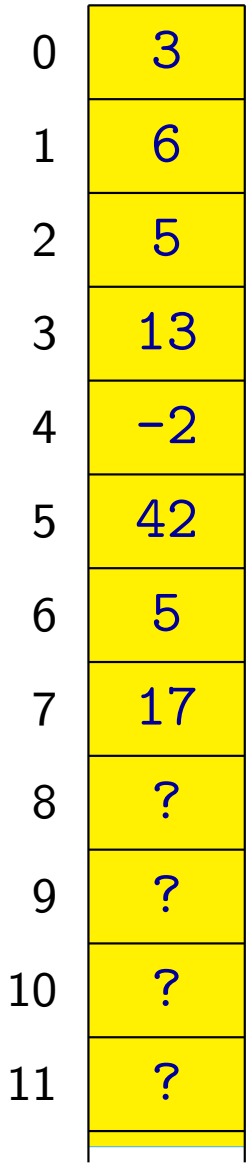


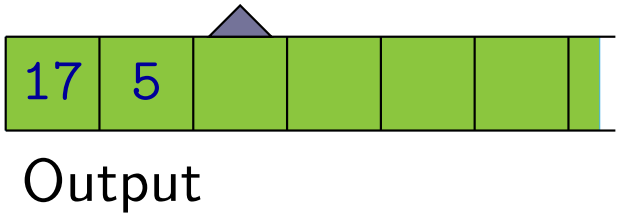


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

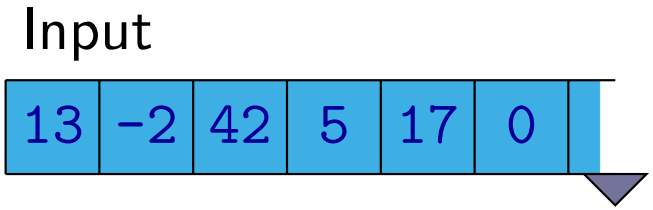


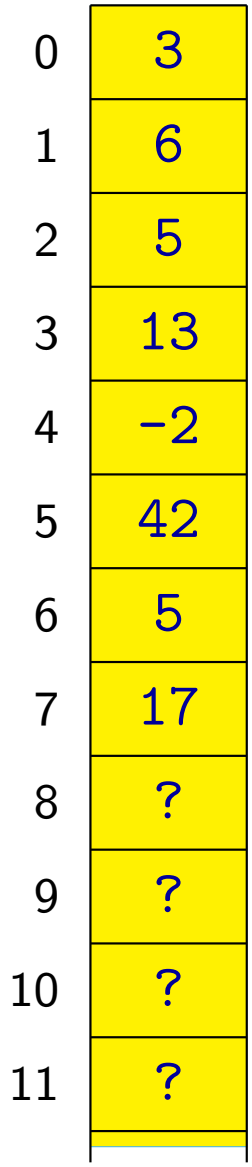


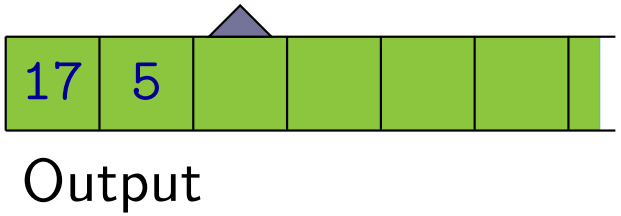


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

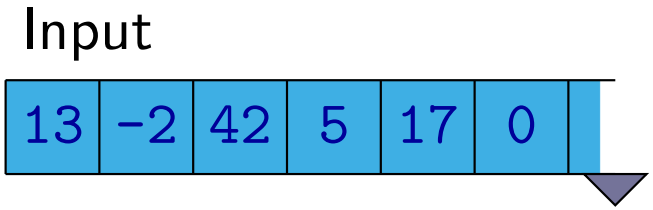


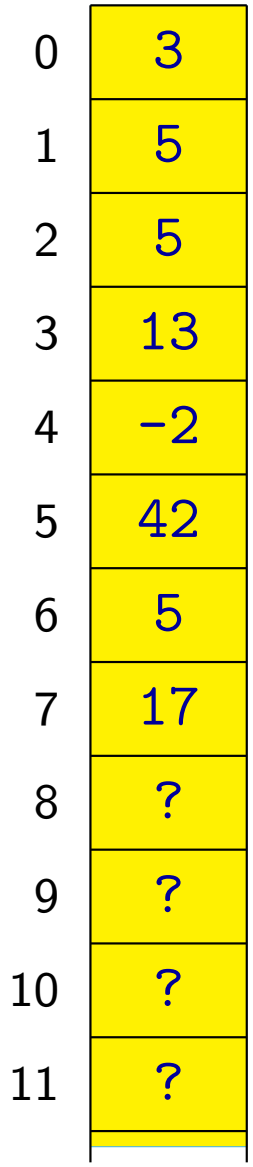


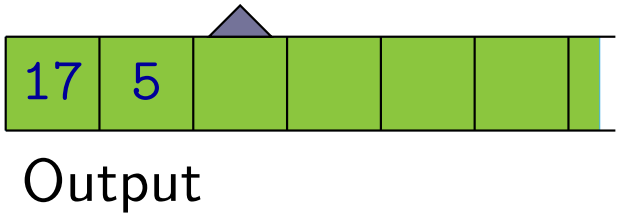


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

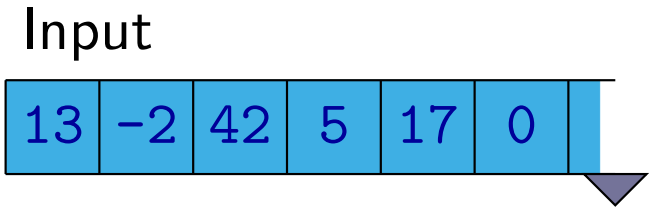


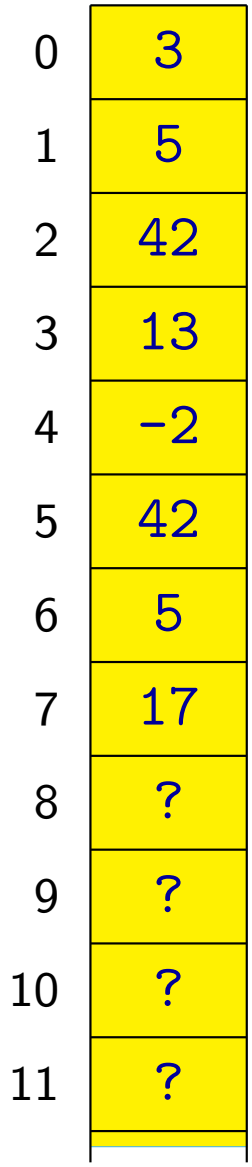


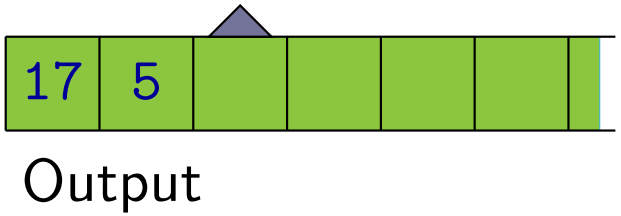


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

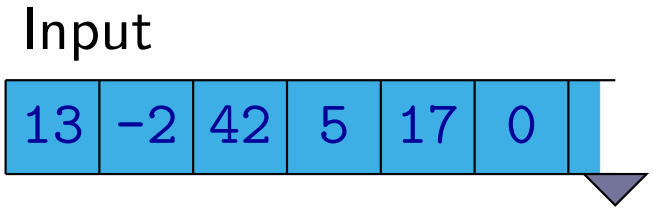


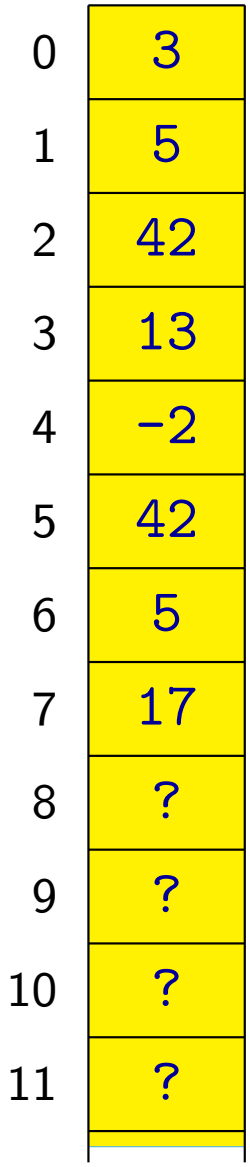


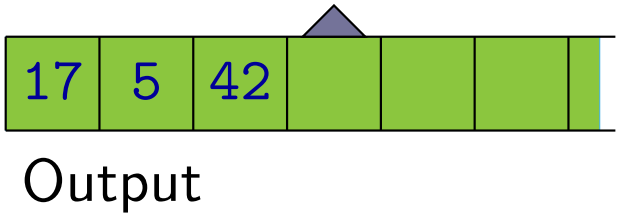


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

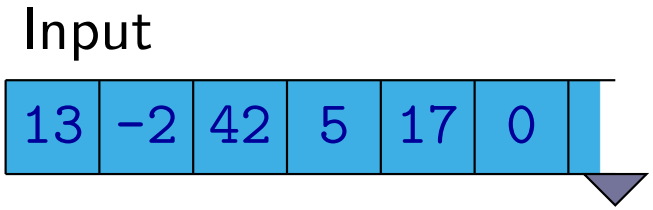


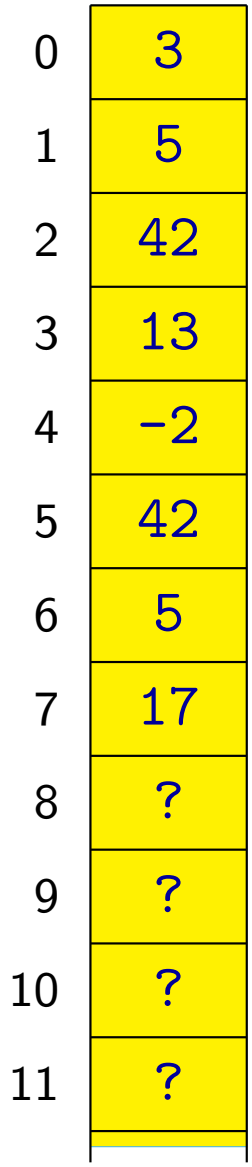


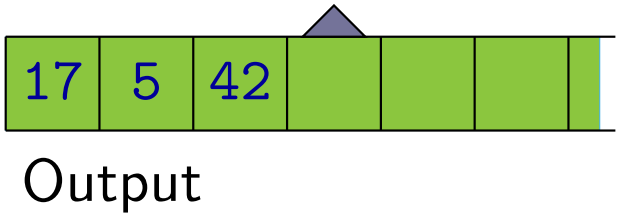


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

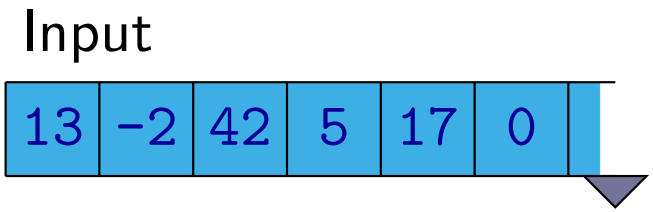


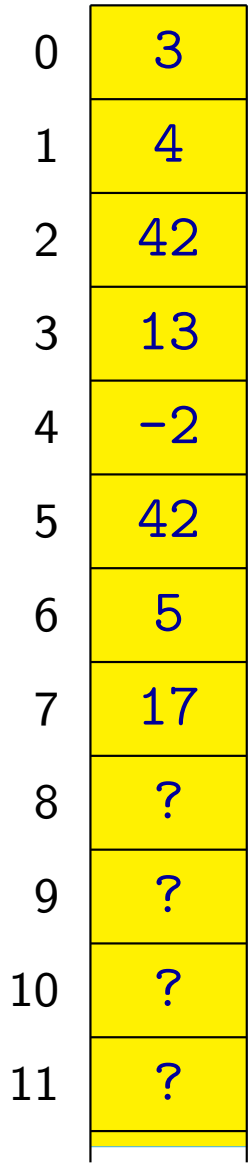


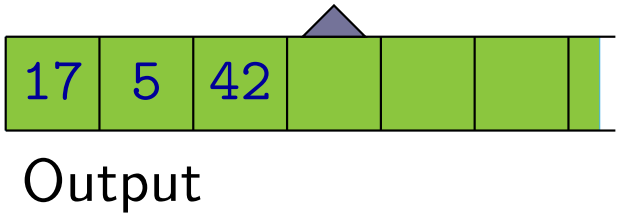


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

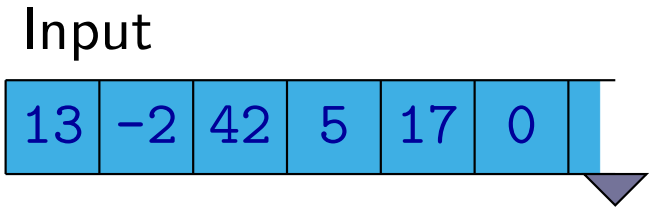


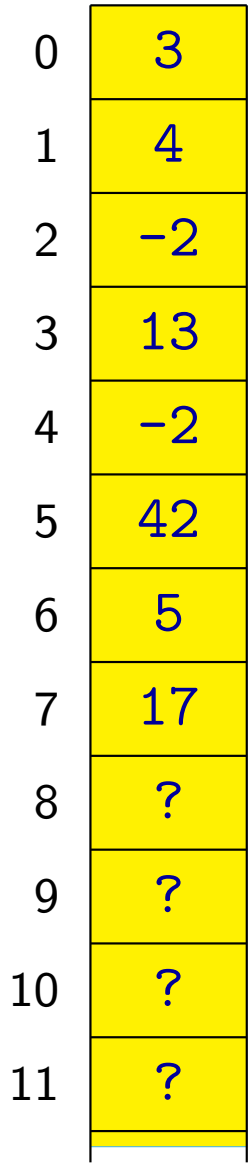


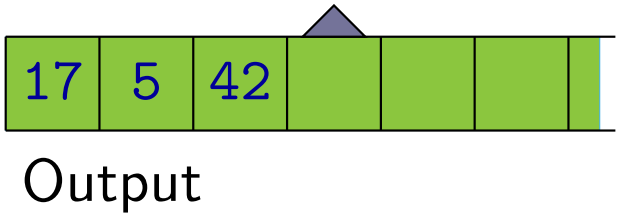


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

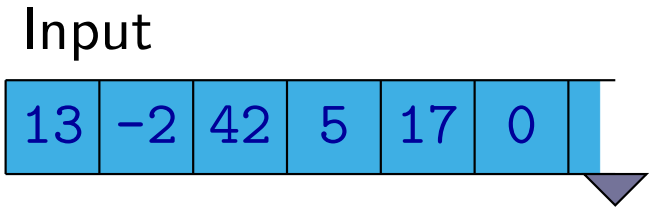


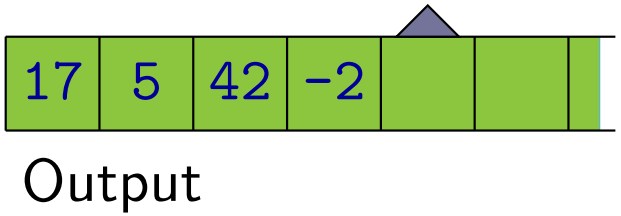


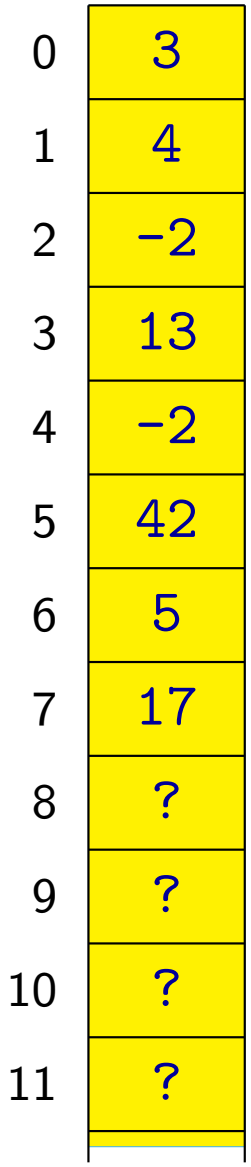


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

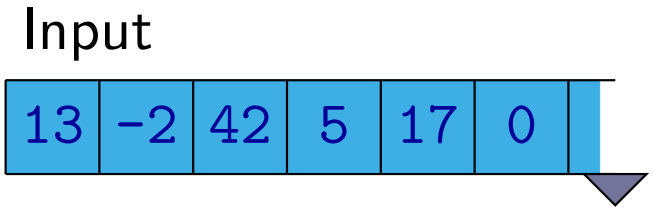


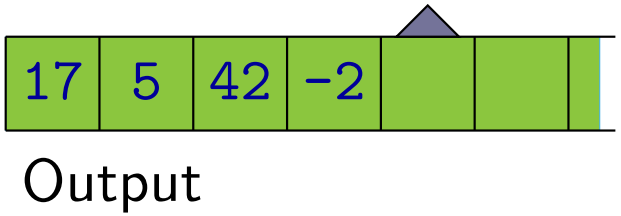


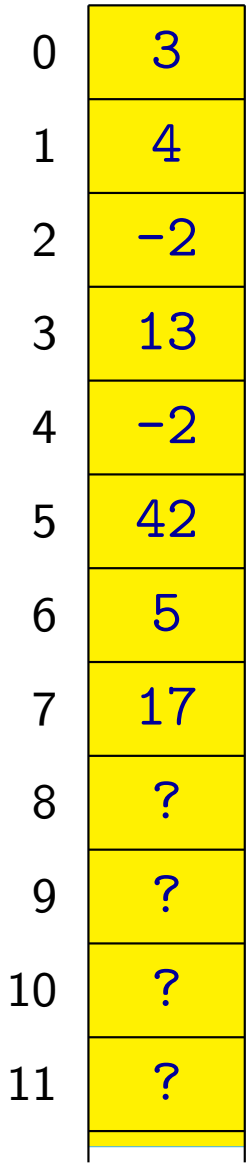


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

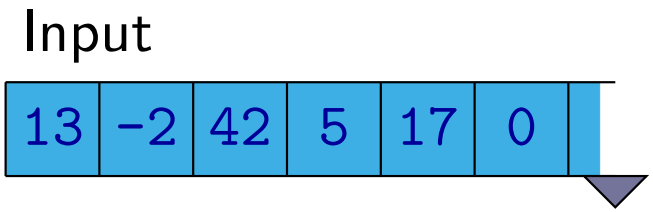


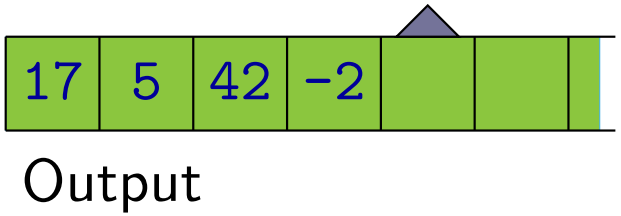


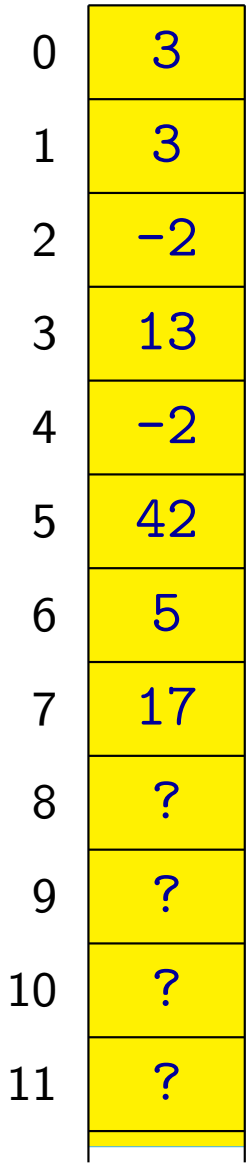


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

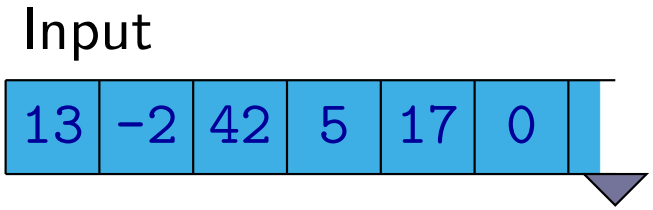


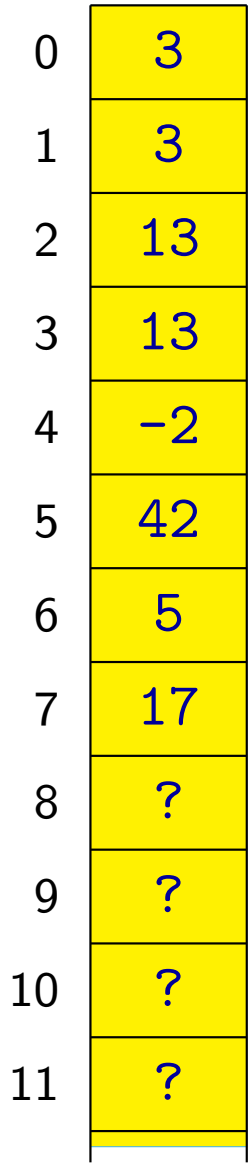


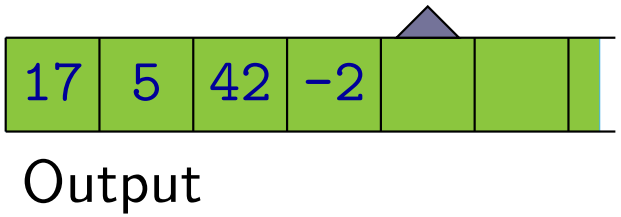


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

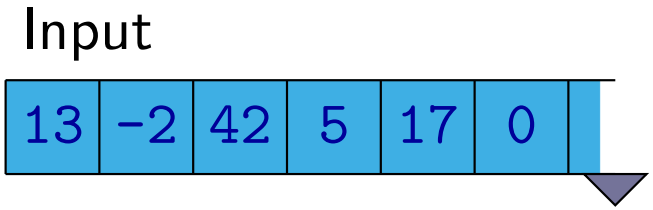


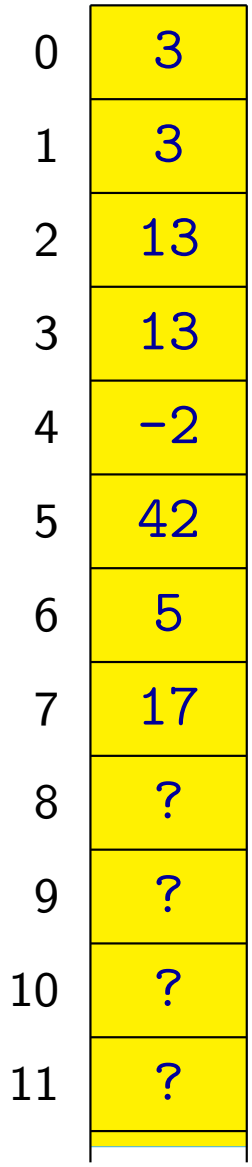


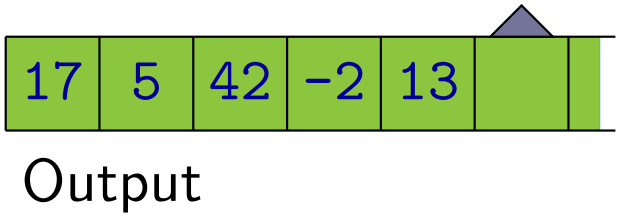


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt

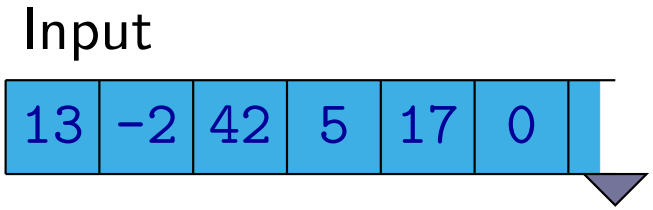


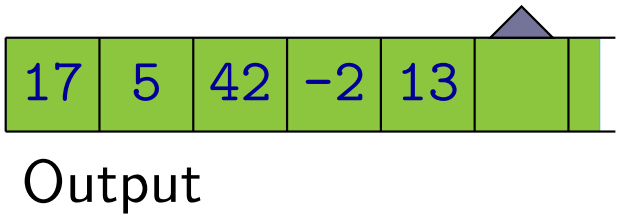


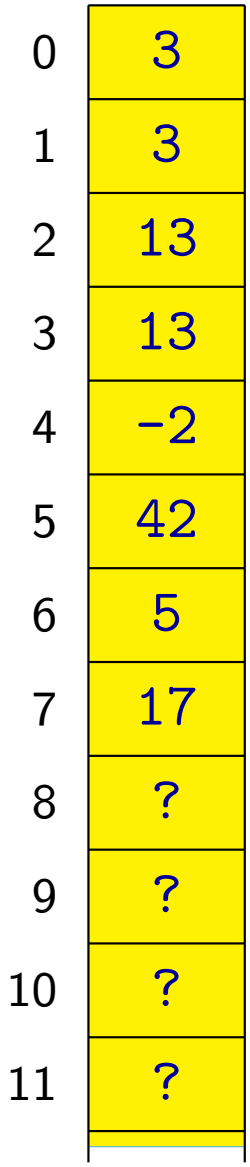


Random Access Machine

$R_0 := 3$
$R_1 := R_0$
$L_1 : R_2 := \text{READ}()$
if $(R_2 = 0)$ **goto** L_3
$[R_1] := R_2$
$R_1 := R_1 + 1$
goto L_1
$L_2 : R_1 := R_1 - 1$
$R_2 := [R_1]$
$\text{WRITE}(R_2)$
$L_3 :$ **if** $(R_1 > R_0)$ **goto** L_2
halt







Random Access Machine

Main differences with respect to real computers:

- The size of memory is not limited (an address can be an arbitrary natural number).
- The size of a content of individual memory cells is not limited (a cell can contain an arbitrary integer).
- It reads data sequantially from an input that consists of a sequence of integers. The input is read-only.
- It writes data sequantially on the output that consists of a sequence of integers. The output is write-only.

Random Access Machine

- Operations like an access to a memory cell with an address less than zero or division by zero result in an error — the computation is stuck.
- For an initial content of memory there are basically two possibilities how to define it:
  - All cells are initialized with value 0.
  - Reading a cell, to which nothing has been written, results in an error.

    Cells at the beginning contain a special value (denoted here by symbol '?') that represents that the given cell has not been initialized yet.
- We could consider also variants of RAMs where memory cells (and cells of input and output) do not contain integers (i.e., the elements of set $\mathbb{Z}$) but they can contain only natural numbers (i.e., elements of set $\mathbb{N}$).

  For example, operation of subtraction ($R_i := R_j - R_k$) then behaves in such a way that whenever the result should be a negative number, then value 0 is assigned as the result.

Random Access Machine

- Different variants of RAMs can differ in what particular operations can be used in arithmetic instructions.

  For example:

  - a support of bitwise operations (and, or, not, xor, ...), bit shifts, ...
  - a variant of RAM that does not have operations for multiplication and division

- We could also consider a variant of RAM where instead of instructions of the form

  $$\textbf{if } (R_i \ rel \ R_j) \textbf{ goto } \ell \qquad \text{nebo} \qquad \textbf{if } (R_i \ rel \ c) \textbf{ goto } \ell$$

  all conditional jumps are of the form

  $$\textbf{if } (R_i \ rel \ 0) \textbf{ goto } \ell$$

  Instead of all relations $\{=, \neq, \leq, \geq, <, >\}$, only a subset of them can be supported, e.g., $\{=, >\}$.

Random Access Machine

- In some variants of RAM, the input and output are not in a form of sequence of numbers.

  Instead, such machine could work with input and output tapes containg sequences of symbols from some alphabet, e.g., $\{0, 1\}$.

  This machine then could have for example some instructions that allow the branch the computation according to a symbol read from the input.

  However, the internal memory even in this variant works with numbers.

- When a machine should produce an answer of the form Yes/No (i.e., to accept or reject the given input), it does not need to have an output tape.

  Instruction **halt** is then replaced with instructions **accept** and **reject**.

Random Access Machine

- In the standard definition of RAM, jump instructions jumping to an adress stored in some memory cell are usualy not considered:

  goto R_i

  RAM could be extended with these instructions.

- For RAMs, a code of a program is usually stored in a separate read-only memory, not in a working memory.

  So the code can not be modified during a computation.

Random Access Machine

- A type of a machine, similar to RAM, but where its program is stored in its working memory (instructions are encoded by numbers) and so it can be modified during a computations, is called **RASP** (**random-access stored program**).

  RASP can simulate behaviour of self-modifying programs.

Implementation of Algorithms as RAM

As a useful intermediate step in the translation from a high-level programming language, we can use **control-flow graph**.

Moreover, the following issues must be resolved:

- Where values of individual variables will be stored in memory (on which particular addresses).
- How more complicated data types (arrays, structs, ...) will be stored in memory — how many cells will they use, what will represent individual cells, etc.
- It is necessary to work with objects in memory using pointers represented by numbers — e.g., accesses to elements of a structure (**struct**) or indexing elements of an array need to use pointer arithmetic.
- More complicated operations on data, which are not directly part of the instruction set, must be simulated using available simpler operations.

Turing Machines

Turing Machine

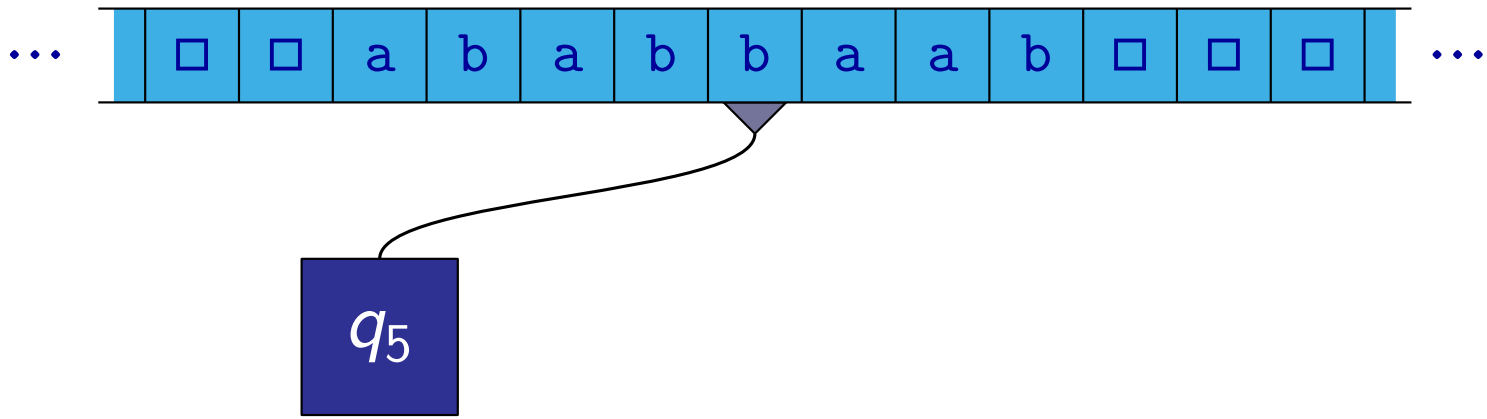
Definition

Formally, **Turing machine** is defined as a tuple $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F)$ where:

- Q is a finite non-empty set of **states**
- Γ is a finite (non-empty) set of **tape symbols** (**tape alphabet**)
- $\Sigma \subseteq \Gamma$ is a finite non-empty set of **input symbols** (**input alphabet**)
- $\delta : (Q - F) \times \Gamma \to Q \times \Gamma \times \{-1, 0, +1\}$ is a **transition function**
- $q_0 \in Q$ is an **initial state**
- $F \subseteq Q$ is a set of **final states**

We assume that $\Gamma - \Sigma$ always contains a special element $\square$ denoting a **blank** symbol.

Configurations of a Turing Machine

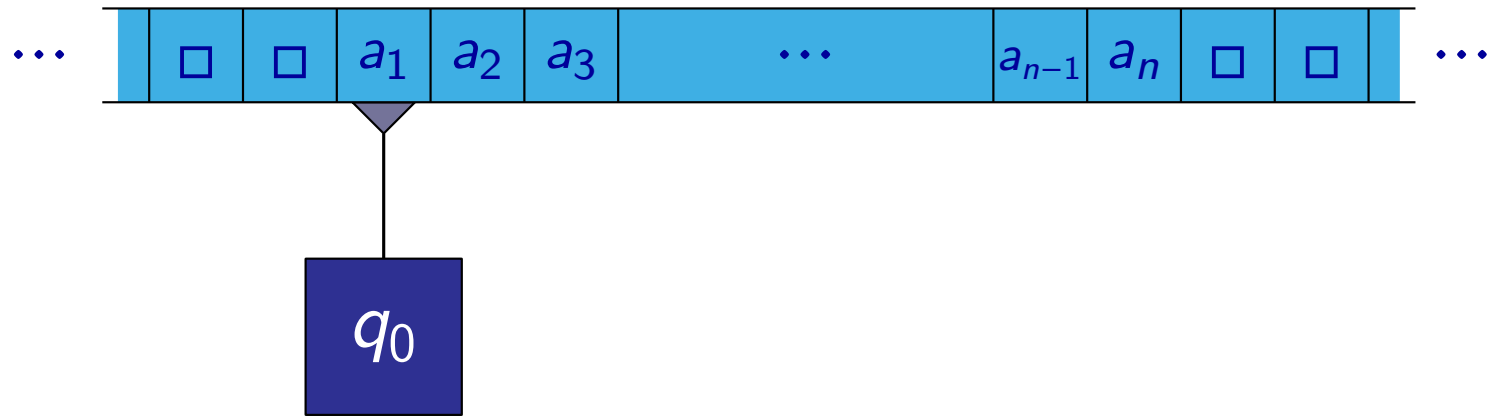


A configuration of a Turing machine is given by:

- a state of its control unit
- a content of the tape
- a position of the head

Configurations of a Turing Machine

A computation of a Turing machine $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F)$ over a word $w \in \Sigma^*$,
where $w = a_1 a_2 \cdots a_n$, starts in an **initial configuration**:



- the state of the control unit is q_0
- word w is written on the tape, remaining cells of the tape are filled with the blank symbols ($\square$)
- the head is on the first symbol of the word w (or on symbol $\square$ when $w = \varepsilon$)

Turing Machine

One step of a Turing machine:

Let us assume that:

- the state of the control unit is q
- the cell of the tape on the position of the head contains symbol b

Let us say that $\delta(q, b) = (q', b', d)$ where $d \in \{-1, 0, +1\}$.

One step of the Turing machine is performed as follows:

- the state of the control unit is changed to q'
- symbol b' is written on the tape cell on the position of the head instead of b
- The head is moved depending on d:
  - for $d = -1$ the head is moved one cell left
  - for $d = +1$ the head is moved one cell right
  - for $d = 0$ the position of the head is not changed

Turing Machine

- A Turing machine performs these steps until a state of its control unit is a state from the set F.
- Those configurations where a state of the control unit belongs to set F are **final configurations**.
- A computation ends in a final configuration.
- A computation of a machine $\mathcal{M}$ over a word w can be infinite.

Turing Machine

We often choose the set of final states $F = \{q_{acc}, q_{rej}\}$.

Then we can define for a word $w \in \Sigma^*$ if a given Turing machine accepts it:

- If the state of the control unit after the computation over the word w is q_{acc}, the machine accepts the word w.
- If the state of the control unit after the computation over the word w is q_{rej}, the machine does not accept the word w.
- The computation of the machine over the word w can be infinite. In this case the machine does not accept the word w.

The language $\mathcal{L}(\mathcal{M})$ of a Turing machine $\mathcal{M}$ is the set of all words accepted by $\mathcal{M}$.

Turing Machine

A language $L \subseteq \Sigma^*$ is **accepted** by a Turing machine $\mathcal{M}$ if:

- for each word $w \in \Sigma^*$ it holds that $w \in L$ iff the computation of $\mathcal{M}$ over w ends in final state q_{acc}.

(So computations over words that do not belong to L can end in state q_{rej} or be infinite.)

Language $L \subseteq \Sigma^*$ is **recognized** by a Turing machine $\mathcal{M}$ if:

- for each word $w \in L$ the computation of machine $\mathcal{M}$ over w ends in final state q_{acc}.
- for every word $w \in (\Sigma^* - L)$ the computation of machine $\mathcal{M}$ over w ends in final state q_{rej}.

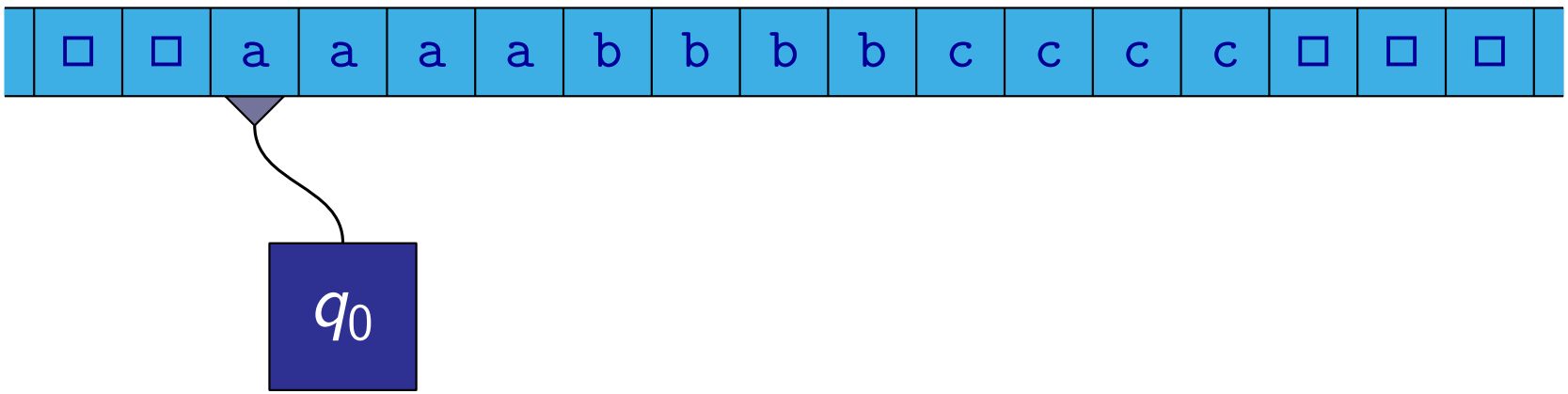
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



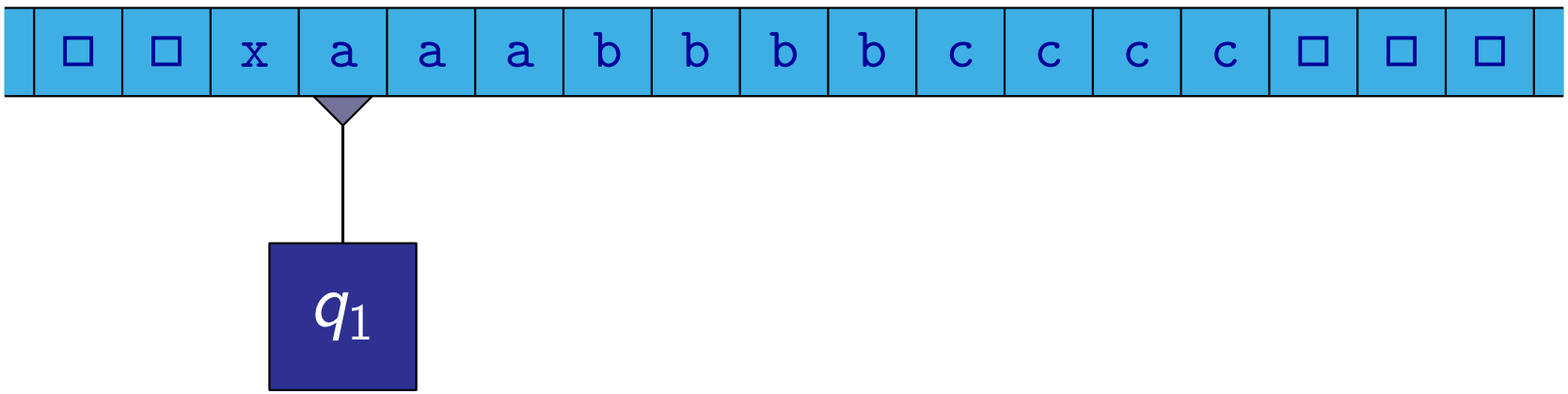
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



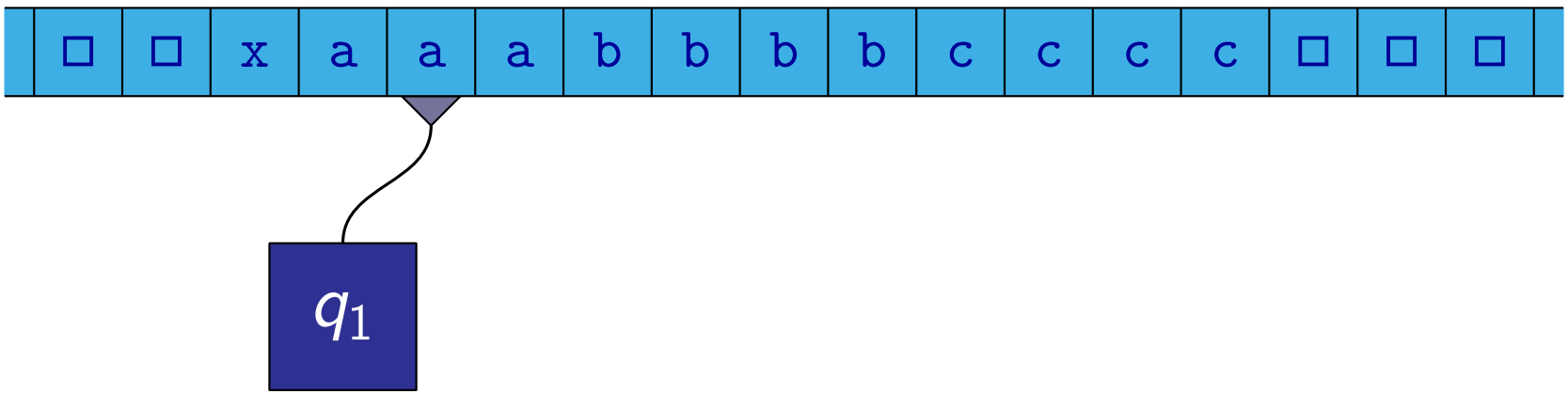
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



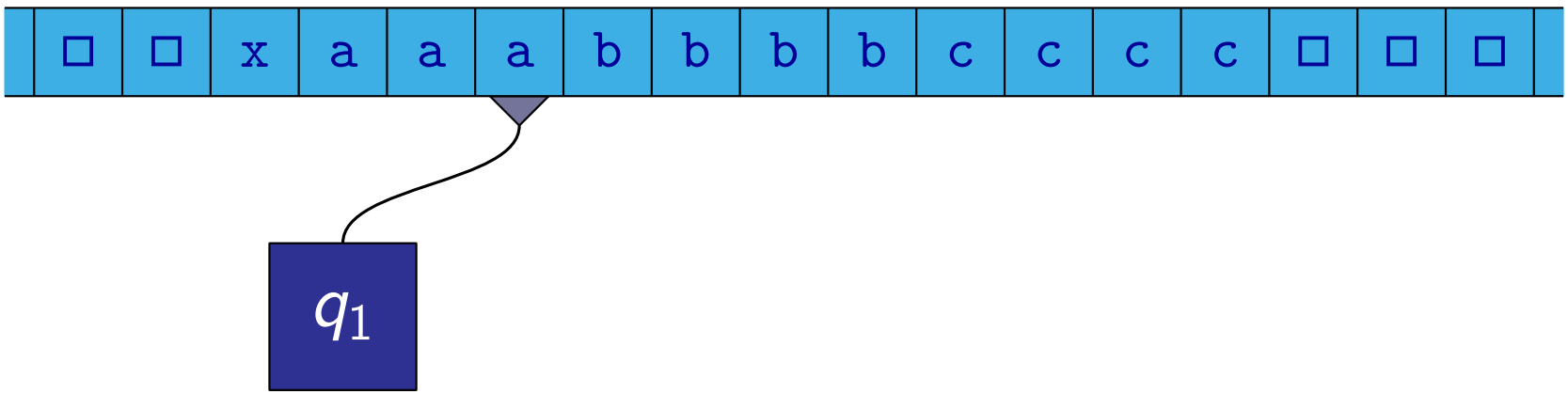
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



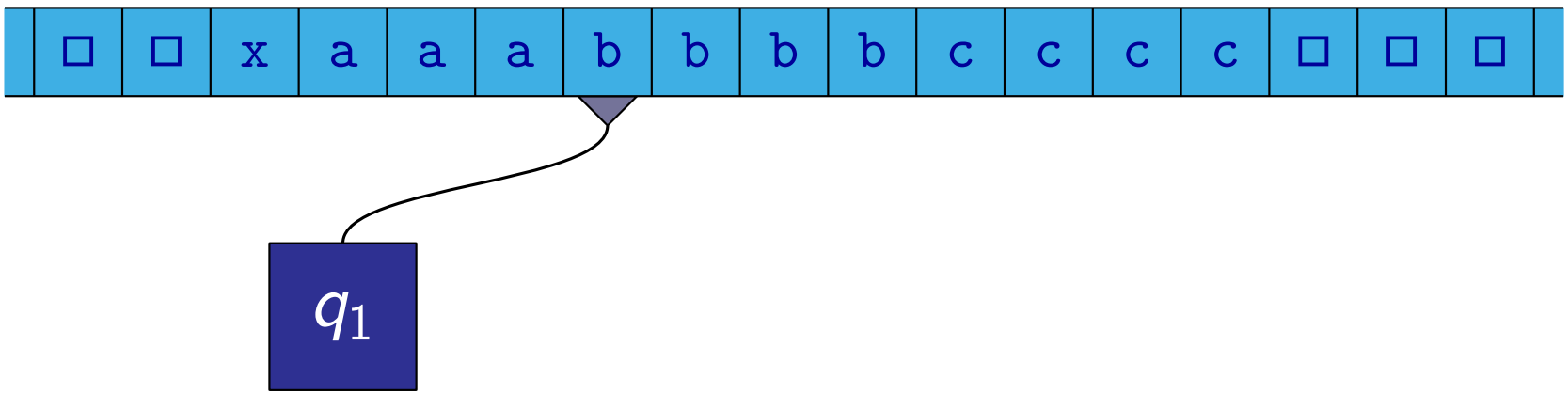
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



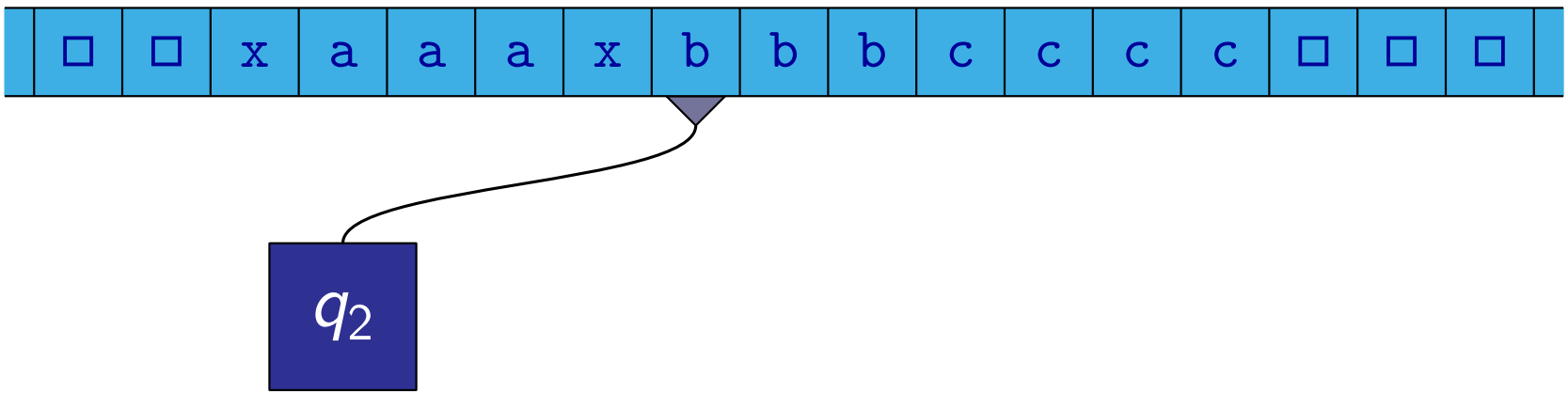
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



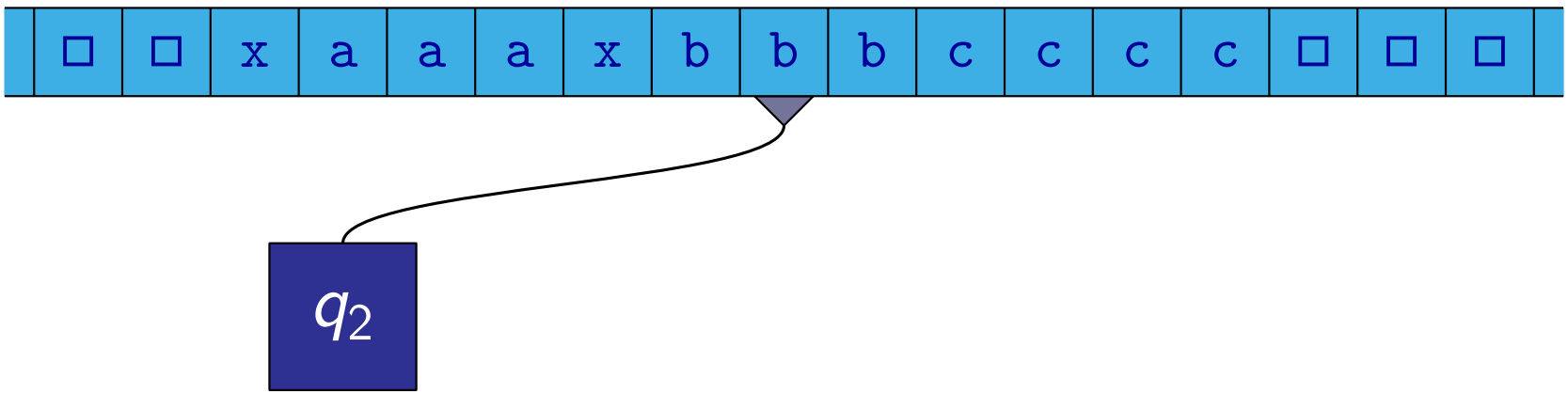
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



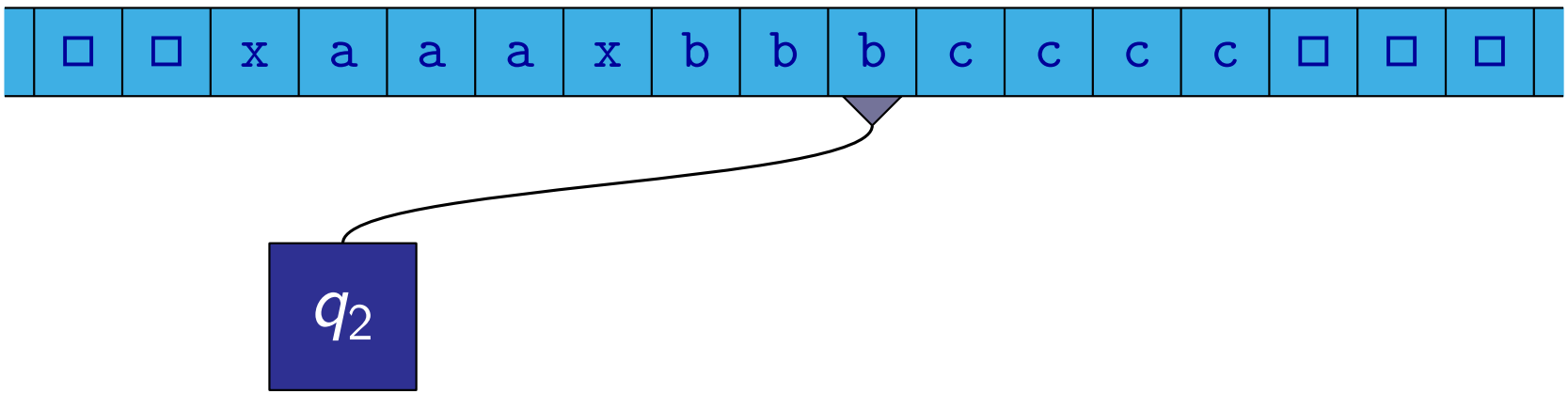
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



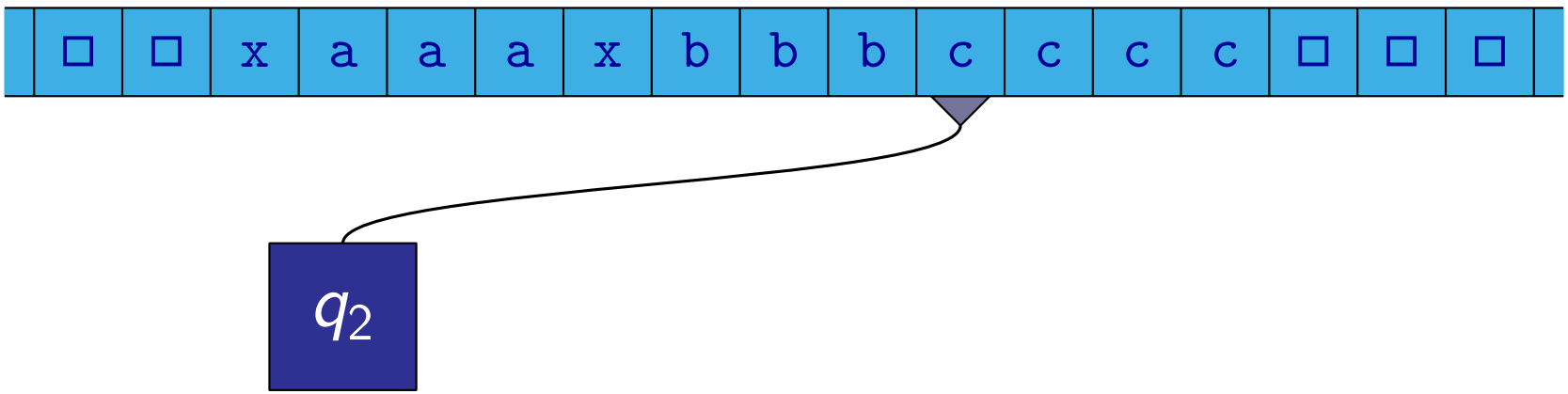
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



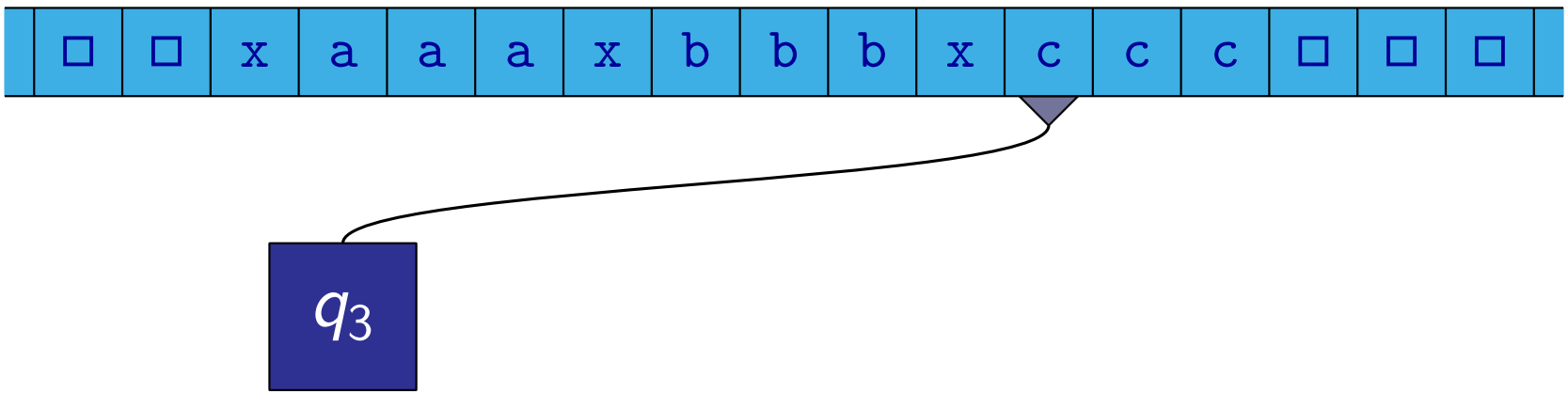
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



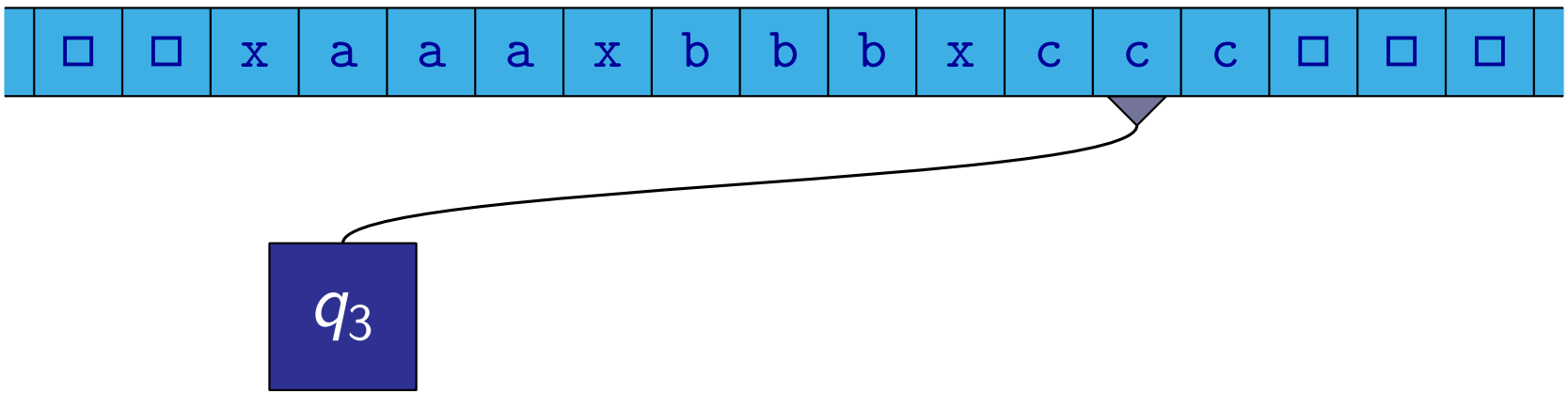
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



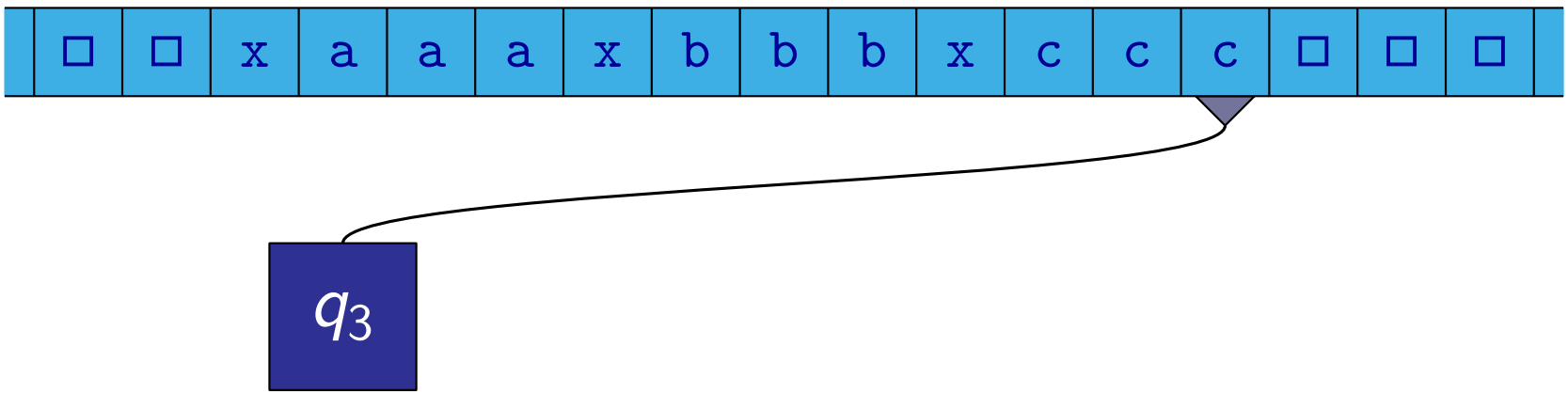
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



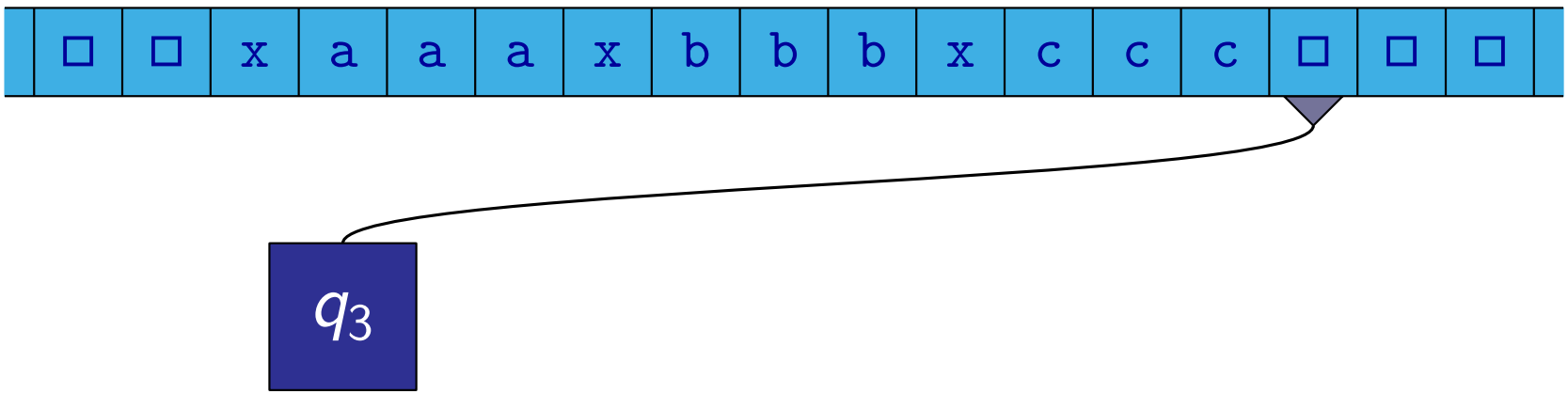
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



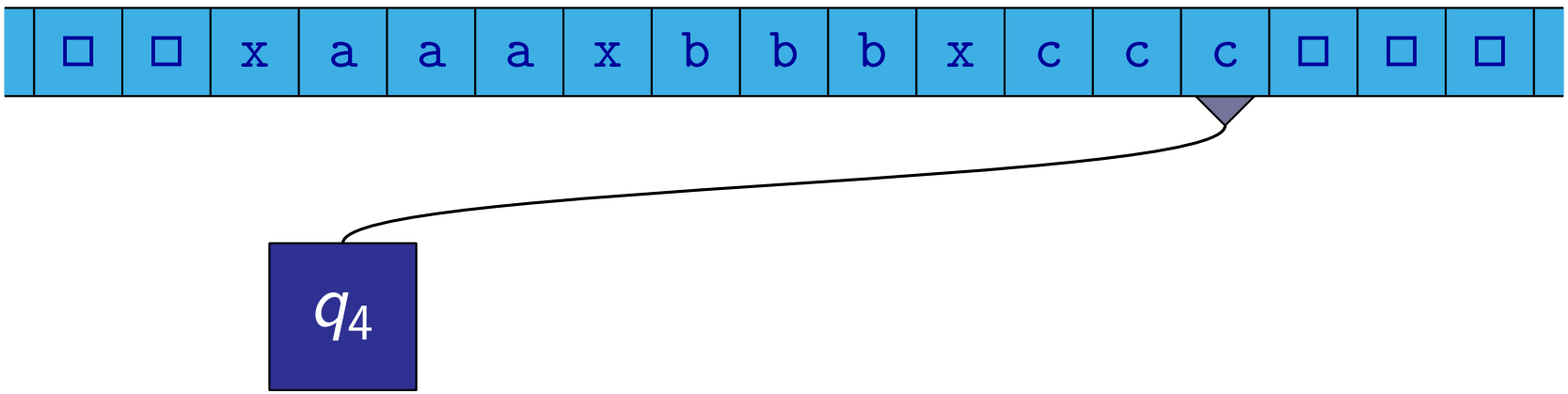
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



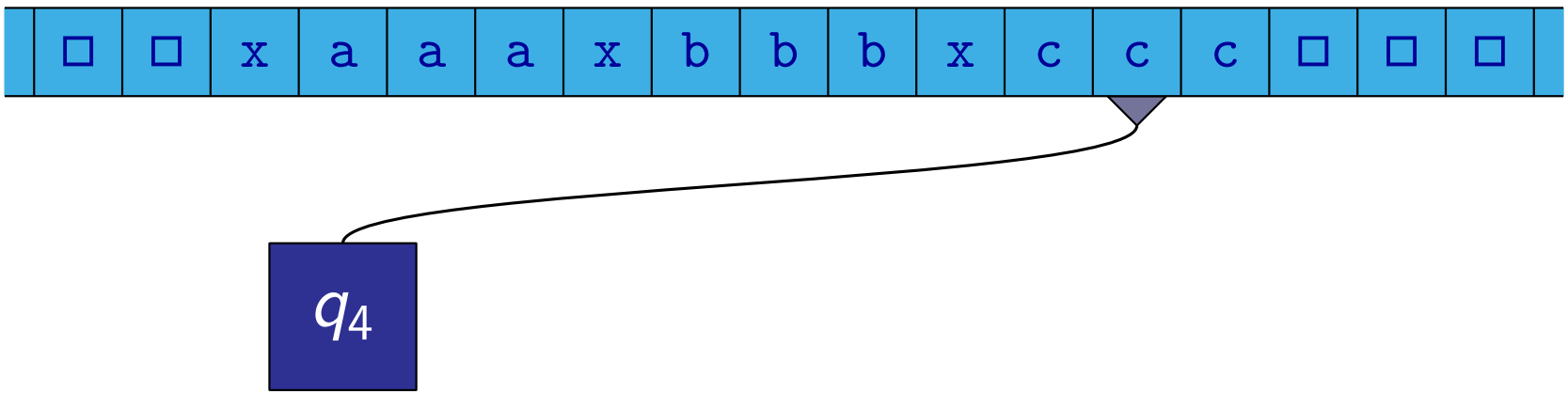
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



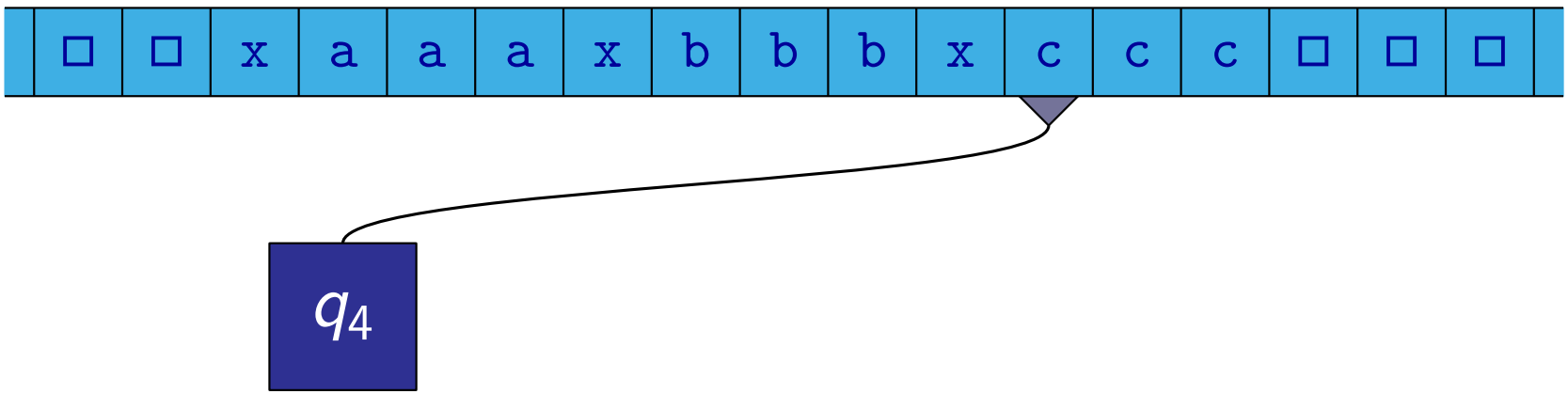
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



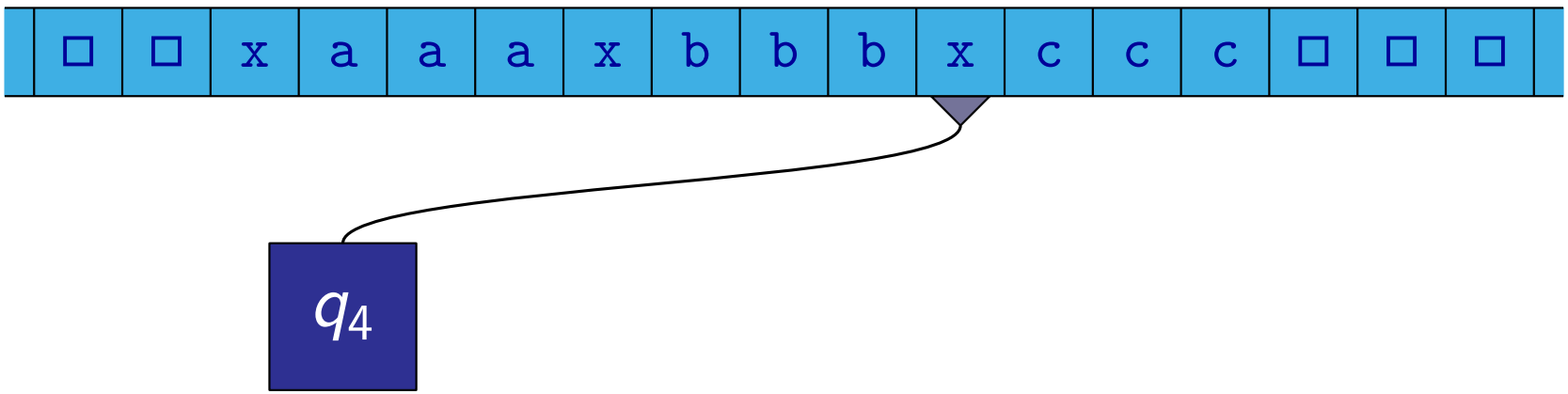
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



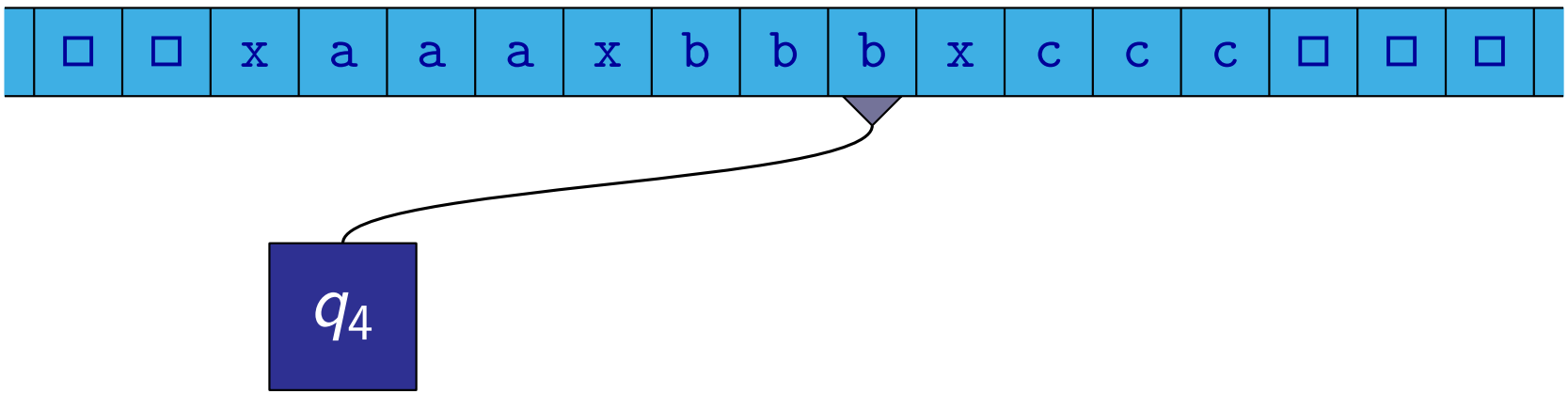
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



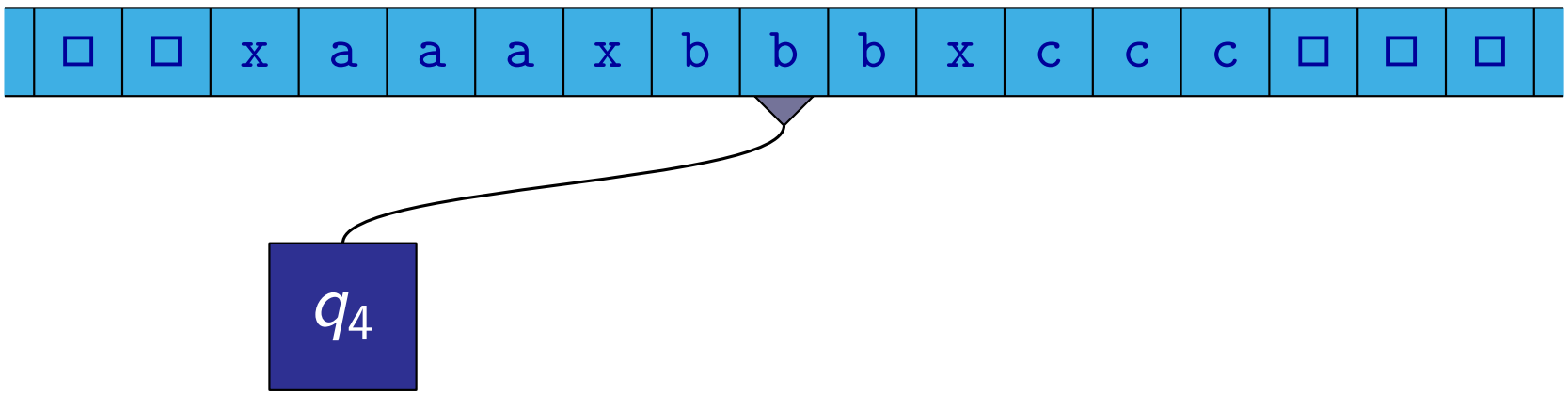
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



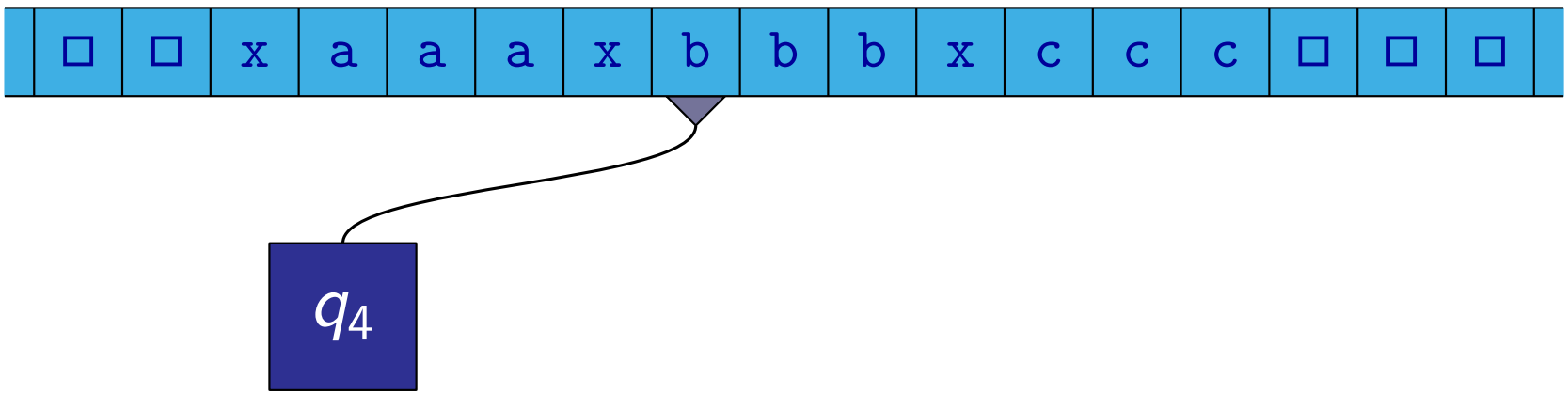
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



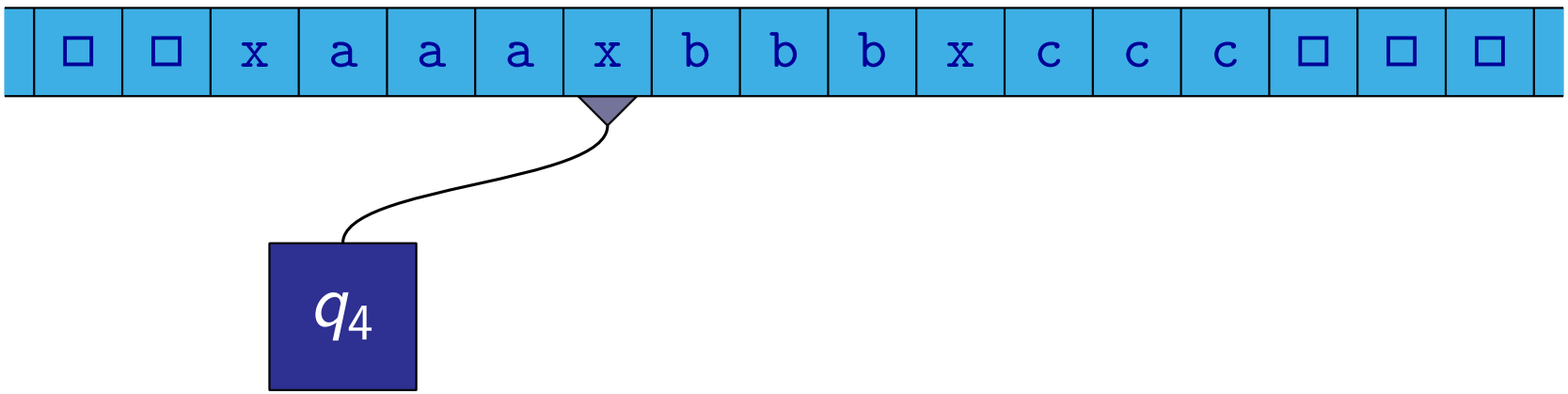
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



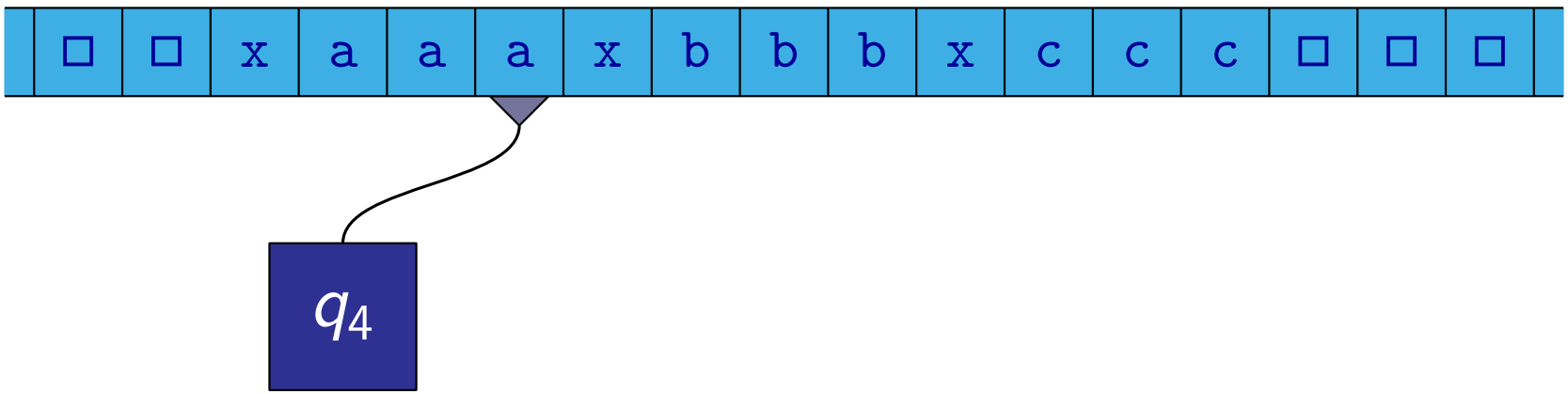
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



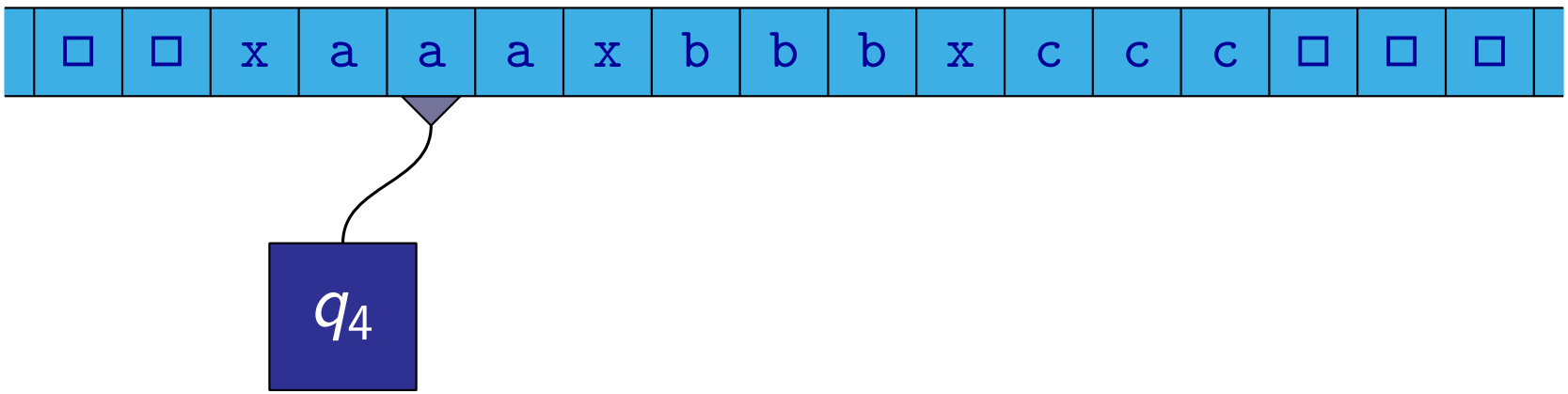
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



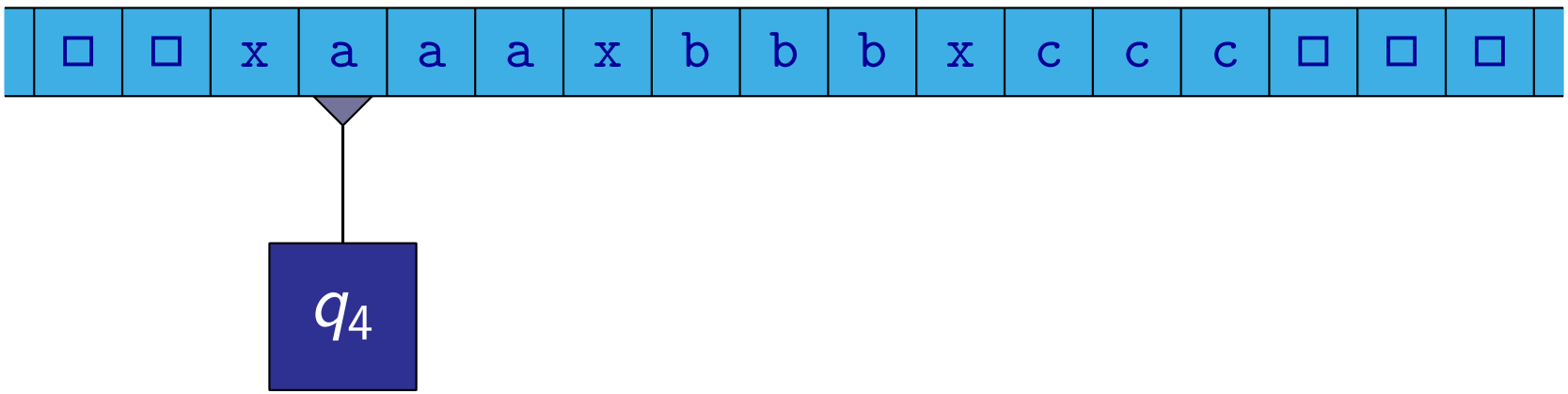
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



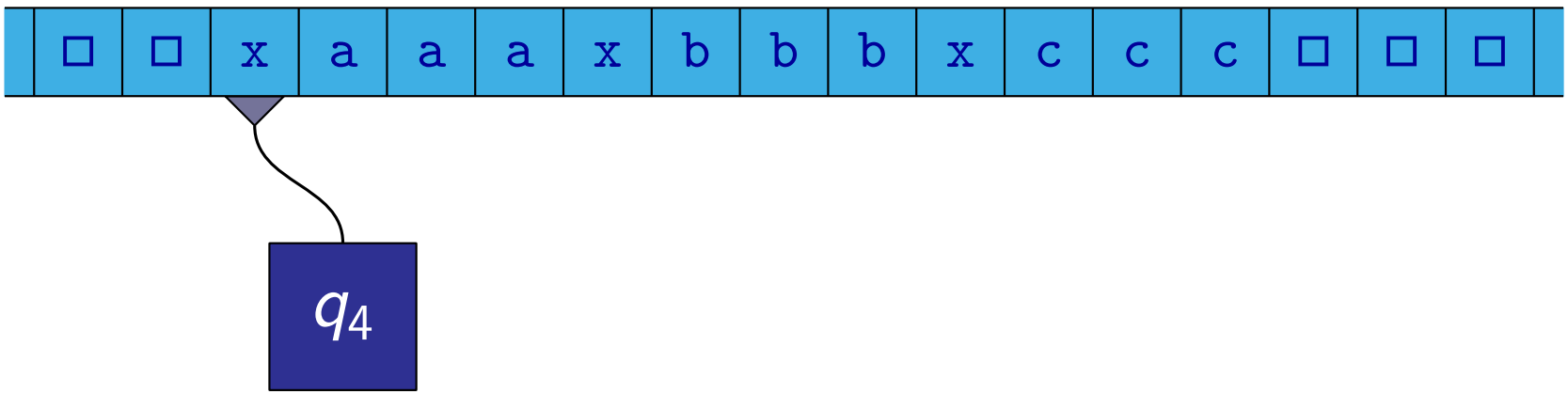
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



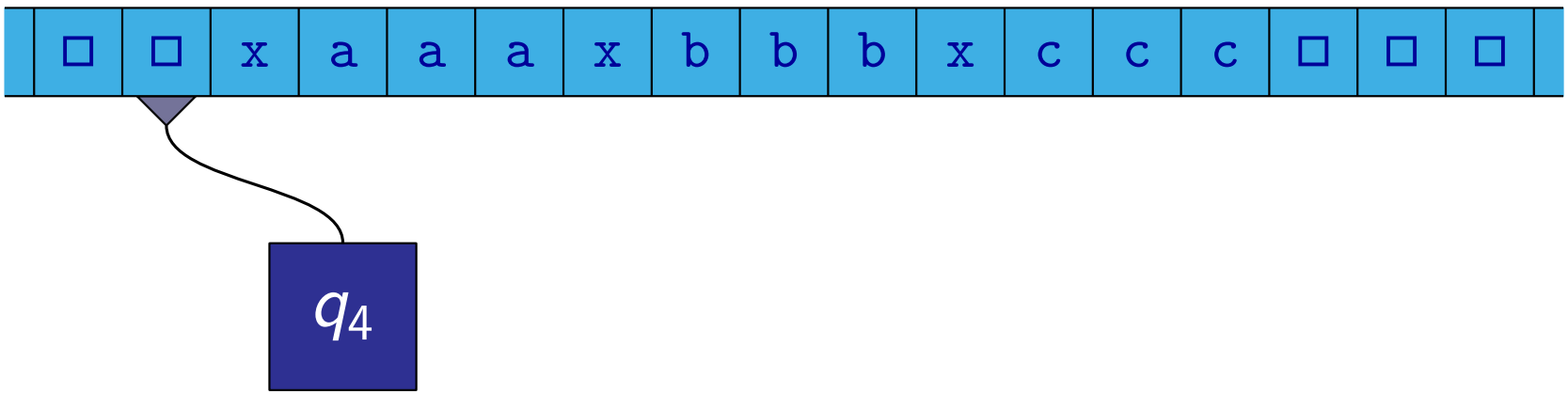
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



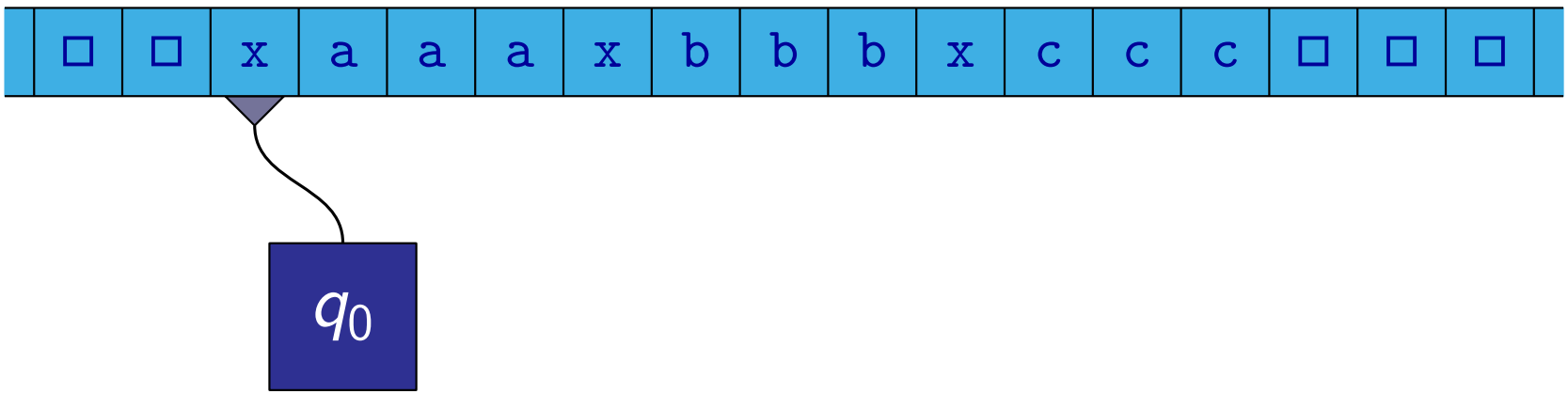
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



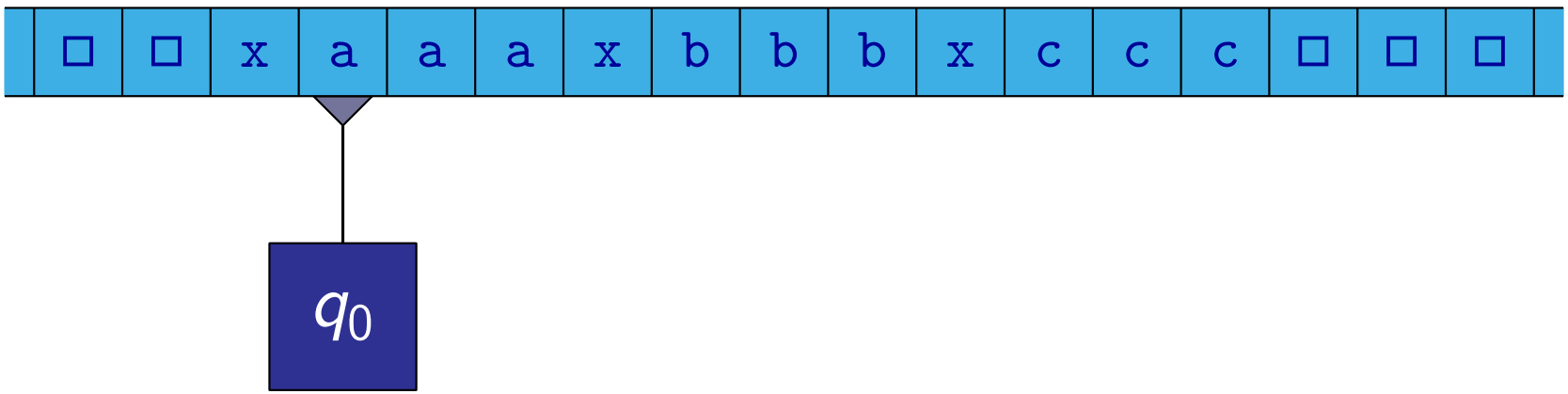
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



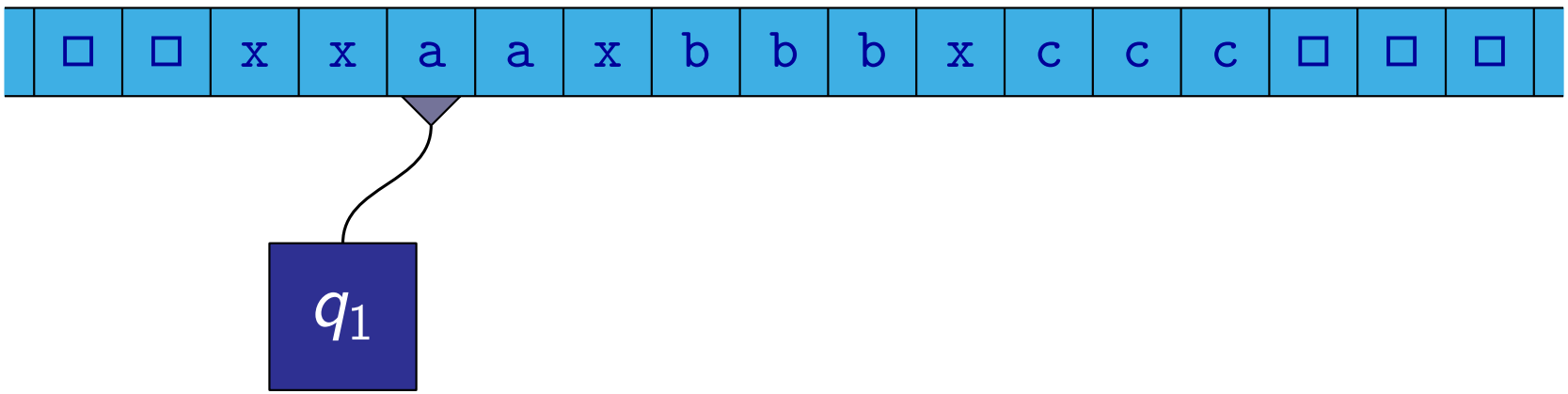
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



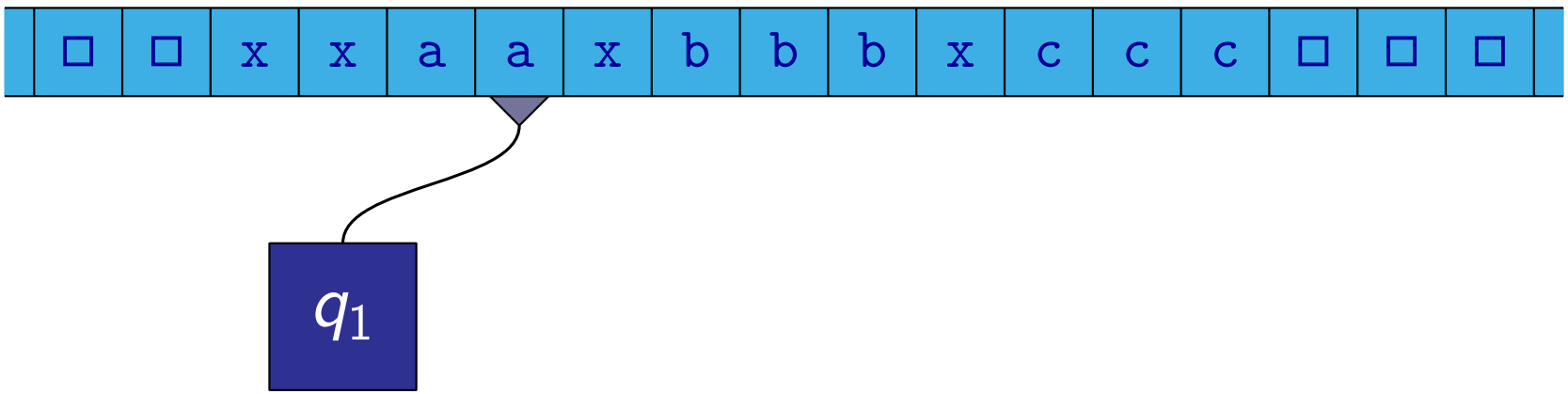
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



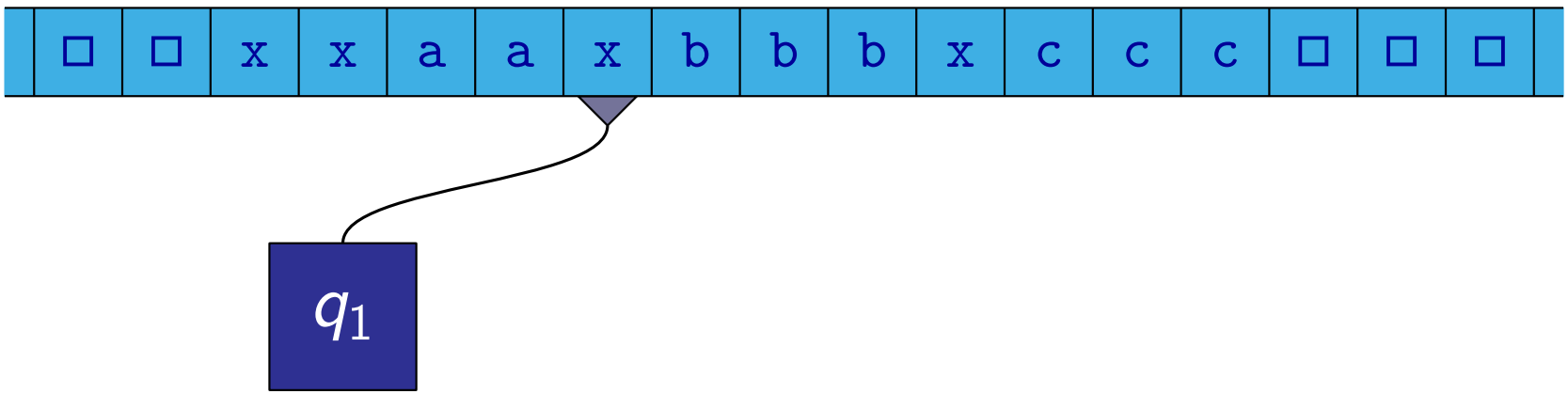
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



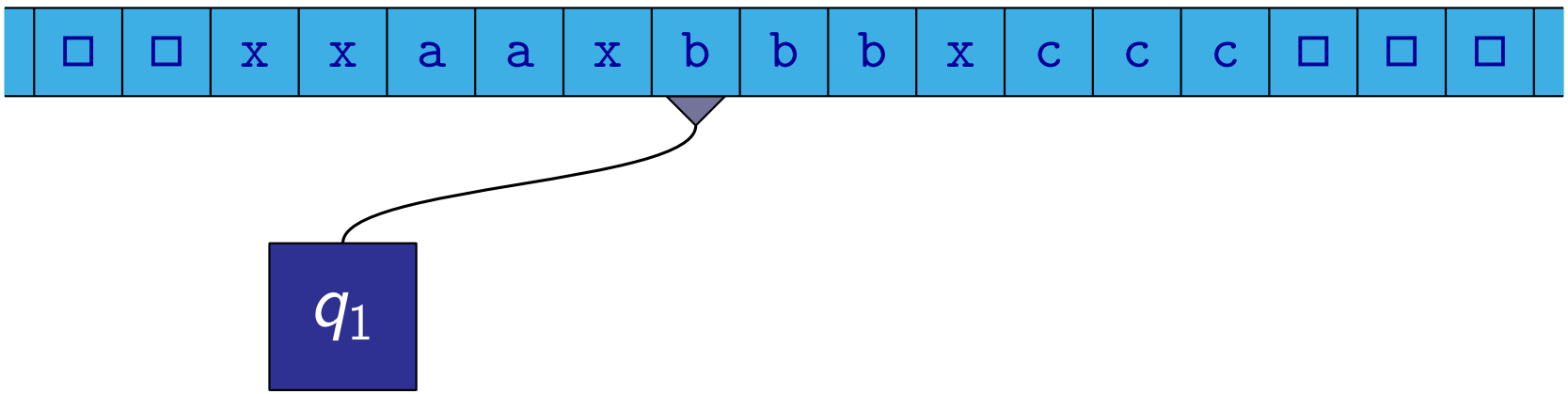
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



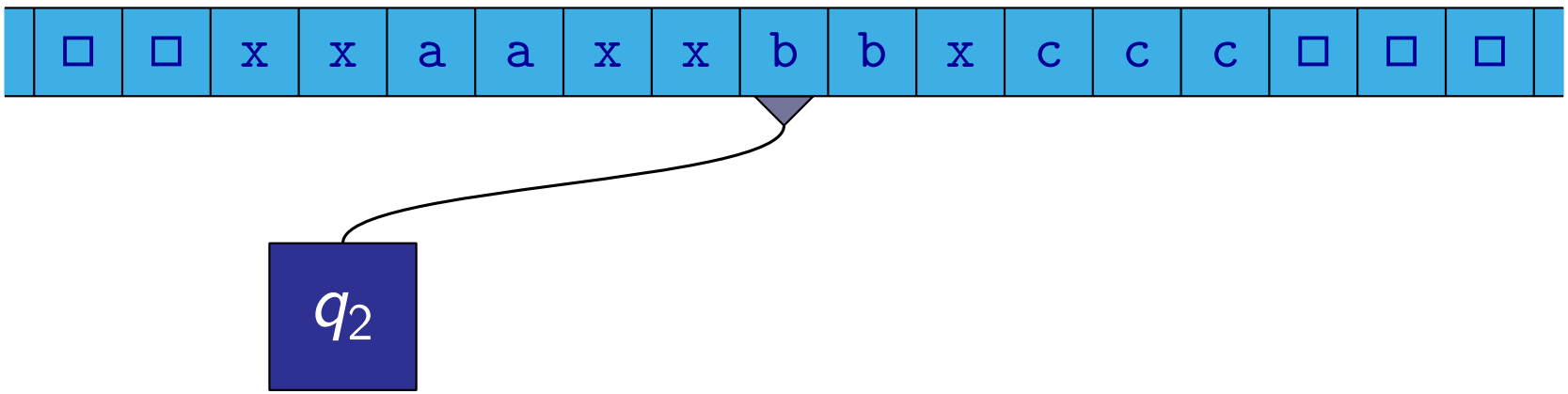
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



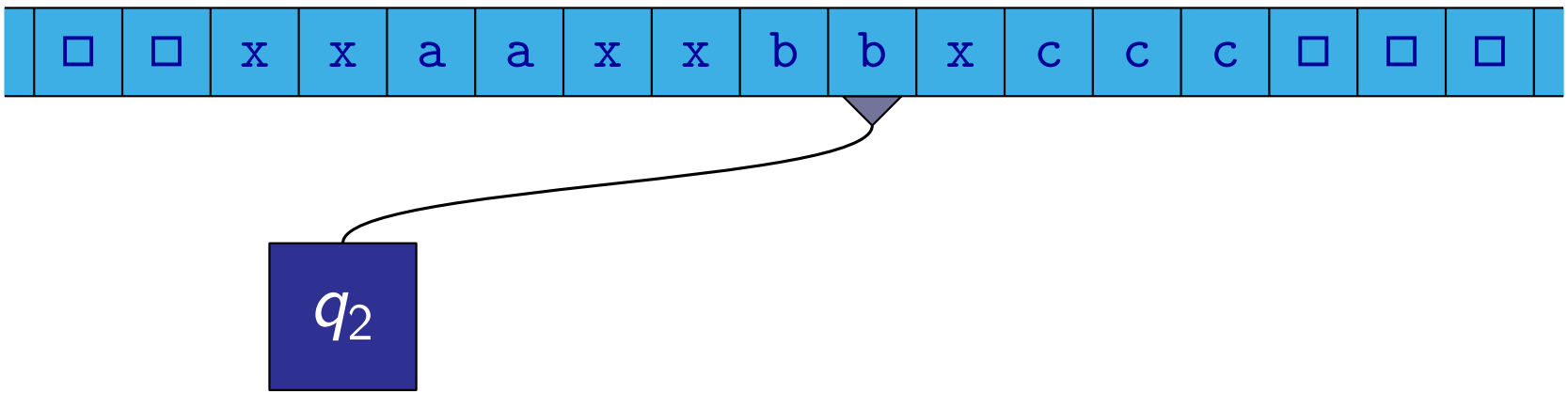
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



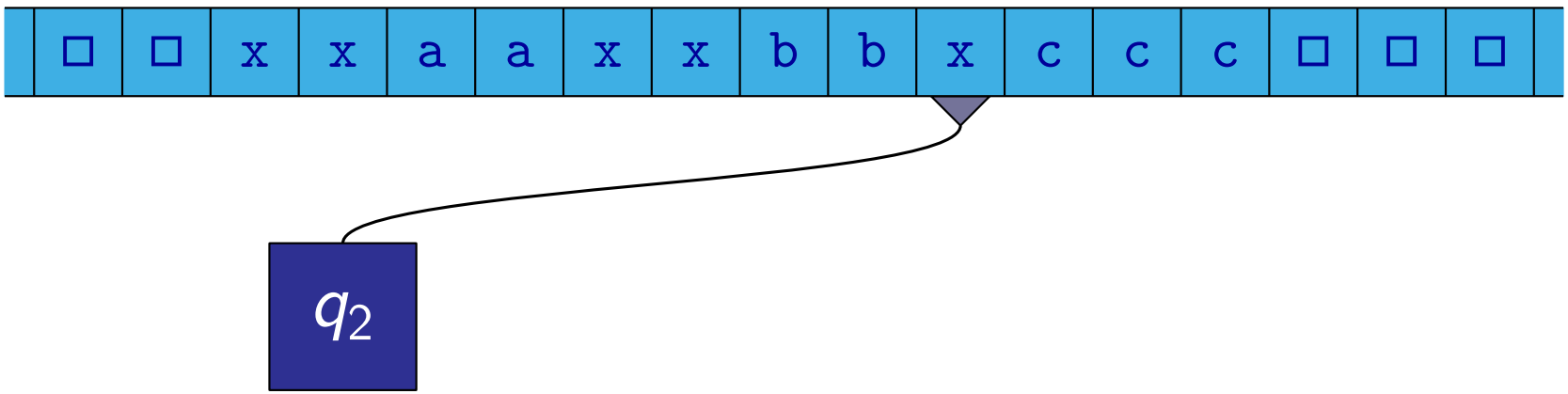
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



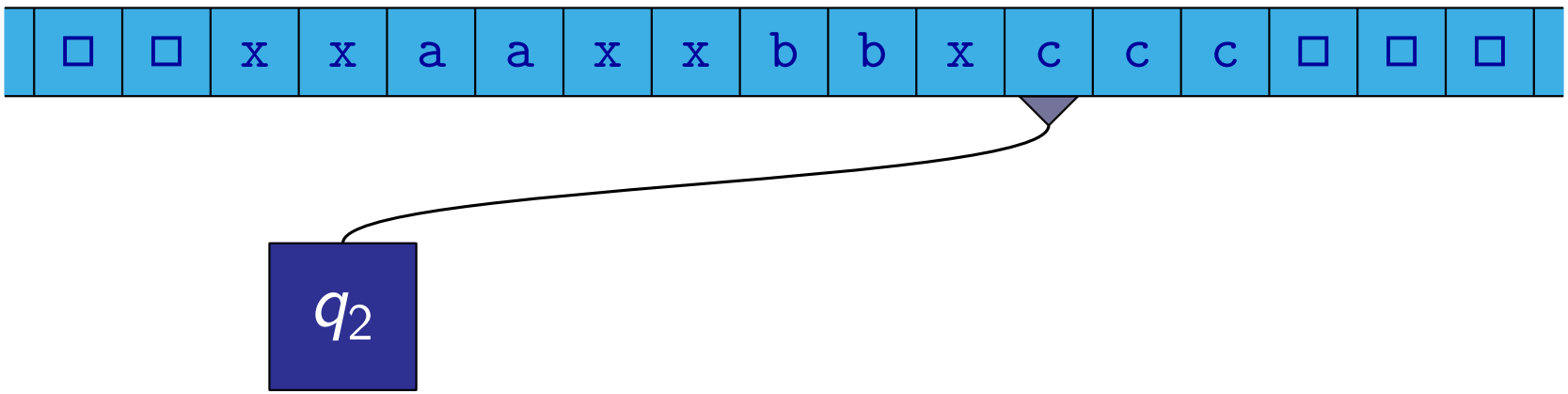
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



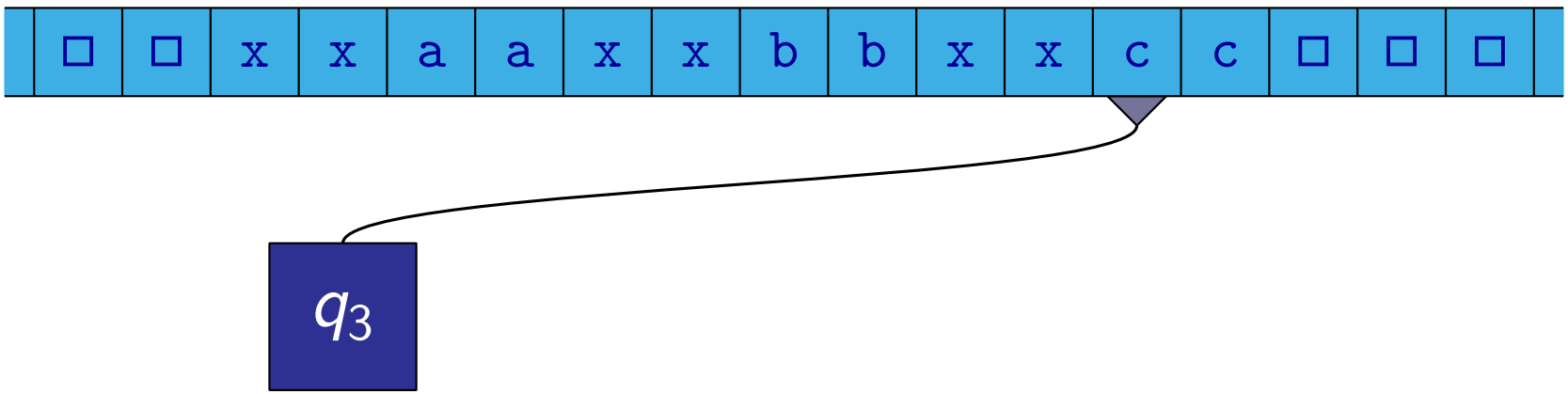
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



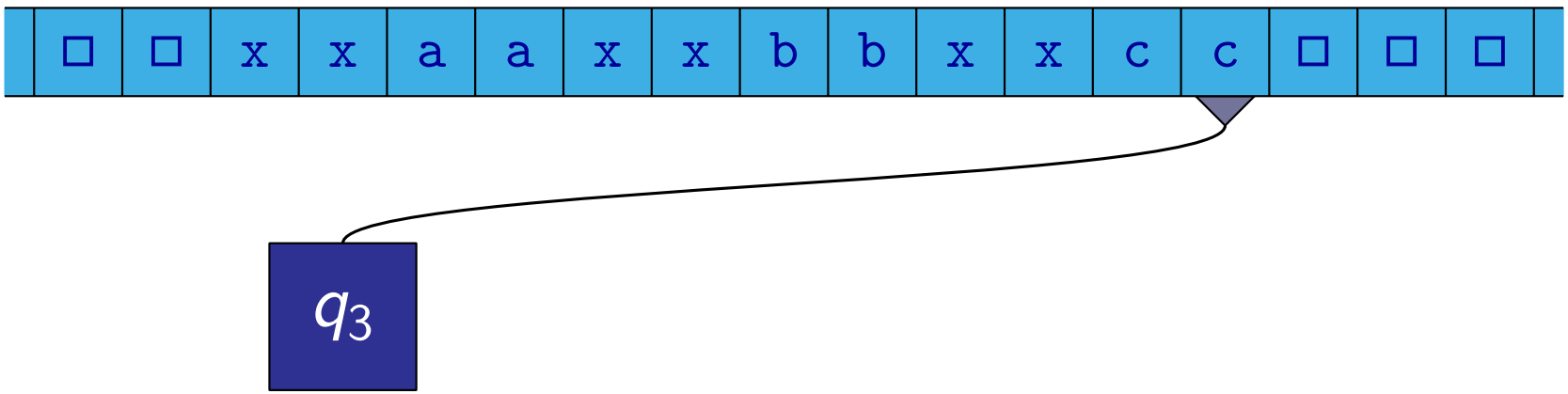
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



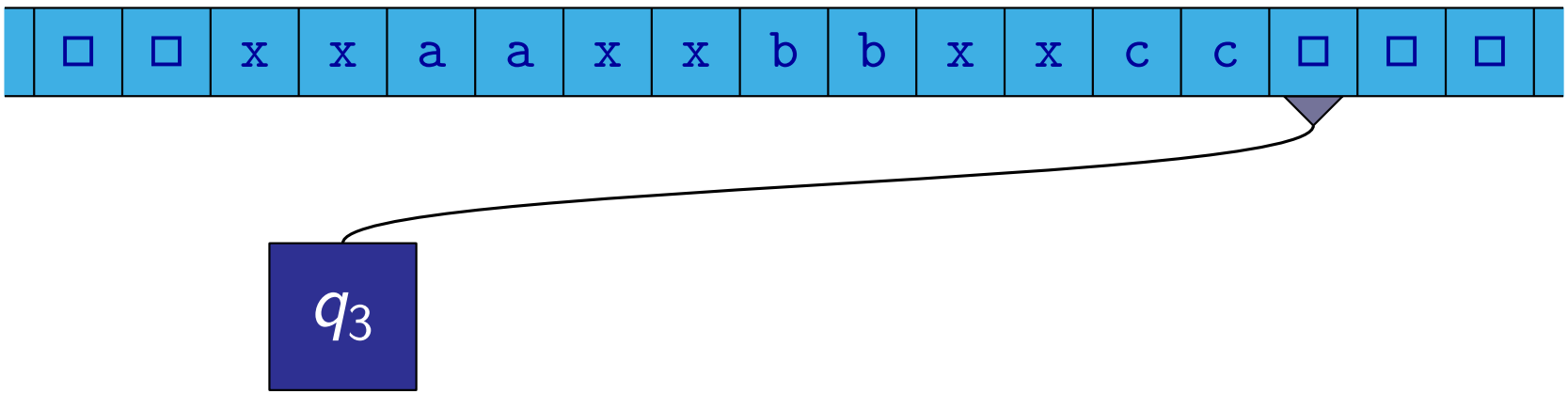
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



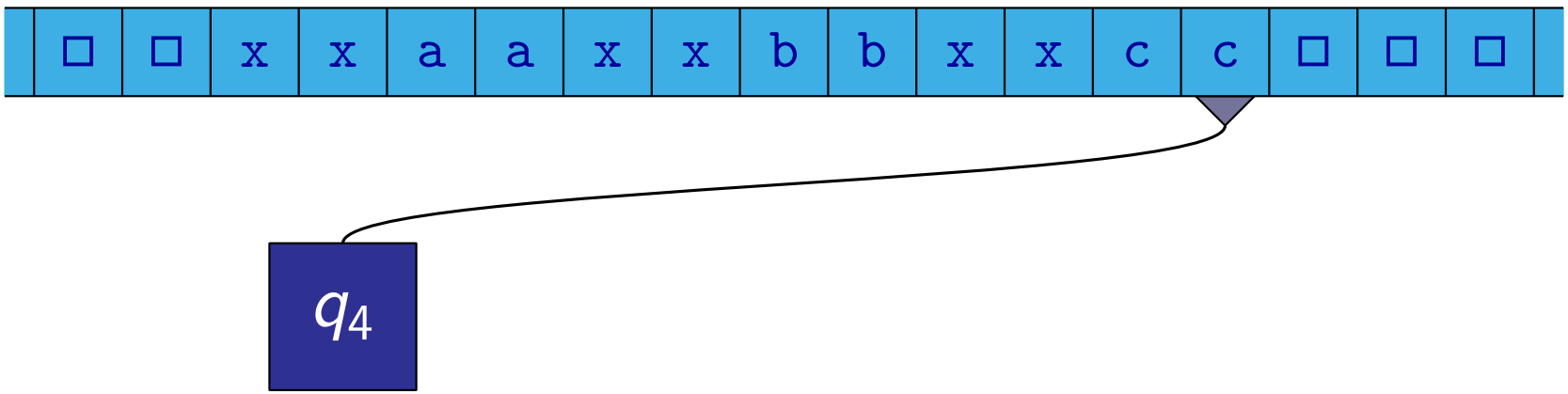
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



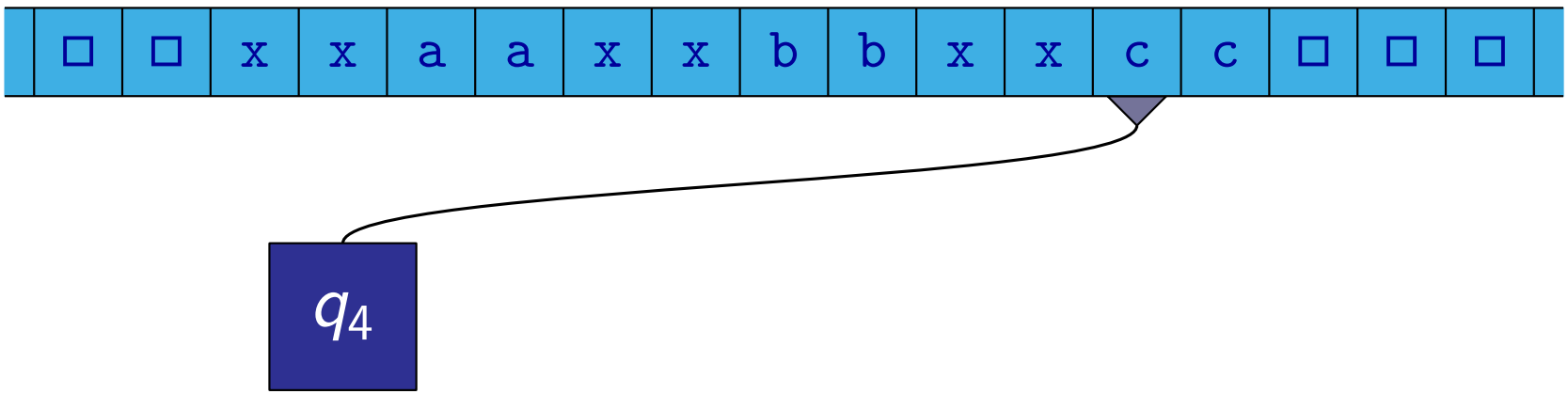
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



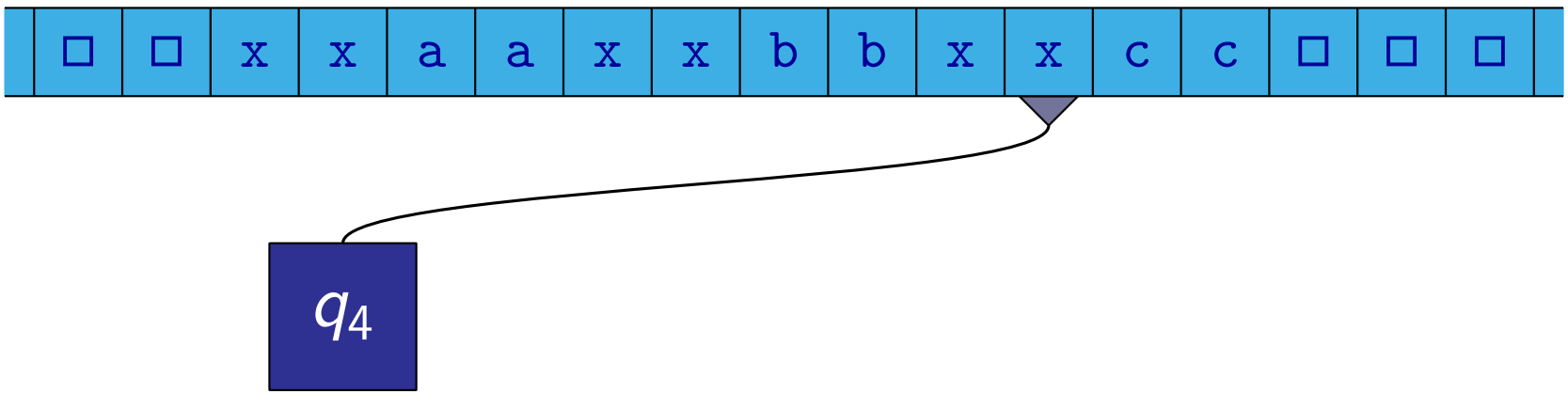
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



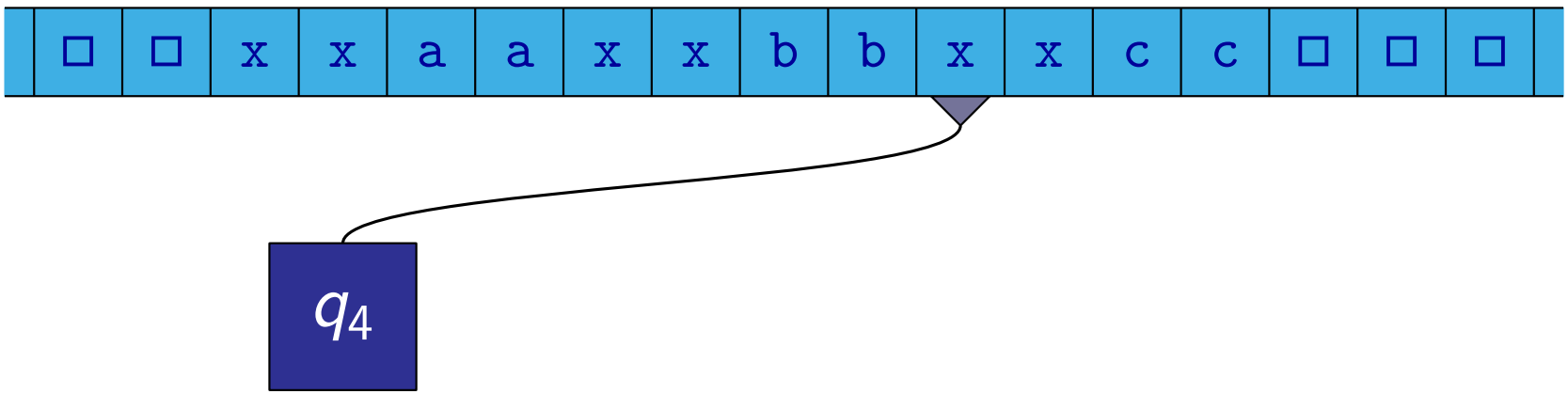
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



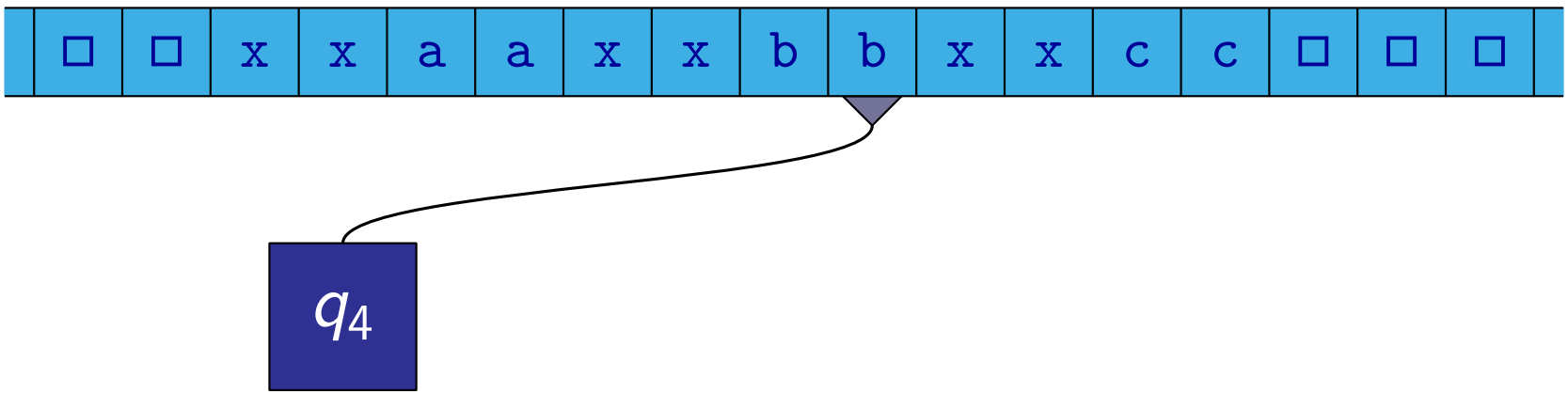
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



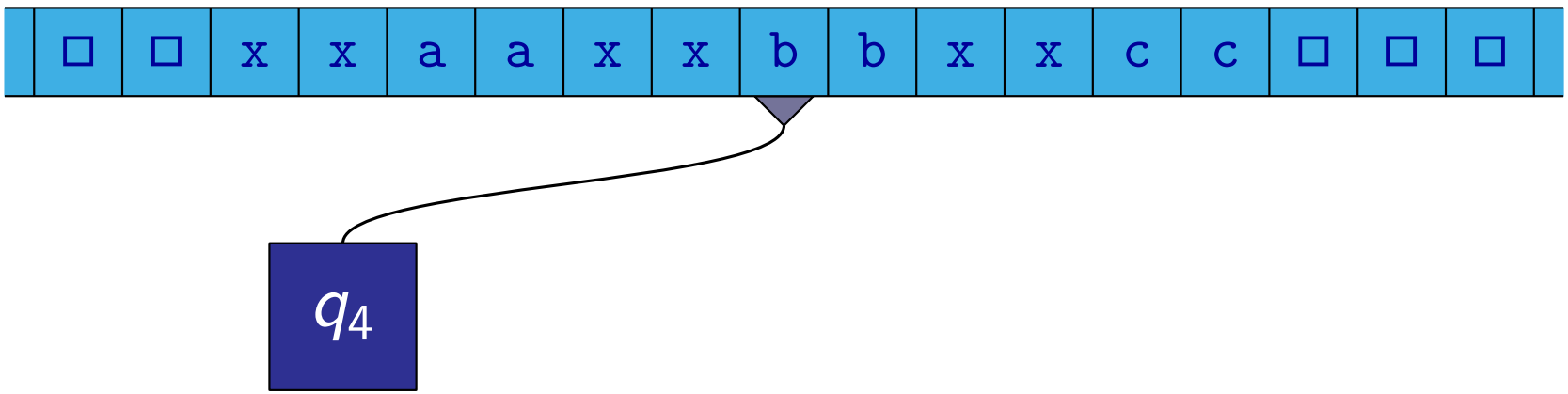
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



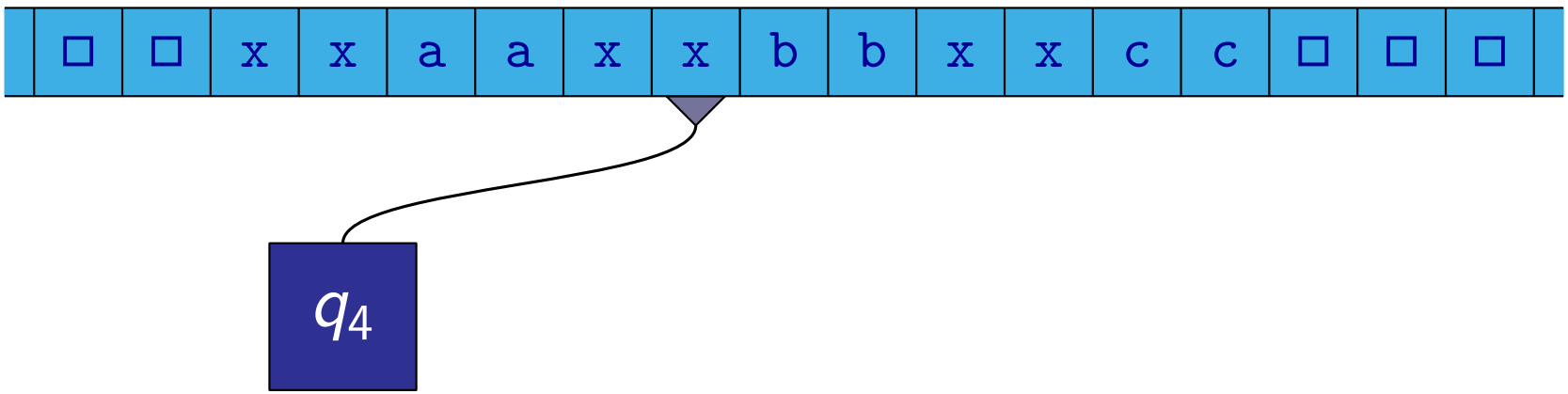
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



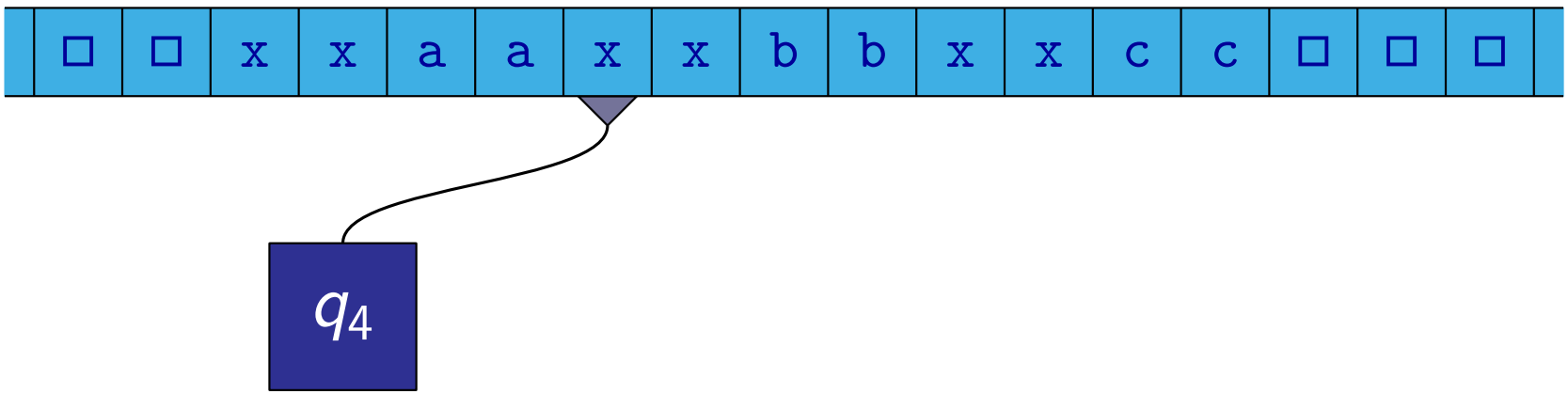
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



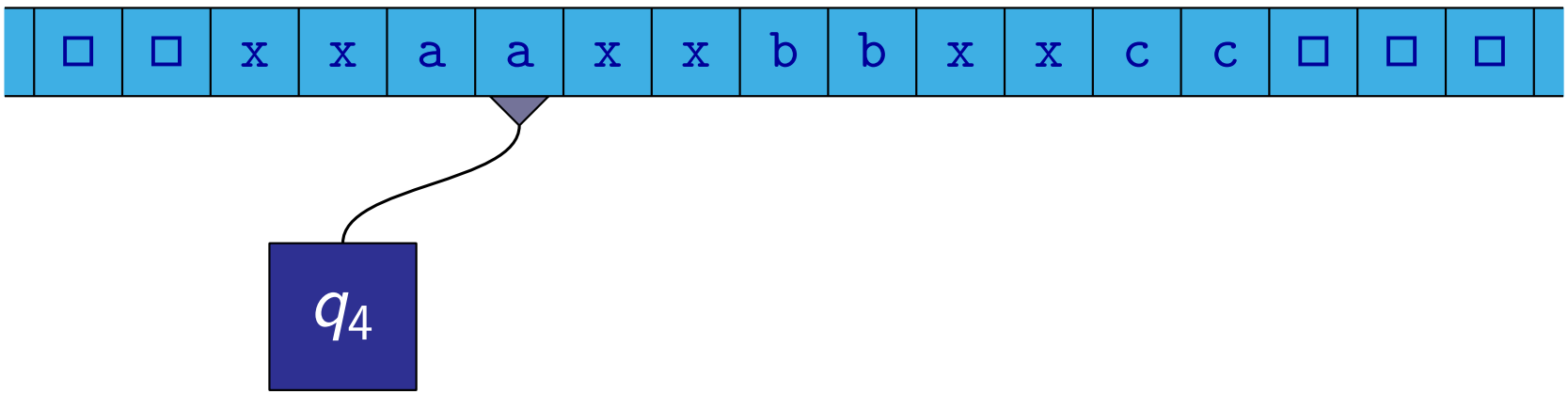
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



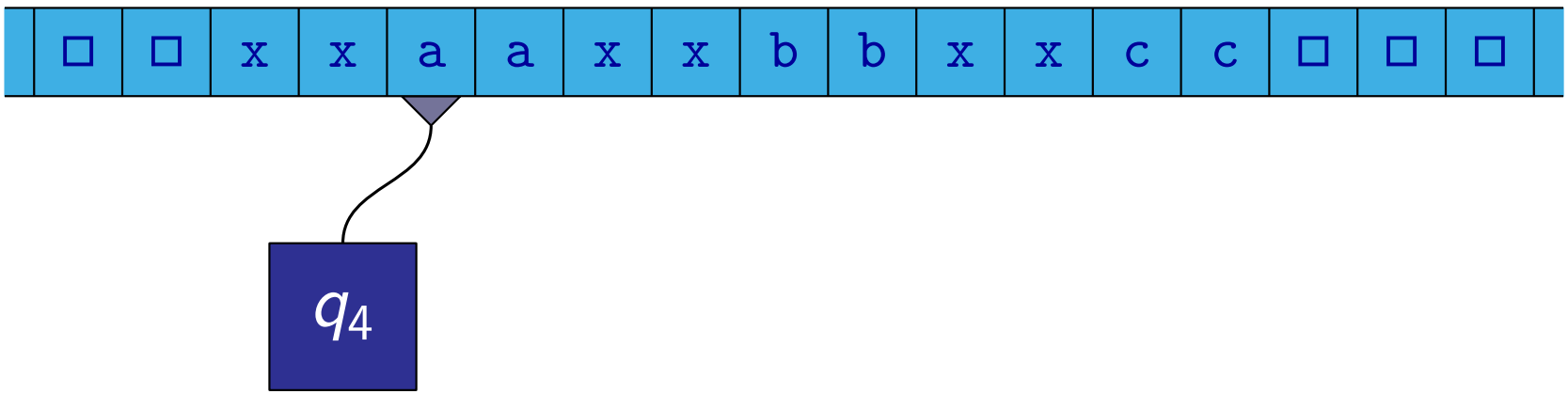
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



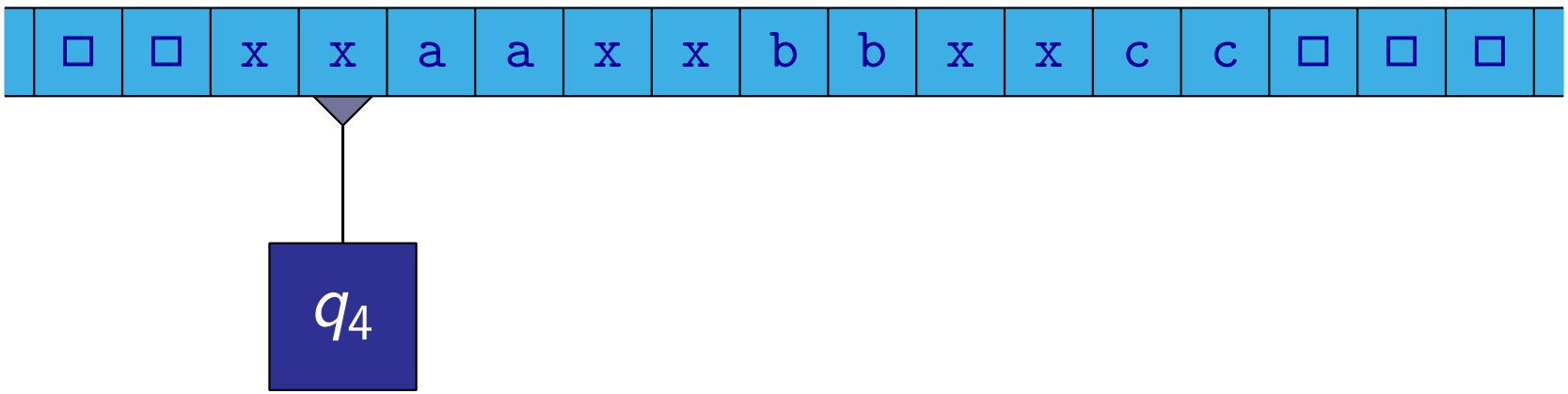
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



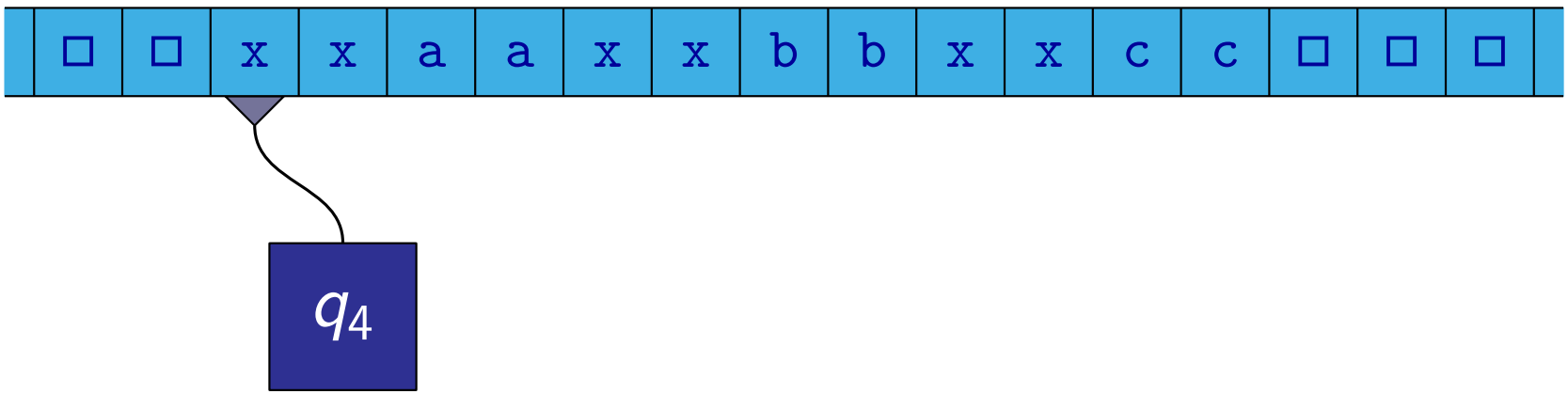
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



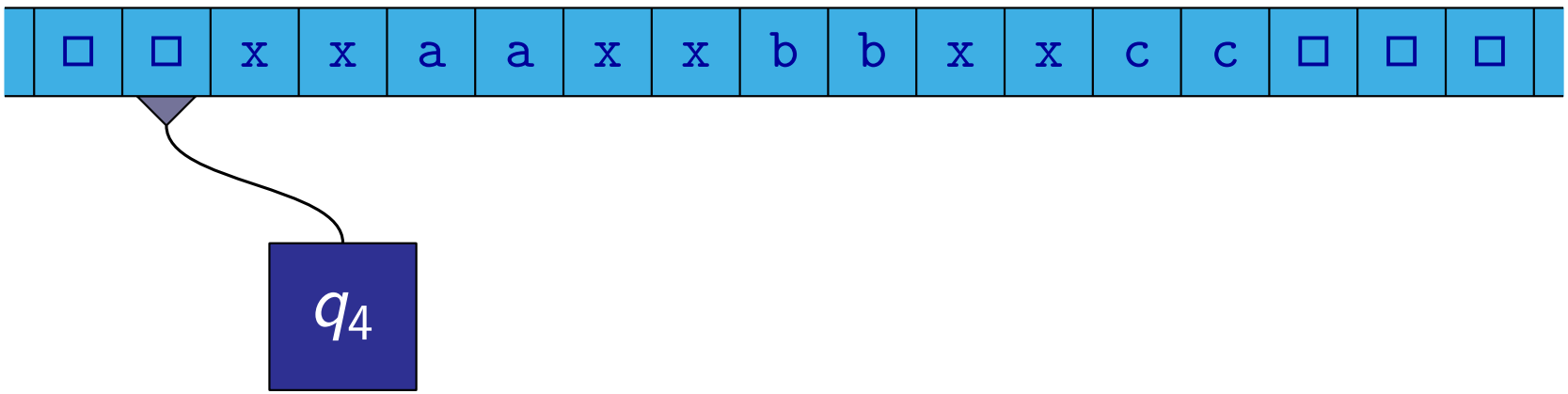
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



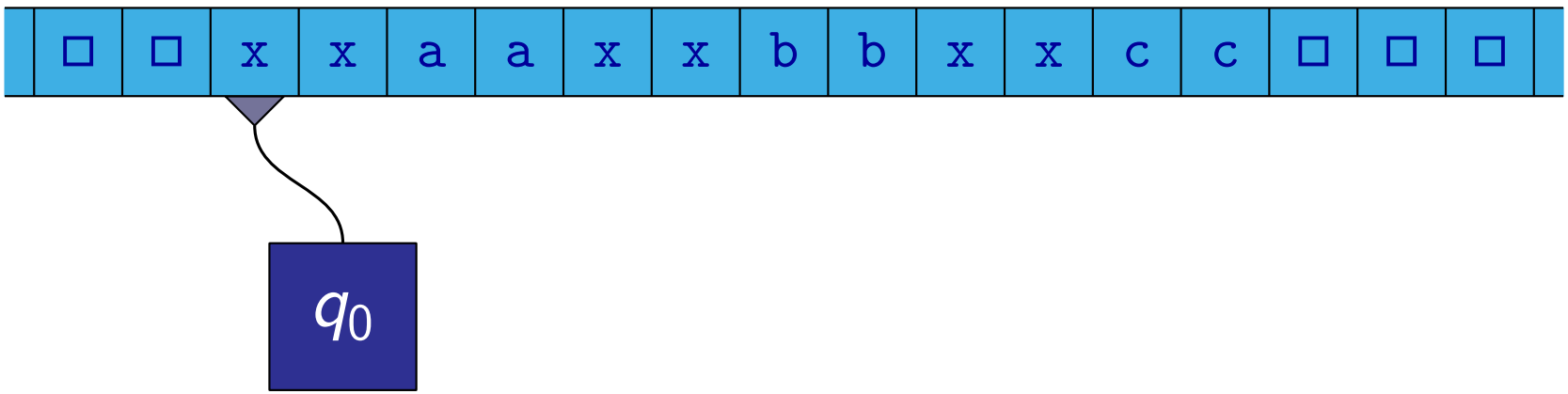
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



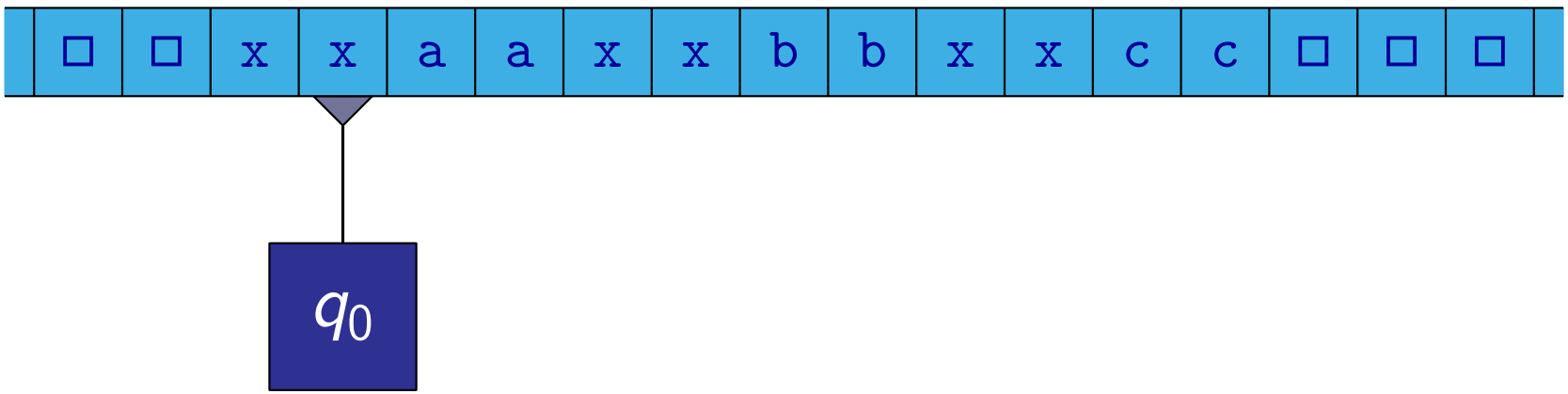
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



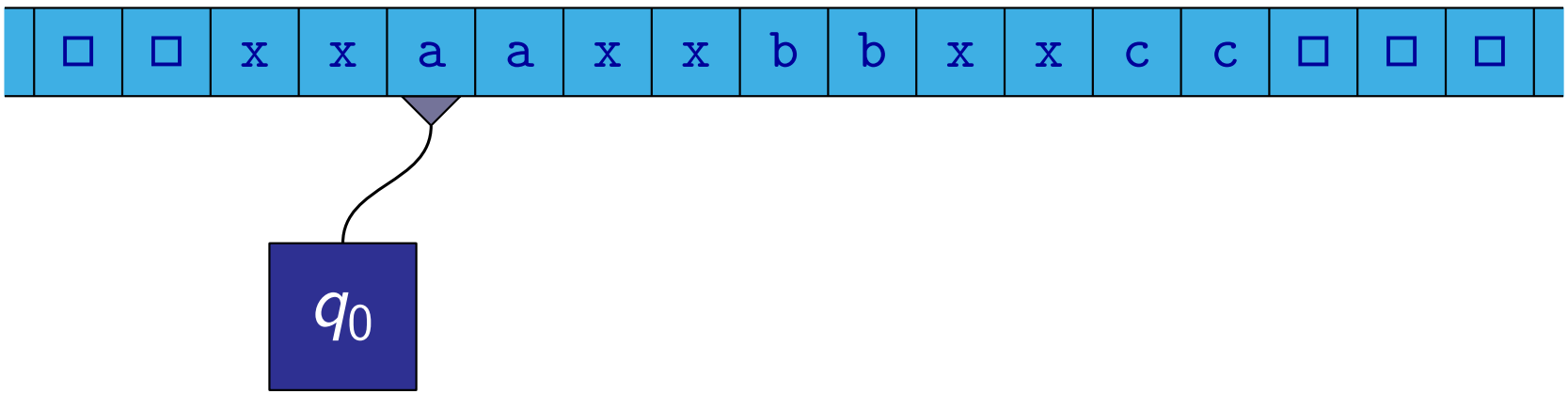
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



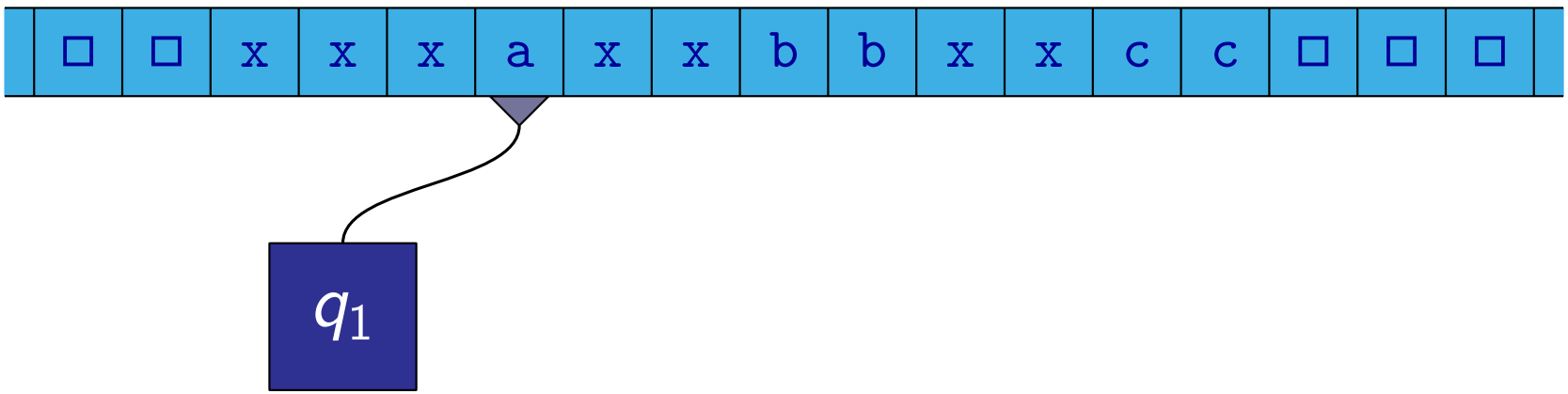
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



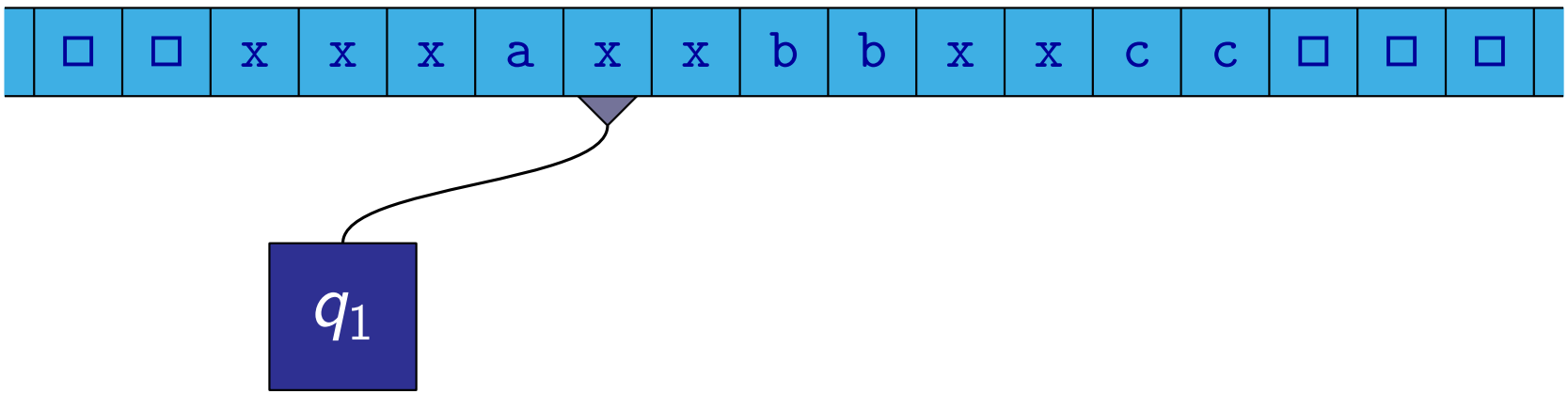
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



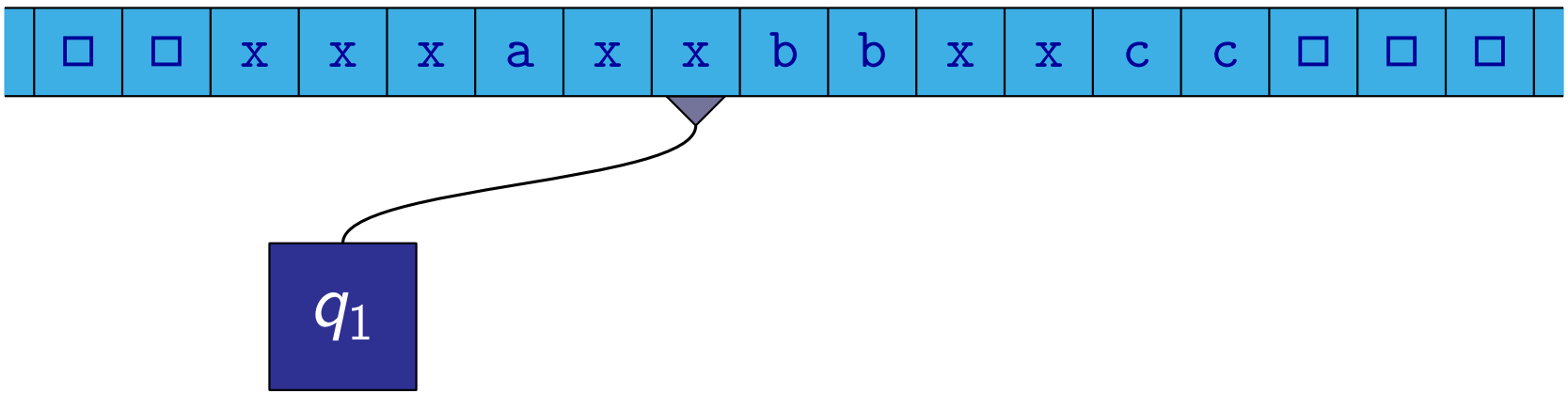
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



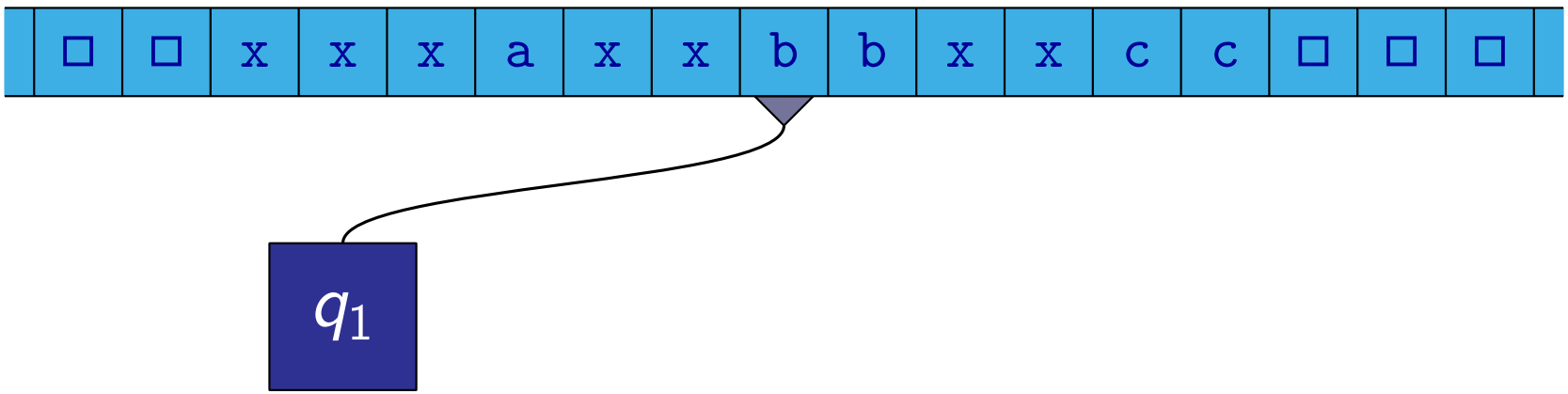
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



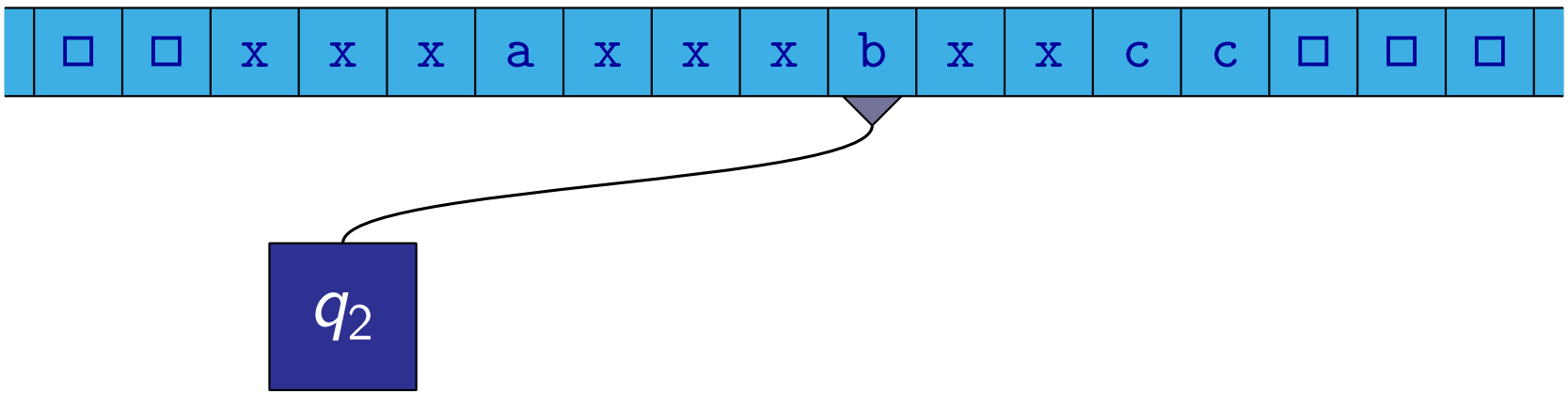
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



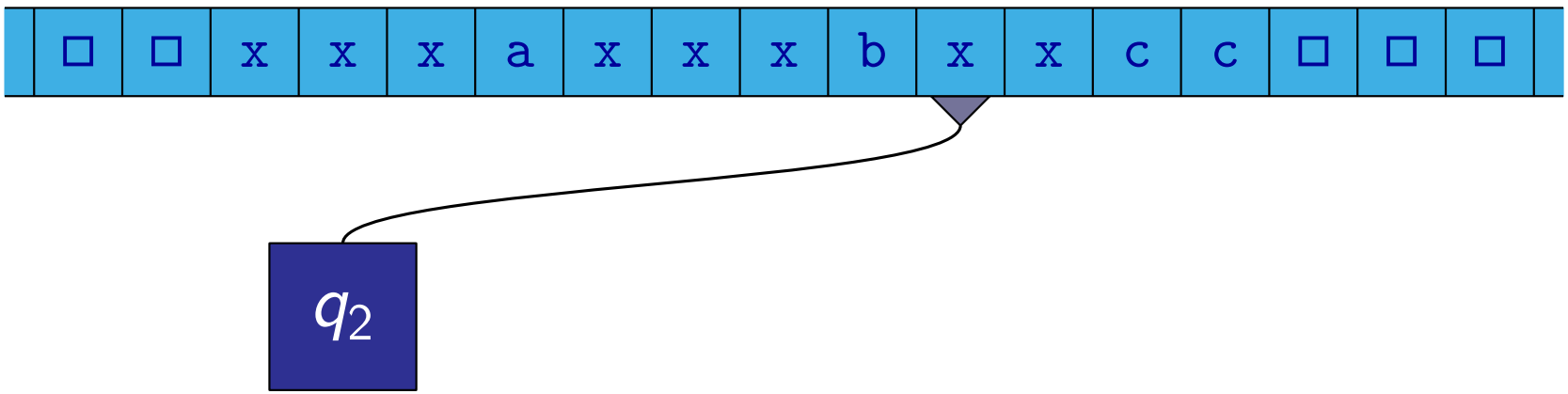
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



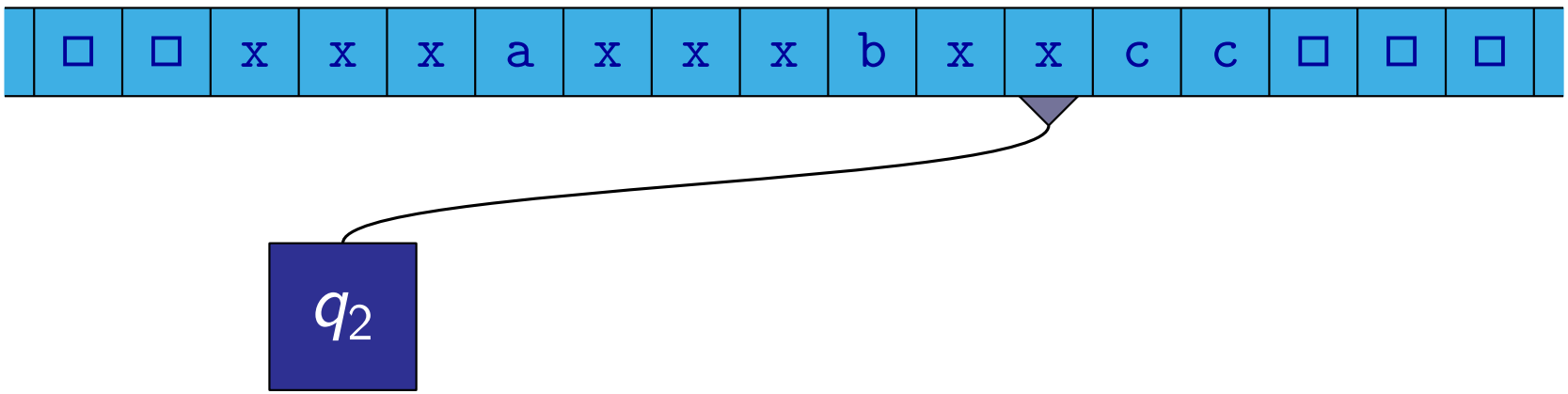
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



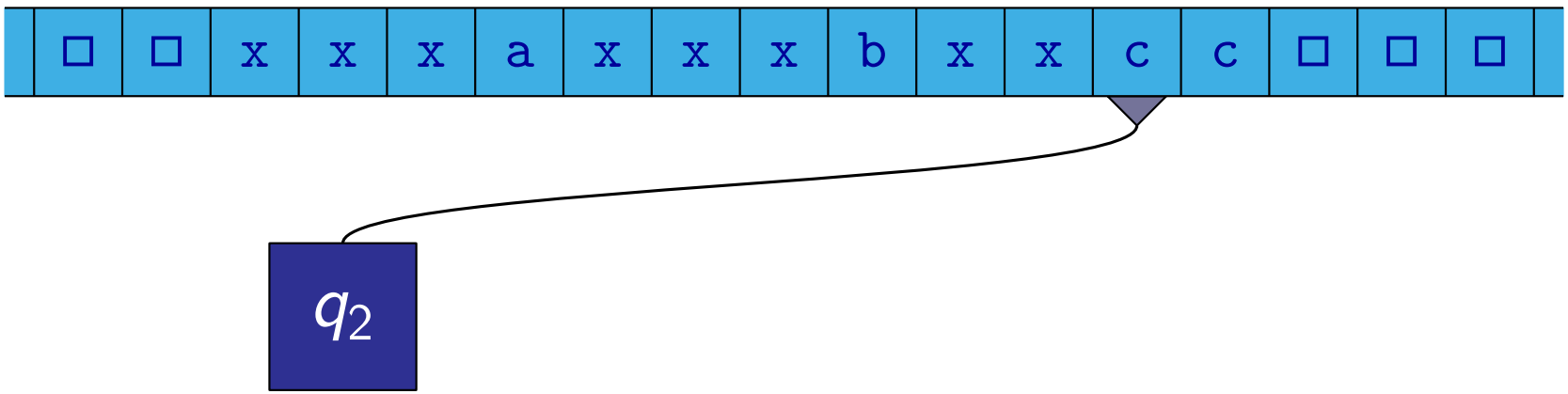
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



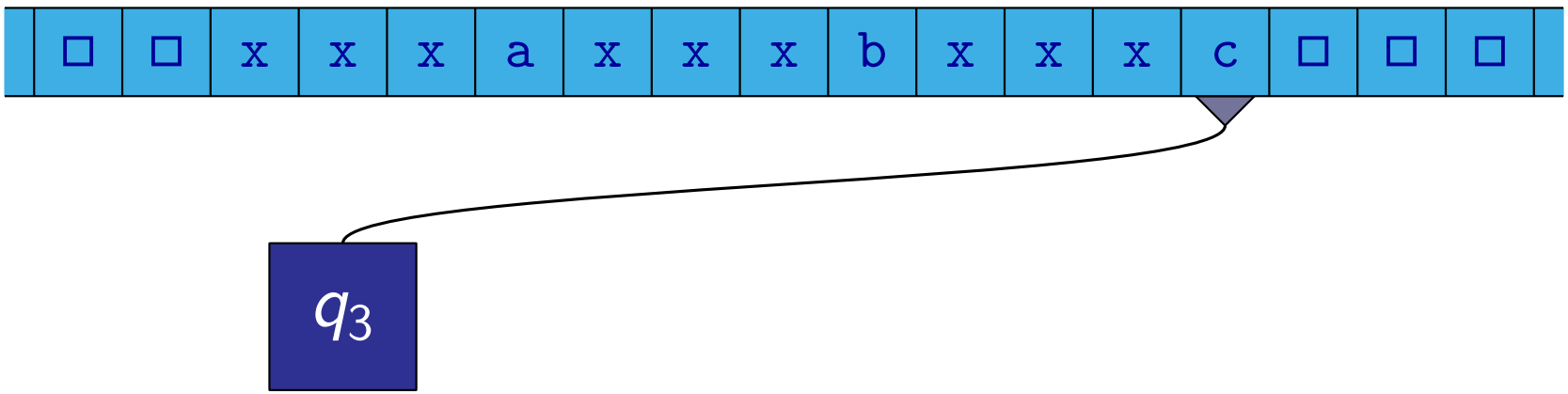
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



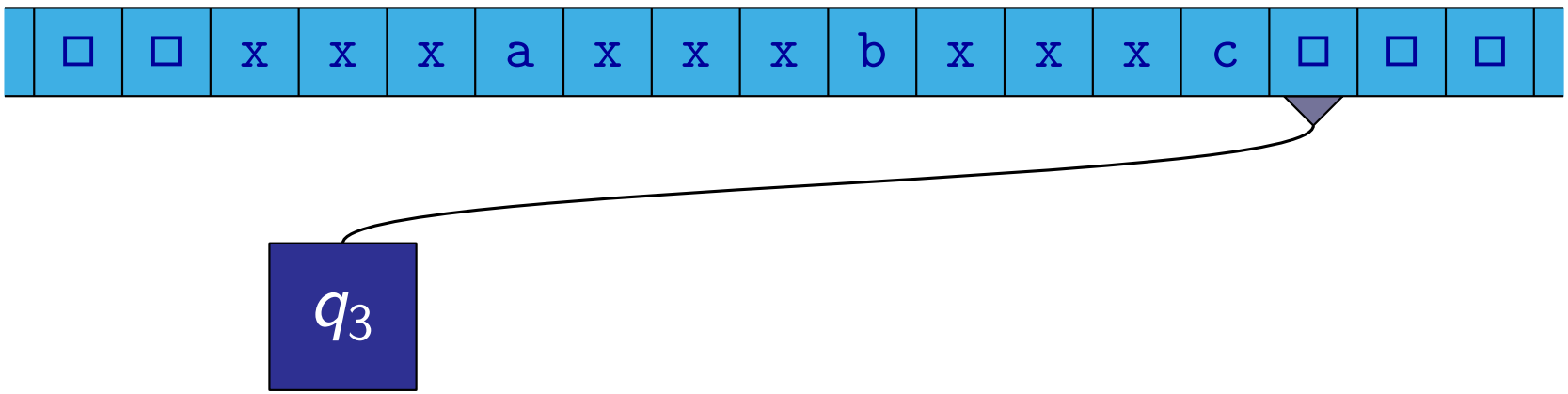
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



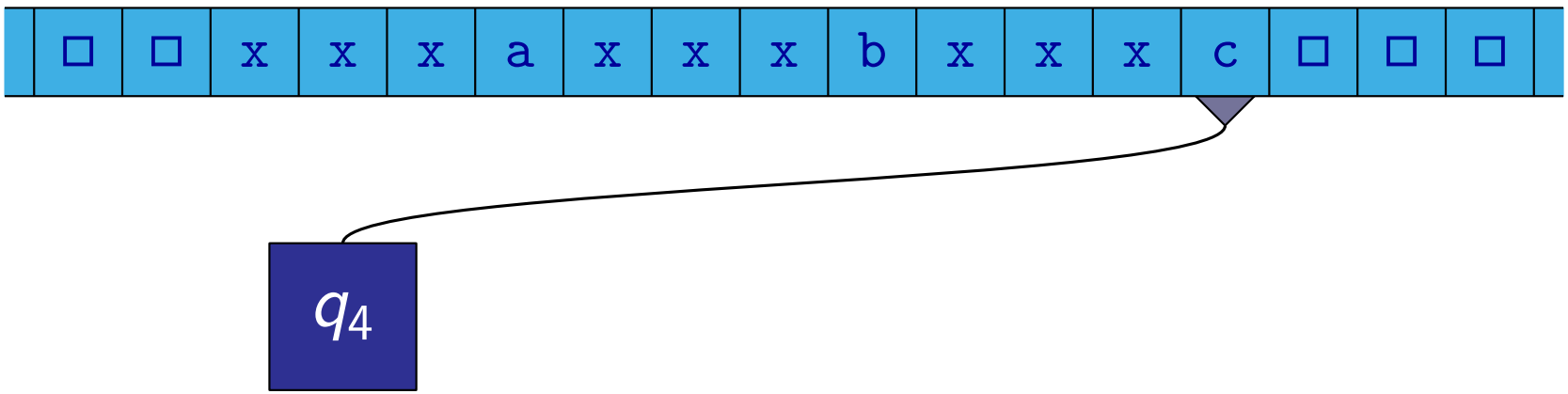
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



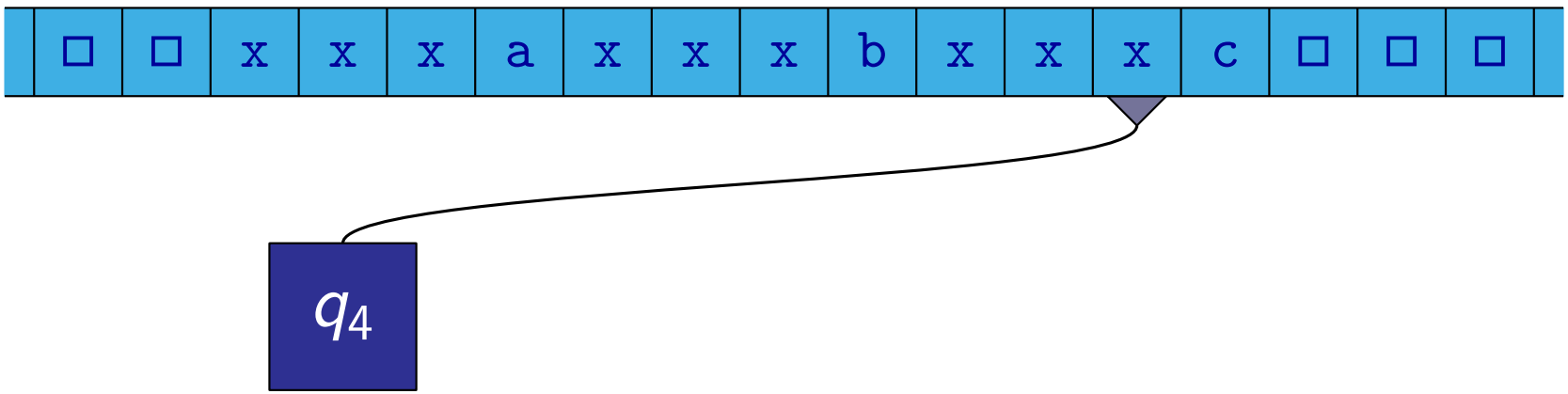
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



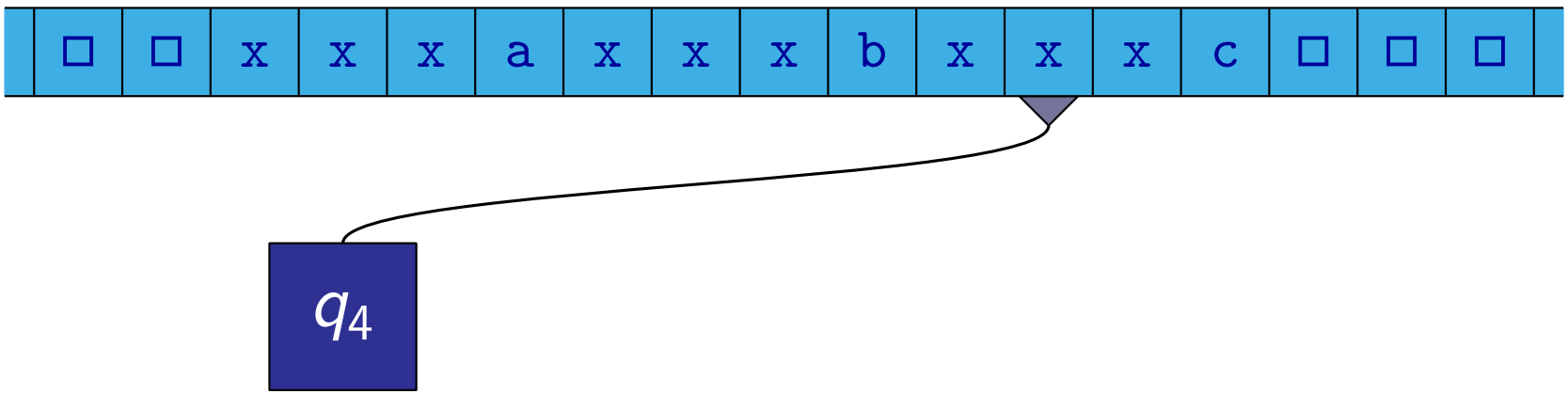
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



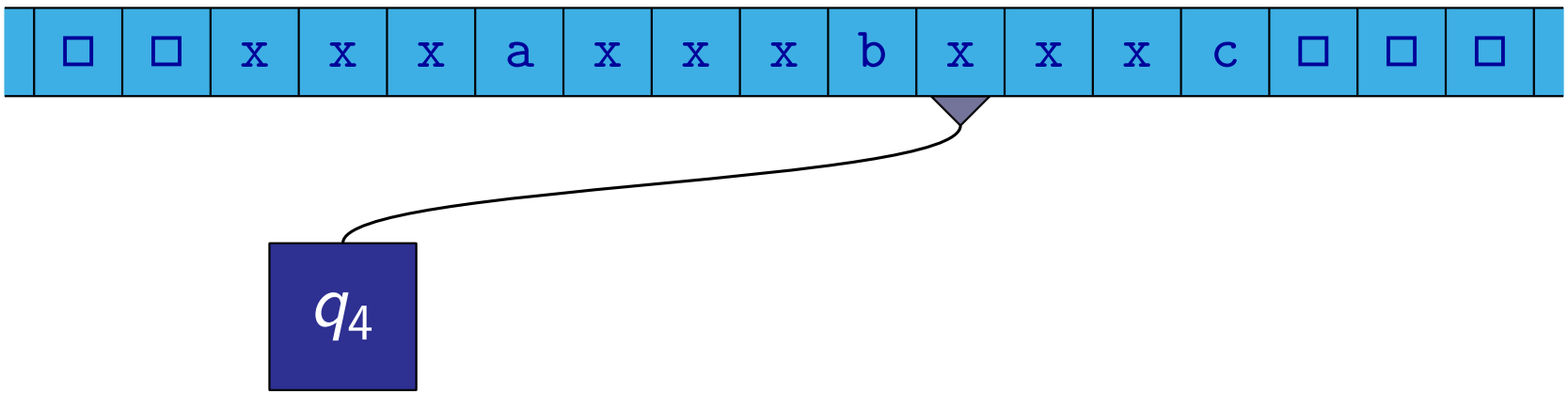
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



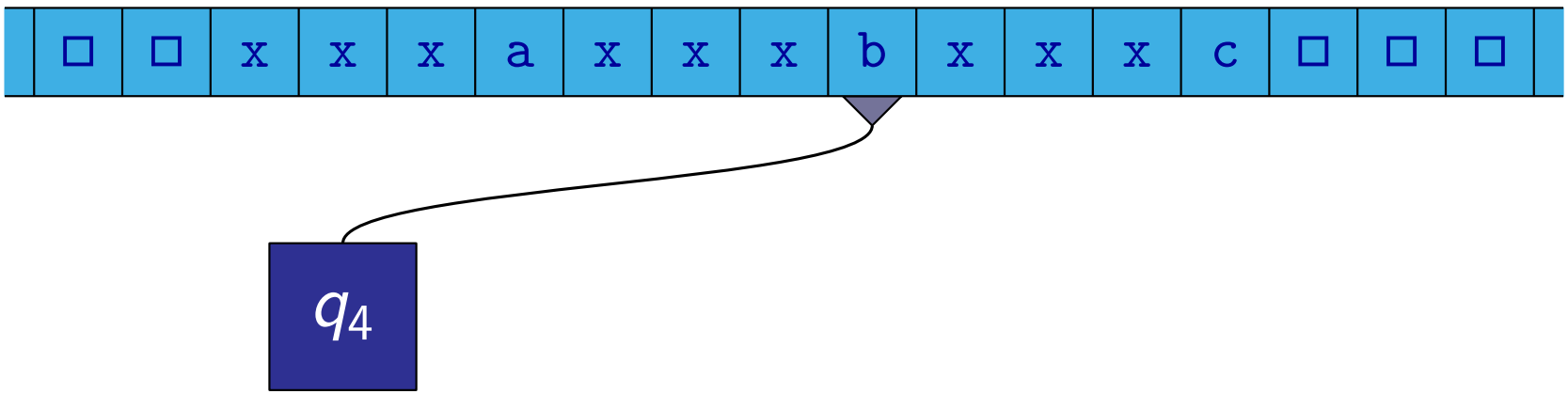
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



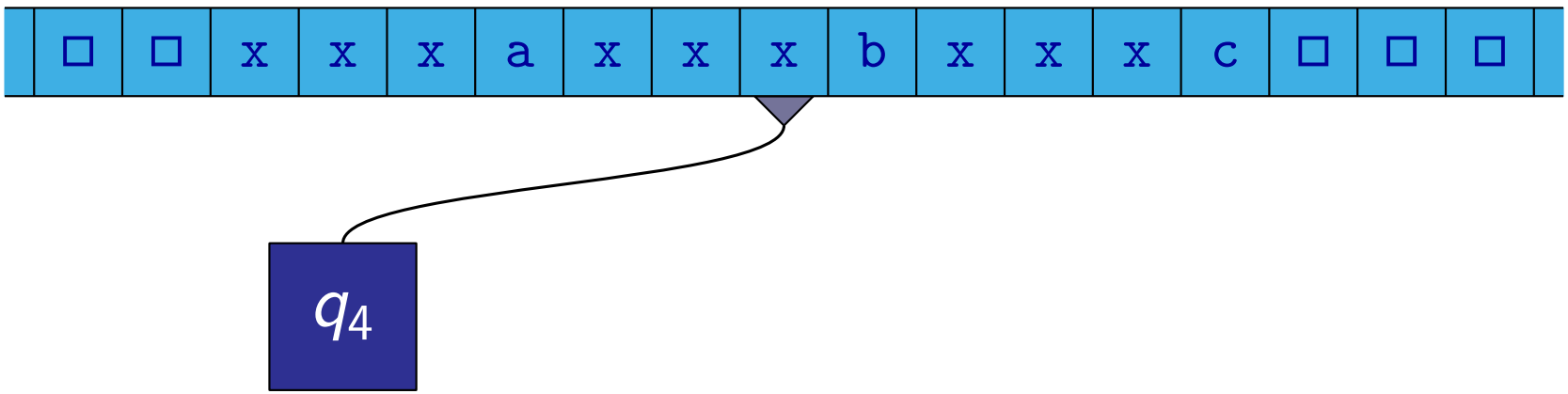
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



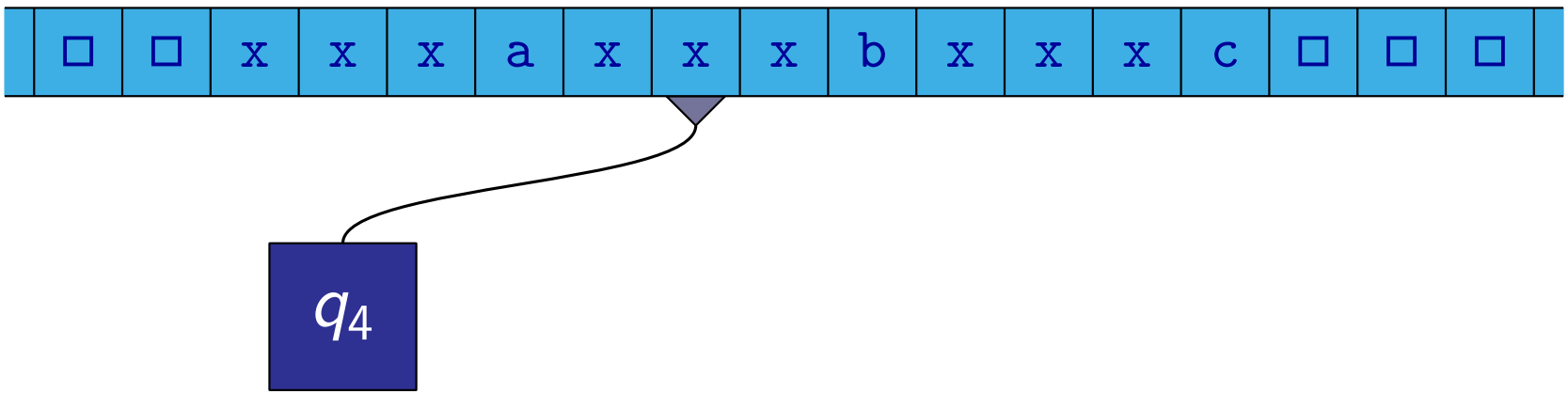
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



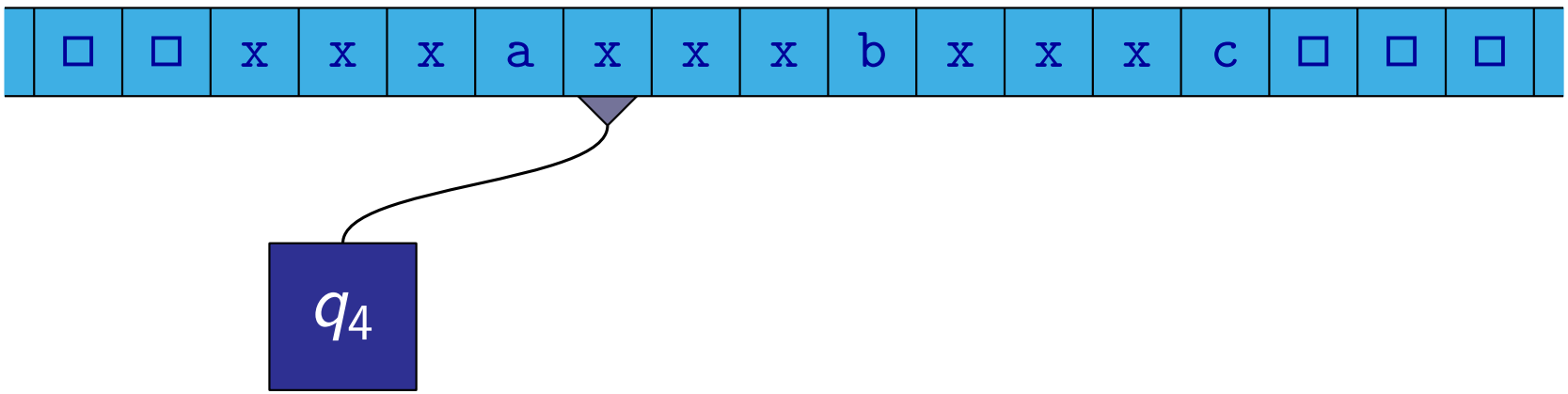
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



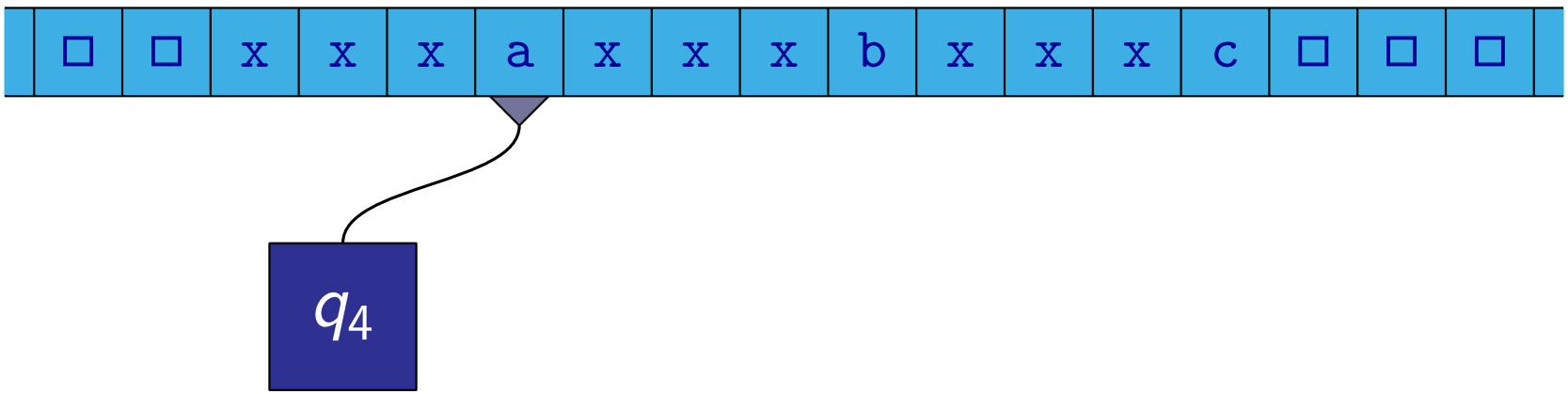
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



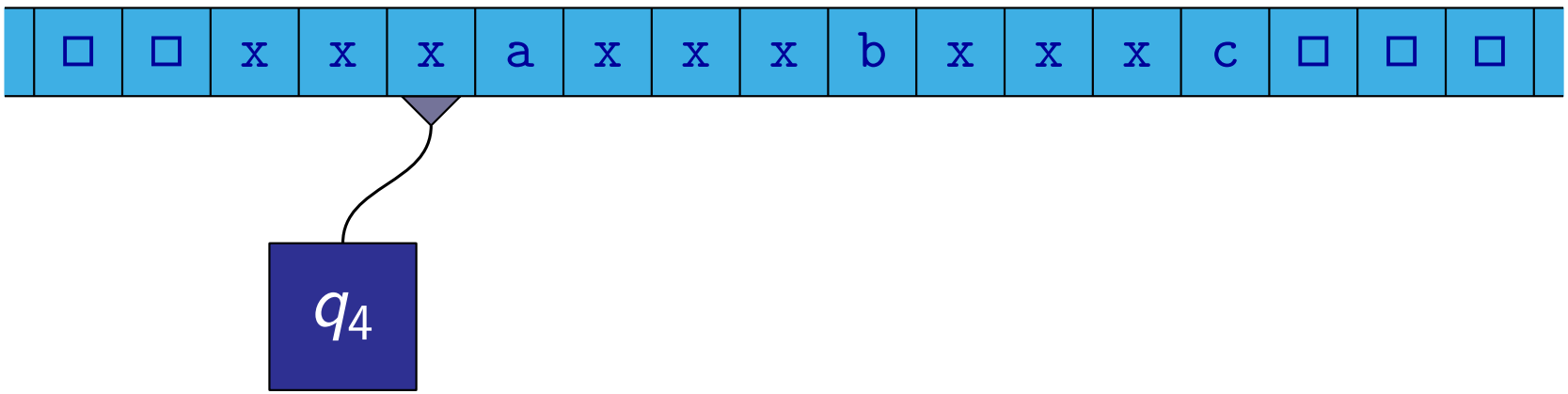
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



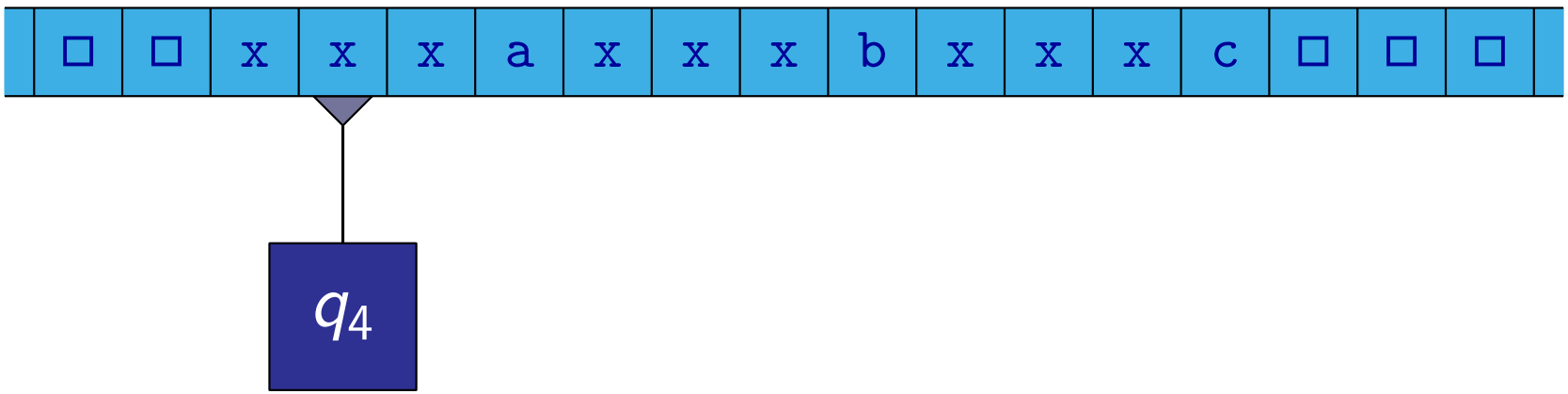
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



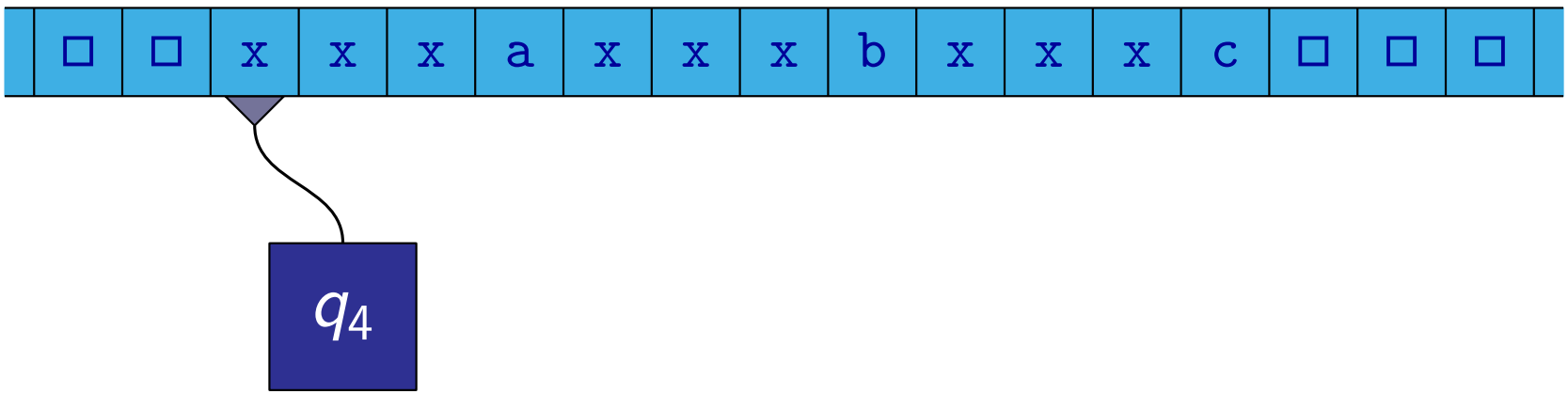
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



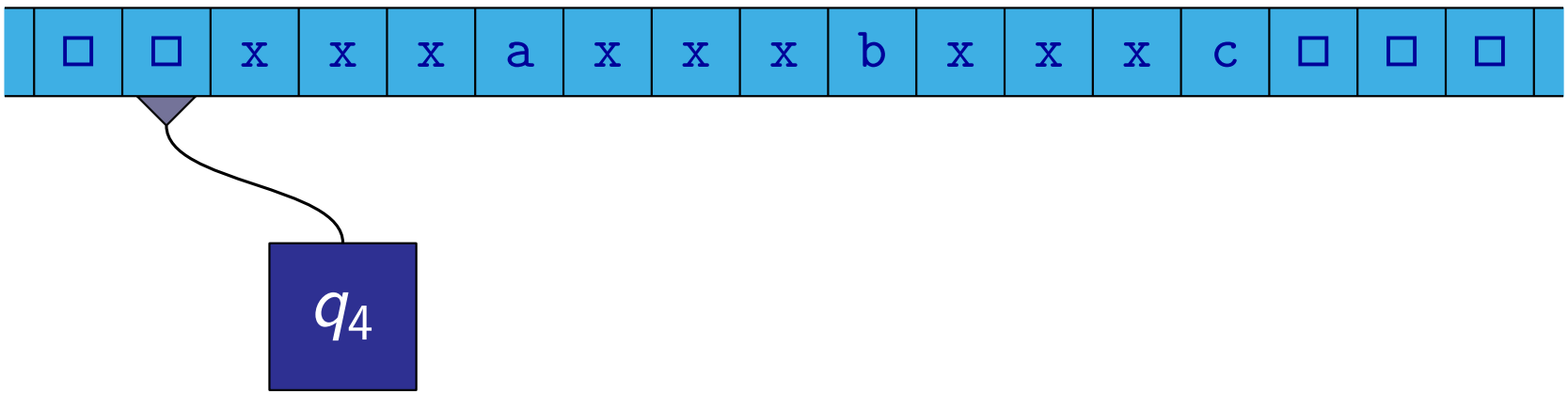
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



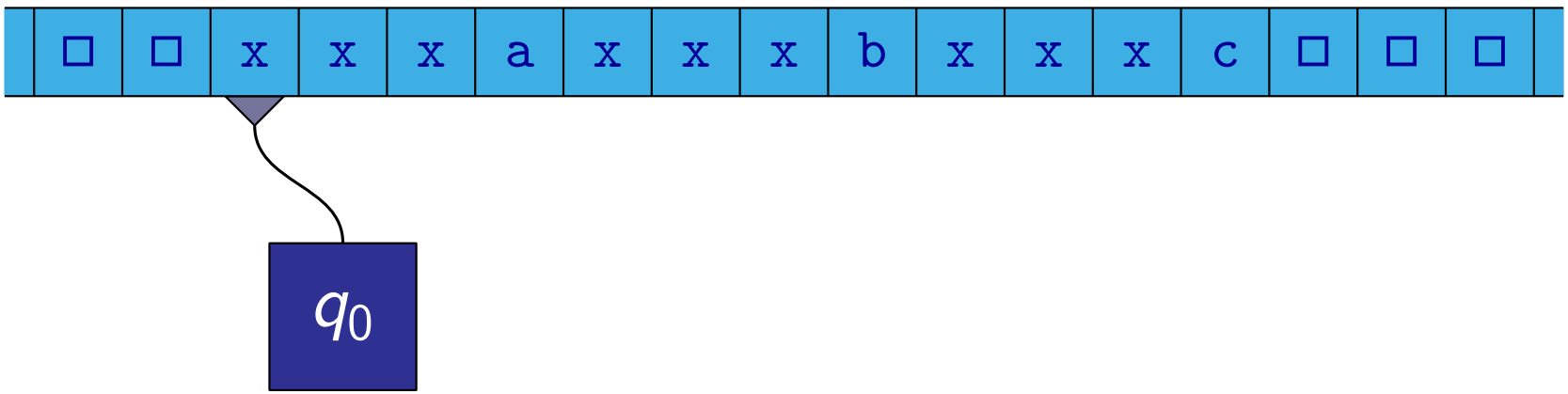
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



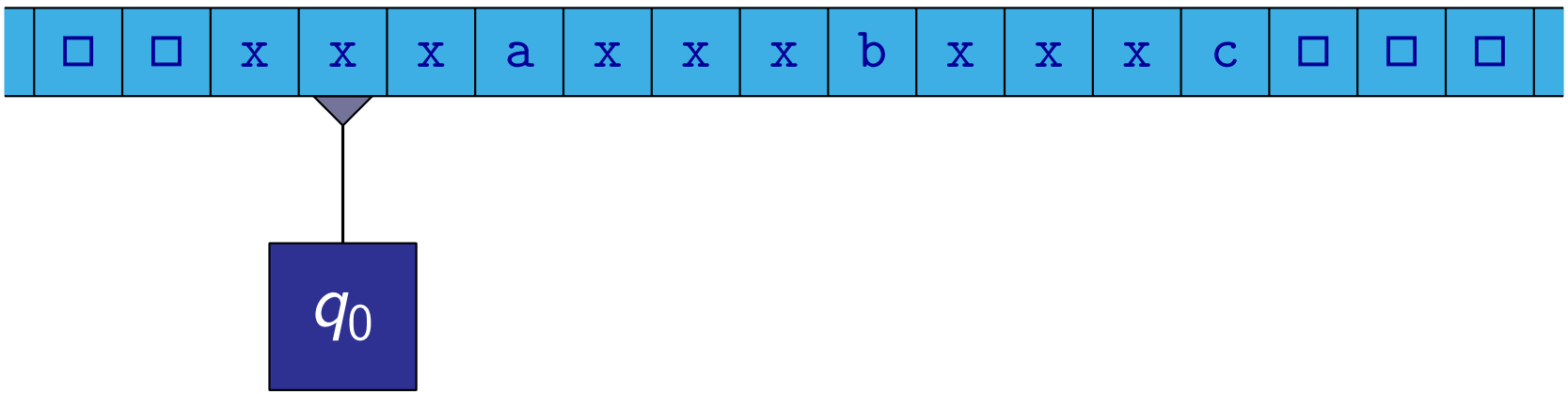
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



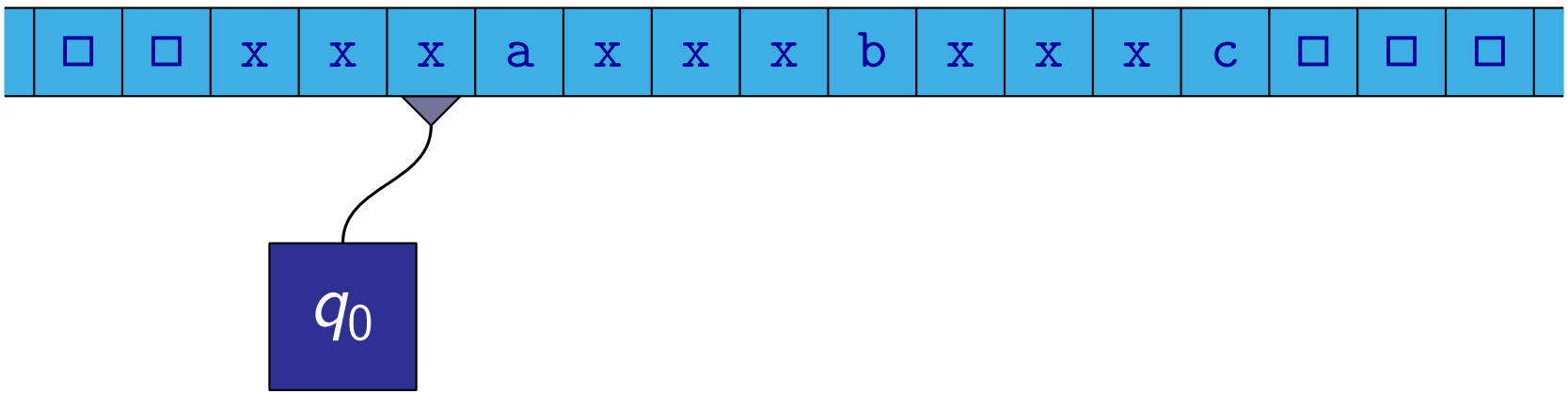
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



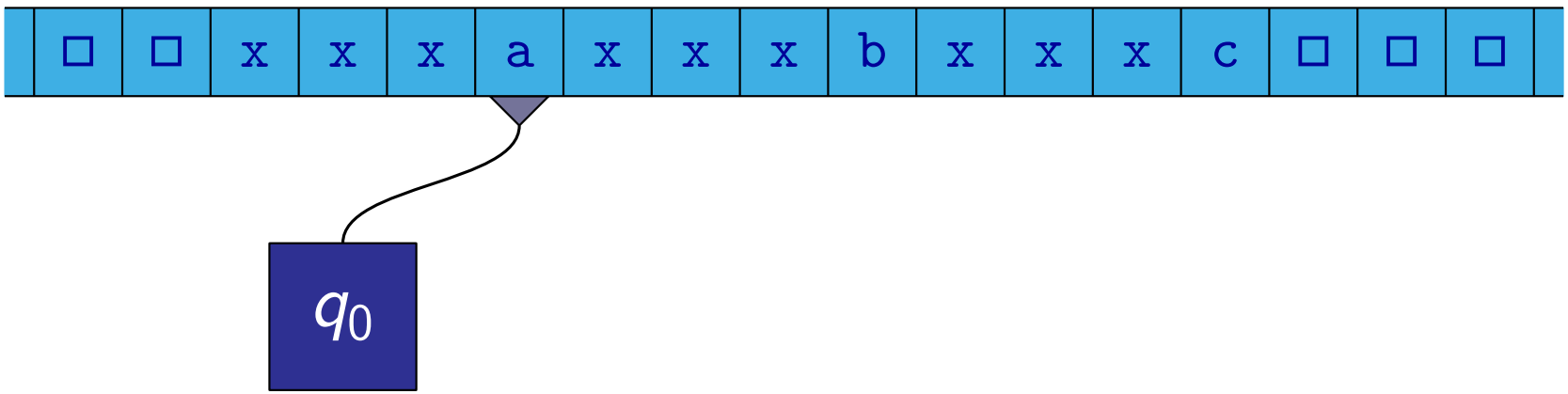
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



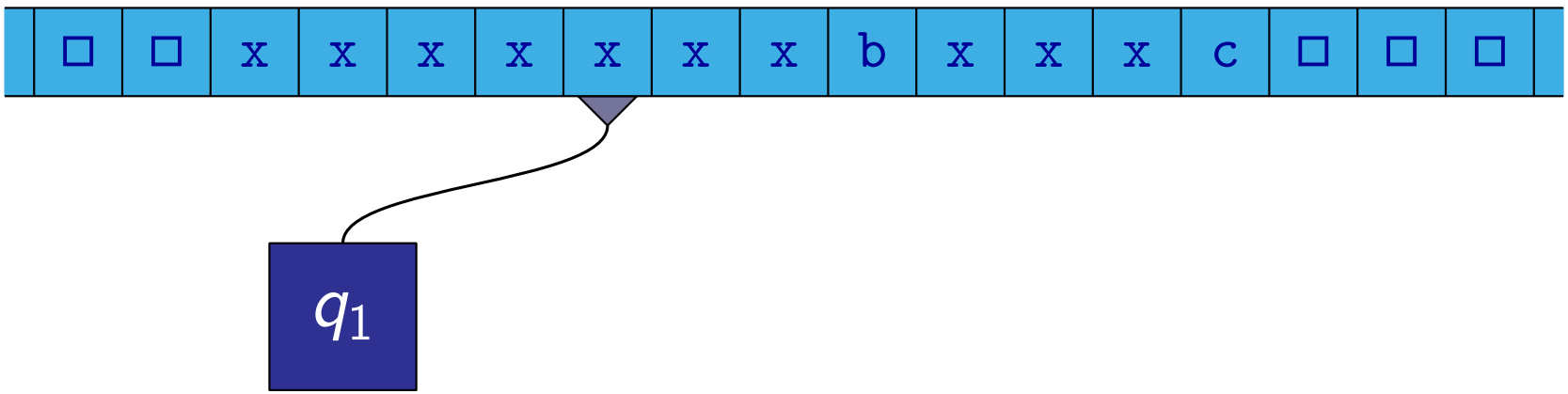
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



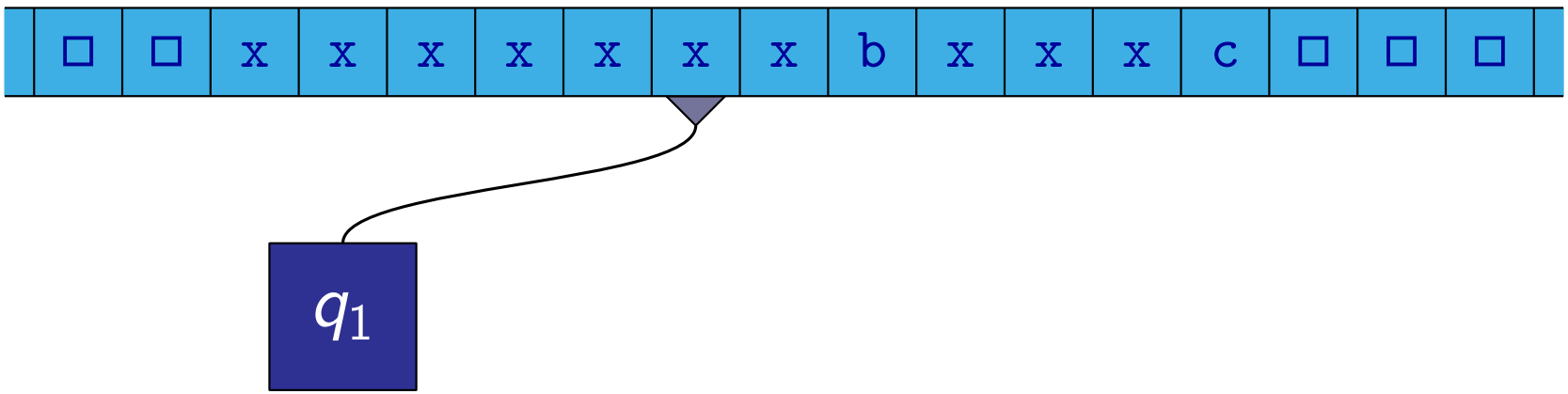
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



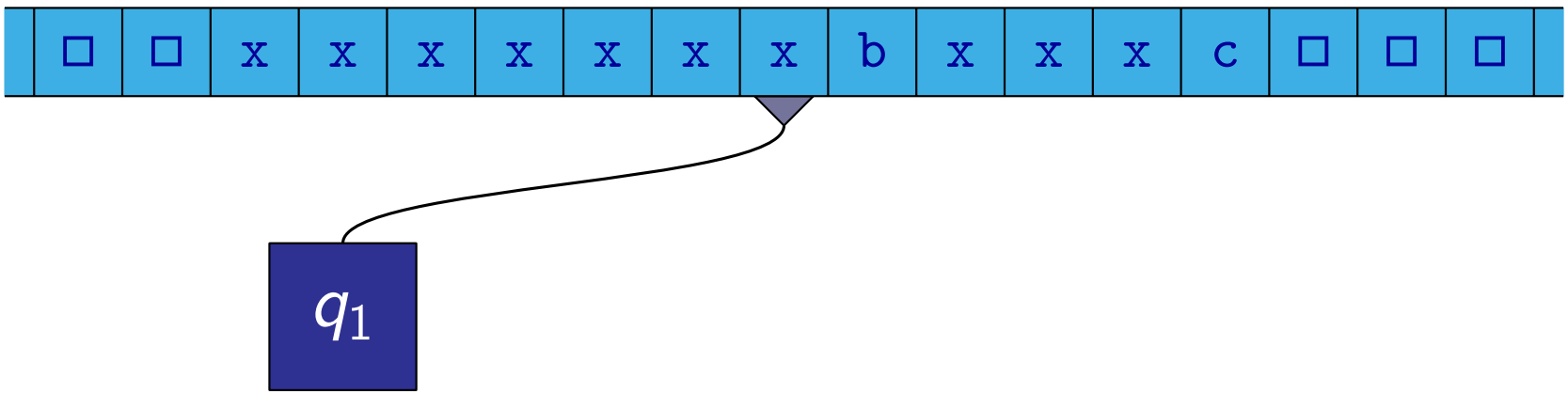
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



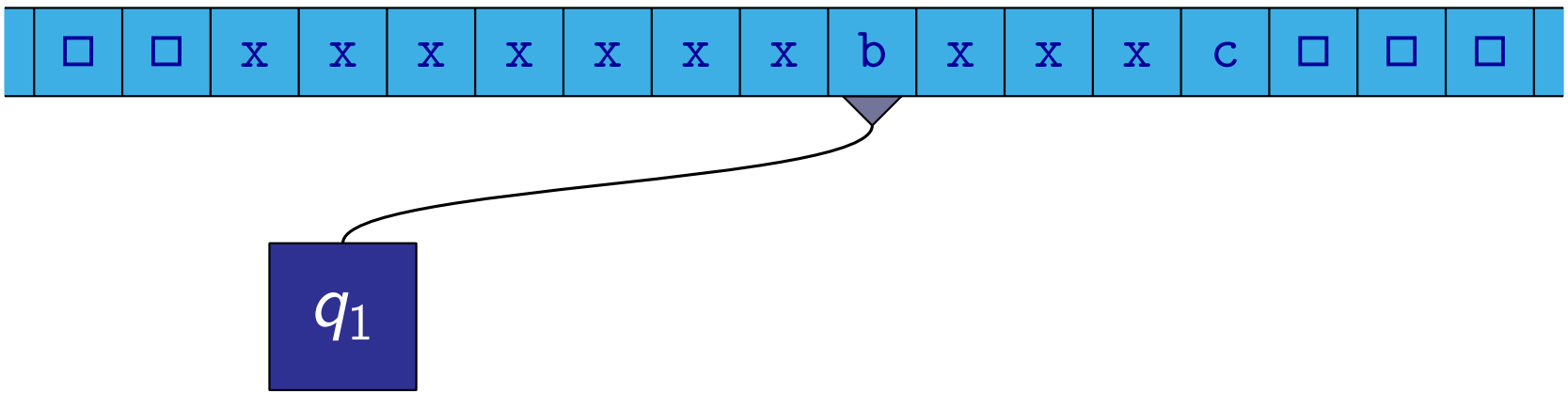
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



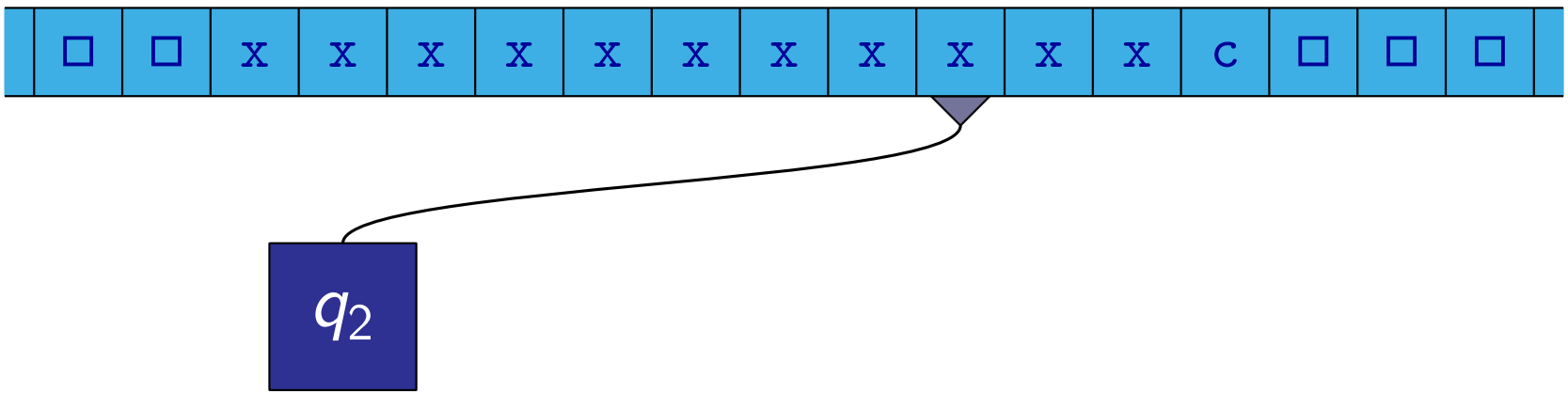
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



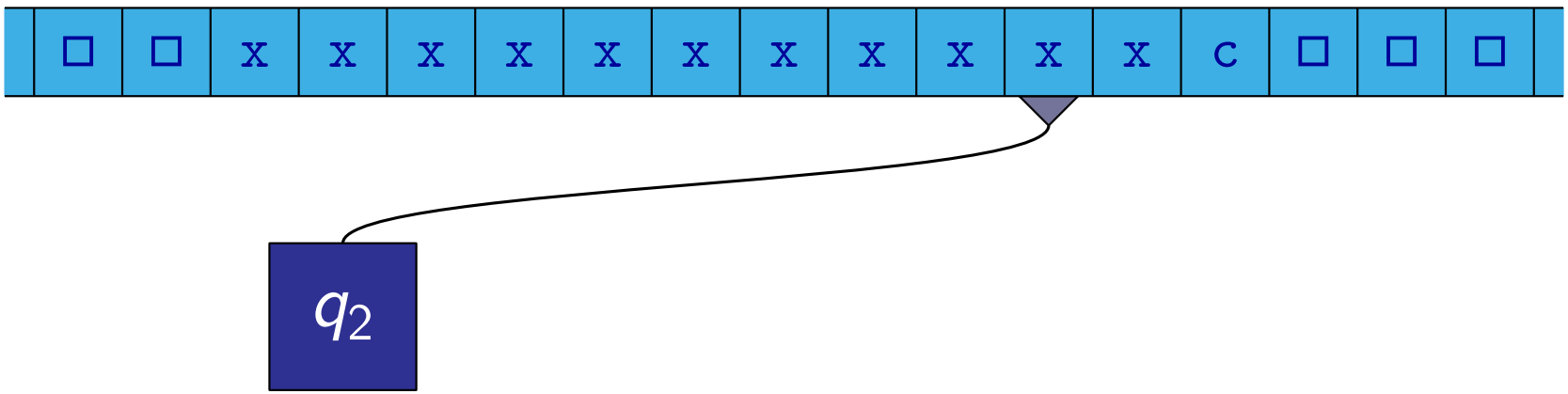
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



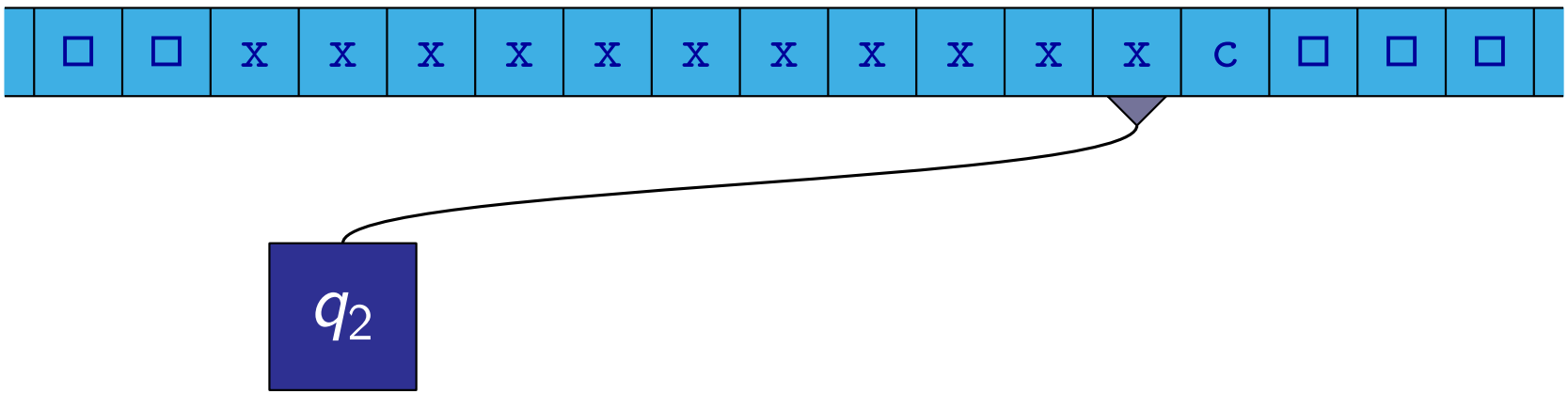
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



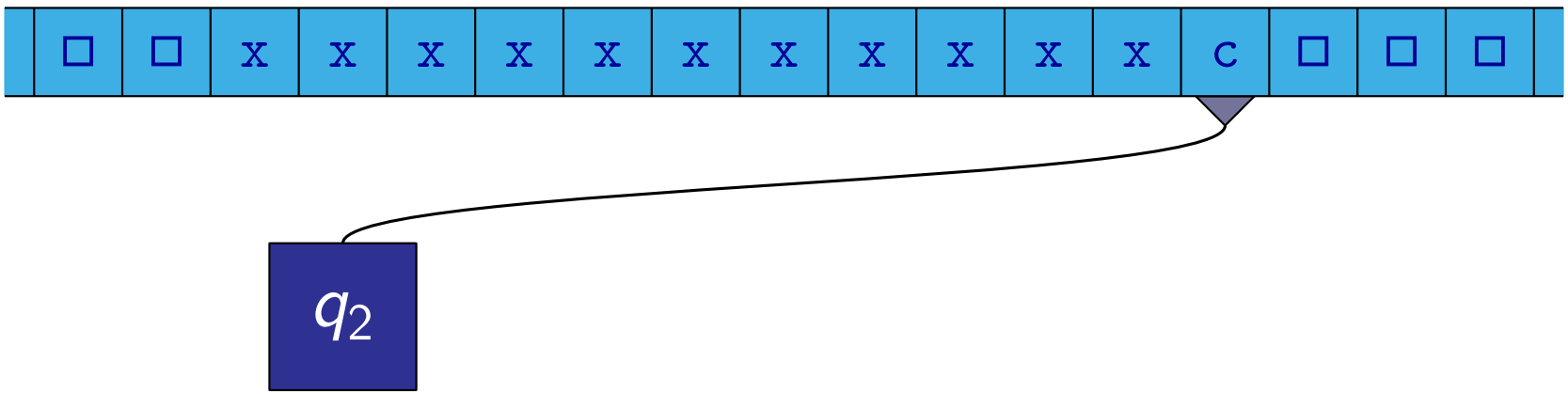
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



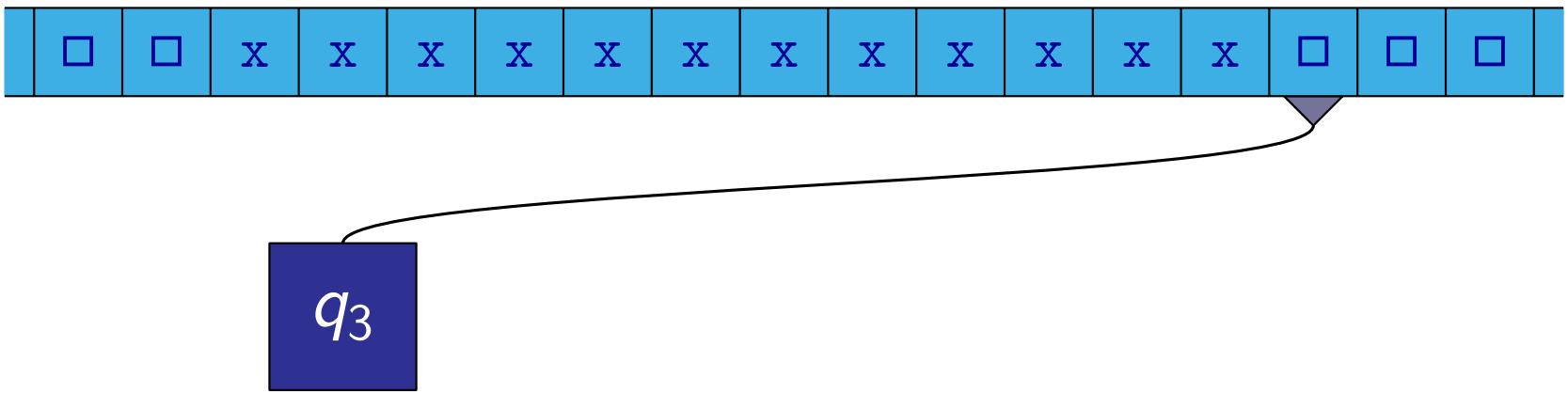
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



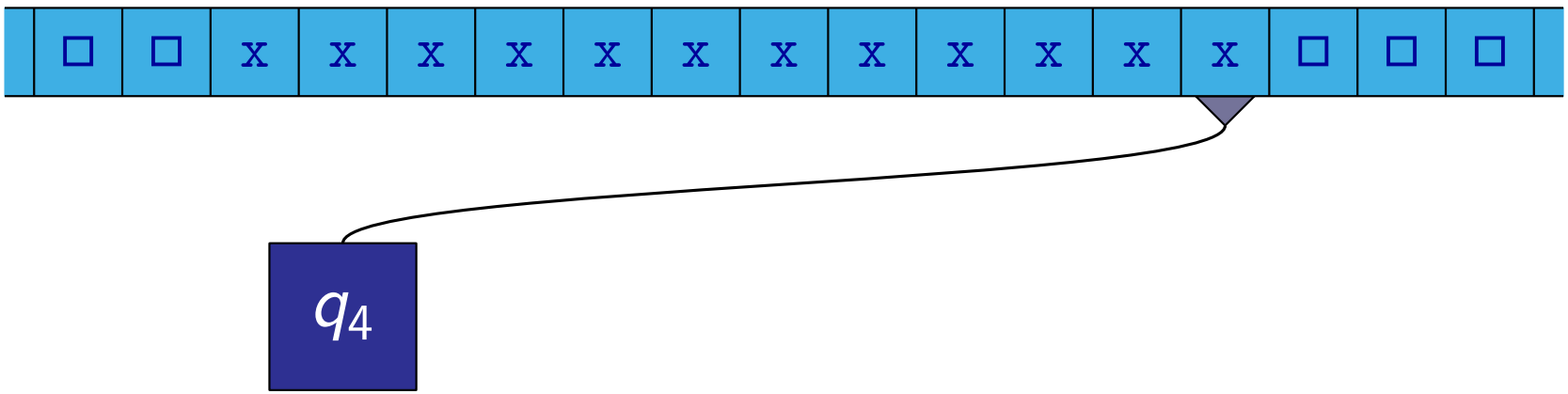
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



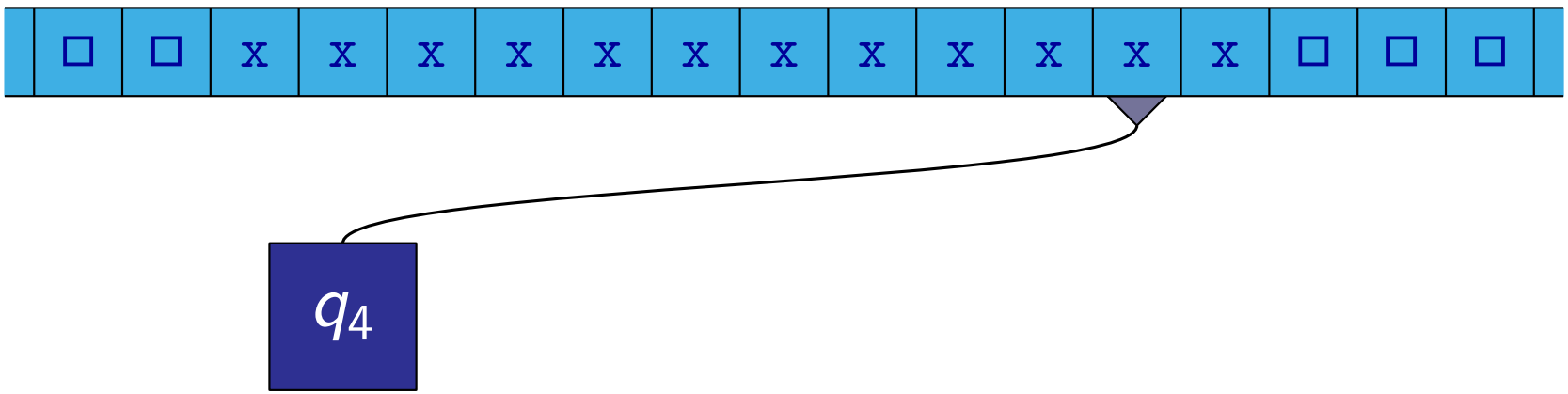
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



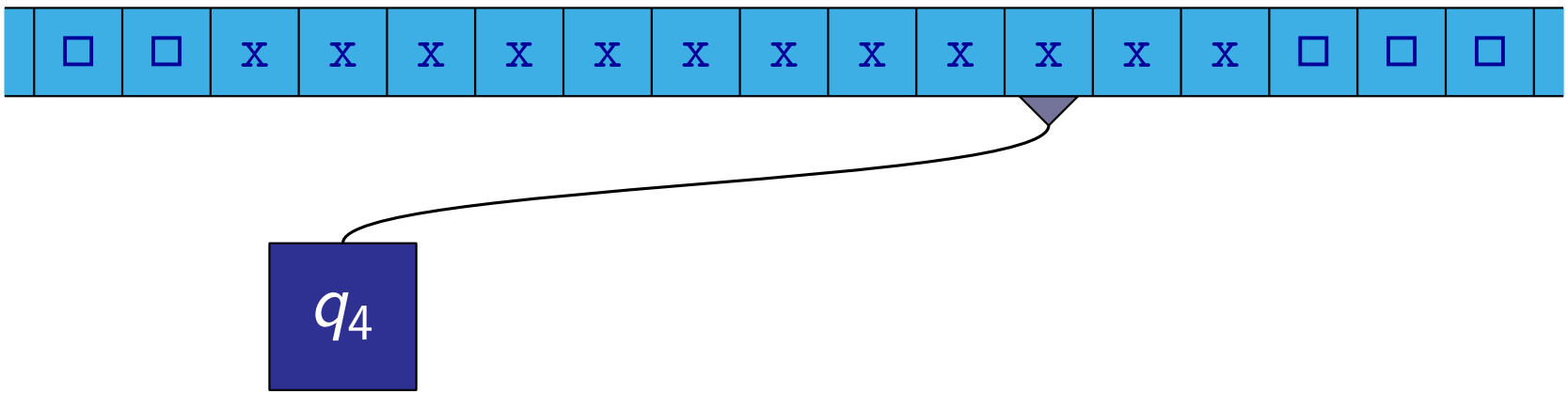
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



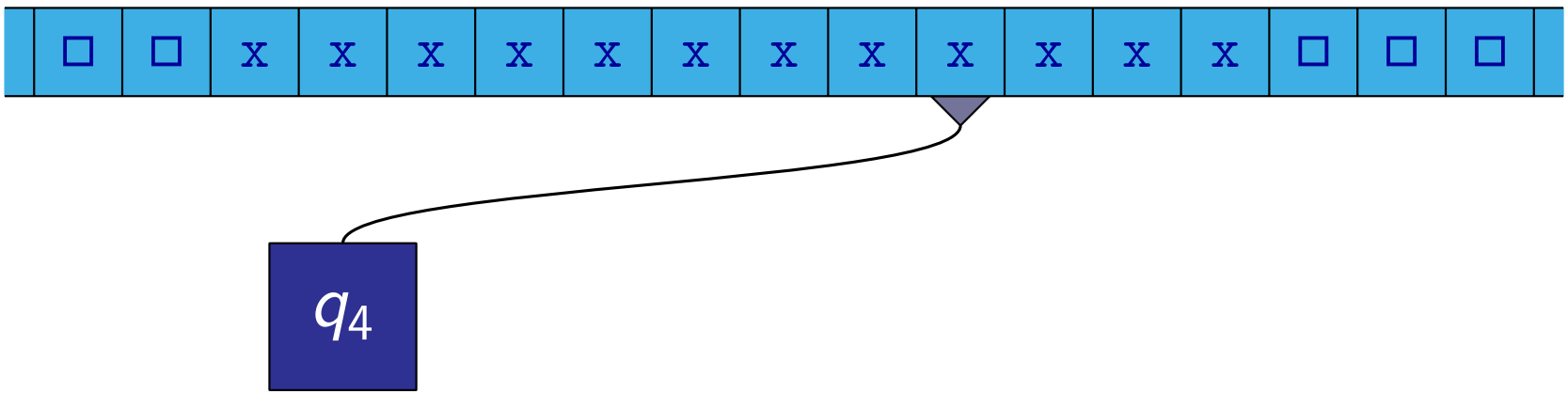
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



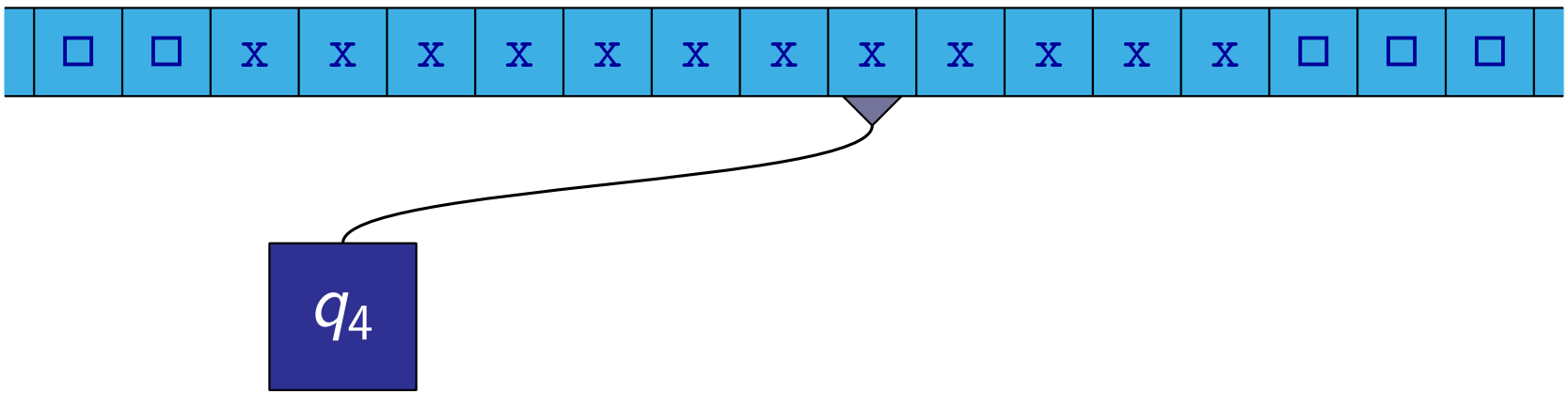
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



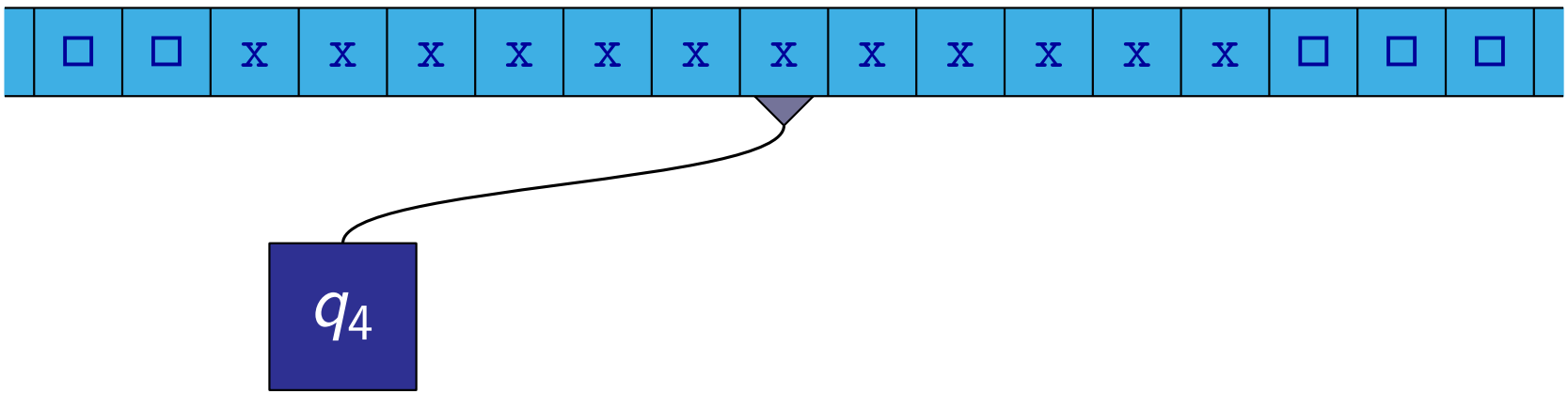
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



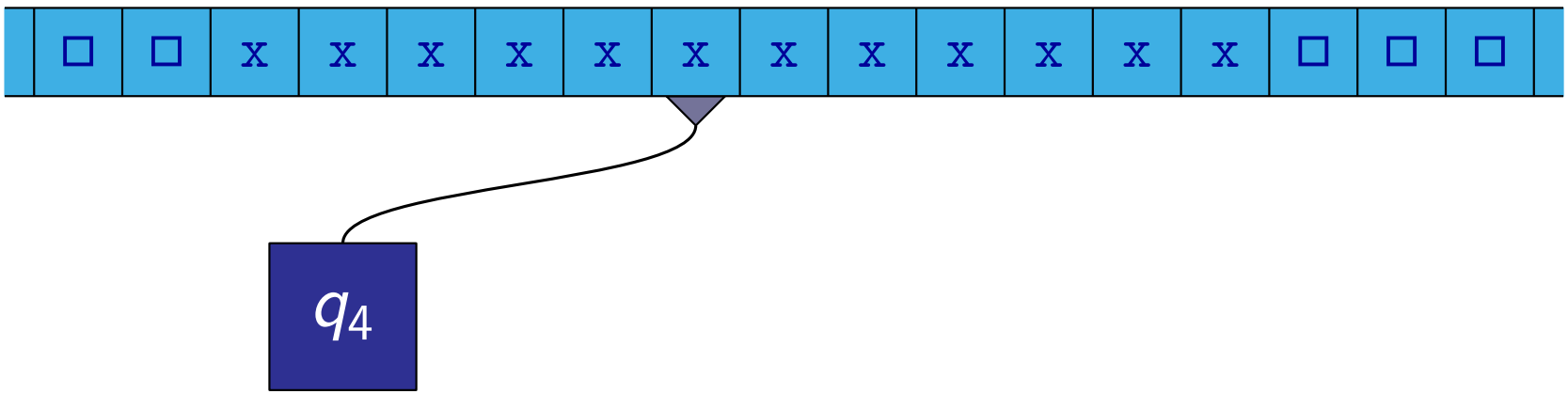
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



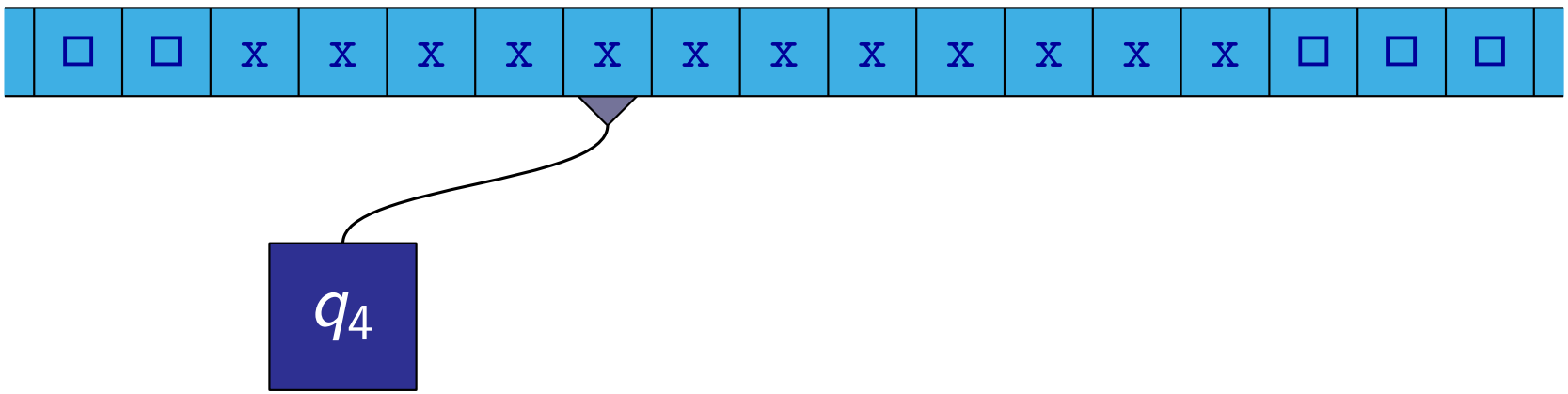
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



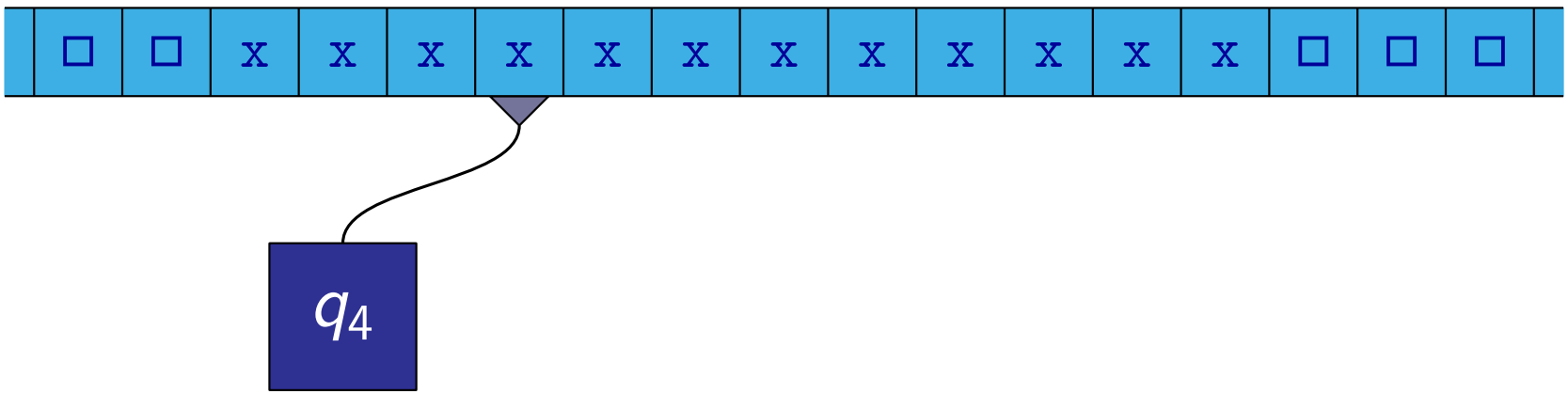
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



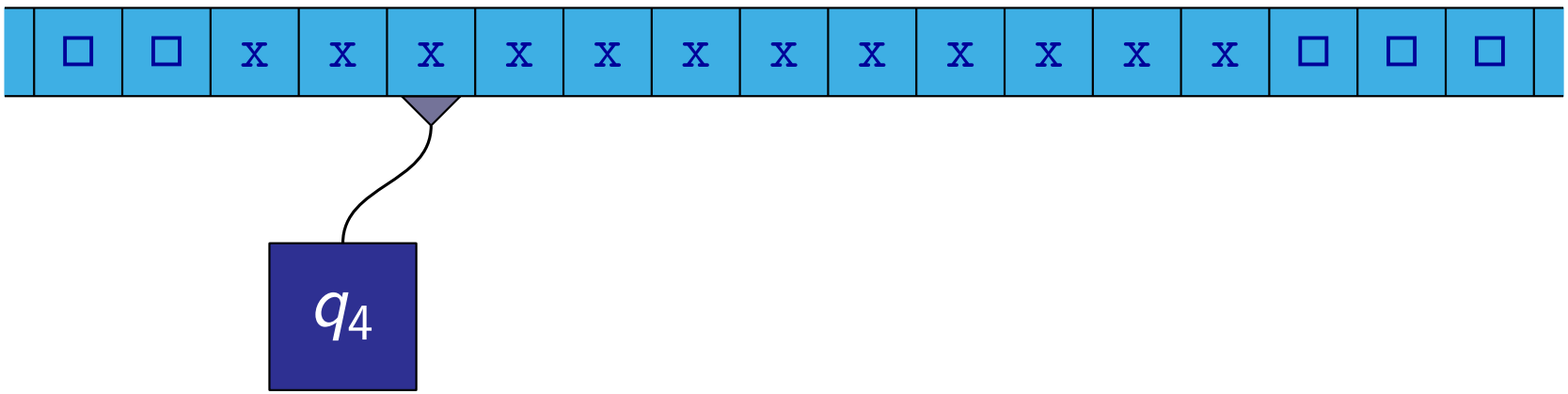
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



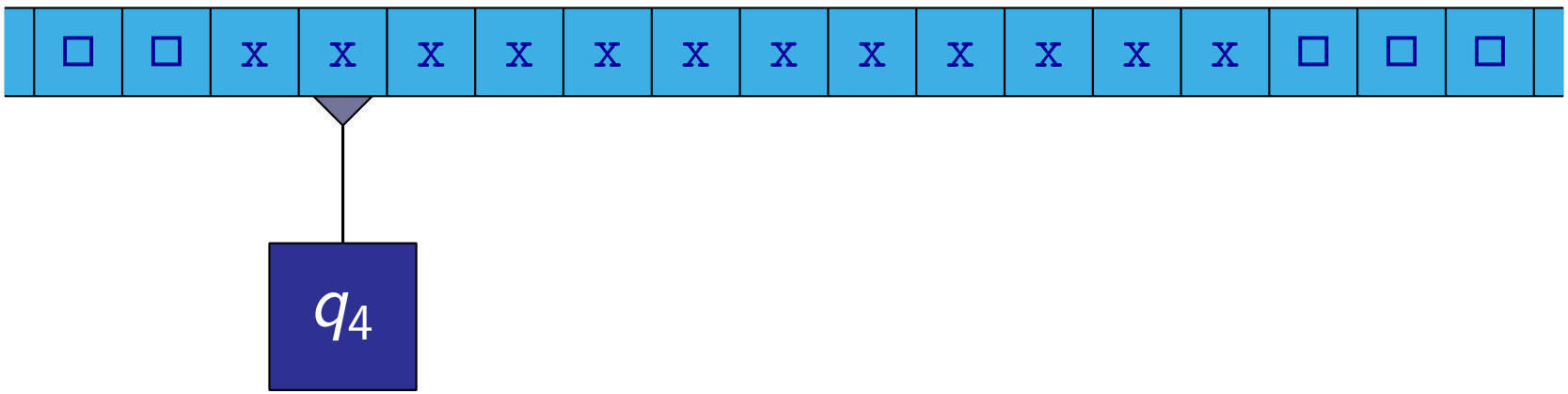
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



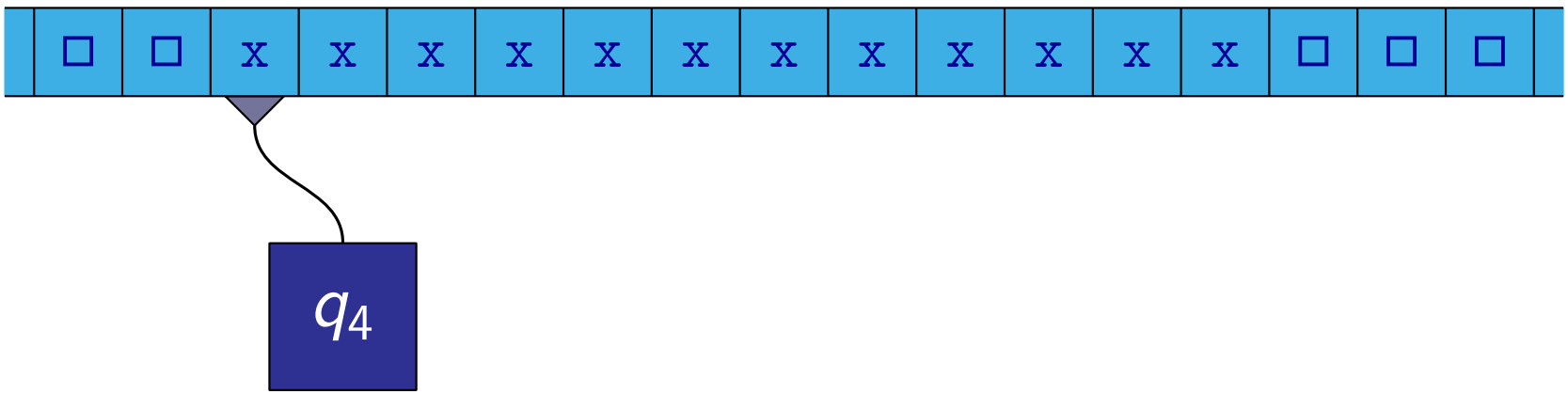
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



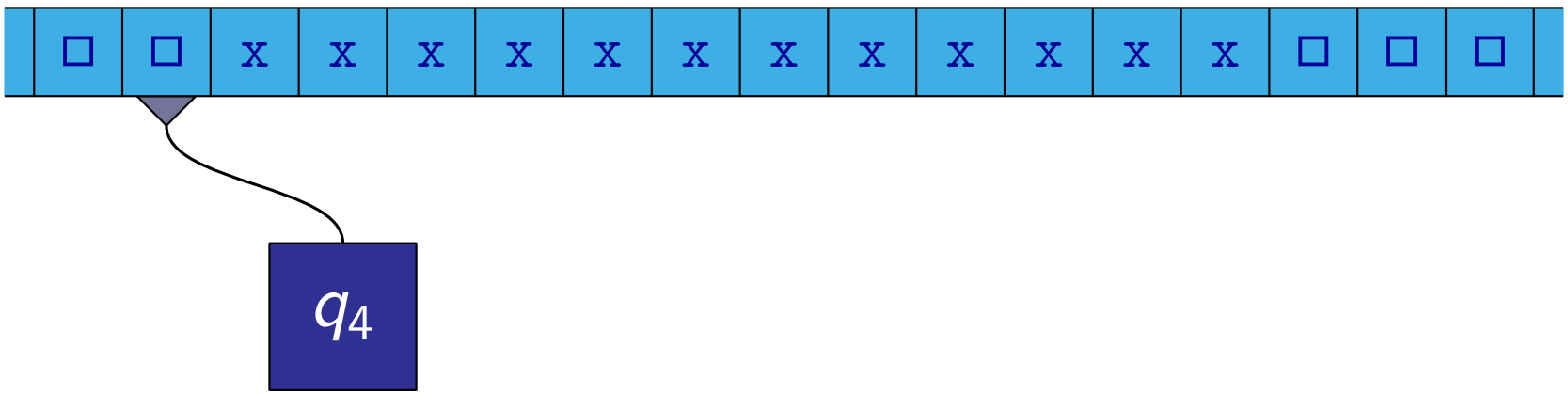
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



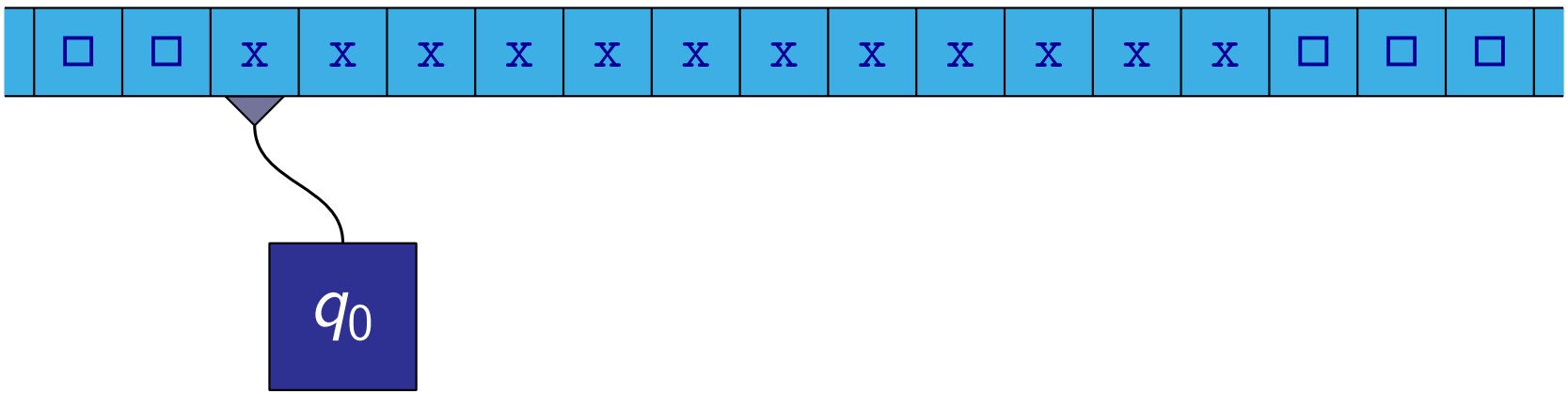
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



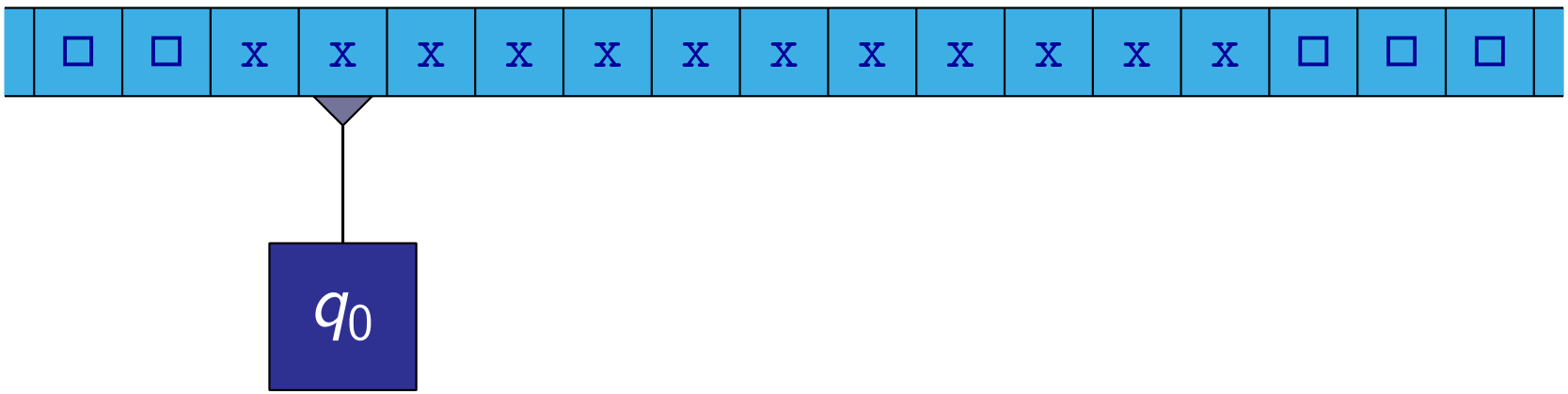
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



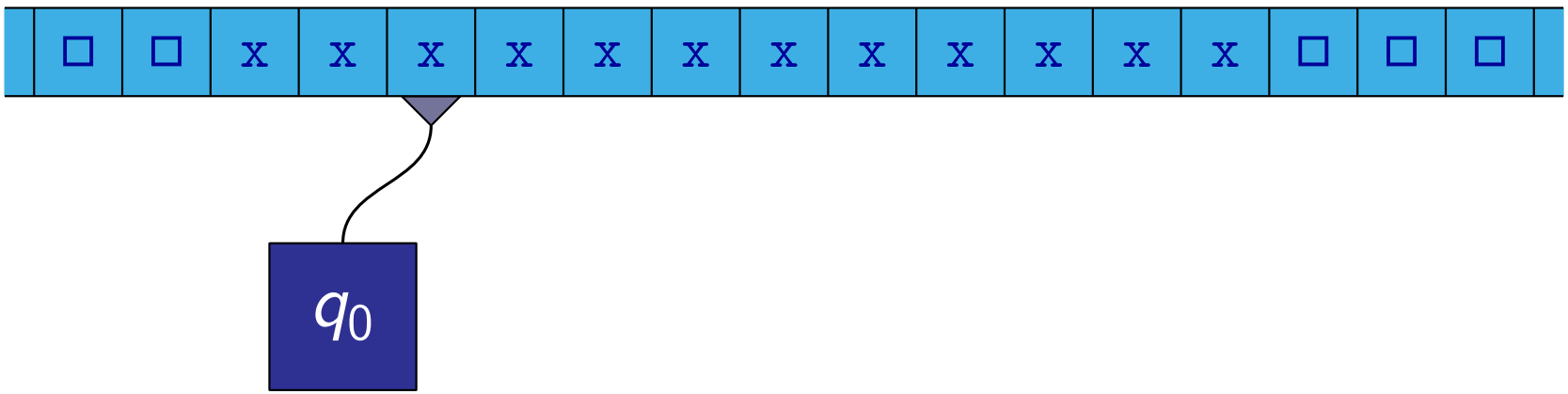
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



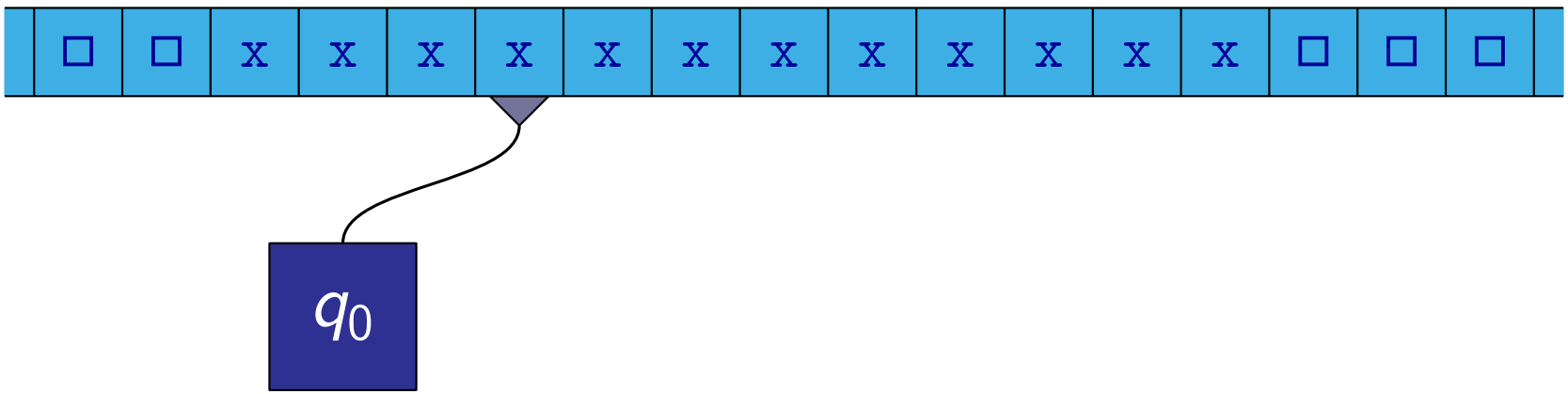
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



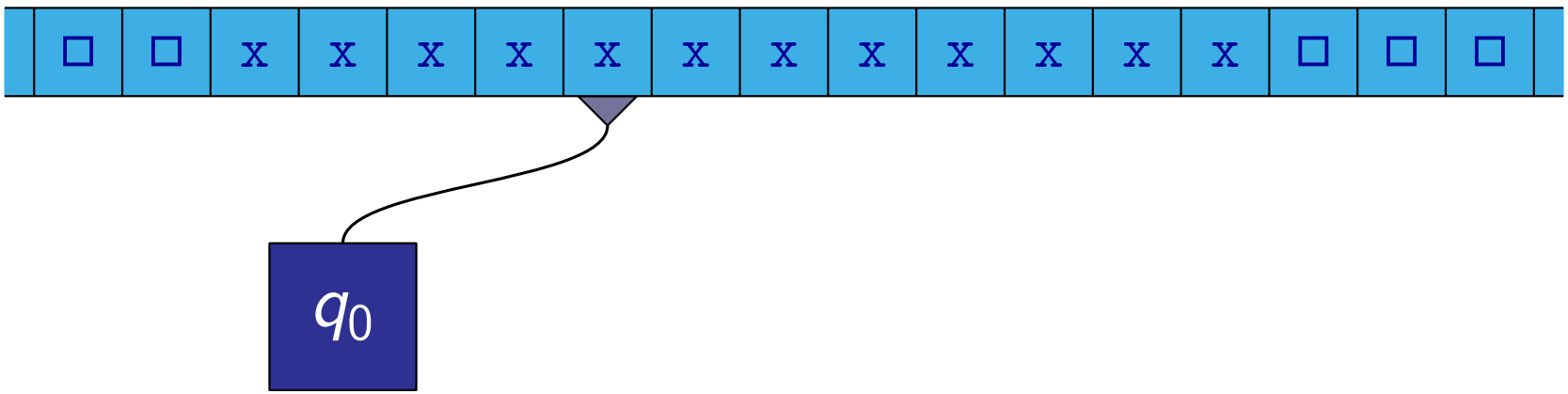
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



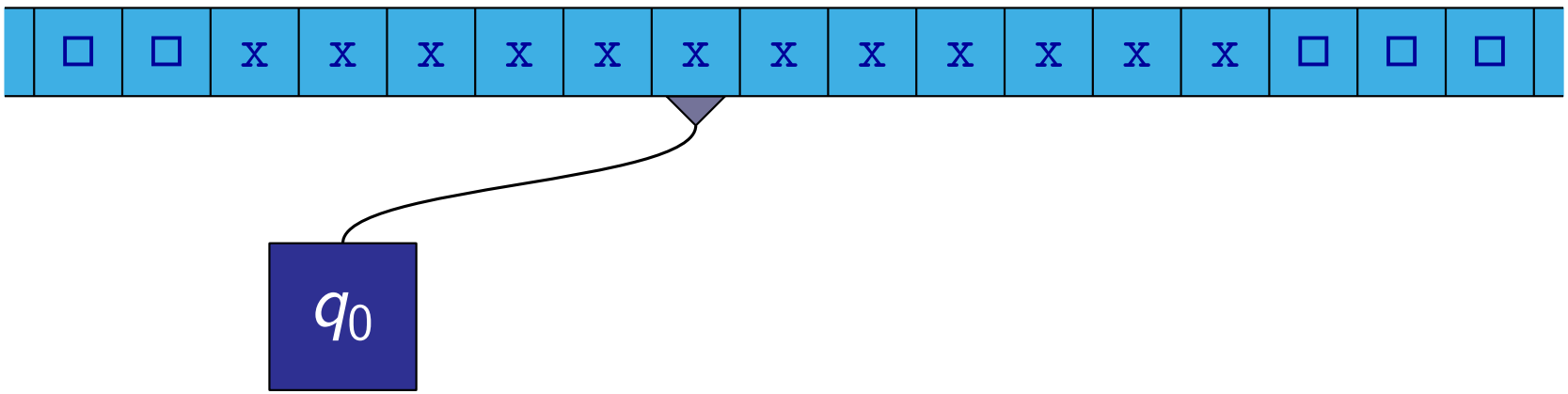
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



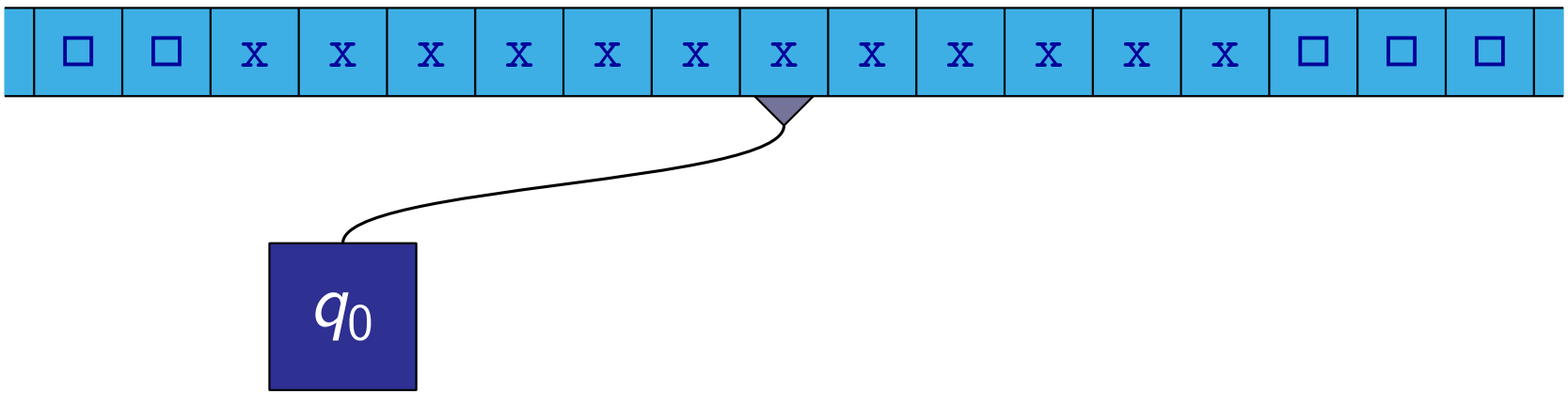
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



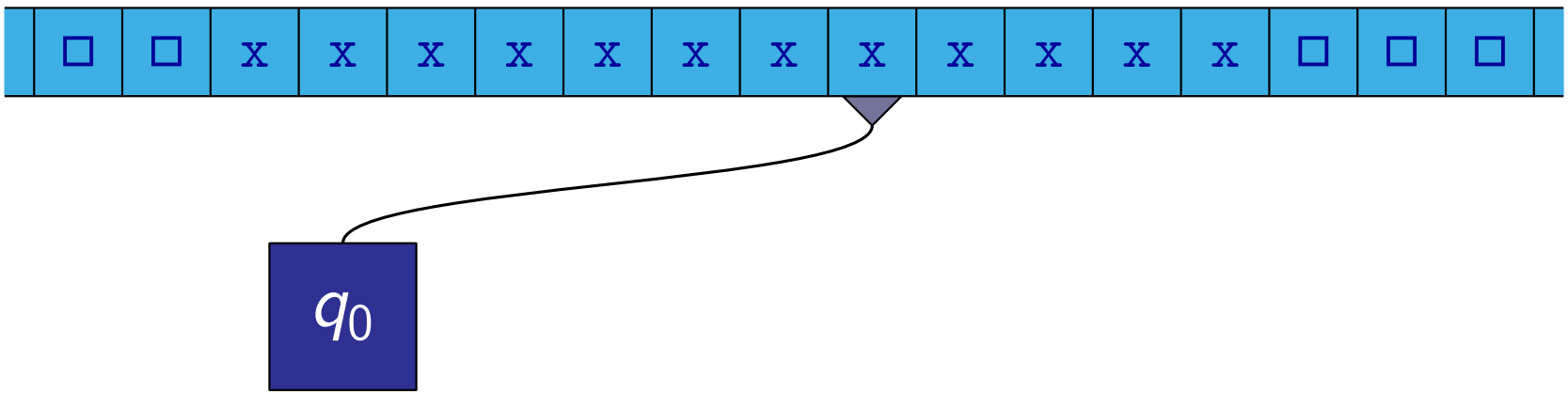
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



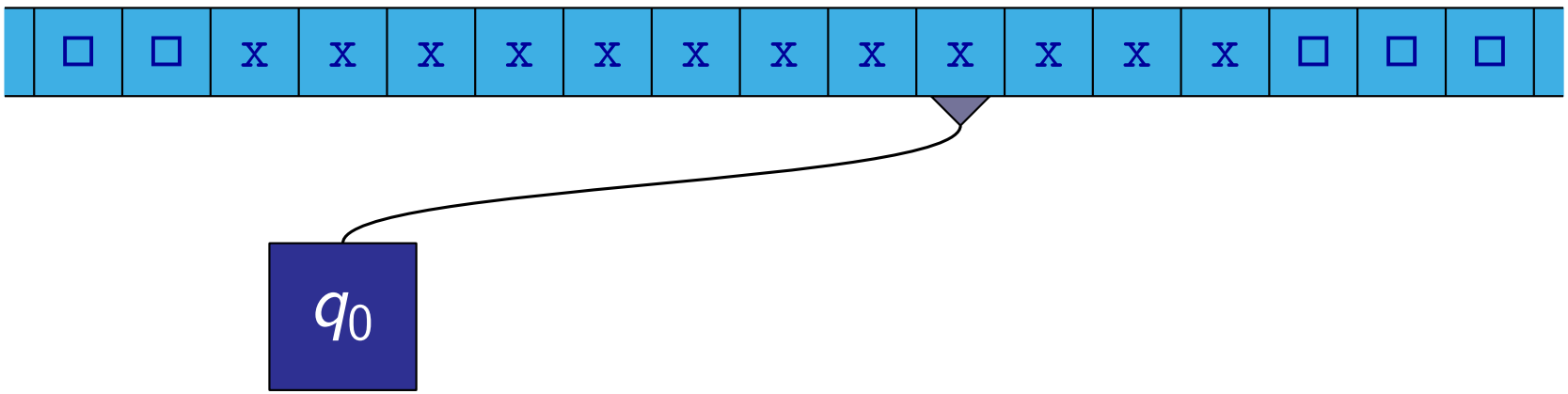
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



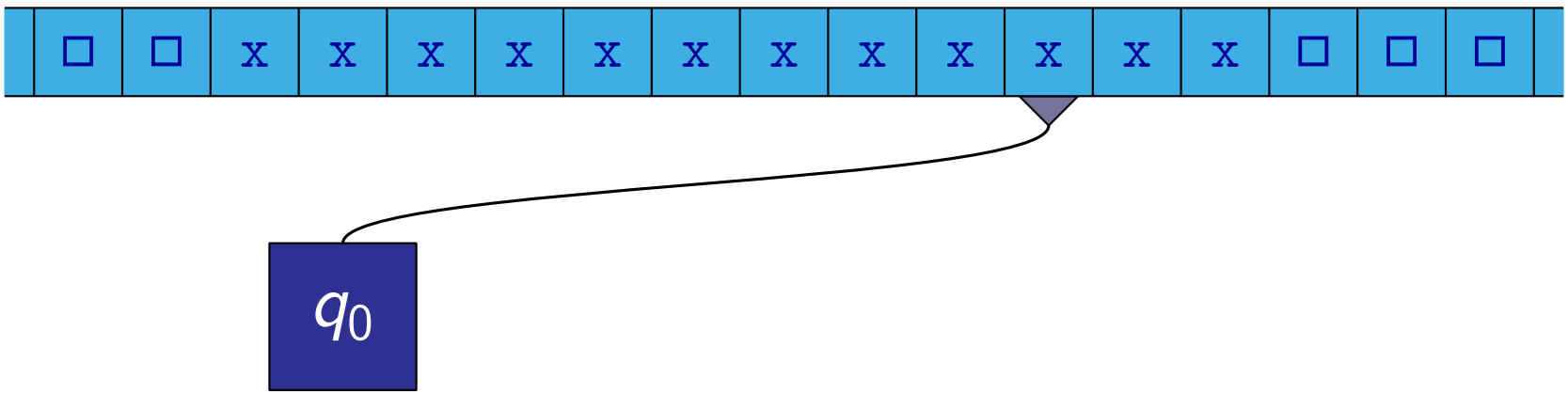
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



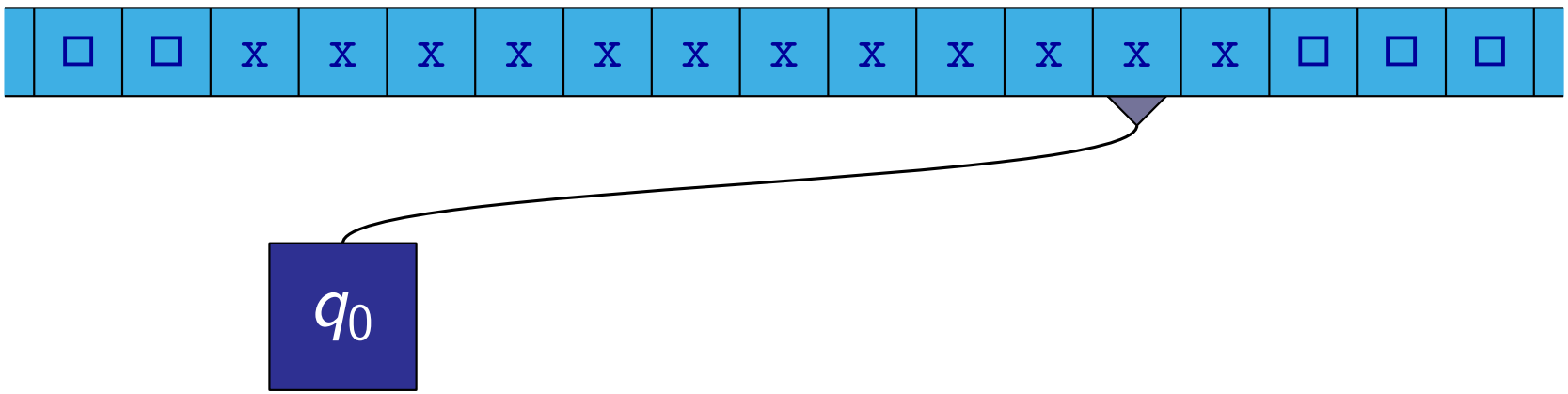
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



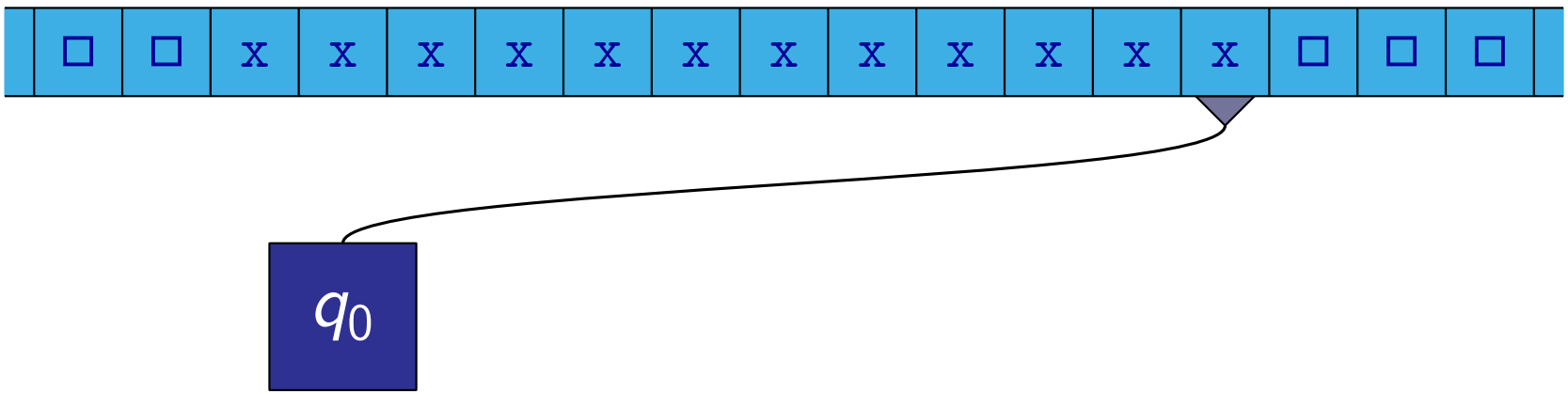
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



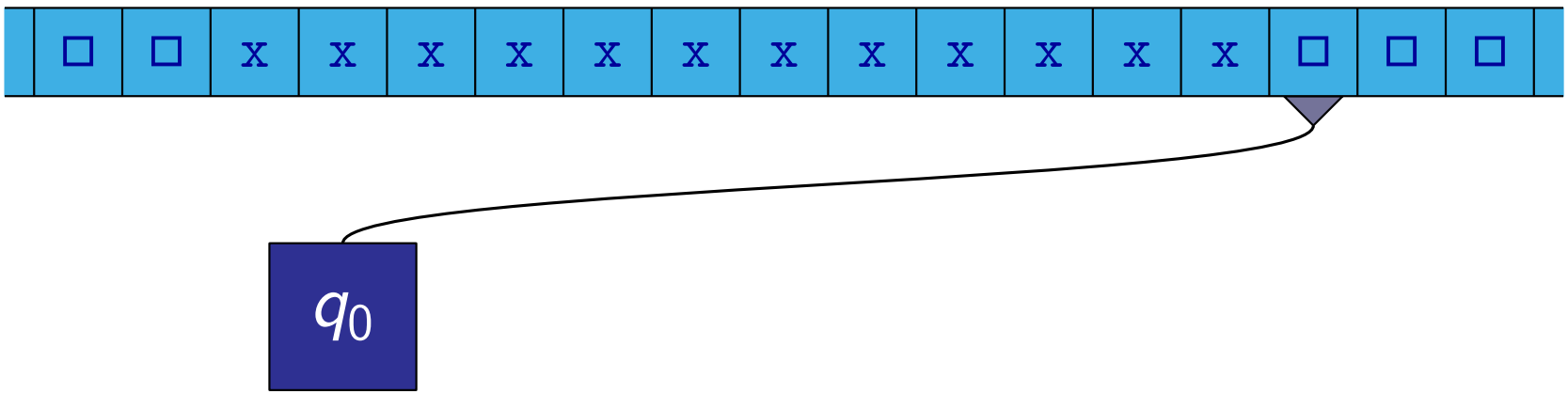
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



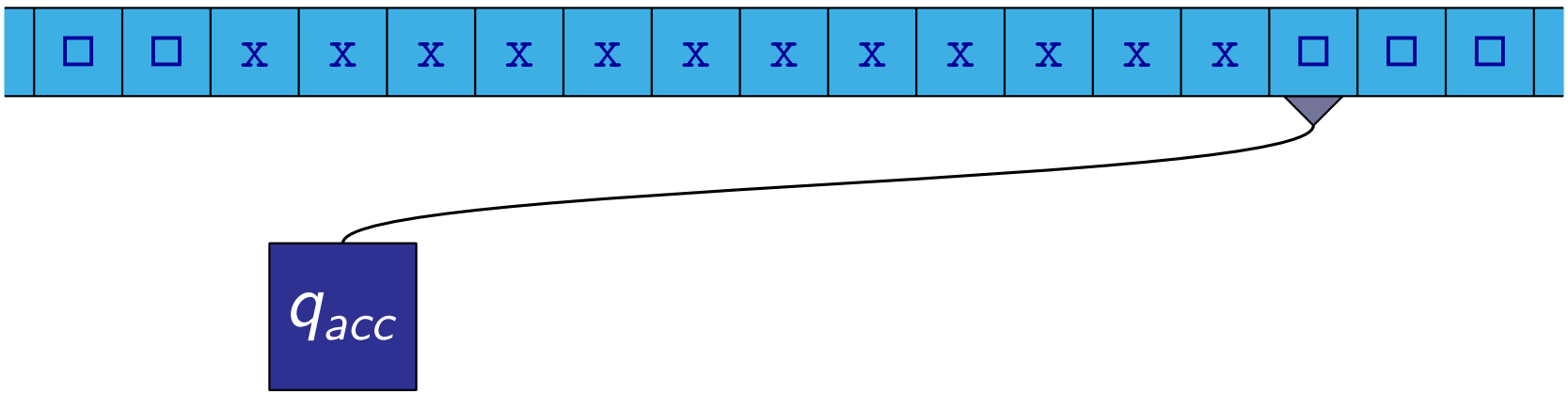
Turing Machine

Language $L = \{a^n b^n c^n \mid n \geq 0\}$

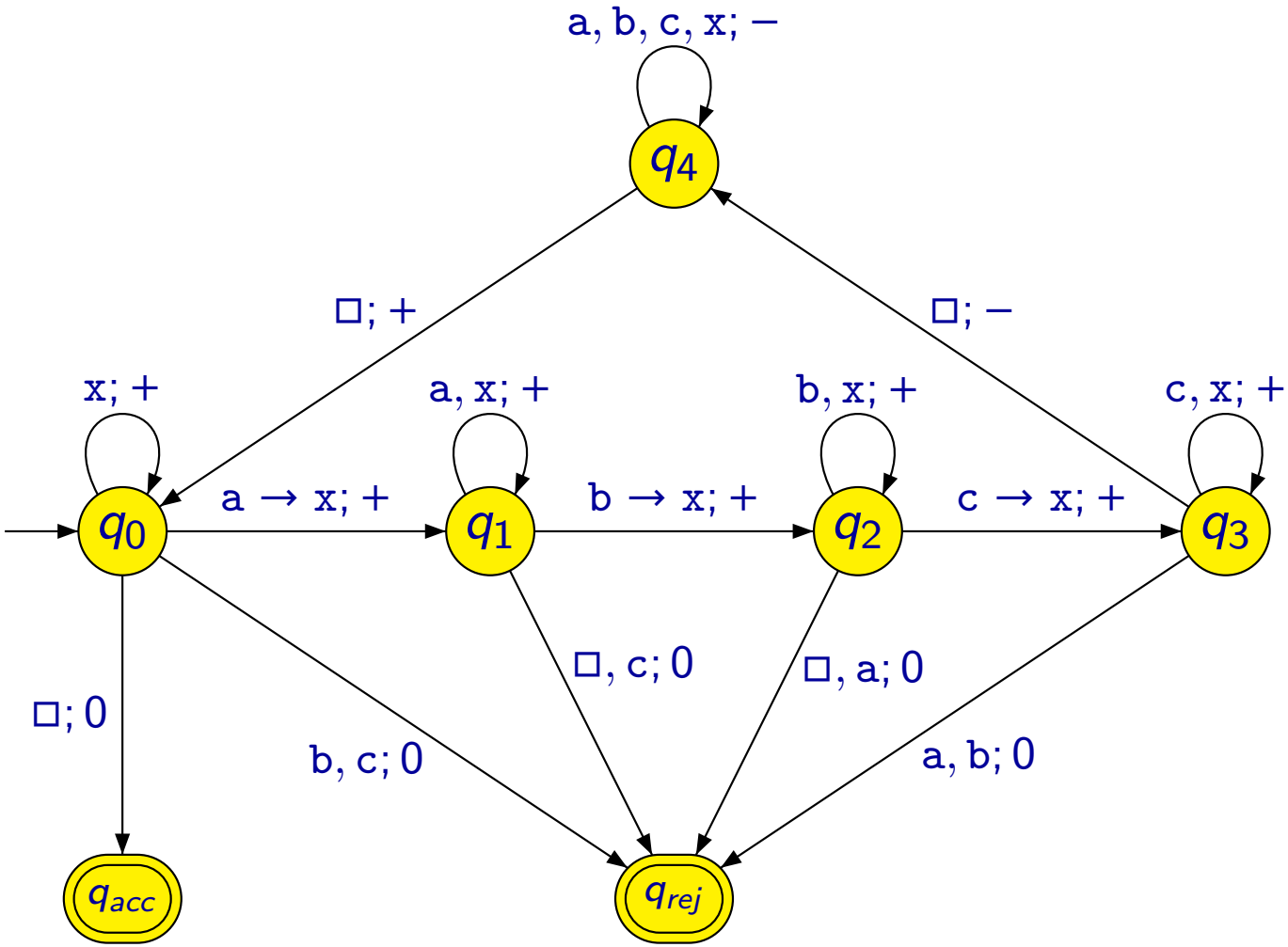
$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$ $\quad F = \{q_{acc}, q_{rej}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{acc}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{rej}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{rej}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{rej}, \square, 0)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{rej}, \mathtt{a}, 0)$	$(q_{rej}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



Turing Machine

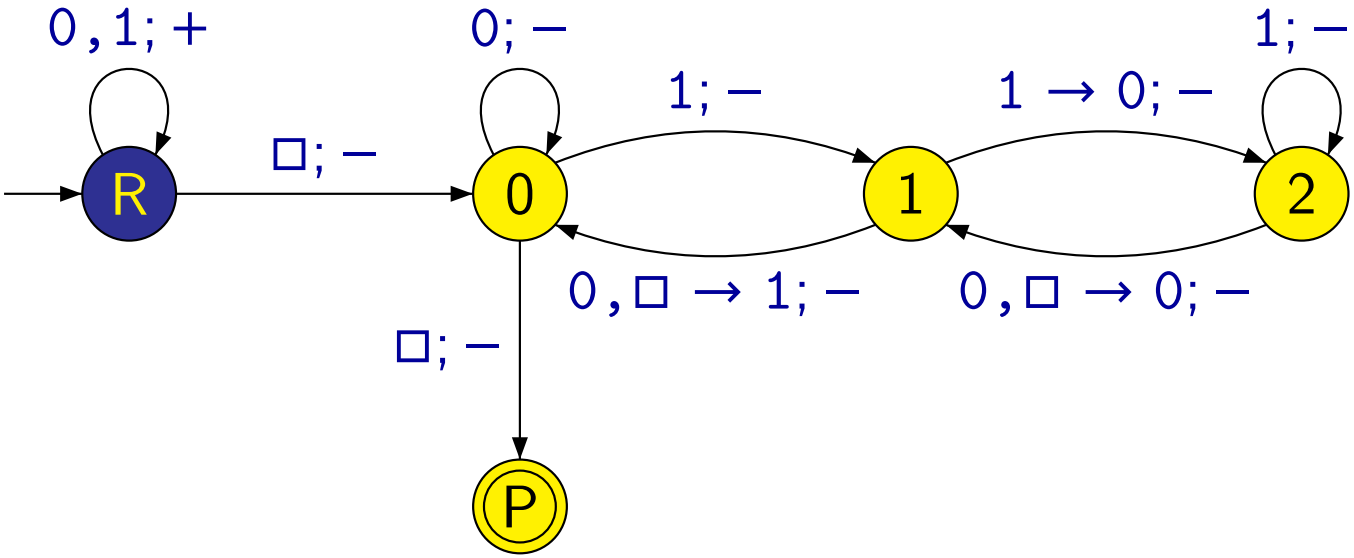


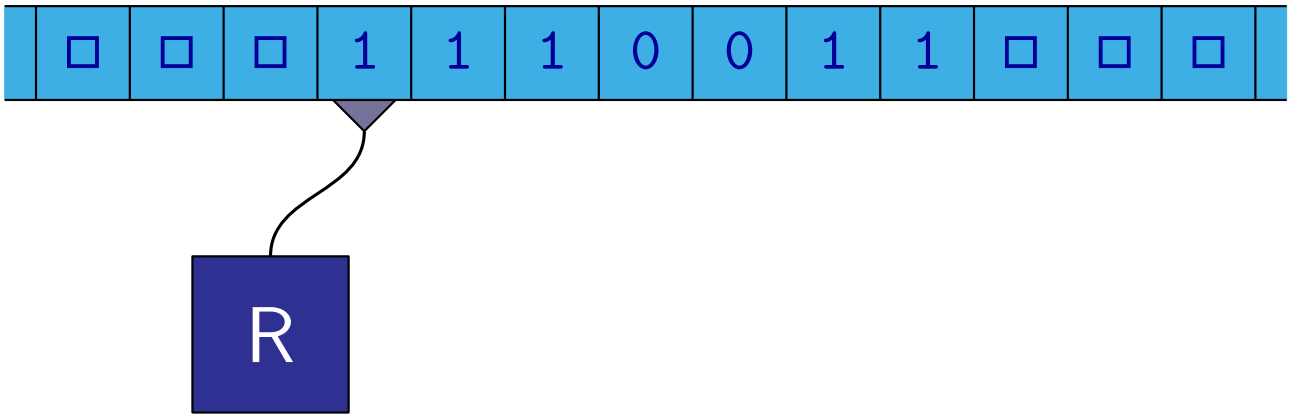
Turing Machine

- A Turing machine can give not only answers YES or NO but it can also compute a function that assigns to each word from Σ^* some other word (from Γ^*).

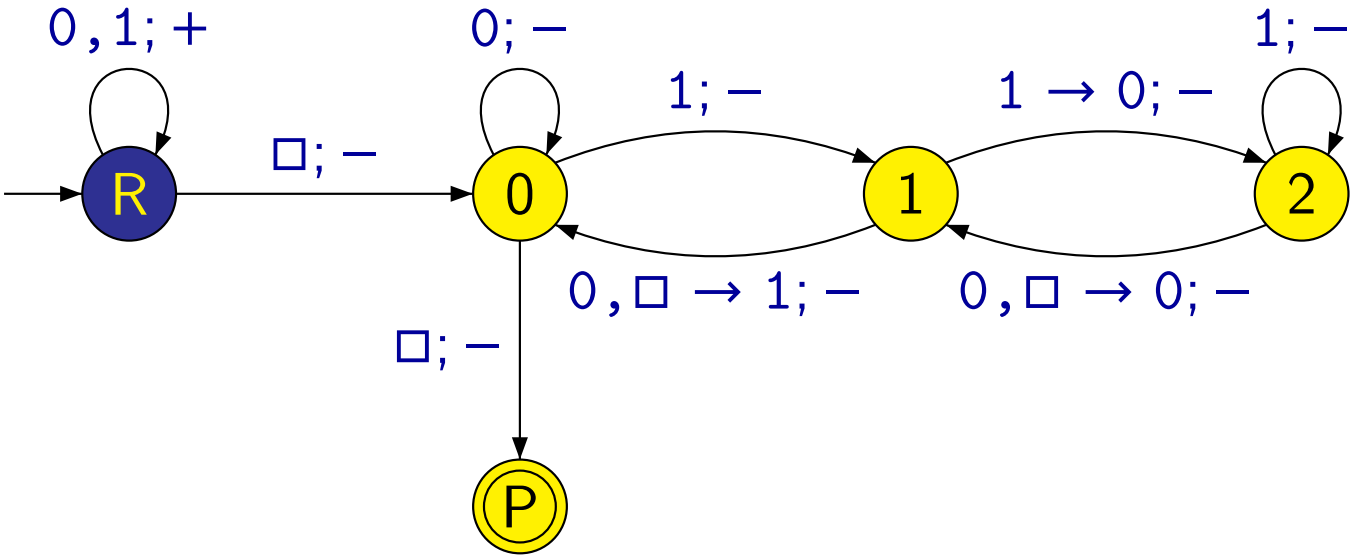
- A word assigned to a word w is the word that remains on the tape after the computation over the word w when we remove all symbols $\square$.

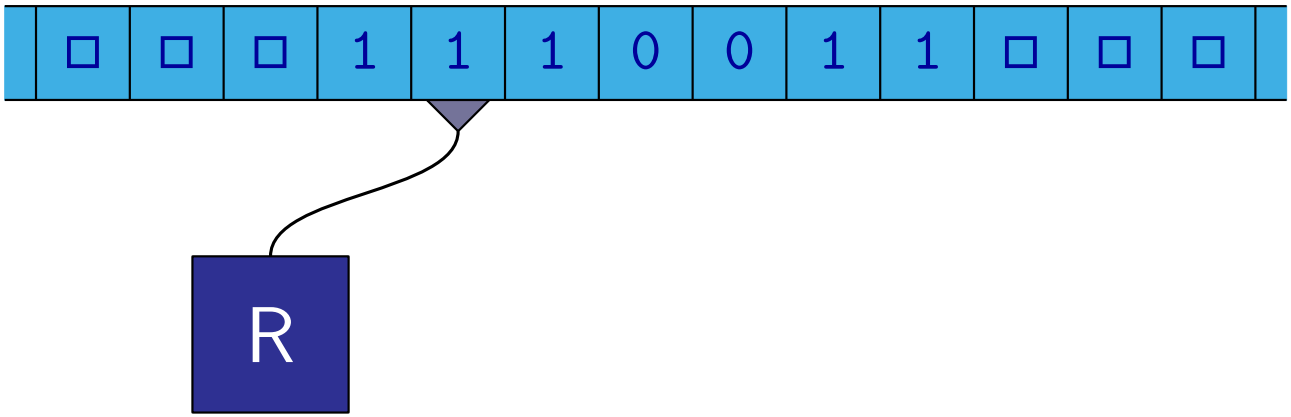
Turing Machine – Multiplication by Three



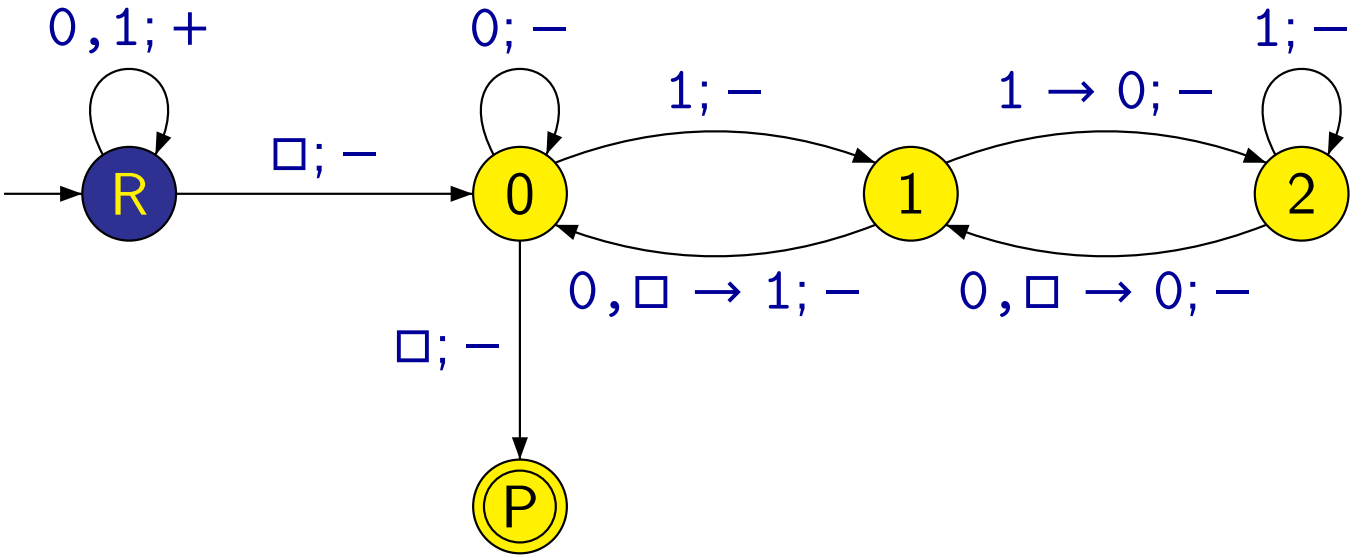


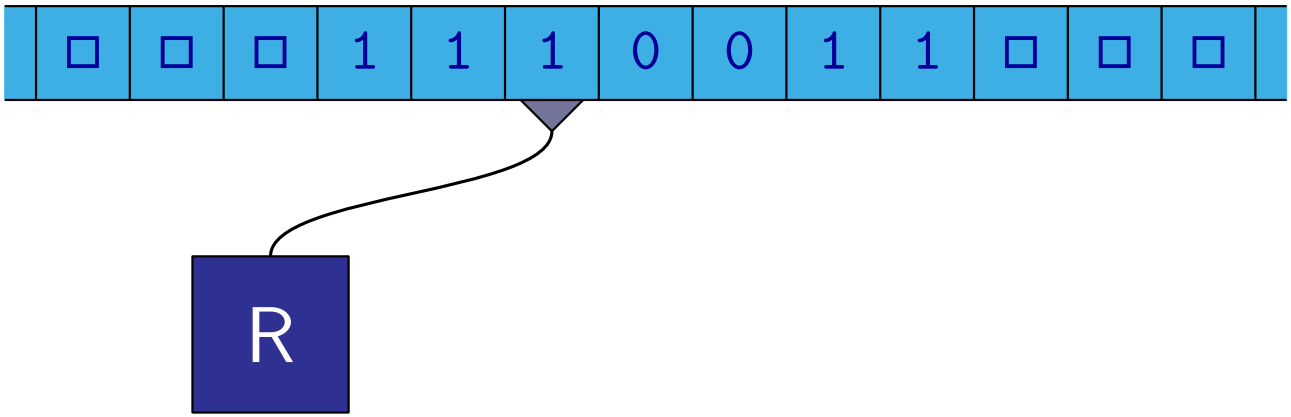
Turing Machine – Multiplication by Three



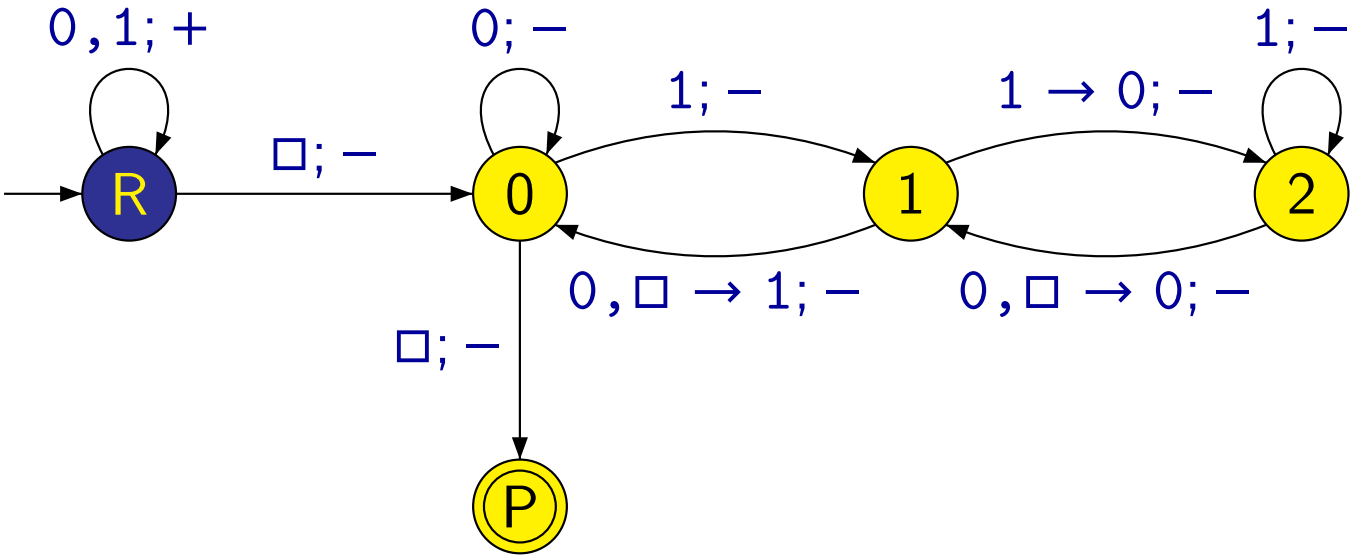


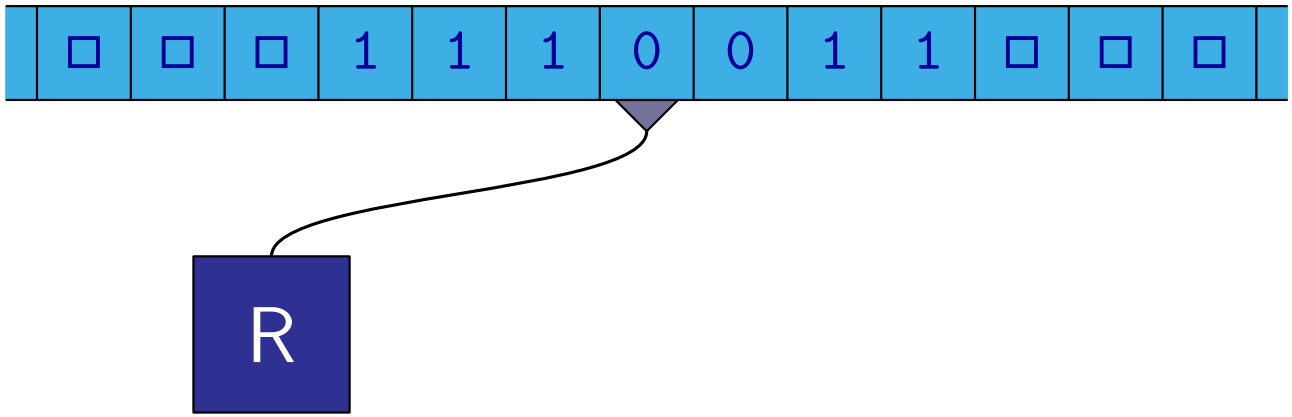
Turing Machine – Multiplication by Three



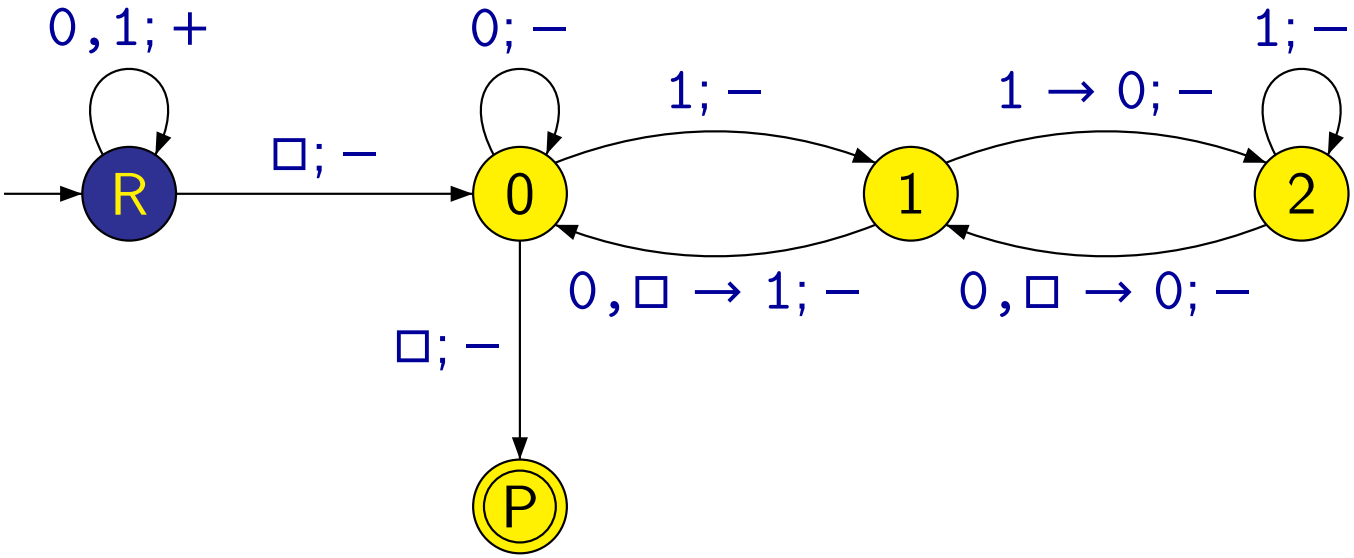


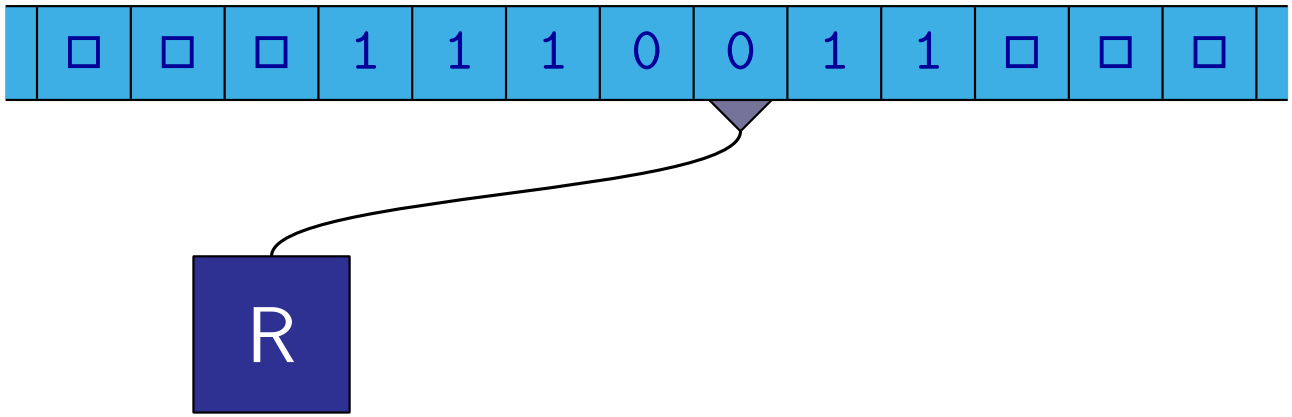
Turing Machine – Multiplication by Three



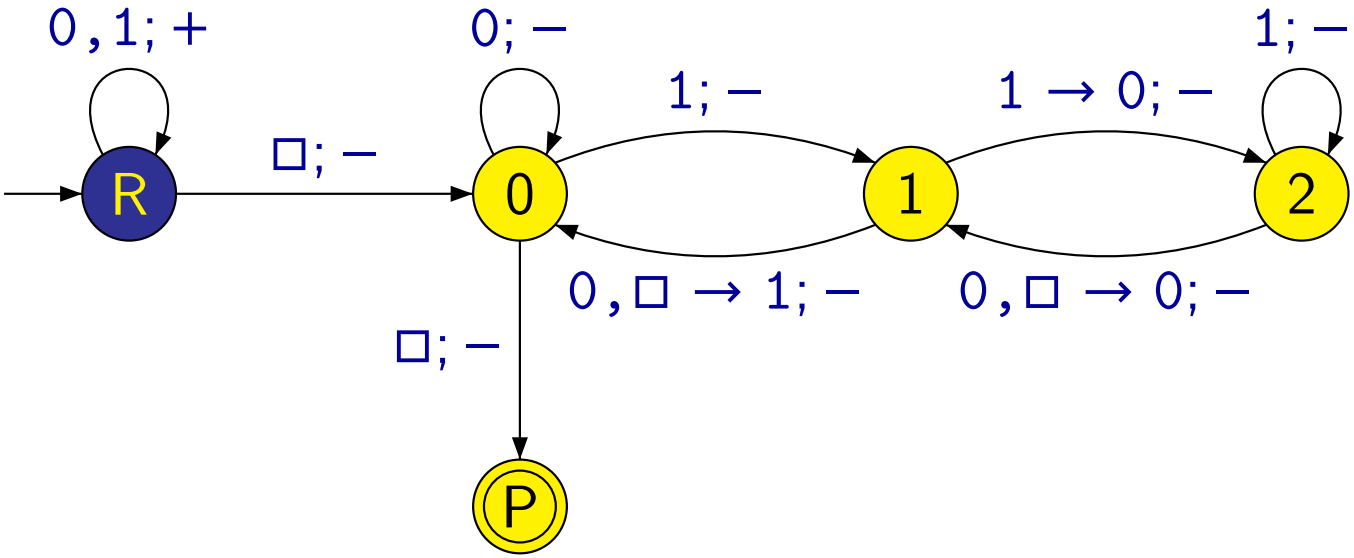


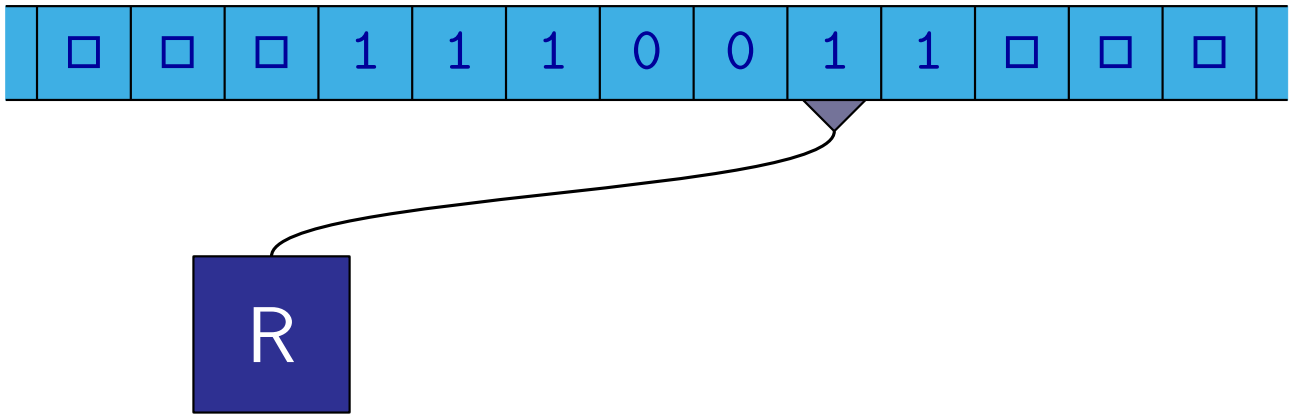
Turing Machine – Multiplication by Three



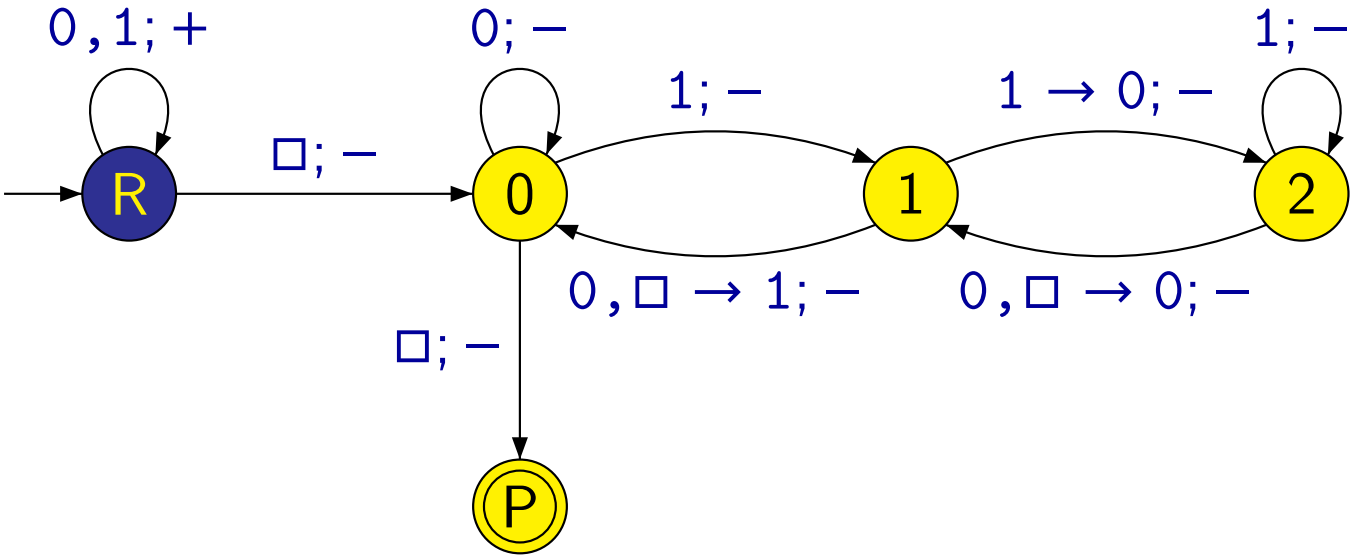


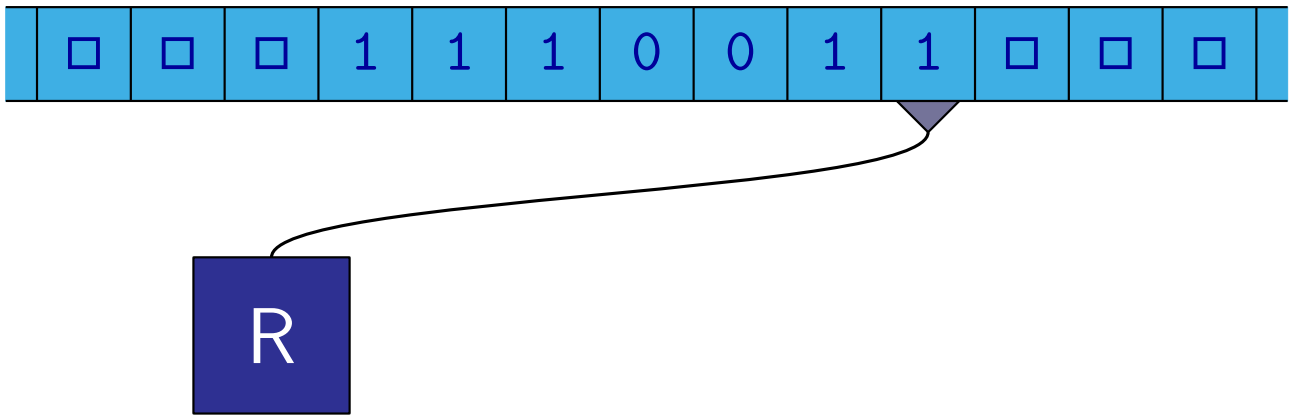
Turing Machine – Multiplication by Three



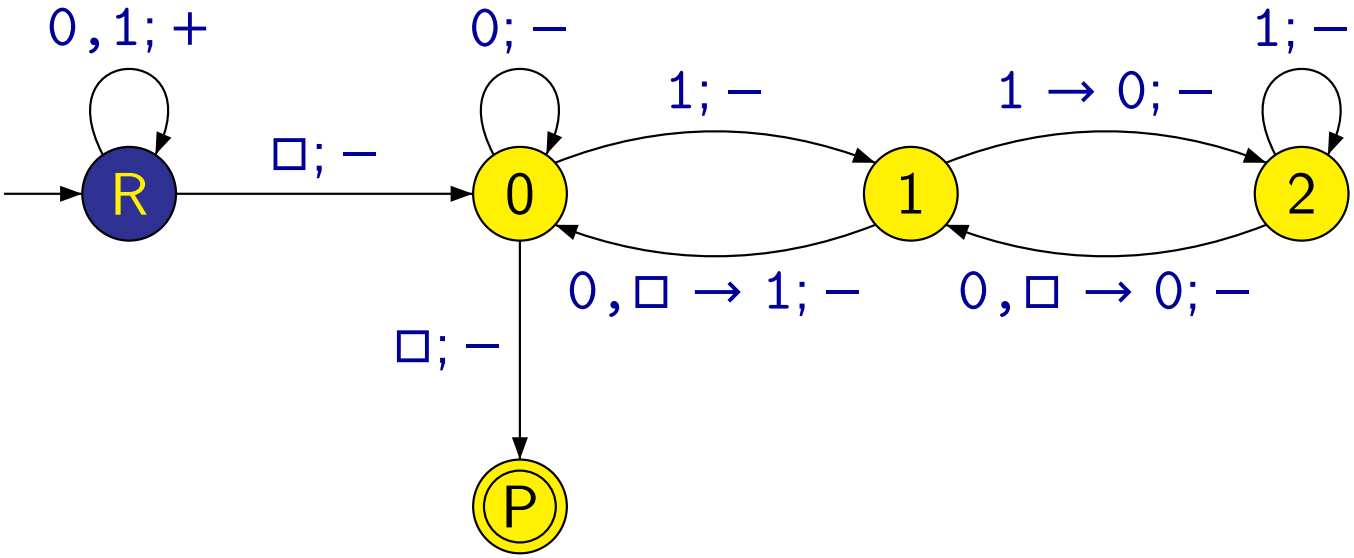


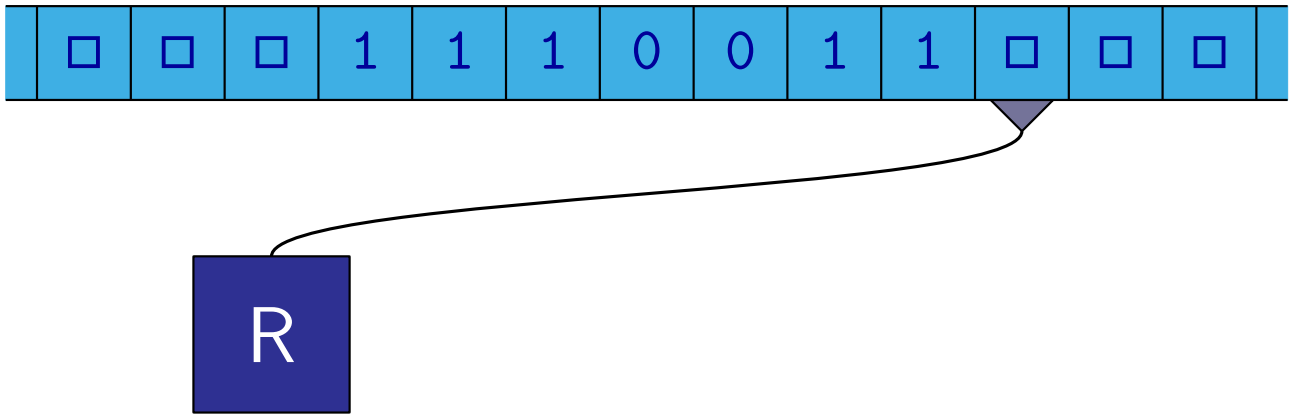
Turing Machine – Multiplication by Three



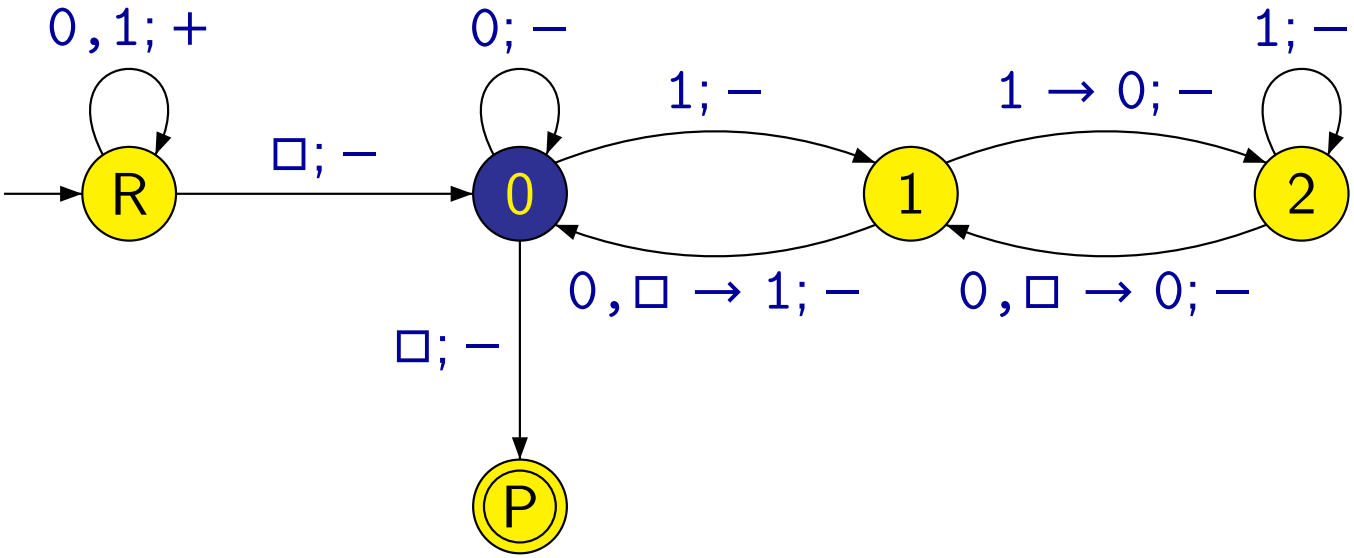


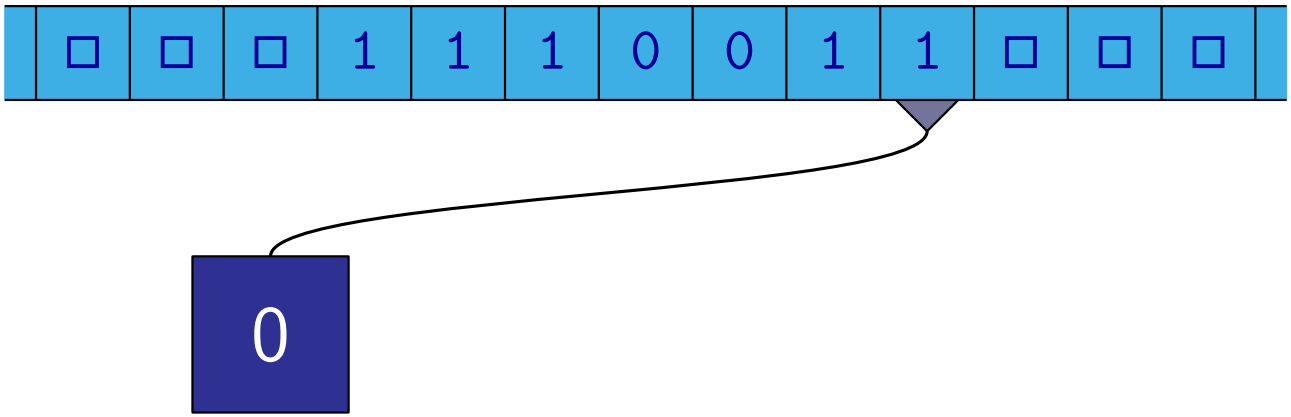
Turing Machine – Multiplication by Three



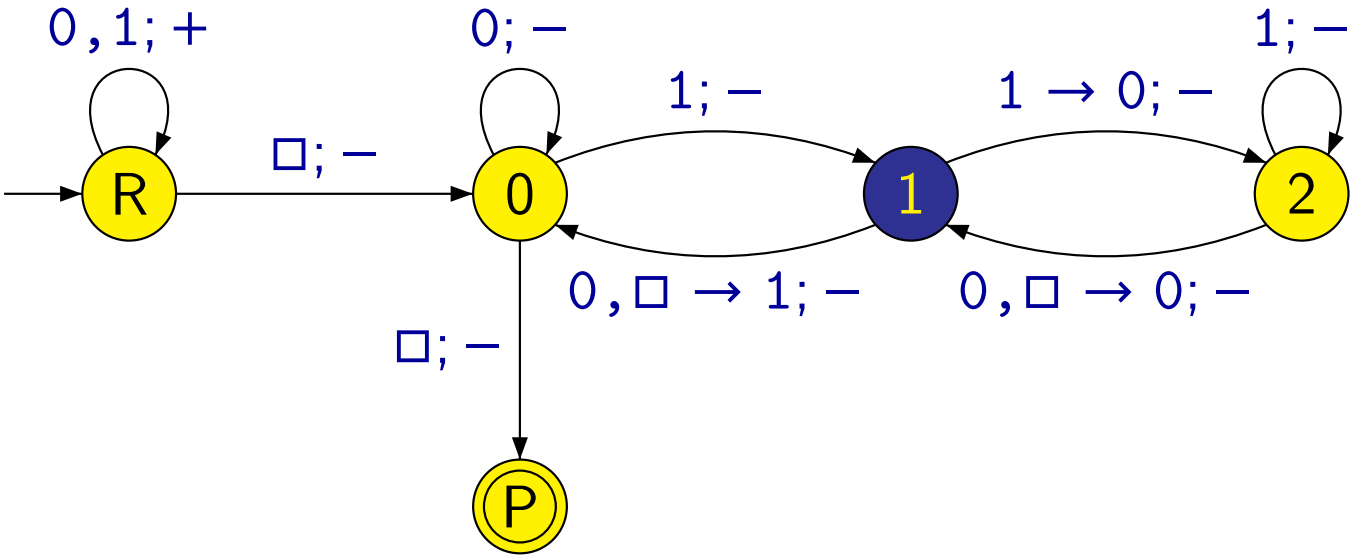


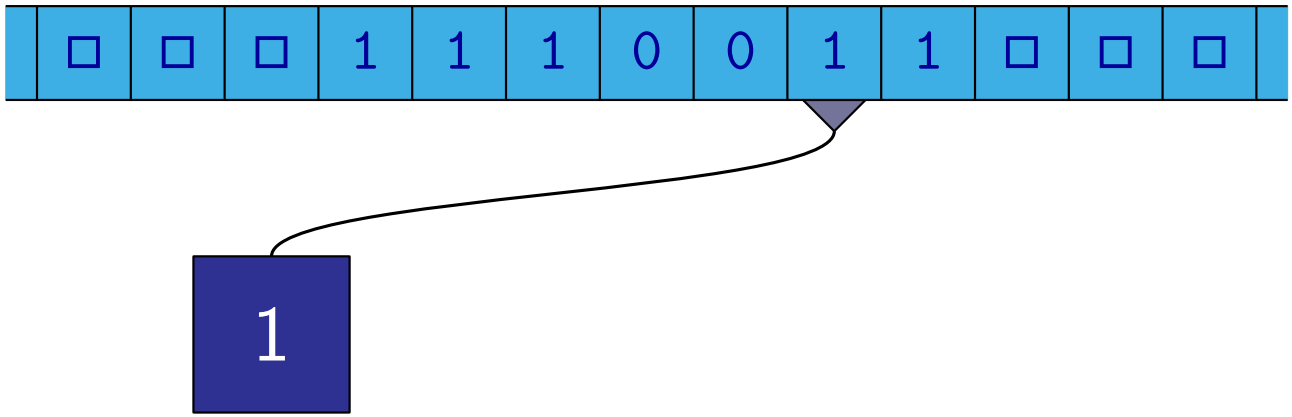
Turing Machine – Multiplication by Three



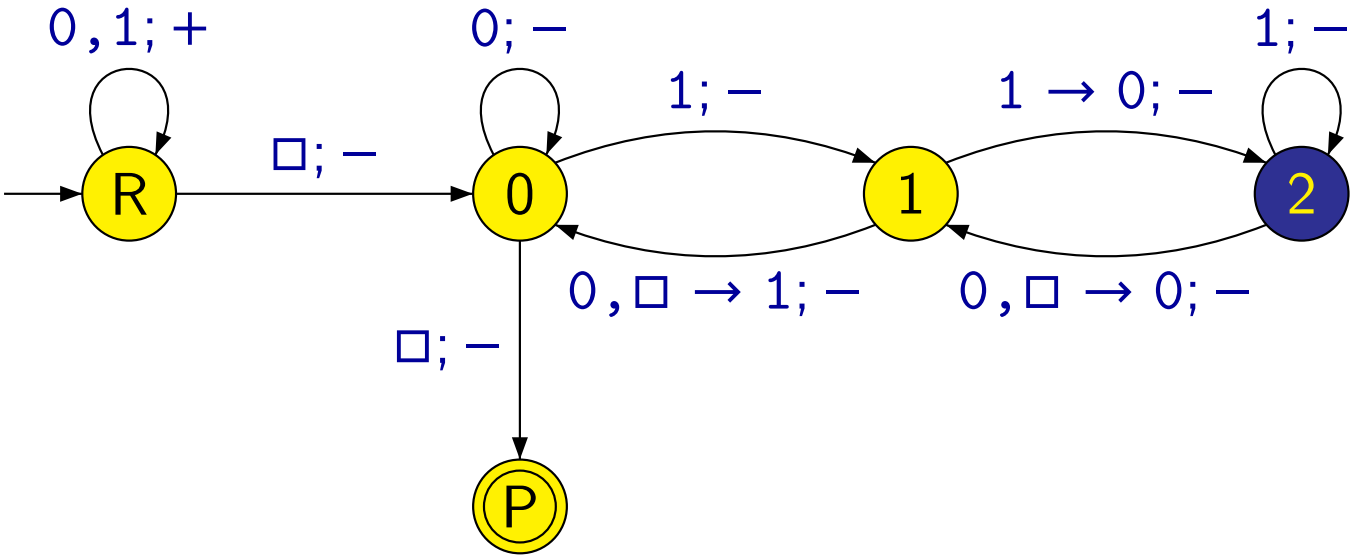


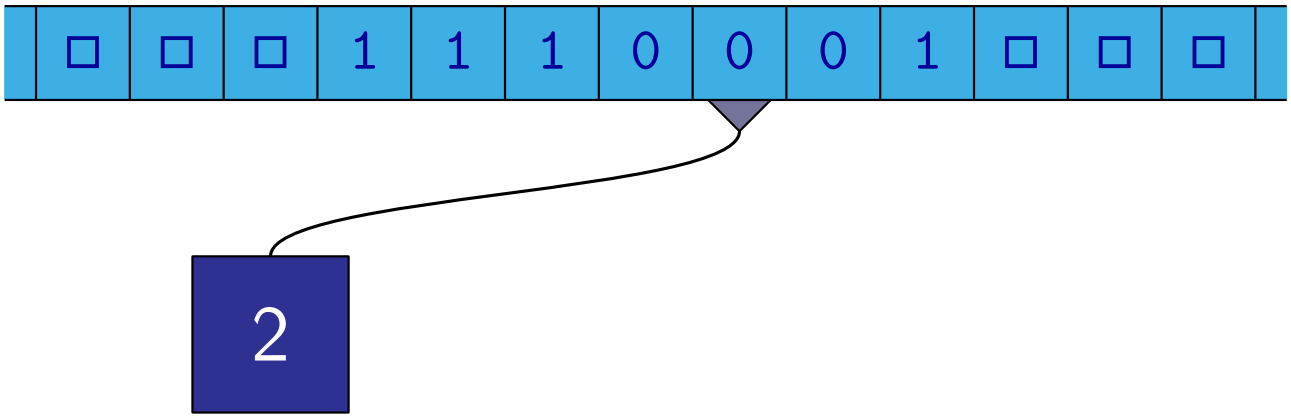
Turing Machine – Multiplication by Three



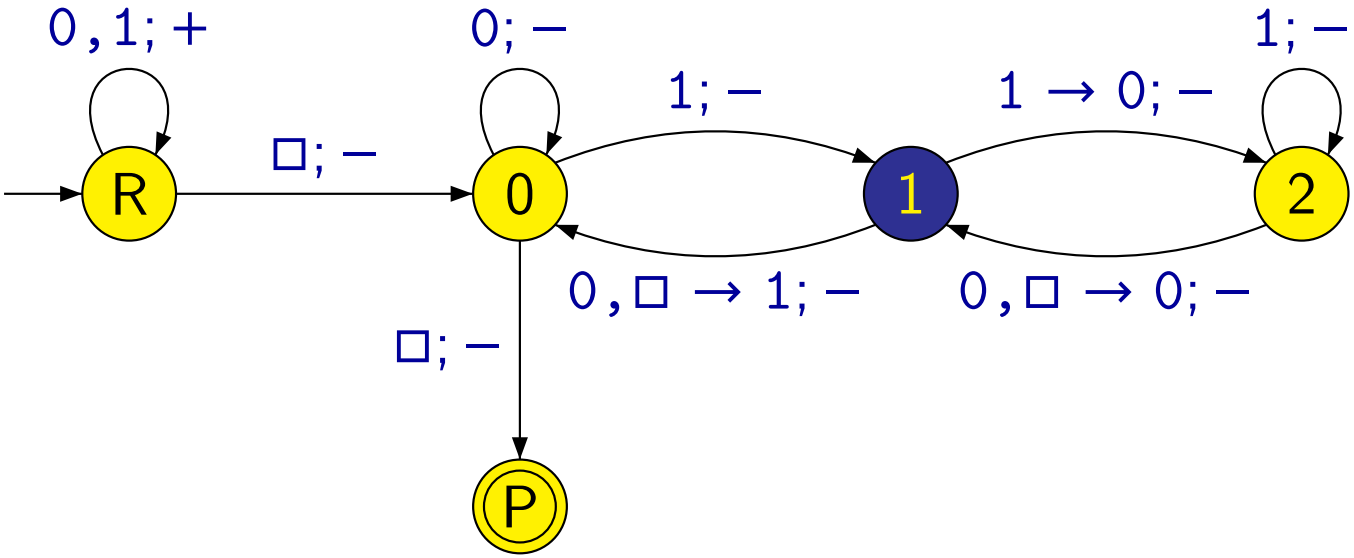


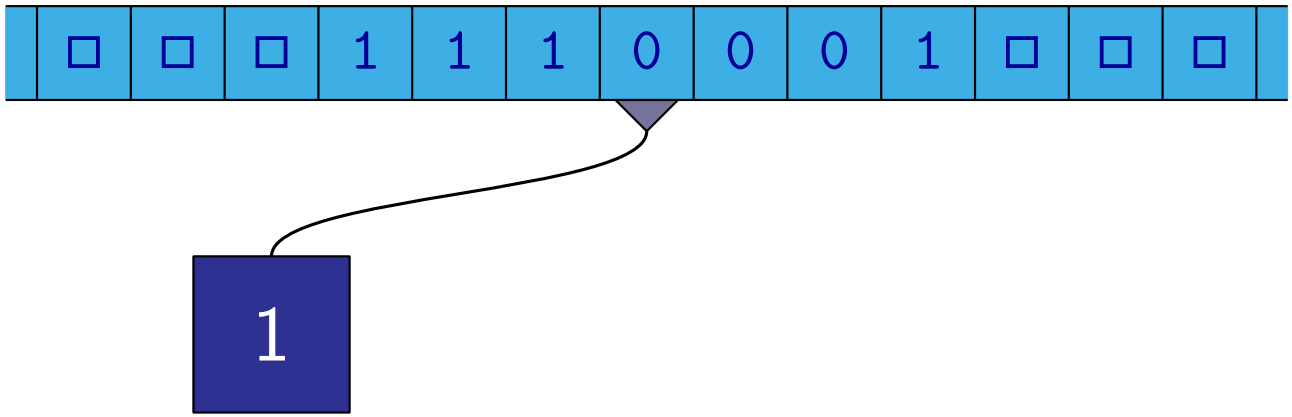
Turing Machine – Multiplication by Three



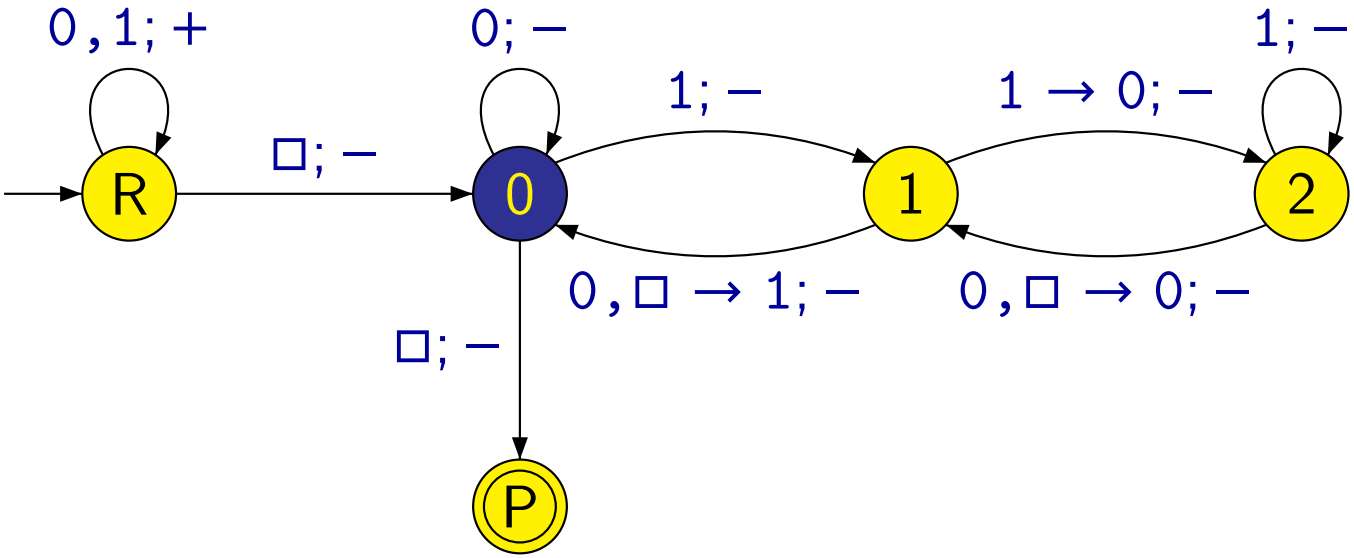


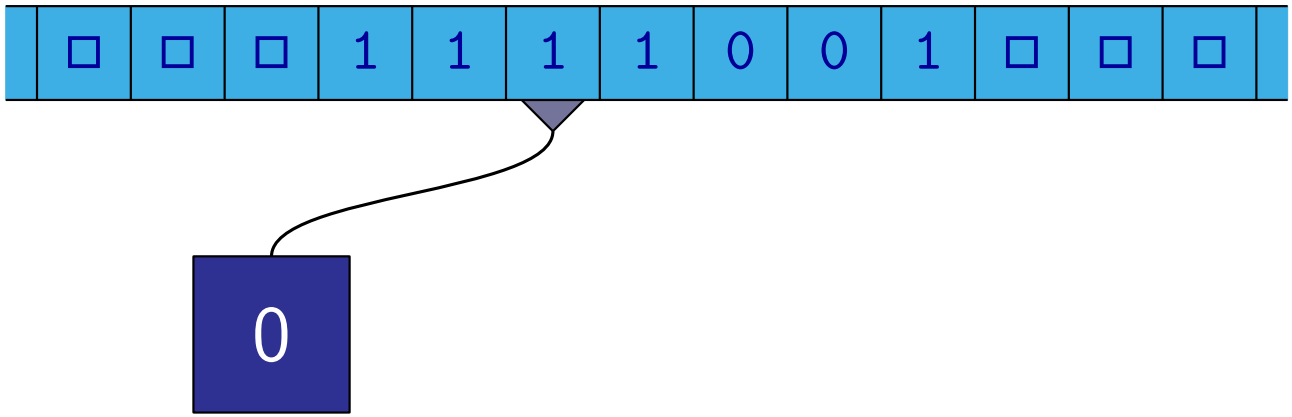
Turing Machine – Multiplication by Three



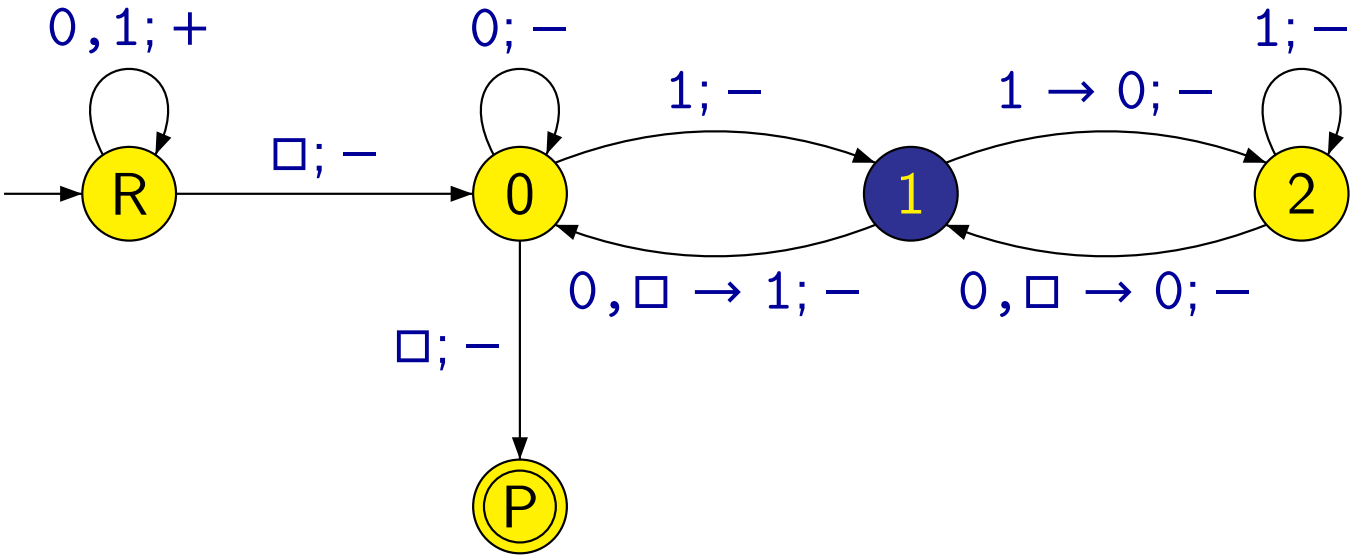


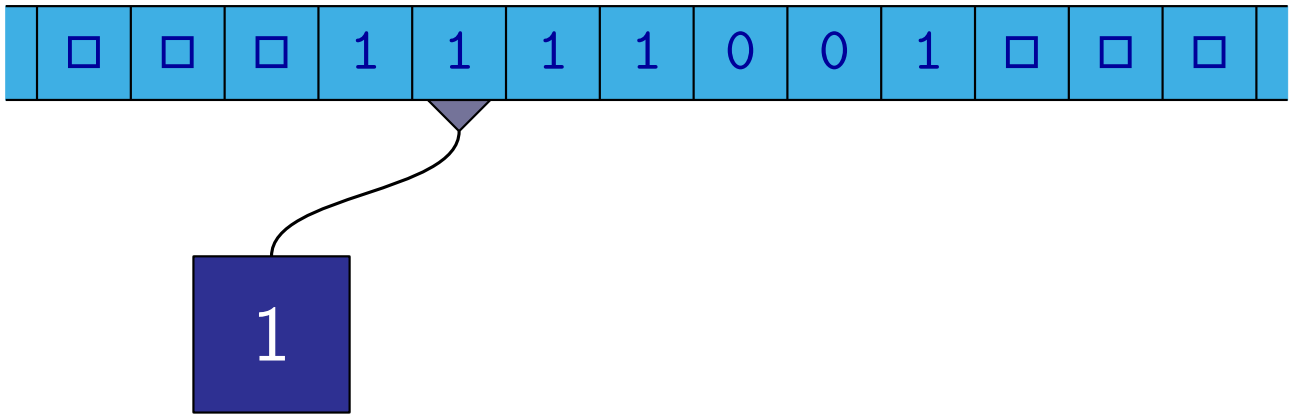
Turing Machine – Multiplication by Three



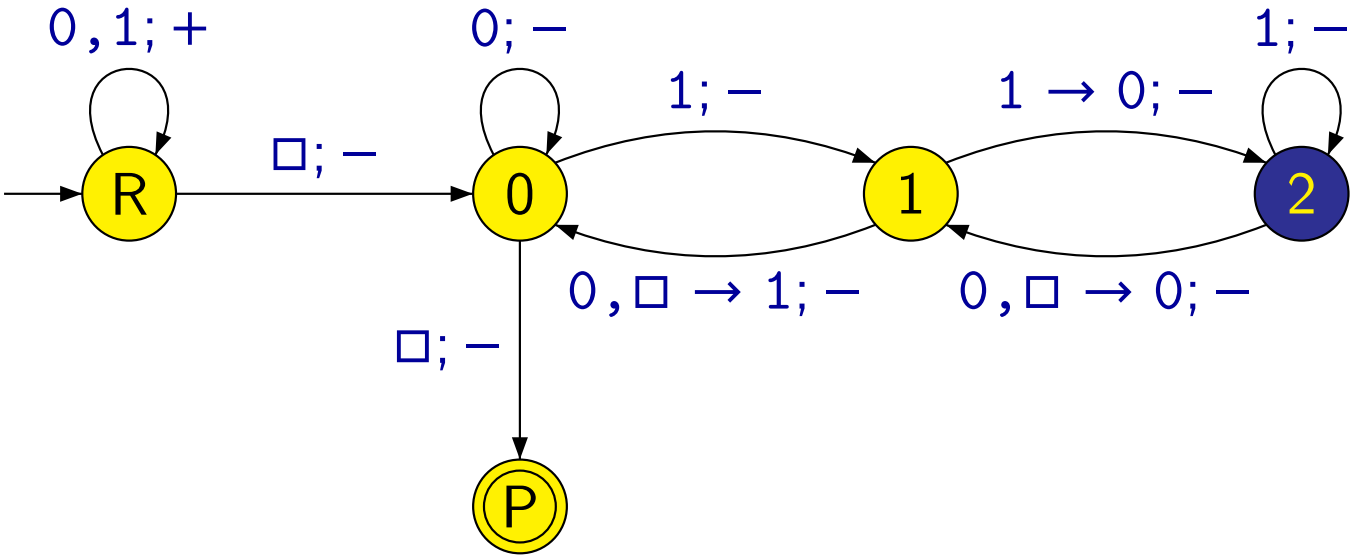


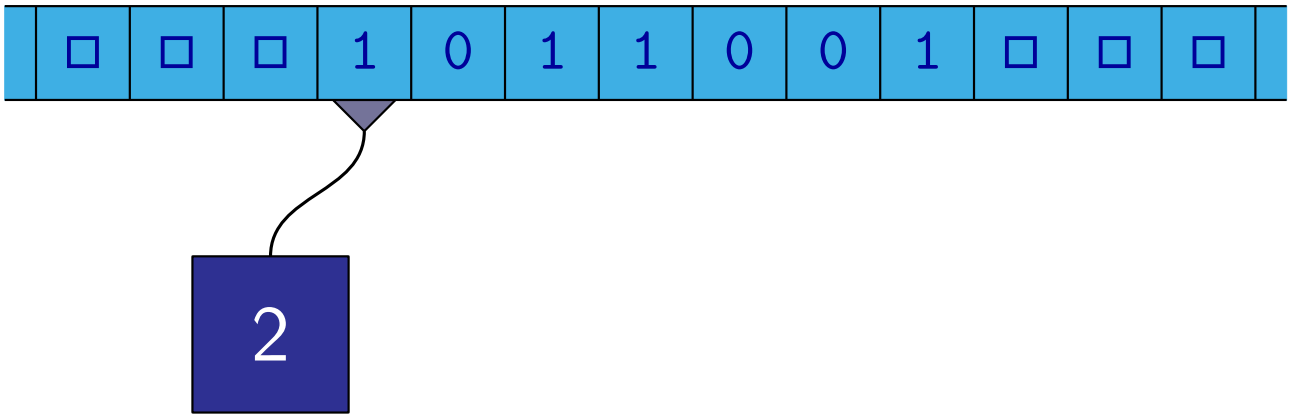
Turing Machine – Multiplication by Three



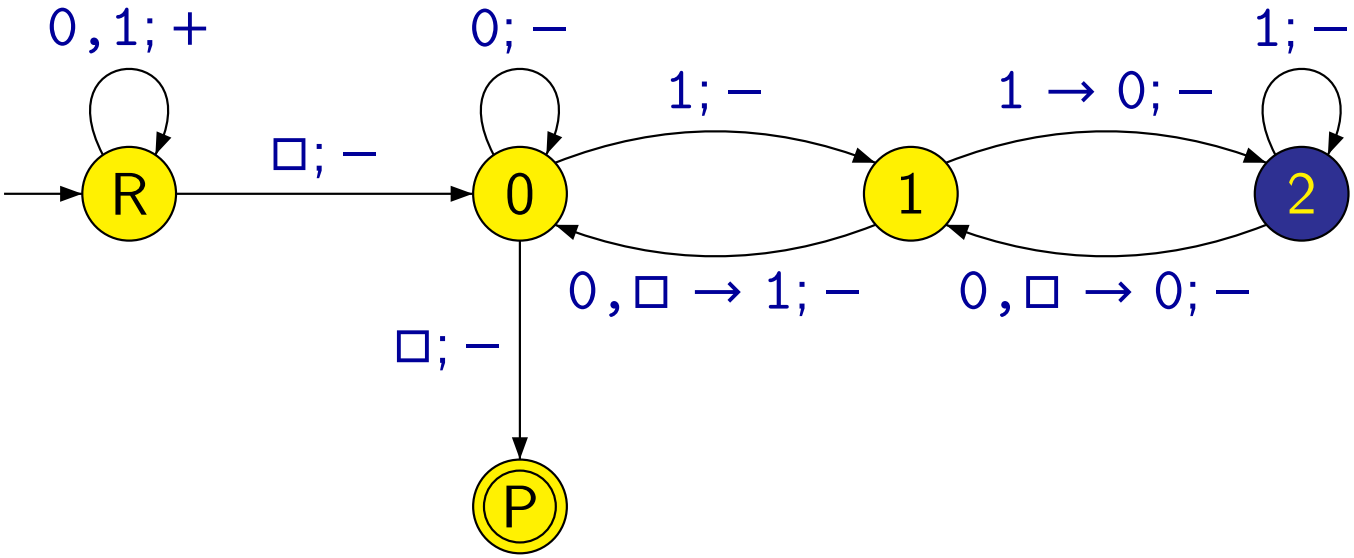


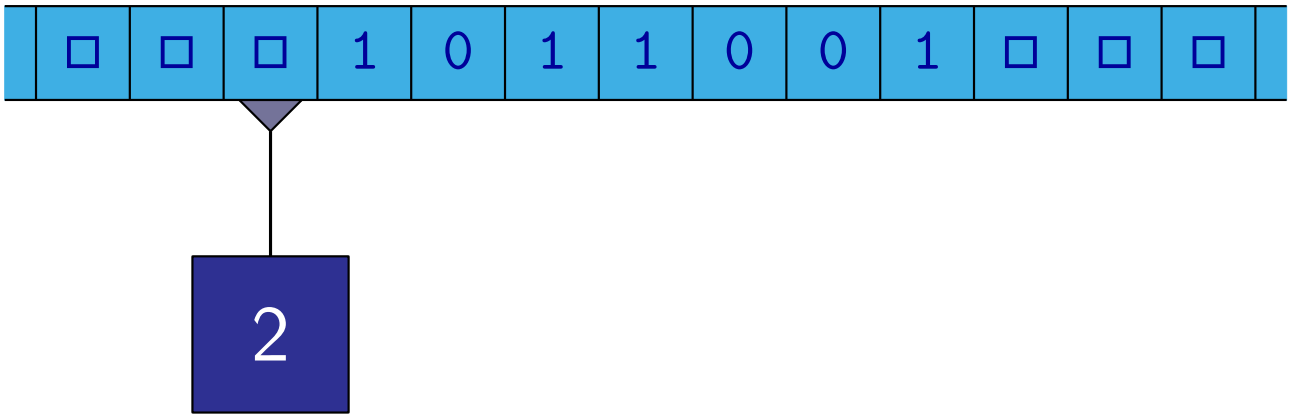
Turing Machine – Multiplication by Three



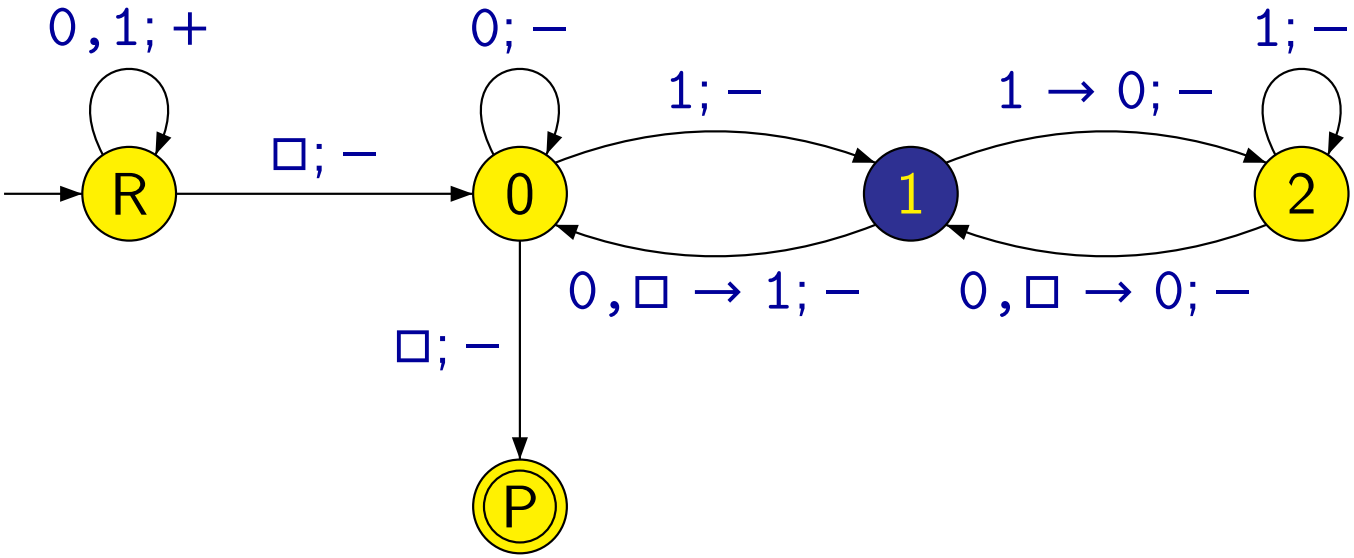


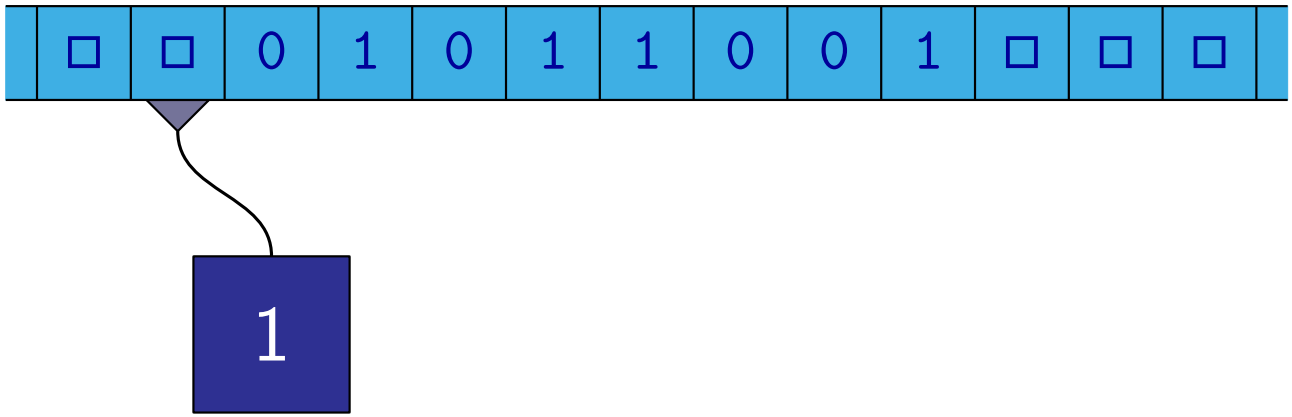
Turing Machine – Multiplication by Three



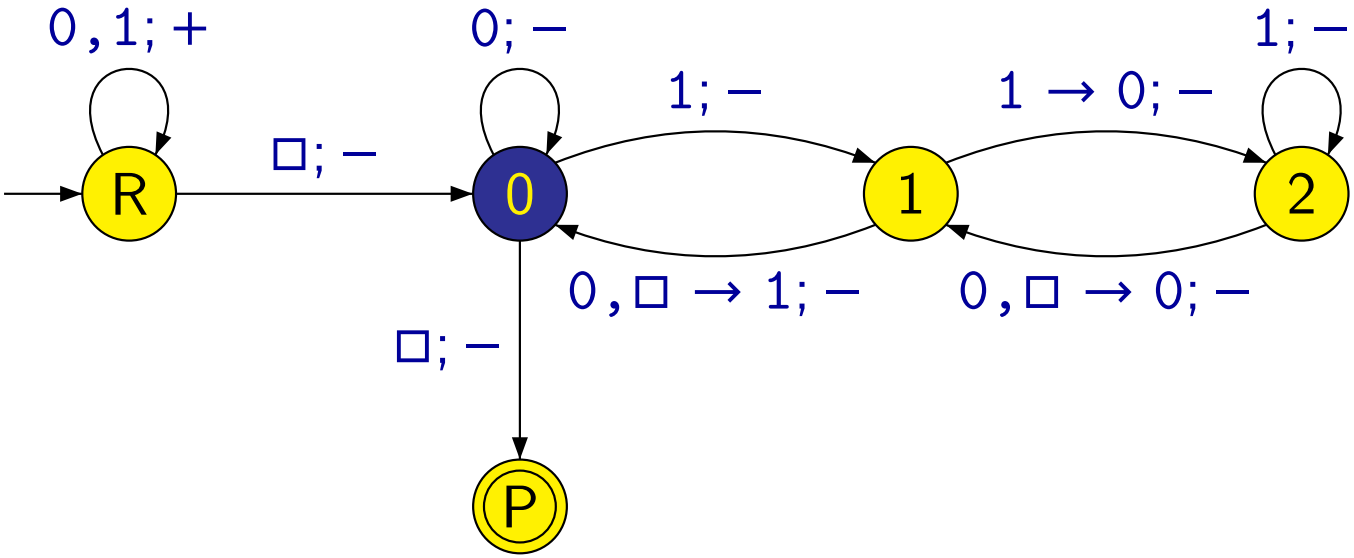


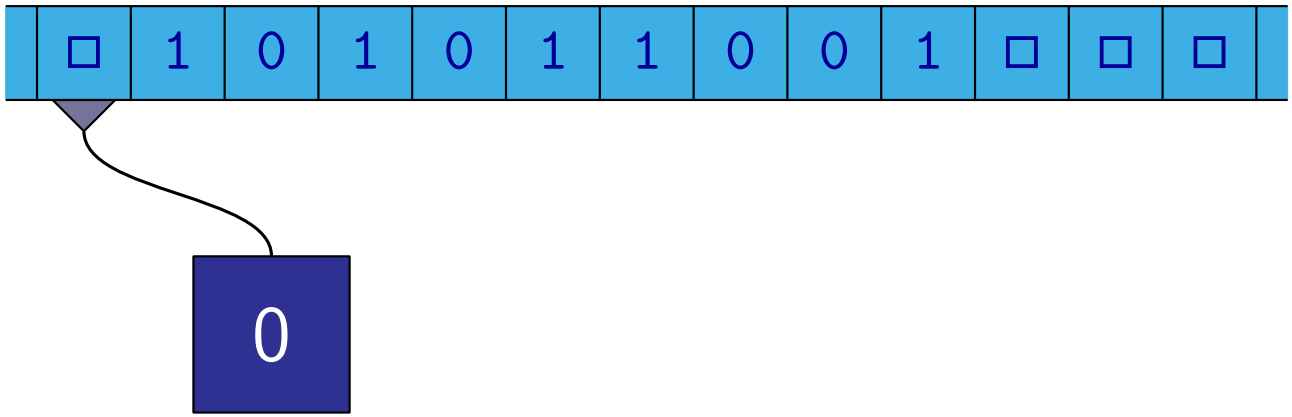
Turing Machine – Multiplication by Three



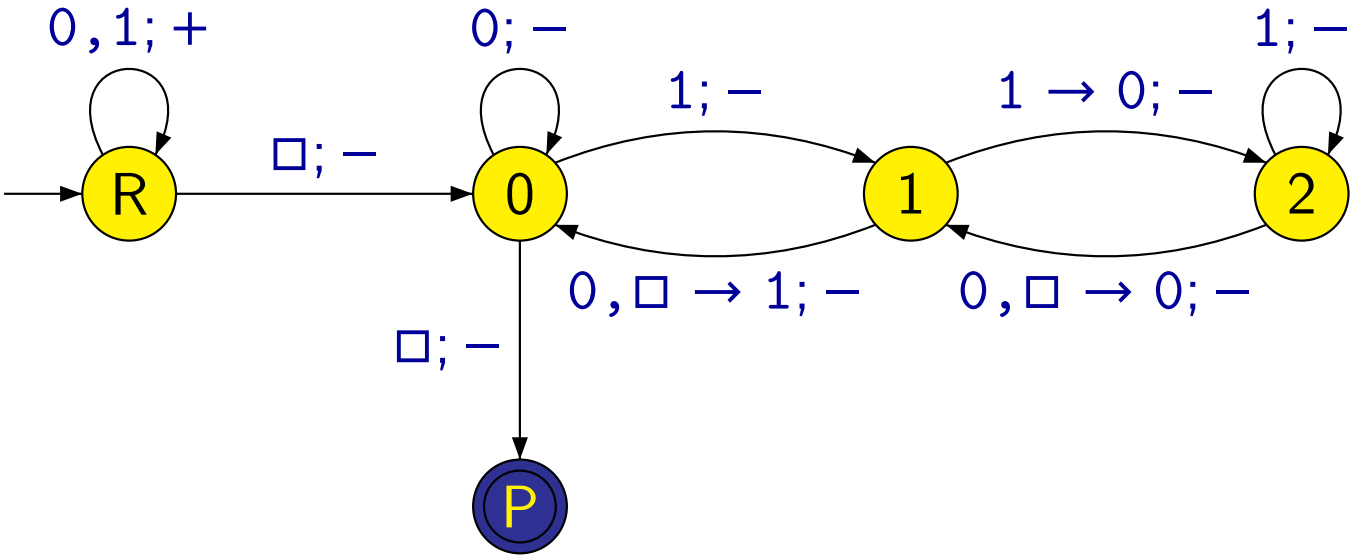


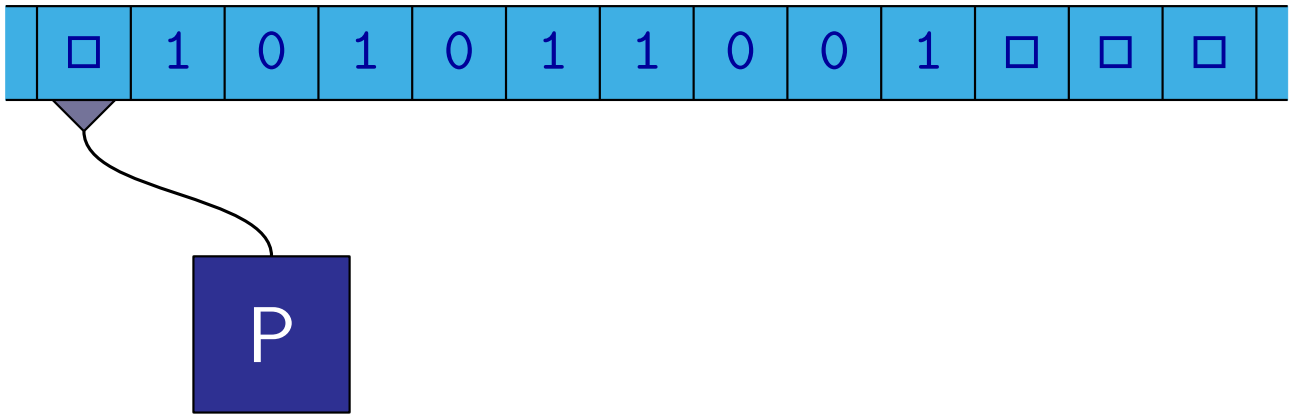
Turing Machine – Multiplication by Three





Turing Machine – Multiplication by Three





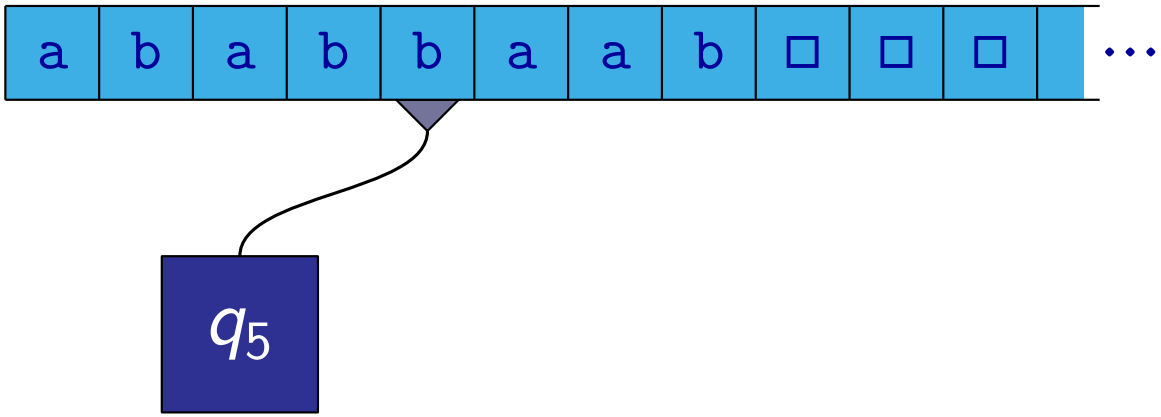
Variants of Turing Machines

- The definition of Turing machine given before is just one of many variants.
- Here we give several examples of differences between different variants of Turing machines.
- Almost all these variants of Turing machines are able to accept or recognize the same languages and to compute the same functions.
- There can be (but need not be) big differences between variants with respect to their running time and an amount of used memory.

Variants of Turing Machines

One-sided or **two-sided** infinite tape:

- In the previous definition, we have considered a tape that is infinite in both directions — to the left and to the right.
- Instead, it is sometimes considered a tape that is infinite only to the right.

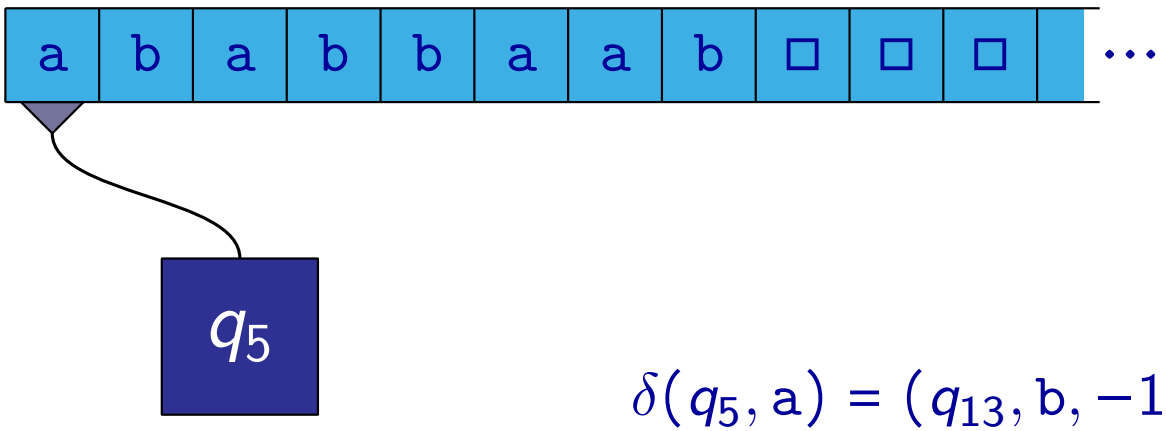


Variants of Turing Machines

It is necessary to define what should happen when the head is on the leftmost cell of the tape and, according to the transition function, it should move to the left.

Two most common possibilities:

- An "error" occurs and the computation is (unsuccesfully) ended:

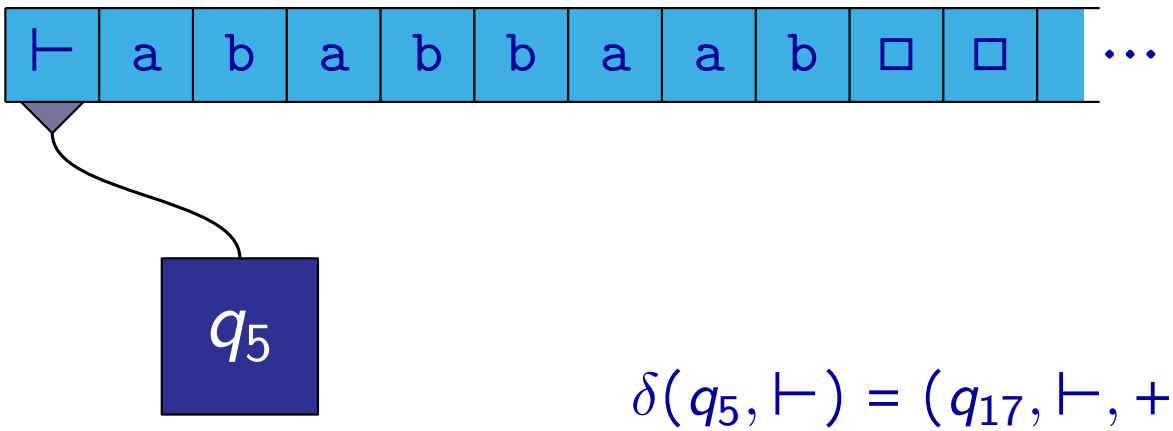


$$\delta(q_5, \mathtt{a}) = (q_{13}, \mathtt{b}, -1)$$

Variants of Turing Machines

- The left end of the tape contains a "marker" represented by a special symbol $\vdash \in (\Gamma - \Sigma)$.

  This marker can not be overwritten and a move to the left is forbidden on this symbol, i.e., for each $q \in Q$ it holds that if $\delta(q, \vdash) = (q', b, d)$ then $b = \vdash$ a $d \in \{0, +1\}$.



$$\delta(q_5, \vdash) = (q_{17}, \vdash, +1)$$

Variants of Turing Machines

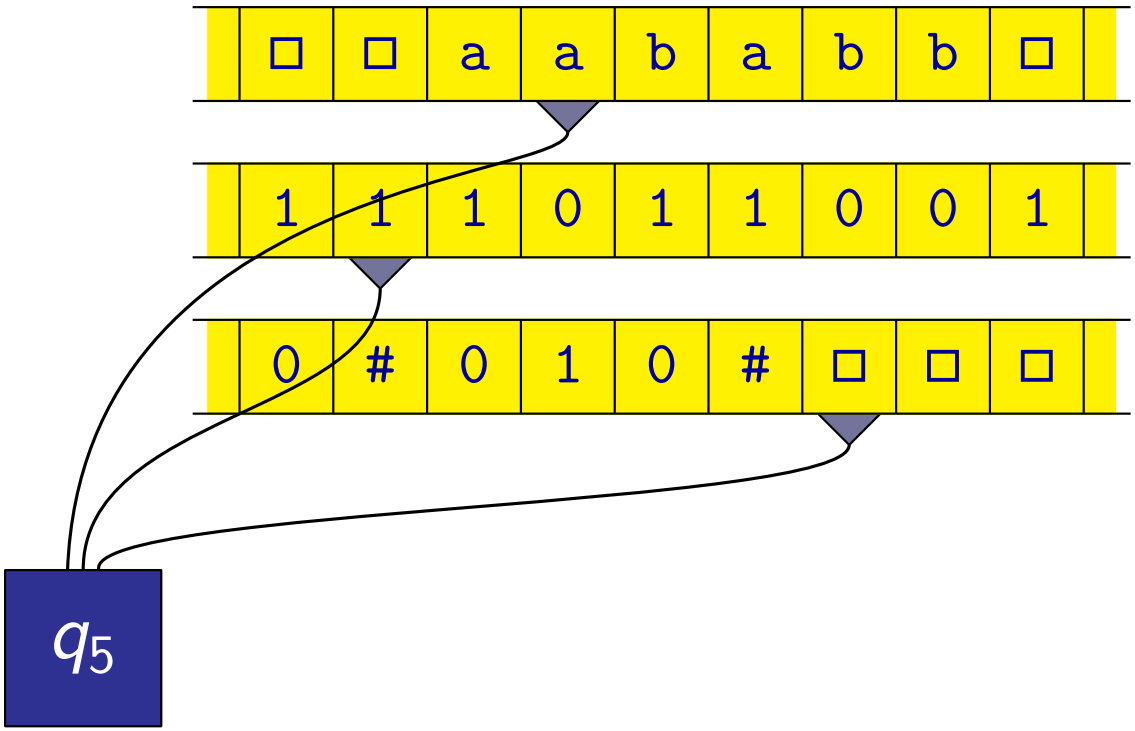
Remark: The possibility that a computation can end unsuccessfully because of an error where it is no possible to continue from the given configuration is quite common also for other types of machines we will consider.

Generally, the following possibilities can happen in a computation:

- The computation ends successfully in a final configuration that corresponds to a correct halting.
- The computation is stuck in a configuration that is not final but it is not possible to continue there — this is considered as a computation ending with an error.
- The computation never halts.

Variants of Turing Machines

Multitape Turing machines are often considered.



Variants of Turing Machines

In the case of a multitape machines:

- Each of k tapes has its own alphabet, i.e., we have tape alphabets Γ_1, Γ_2, ..., Γ_k.
- The transition function δ is of the type

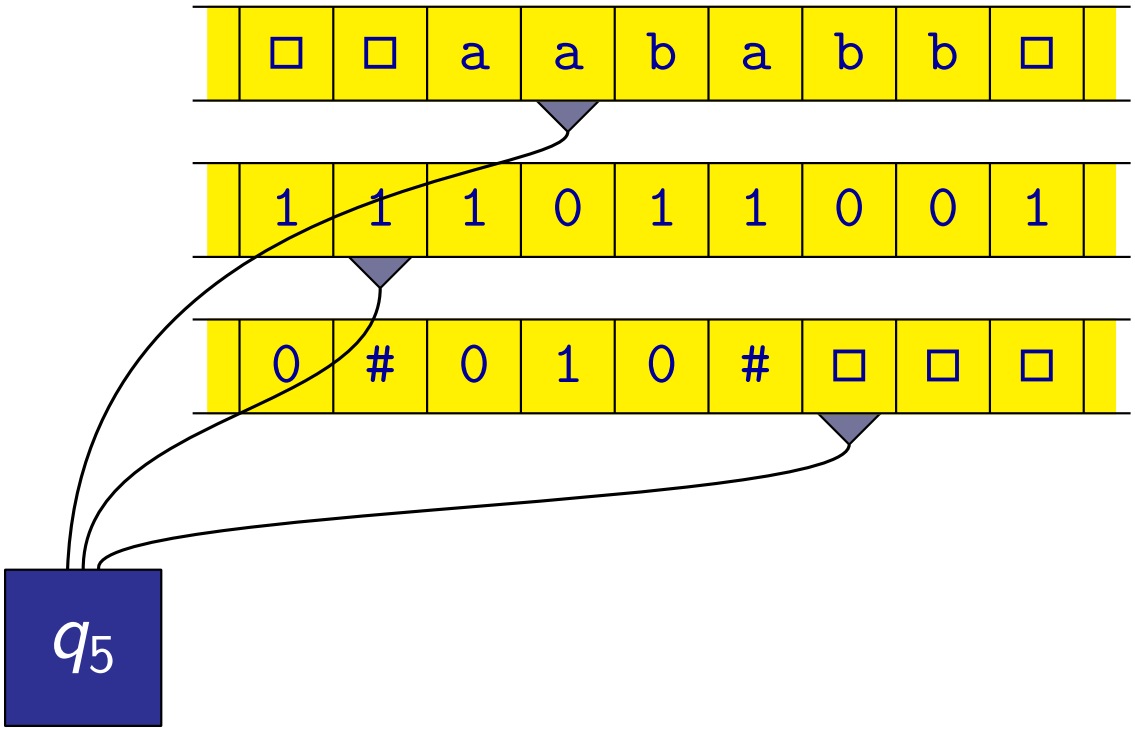
$$(Q - F) \times \Gamma_1 \times \cdots \times \Gamma_k \rightarrow Q \times \Gamma_1 \times \{-1, 0, +1\} \times \cdots \times \Gamma_k \times \{-1, 0, +1\}$$

Example:

$$\delta(q_5, \mathtt{a}, 1, \square) = (q_{12}, \mathtt{a}, -1, \mathtt{x}, 0, 1, +1)$$

Variants of Turing Machines

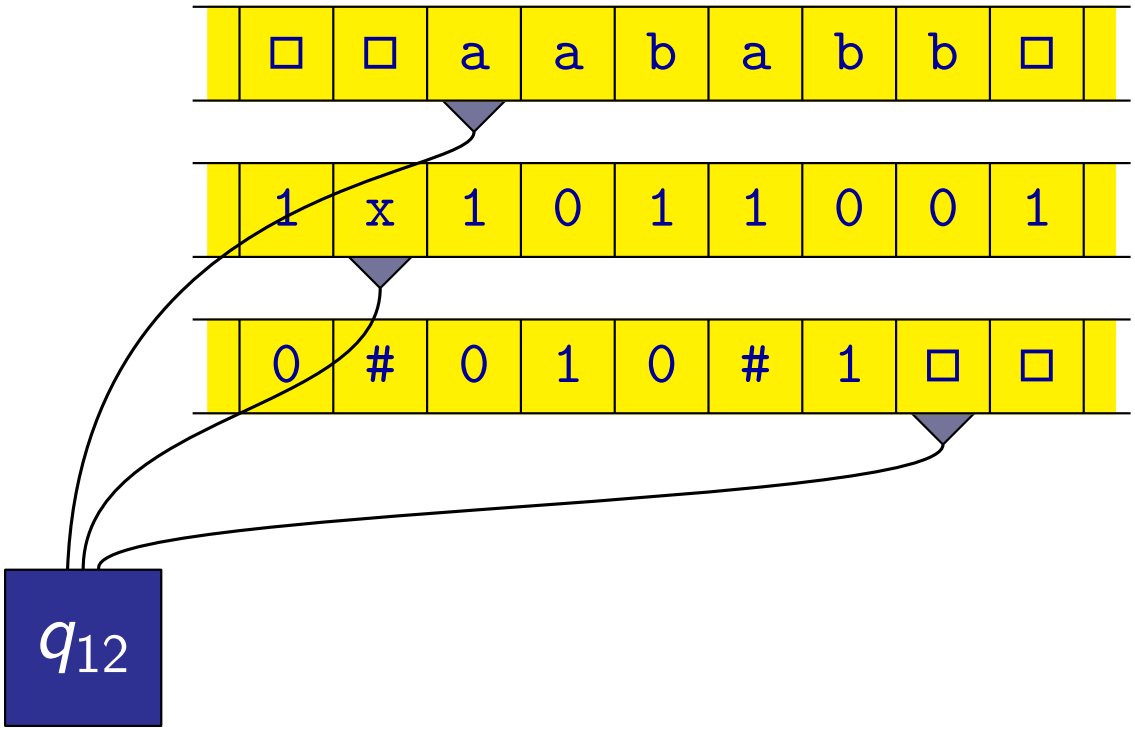
Example:



$$\delta(q_5, \mathtt{a}, 1, \square) = (q_{12}, \mathtt{a}, -1, \mathtt{x}, 0, 1, +1)$$

Variants of Turing Machines

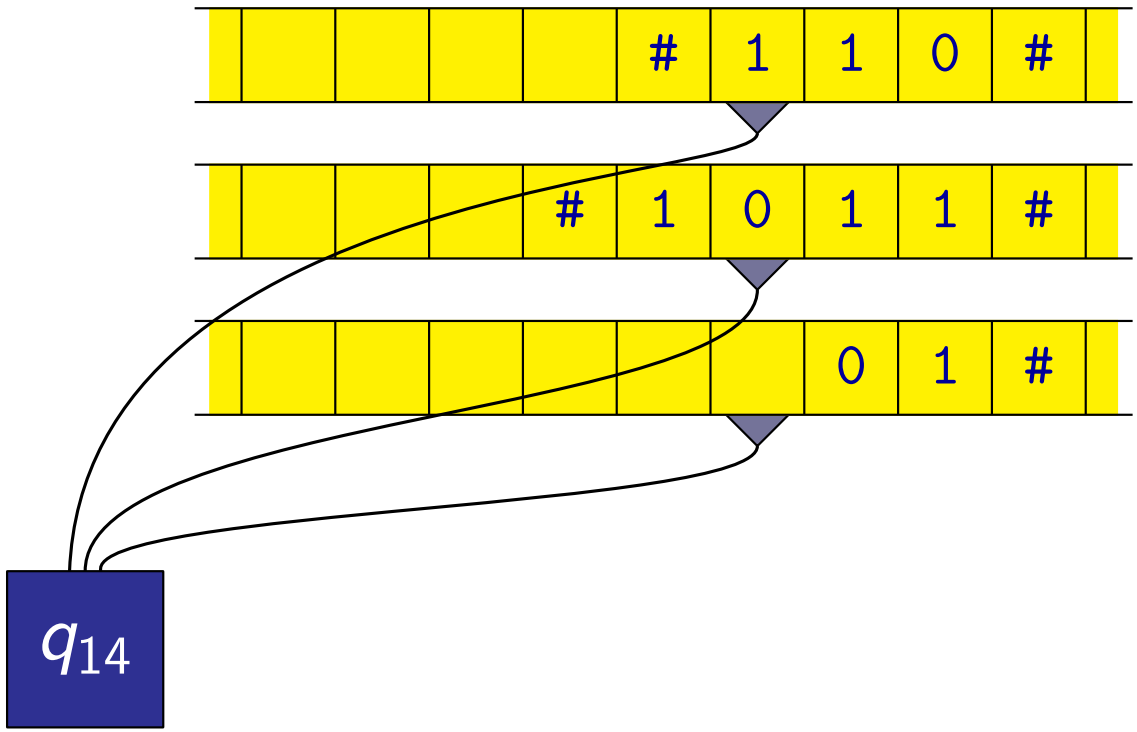
Example:



$$\delta(q_5, \mathtt{a}, 1, \square) = (q_{12}, \mathtt{a}, -1, \mathtt{x}, 0, 1, +1)$$

Variants of Turing Machines

Example: A machine that gets as an input two natural numbers written in binary and separated by symbols # (e.g., number 6 and 11 will be written as "#110#" a "#1011#").

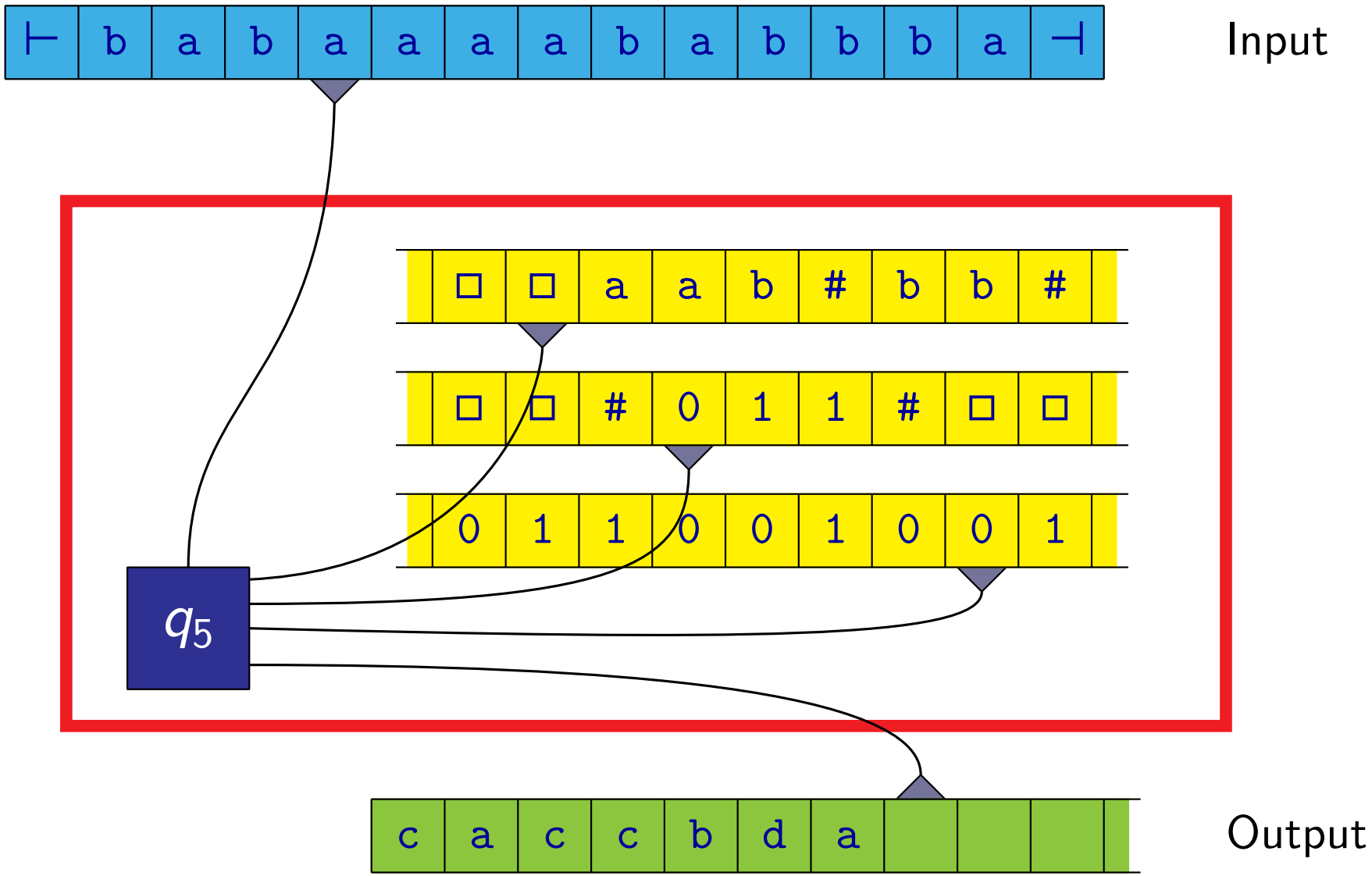


Variants of Turing Machines

Multitape machines often use one of its tapes as an input tape and one of its tapes as an output tape. Other tapes are used as working tapes:

- **Input tape** — it contains an input word, the machine can not write on it (it is read-only), it is not infinite
- **Working tapes** — the machine can read from them and write on them (they are read/write), at the beginning of a computation they are empty (they contain only symbols $\square$)
- **Output tape** — the machine can only write on it (it is write-only), it can not read from it, it is empty at the beginning of a computation, the head can move only from the left to the right

Variants of Turing Machines



Variants of Turing Machines

If a machine has a special separate input tape (which is read-only), the following two variants are typically used:

- The head on this tape can move to the left and to the right.

  In this case, an input word $w \in \Sigma^*$ is bounded from the left and from the right using "endmarkers", i.e., special symbols $\vdash, \dashv \in (\Gamma - \Sigma)$.

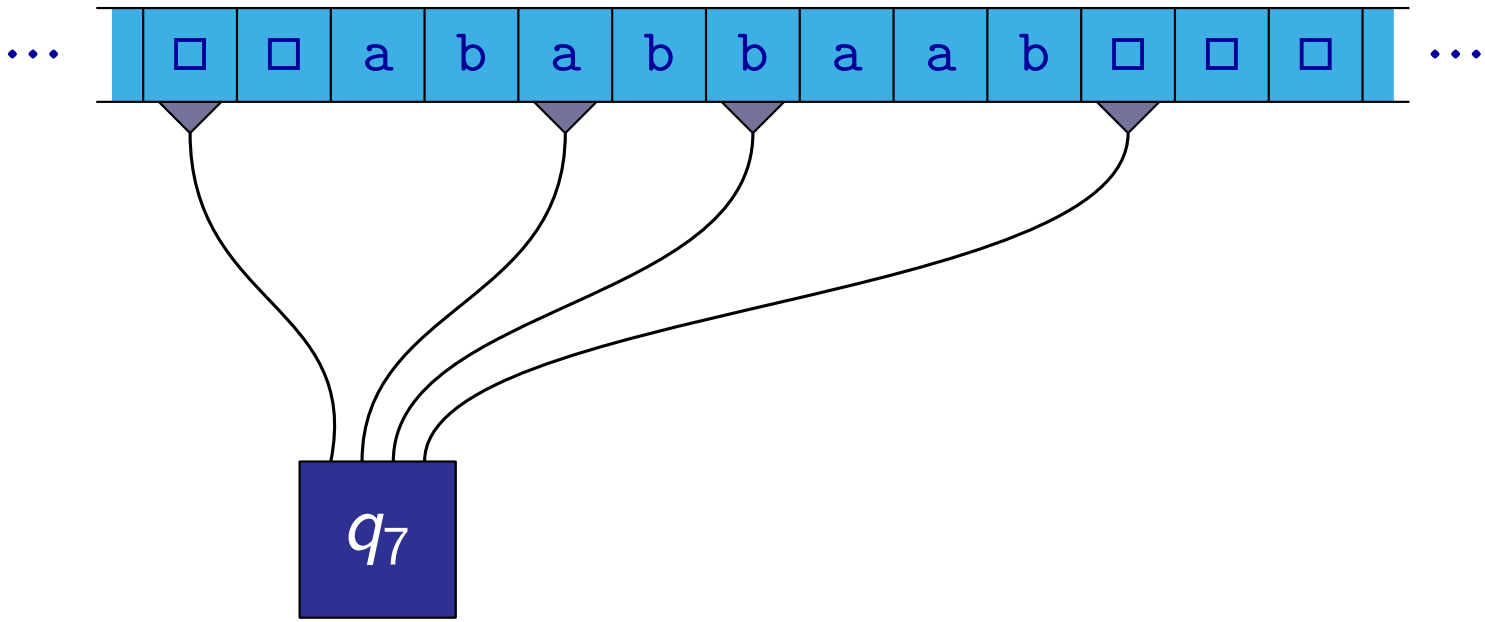
- The head can move only from left to right.

Remark: The variant with possible movement in both directions and endmarkers is more common.

If it is not specified otherwise, we will consider this variant.

Variants of Turing Machines

Instead of several tapes, we can consider **several heads** on one tape:



Variants of Turing Machines

In the variant with several heads on one tape, it is necessary to specify:

- If there can be more than one head in the same time on one tape cell.
- If this is the case, what is the behaviour of the machine if several heads occurring on the same cell want to write different symbols on this cell.
- Whether the given machine can detect the situation when several head are on the same cell.

Remark: Of course, in general we can consider machines with several tapes where each of these tapes is equipped with several heads.

Variants of Turing Machines

Consider a machine with several tapes and with arbitrary number of heads on each tape.

Instead of describing a transition function that works with all heads in each step, we can alternatively describe the behaviour of the machine by a **program** consisting of simpler instructions of the following types:

- to move a given head by one cell to the left
- to move a given head by one cell to the right
- to write a specified symbol on the given position of a specified head on a tape
- to read one symbol from a position of a given head and to branch the program according this symbol (i.e., to go to different states of the control unit)

Variants of Turing Machines

So far we considered only **linear** (one-dimensitional) tapes.

Instead, the memory with cells (where every cell contains one symbol from some alphabet) can have some other structure.

For example:

- two-dimensional **square grid**
  — a movement of a head into four directions: left, right, up, down
- d-dimensional memory for some $d = 3, 4, \ldots$
  (three-dimensional, four-dimensional, etc.)
- a memory organized in a form of an (infinite) tree
- ...

Simulations between different kinds of machines

Simulation of a Computation

Explanation what it means that a machine $\mathcal{M}$ is **simulated** by a machine $\mathcal{M}'$:

- A computation of machine $\mathcal{M}$ for input w is a (finite or infinite) sequence of configurations of machine $\mathcal{M}$

$$\alpha_0 \longrightarrow \alpha_1 \longrightarrow \alpha_2 \longrightarrow \cdots$$

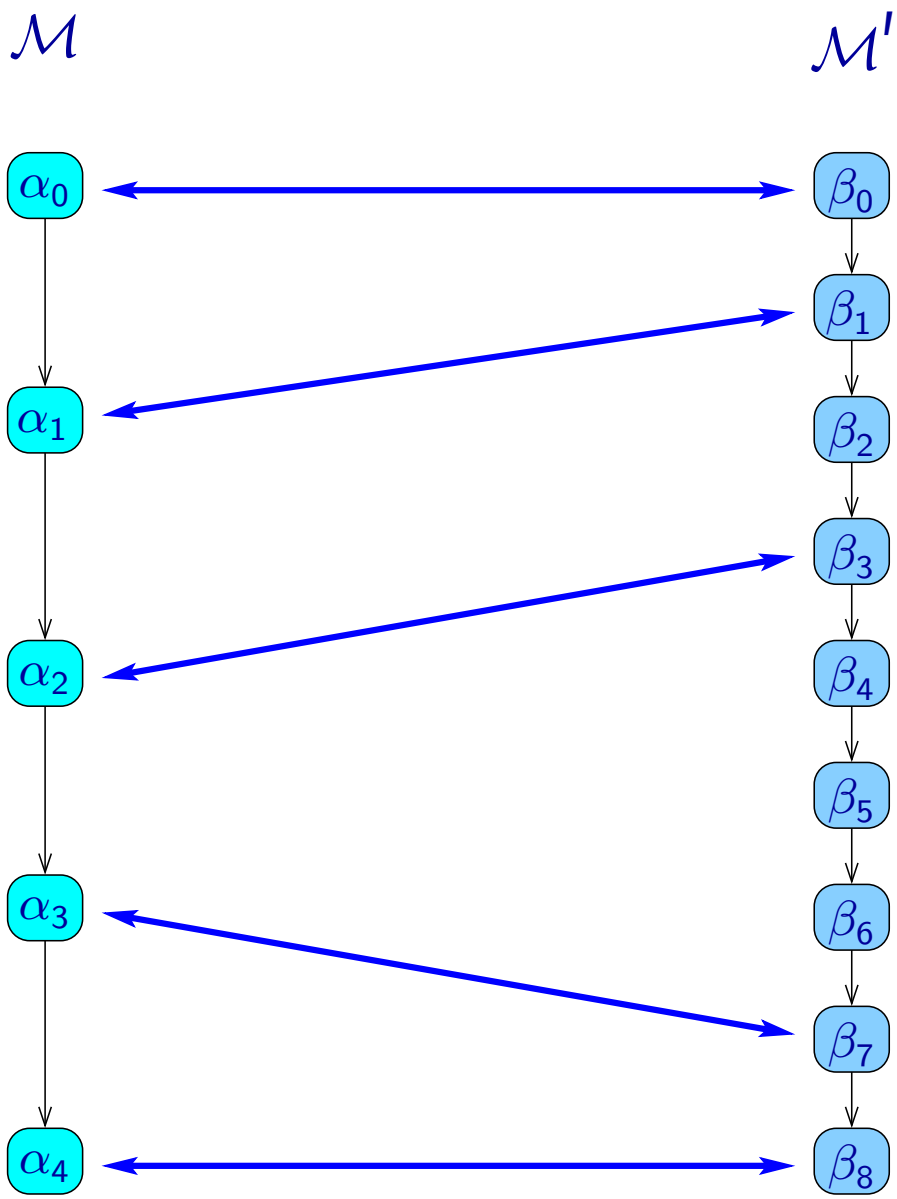
- For this computation, there is a corresponding computation of machine $\mathcal{M}'$ consisting of configurations

$$\beta_0 \longrightarrow \beta_1 \longrightarrow \beta_2 \longrightarrow \cdots$$

  where for every configuration α_i there is some corresponding configuration $\beta_{f(i)}$ where $f : \mathbb{N} \to \mathbb{N}$ is a function, for which $f(i) \leq f(j)$ for every i and j where $i < j$.

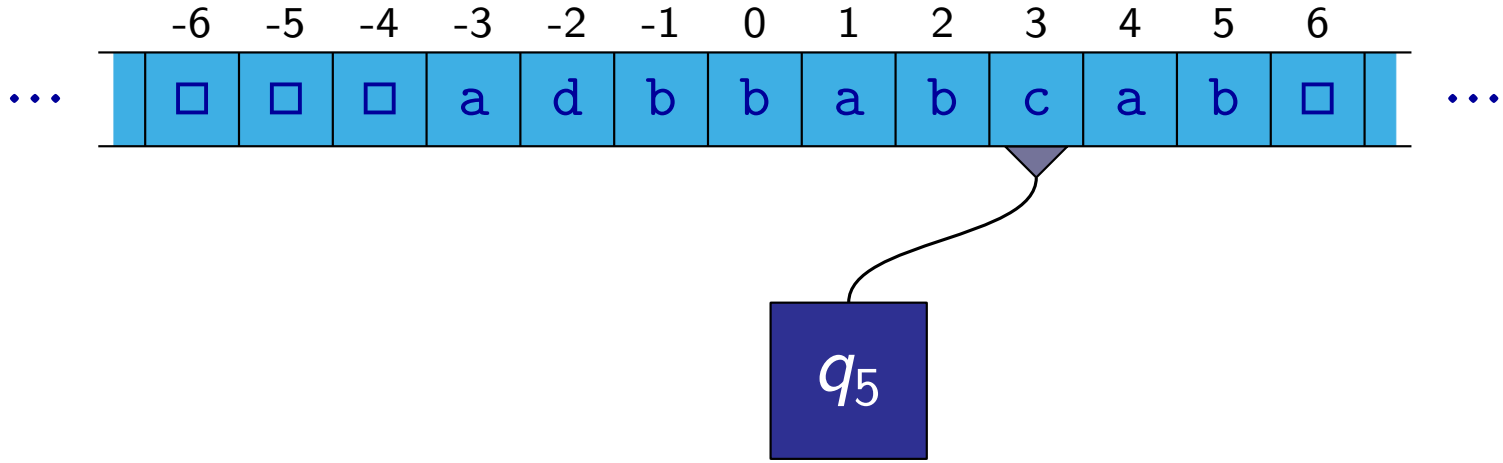
- There are functions mapping an input w to corresponding initial configurations α_0 and β_0 and analogously functions mapping final configurations to a result of computation.

Simulation of a Computation

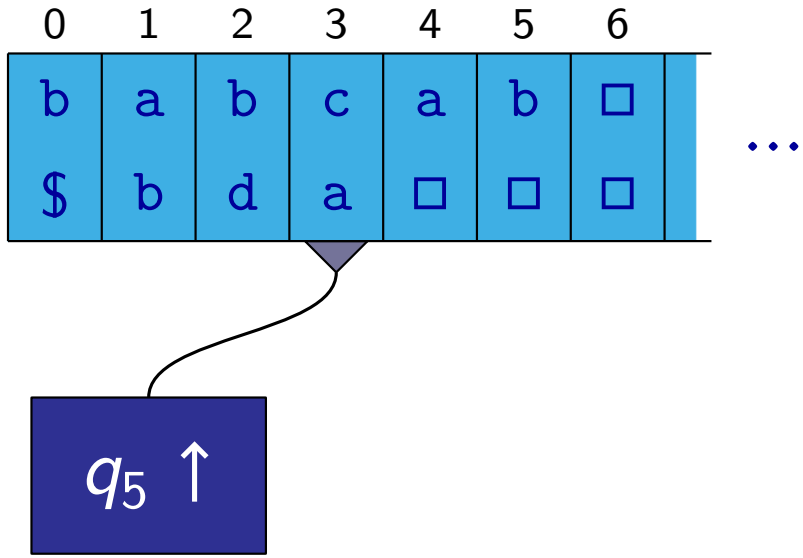


Two-sided Infinite Tape by One-sided Infinite

A tape infinite in both directions:

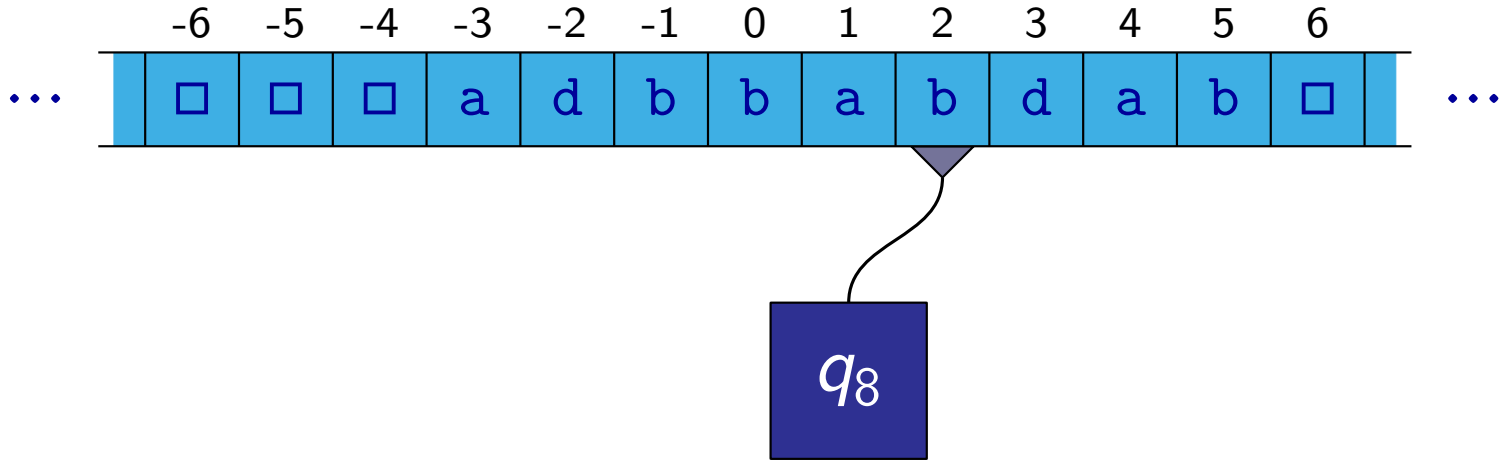


One-sided infinite tape:

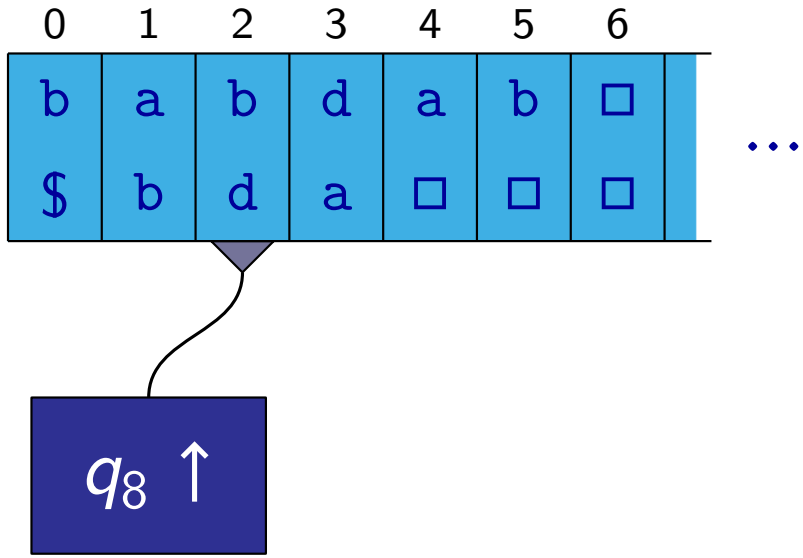


Two-sided Infinite Tape by One-sided Infinite

A tape infinite in both directions:

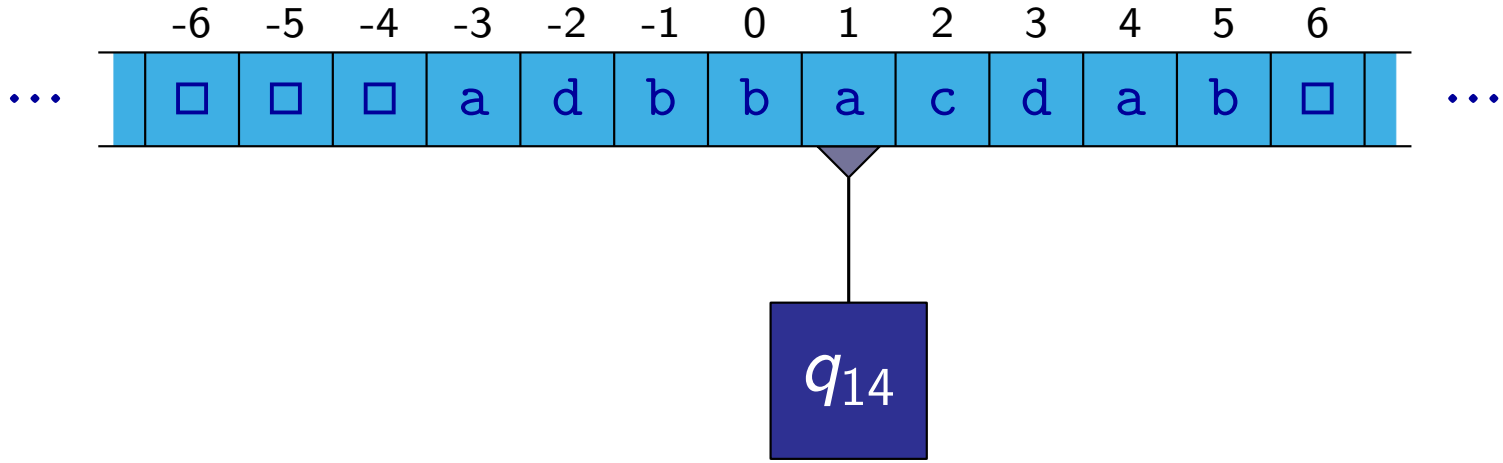


One-sided infinite tape:

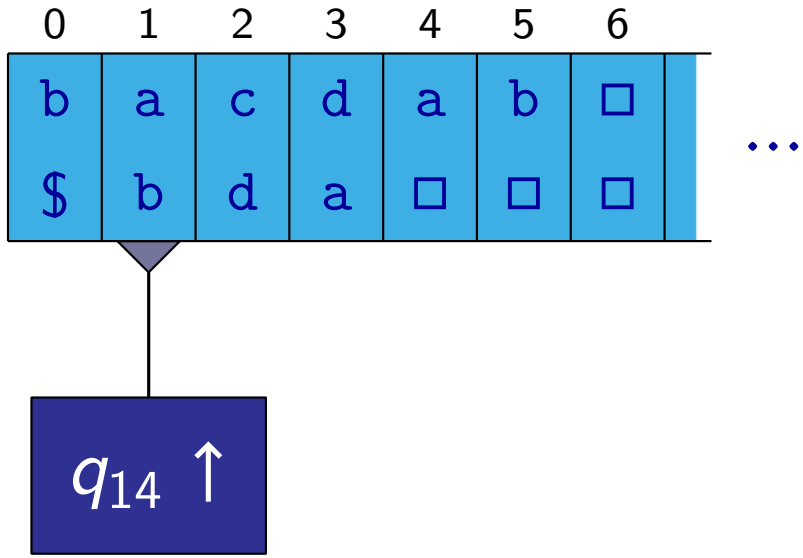


Two-sided Infinite Tape by One-sided Infinite

A tape infinite in both directions:

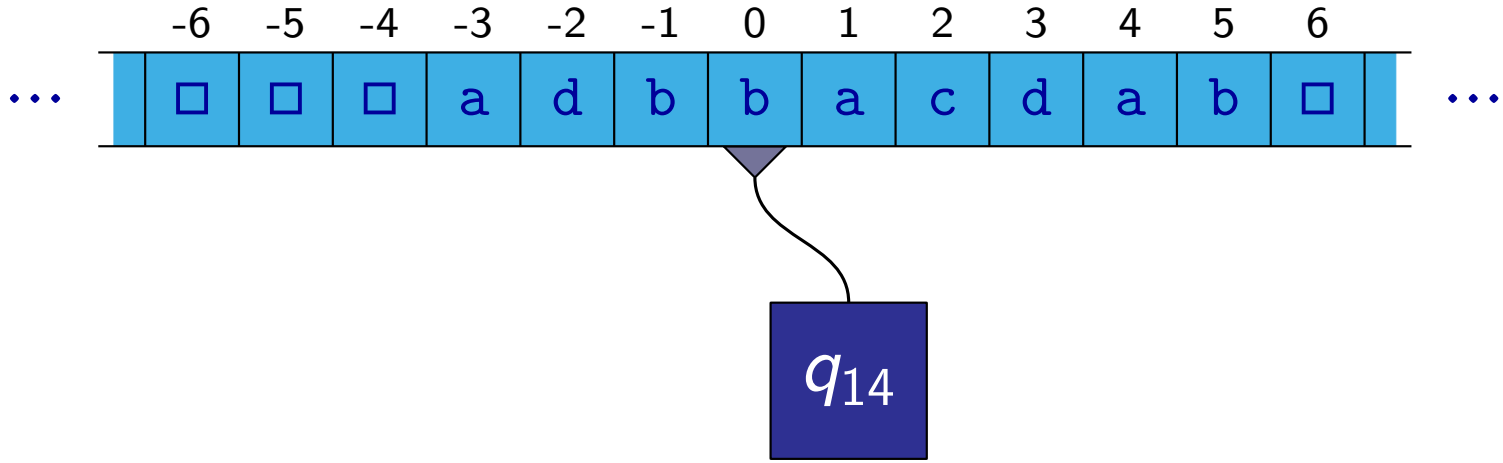


One-sided infinite tape:

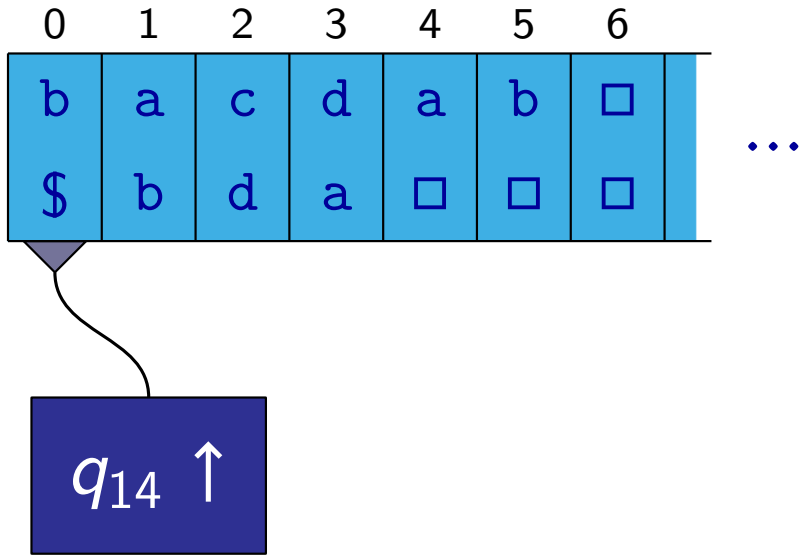


Two-sided Infinite Tape by One-sided Infinite

A tape infinite in both directions:

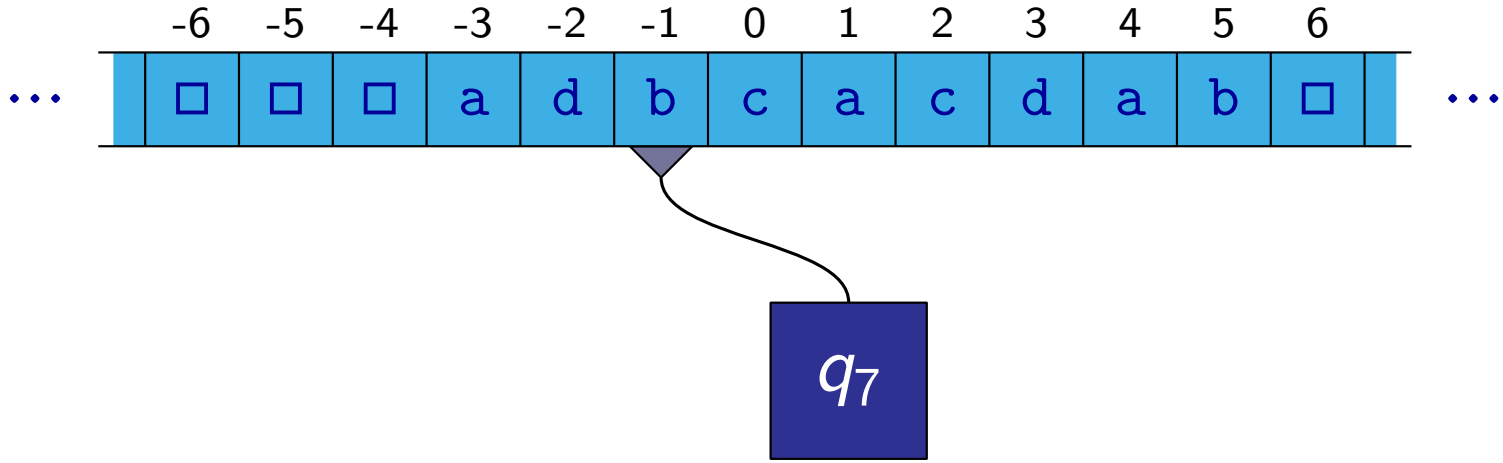


One-sided infinite tape:

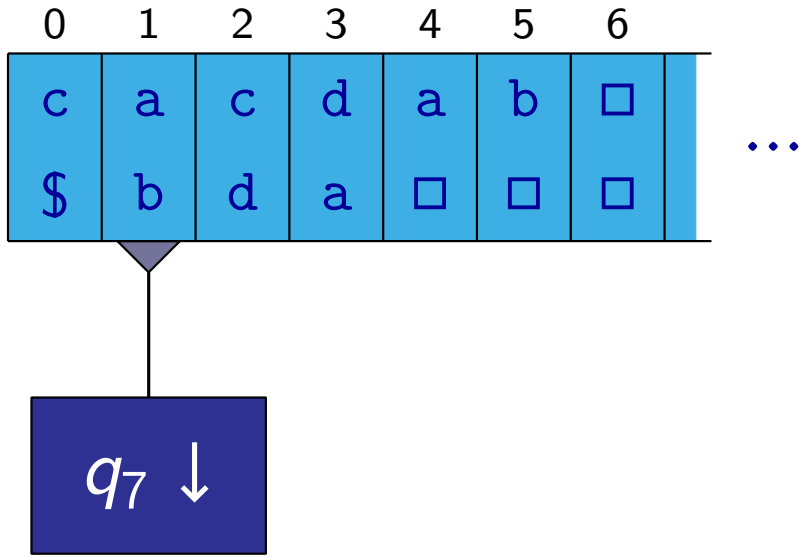


Two-sided Infinite Tape by One-sided Infinite

A tape infinite in both directions:

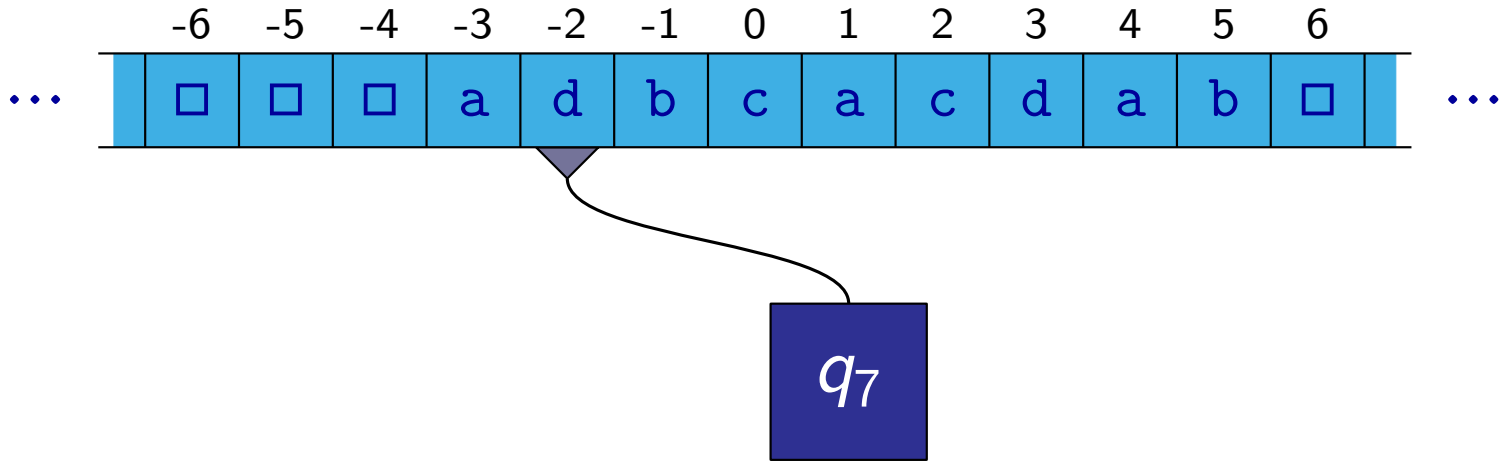


One-sided infinite tape:

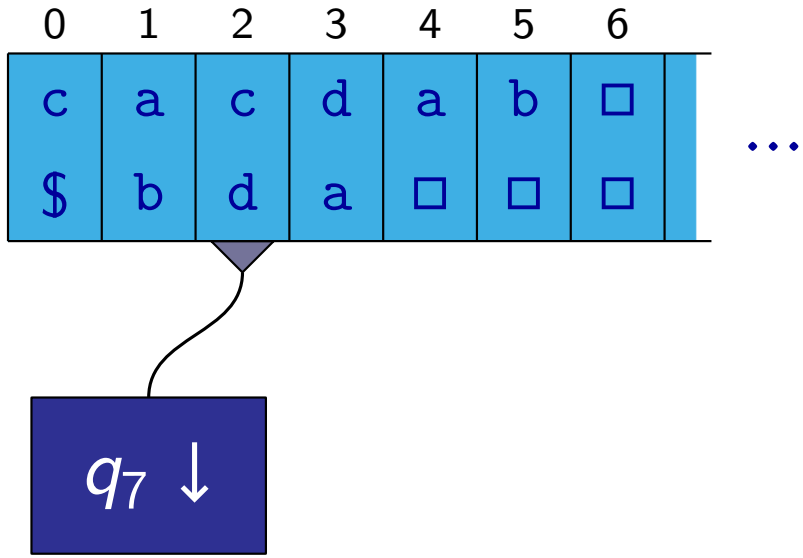


Two-sided Infinite Tape by One-sided Infinite

A tape infinite in both directions:

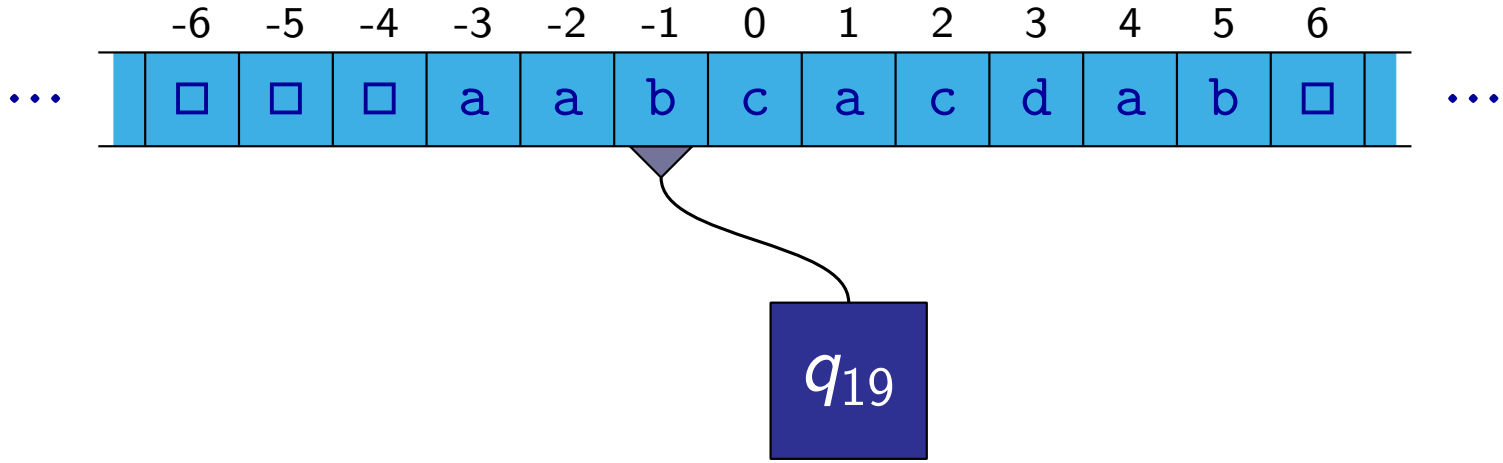


One-sided infinite tape:

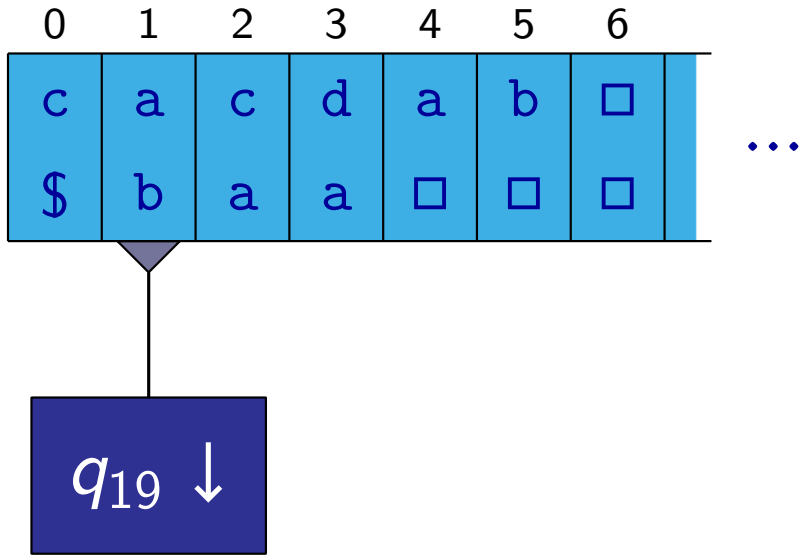


Two-sided Infinite Tape by One-sided Infinite

A tape infinite in both directions:

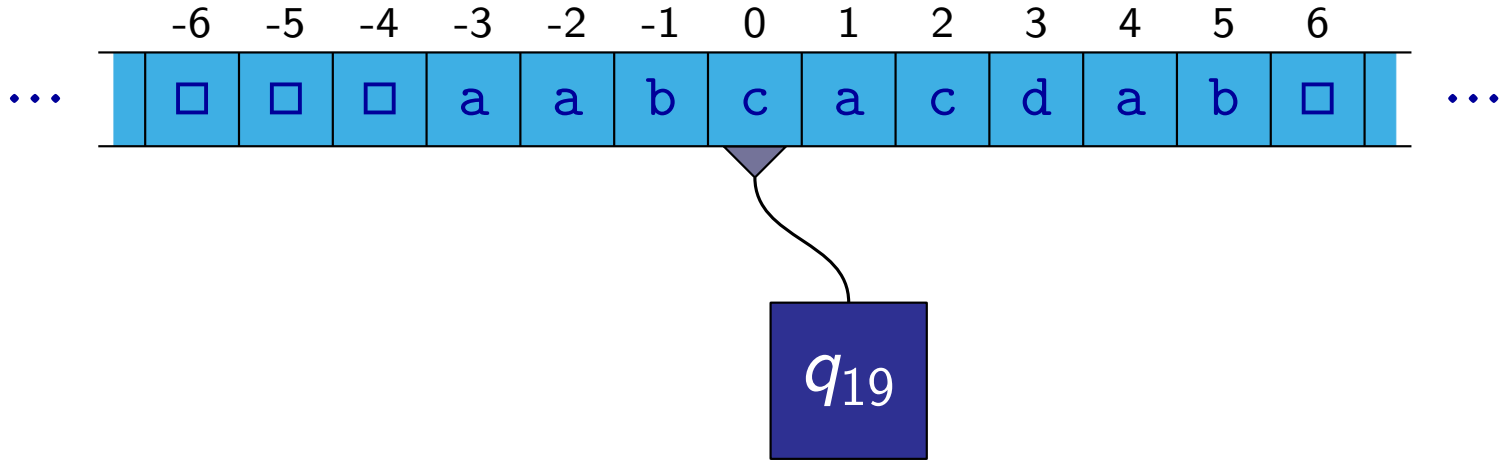


One-sided infinite tape:

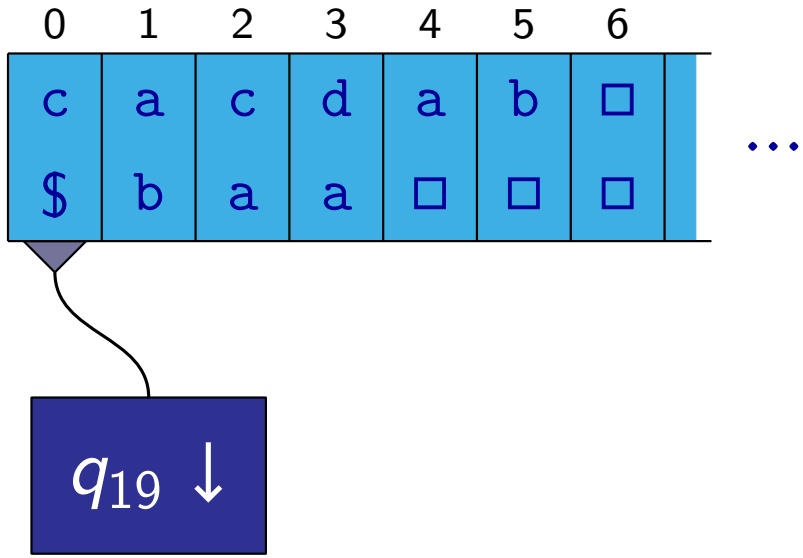


Two-sided Infinite Tape by One-sided Infinite

A tape infinite in both directions:

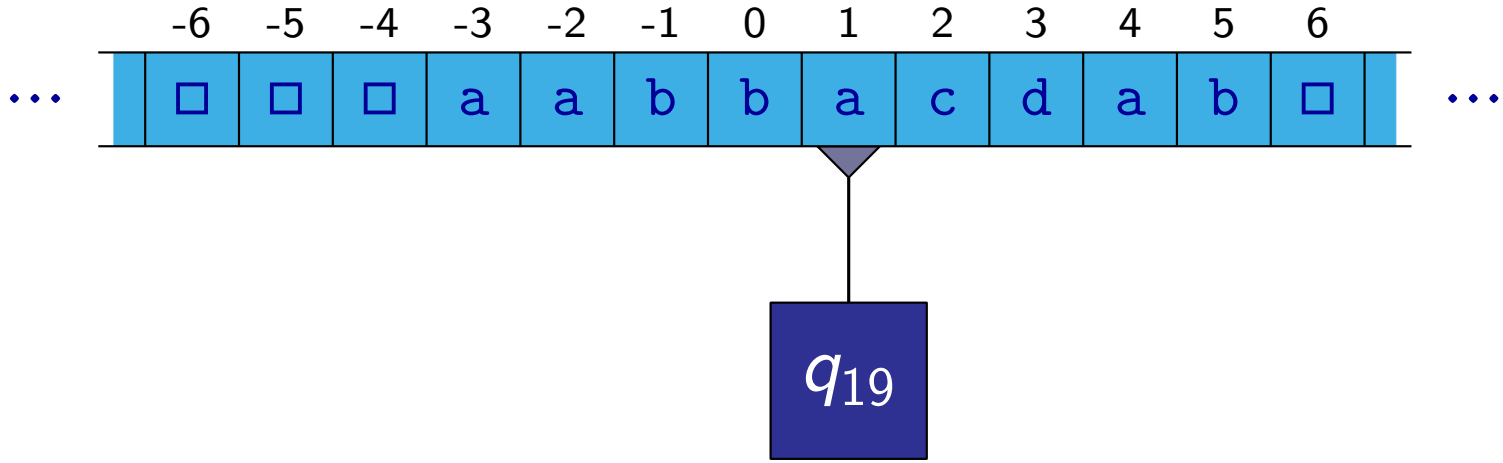


One-sided infinite tape:

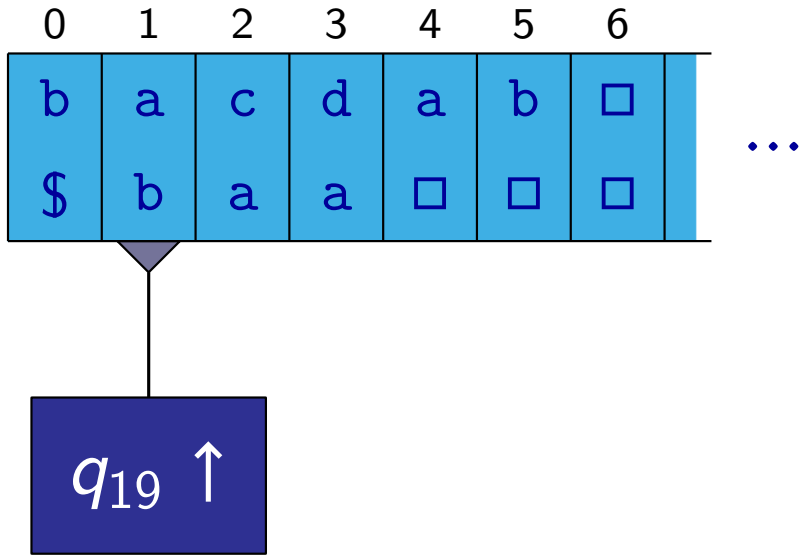


Two-sided Infinite Tape by One-sided Infinite

A tape infinite in both directions:

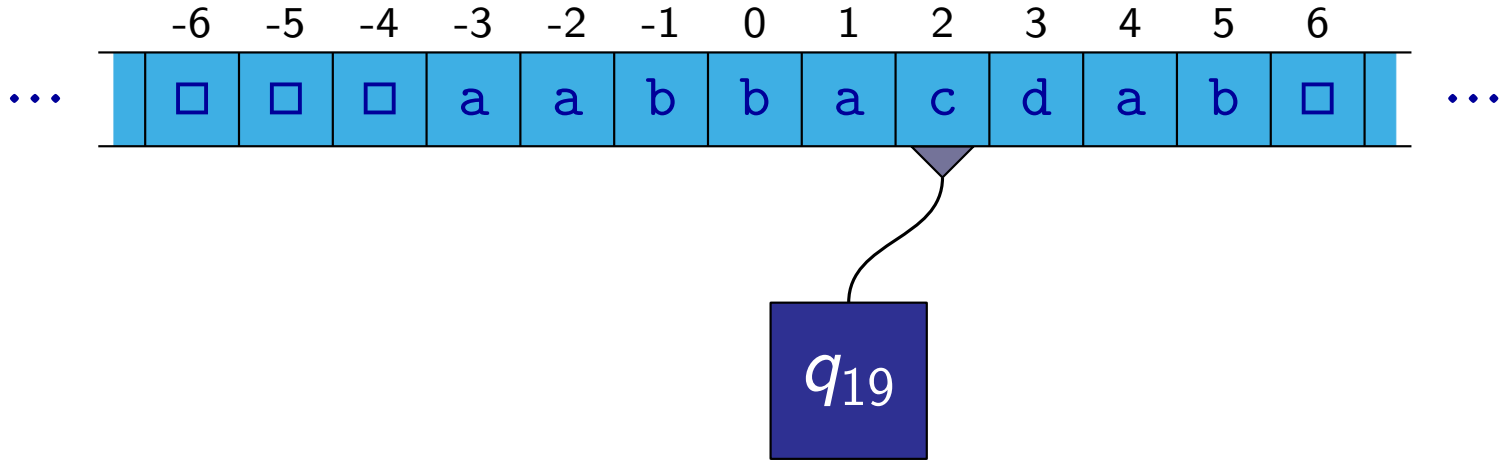


One-sided infinite tape:

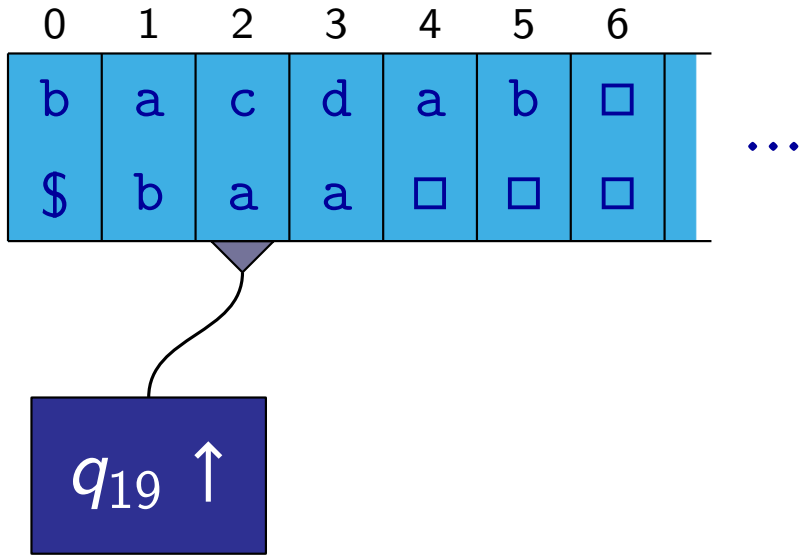


Two-sided Infinite Tape by One-sided Infinite

A tape infinite in both directions:



One-sided infinite tape:



Alphabet $\{0, 1\}$

A behaviour of a machine with an arbitrary tape alphabet Γ can be simulated by a machine with the tape alphabet $\{0, 1\}$.

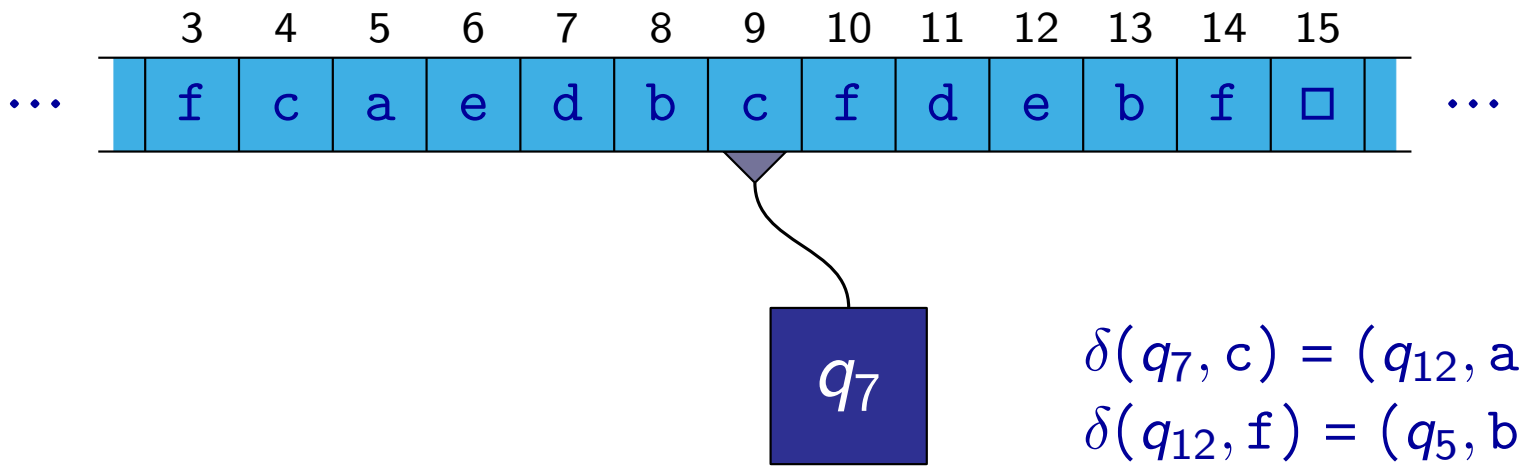
We just need to choose some suitable encoding of symbols Γ as k-bit sequences.

Example: Tape alphabet $\Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}, \mathtt{g}\}$

$$
\begin{array}{ccc}
\square & \leftrightarrow & 000 \\
\mathtt{a} & \leftrightarrow & 001 \\
\mathtt{b} & \leftrightarrow & 010 \\
\mathtt{c} & \leftrightarrow & 011 \\
\mathtt{d} & \leftrightarrow & 100 \\
\mathtt{e} & \leftrightarrow & 101 \\
\mathtt{f} & \leftrightarrow & 110 \\
\mathtt{g} & \leftrightarrow & 111
\end{array}
$$

Alphabet $\{0, 1\}$

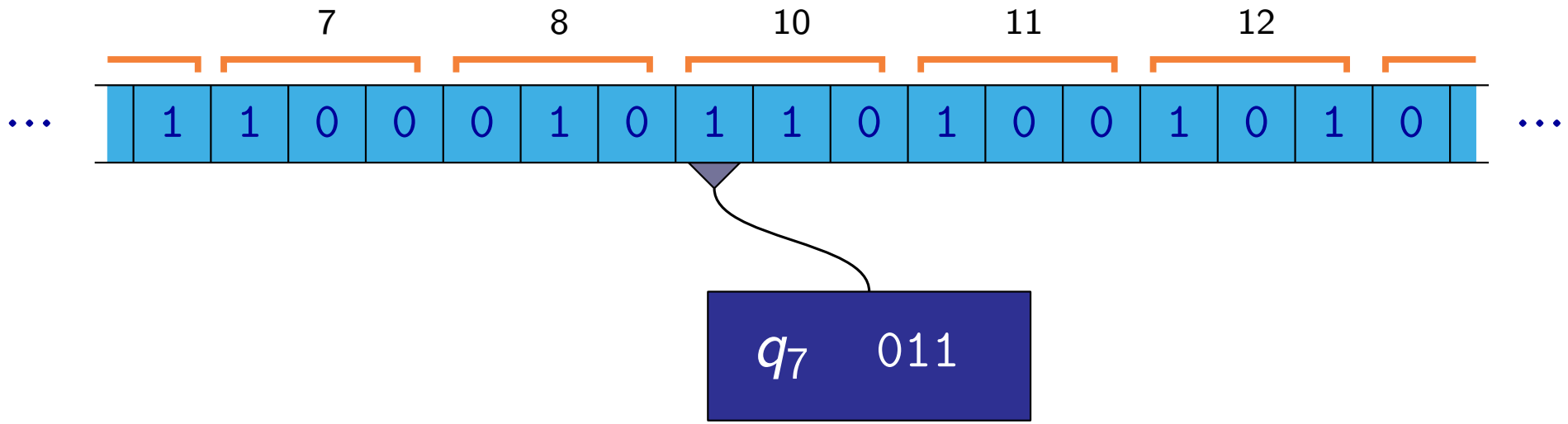
A machine with tape alphabet Γ:



$\delta(q_7, \mathtt{c}) = (q_{12}, \mathtt{a}, +1)$

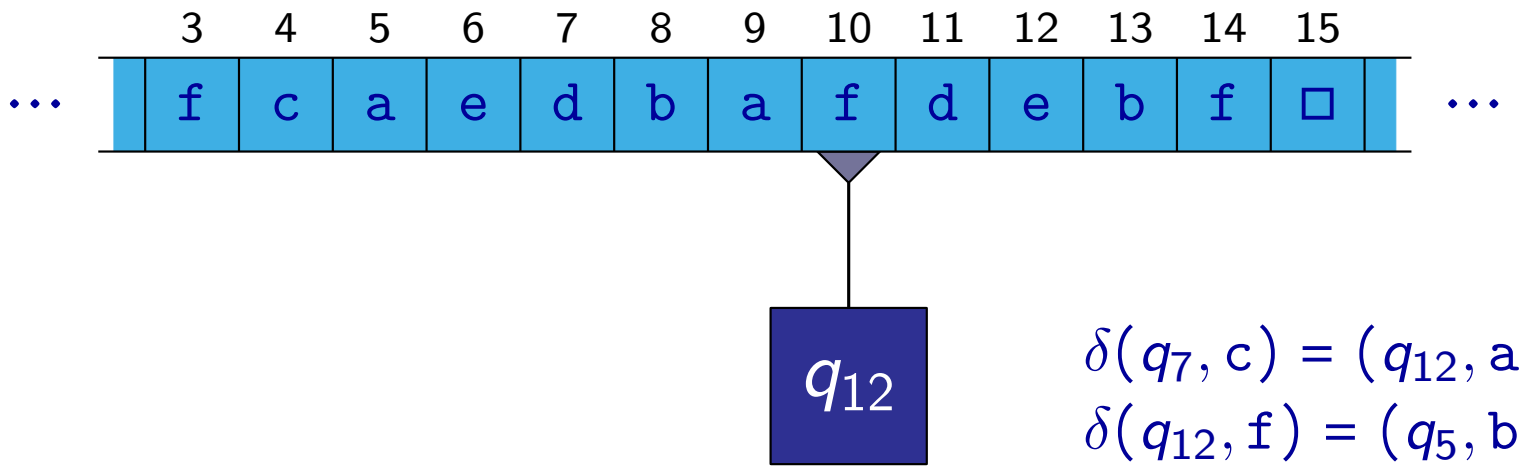
$\delta(q_{12}, \mathtt{f}) = (q_5, \mathtt{b}, -1)$

A machine with alphabet $\{0, 1\}$:



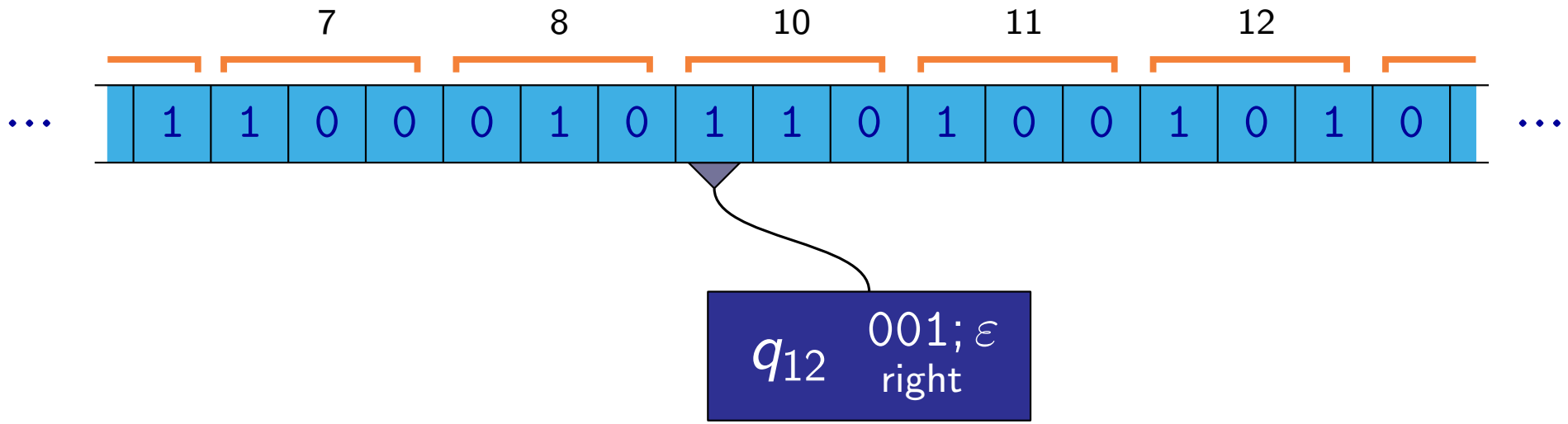
Alphabet $\{0, 1\}$

A machine with tape alphabet Γ:



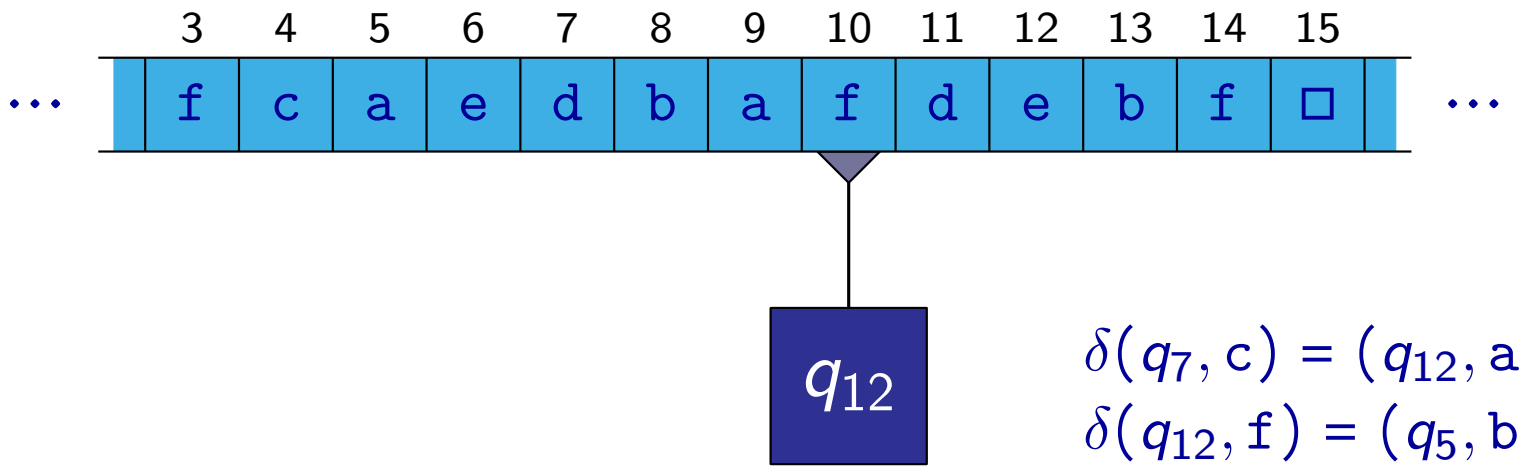
$$\delta(q_7, \mathtt{c}) = (q_{12}, \mathtt{a}, +1)$$
$$\delta(q_{12}, \mathtt{f}) = (q_5, \mathtt{b}, -1)$$

A machine with alphabet $\{0, 1\}$:



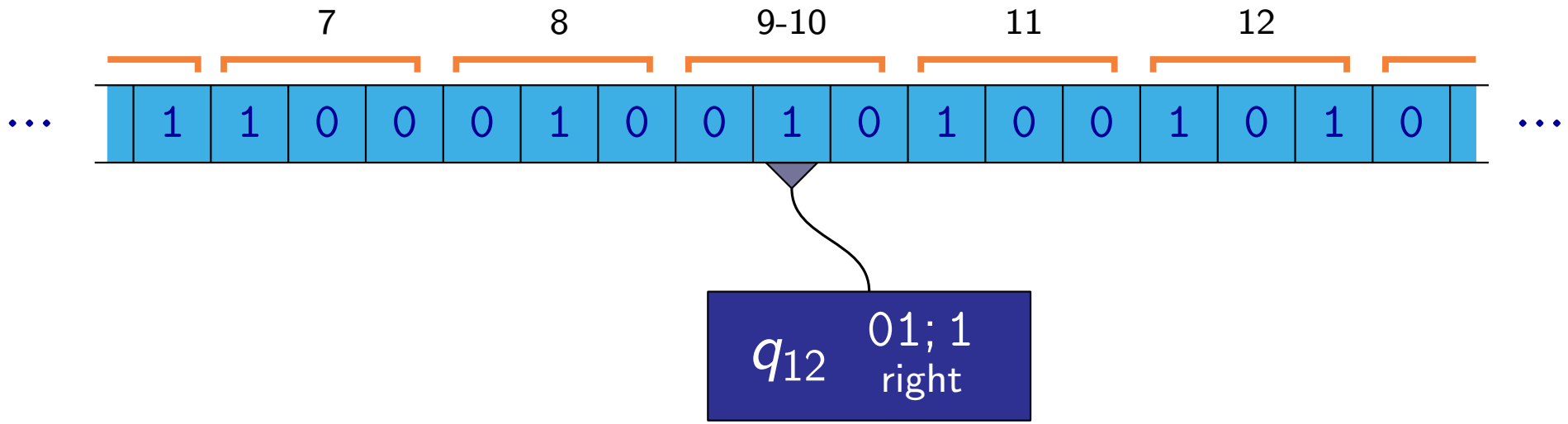
Alphabet $\{0, 1\}$

A machine with tape alphabet Γ:



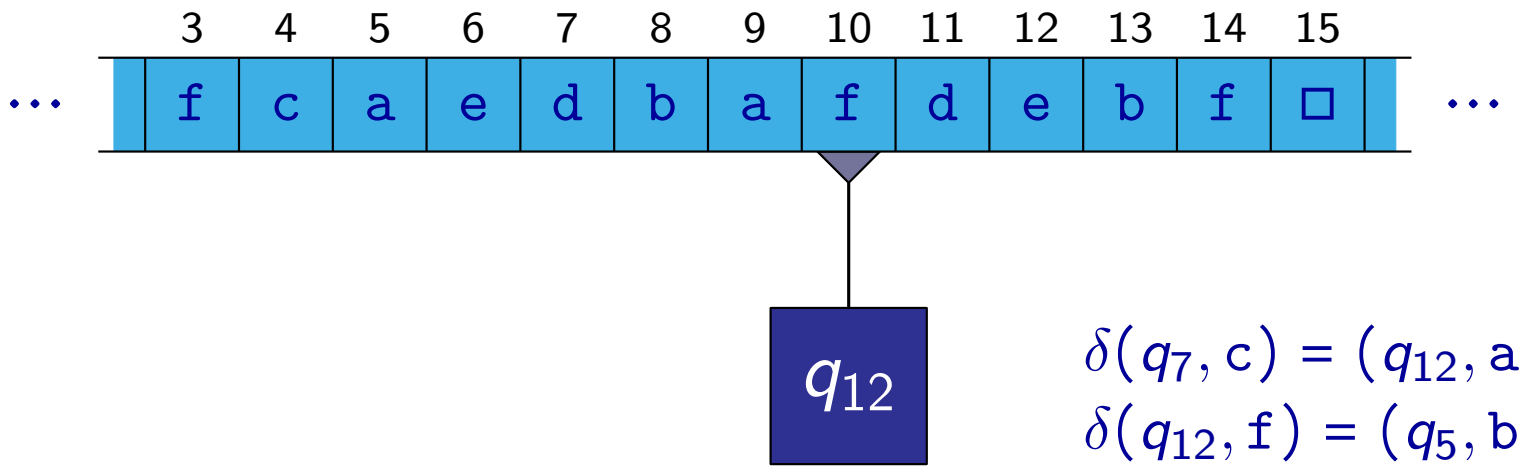
$$\delta(q_7, \mathtt{c}) = (q_{12}, \mathtt{a}, +1)$$
$$\delta(q_{12}, \mathtt{f}) = (q_5, \mathtt{b}, -1)$$

A machine with alphabet $\{0, 1\}$:



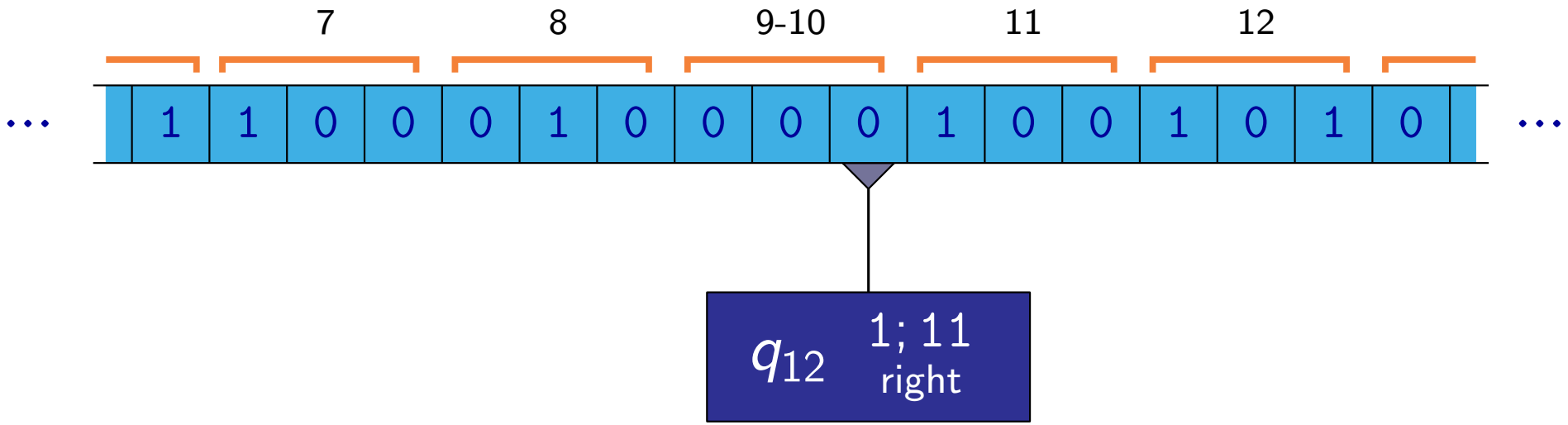
Alphabet $\{0, 1\}$

A machine with tape alphabet Γ:



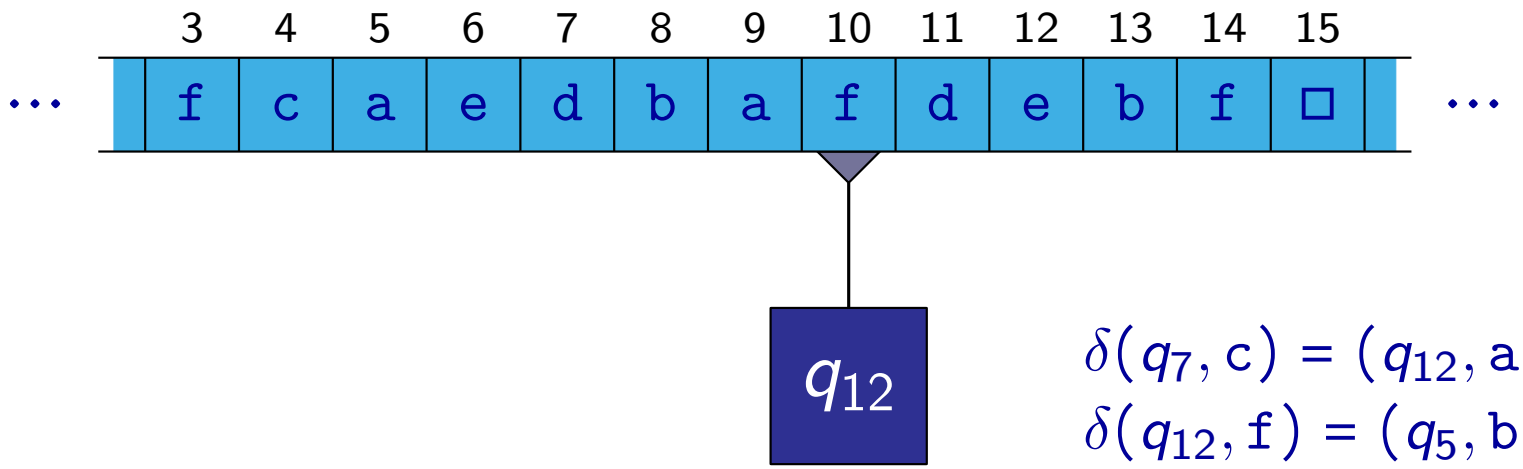
$$\delta(q_7, \mathtt{c}) = (q_{12}, \mathtt{a}, +1)$$
$$\delta(q_{12}, \mathtt{f}) = (q_5, \mathtt{b}, -1)$$

A machine with alphabet $\{0, 1\}$:



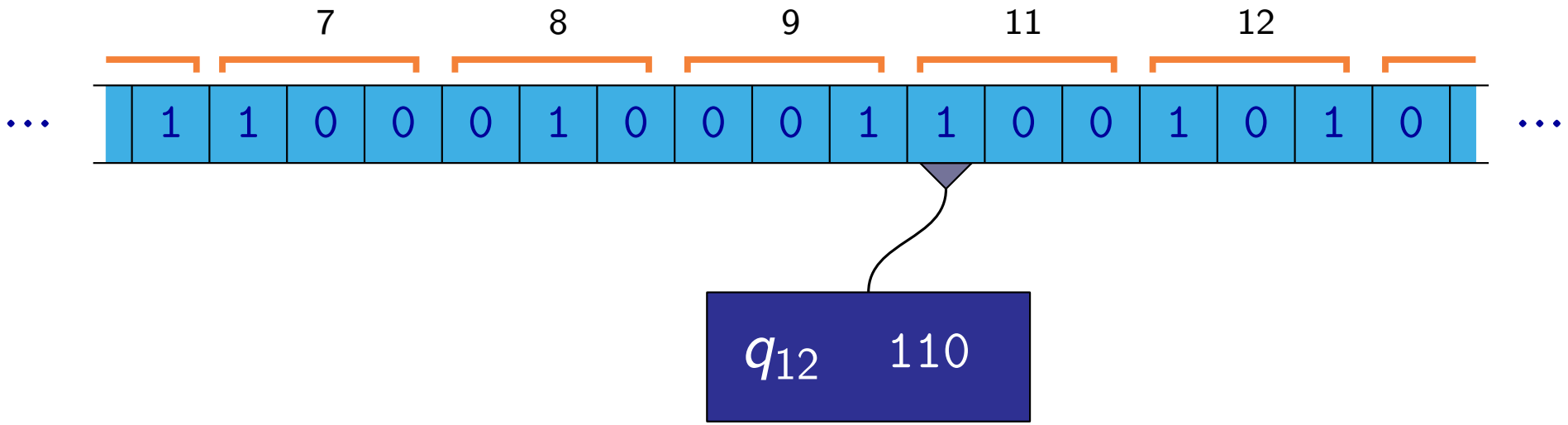
Alphabet $\{0, 1\}$

A machine with tape alphabet Γ:



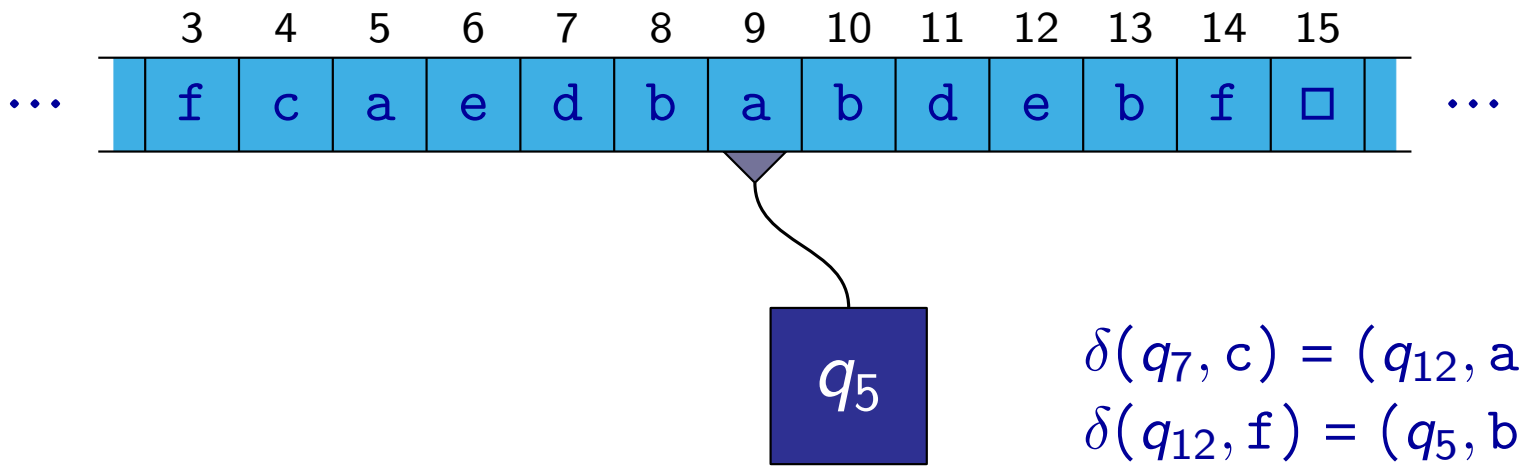
$$\delta(q_7, \mathtt{c}) = (q_{12}, \mathtt{a}, +1)$$
$$\delta(q_{12}, \mathtt{f}) = (q_5, \mathtt{b}, -1)$$

A machine with alphabet $\{0, 1\}$:



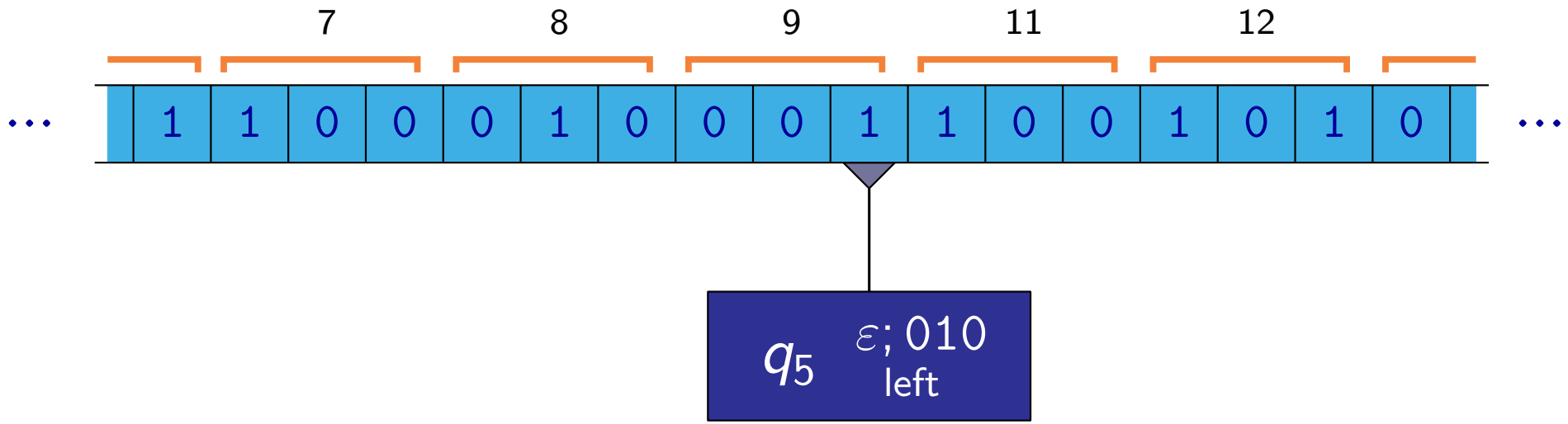
Alphabet $\{0, 1\}$

A machine with tape alphabet Γ:



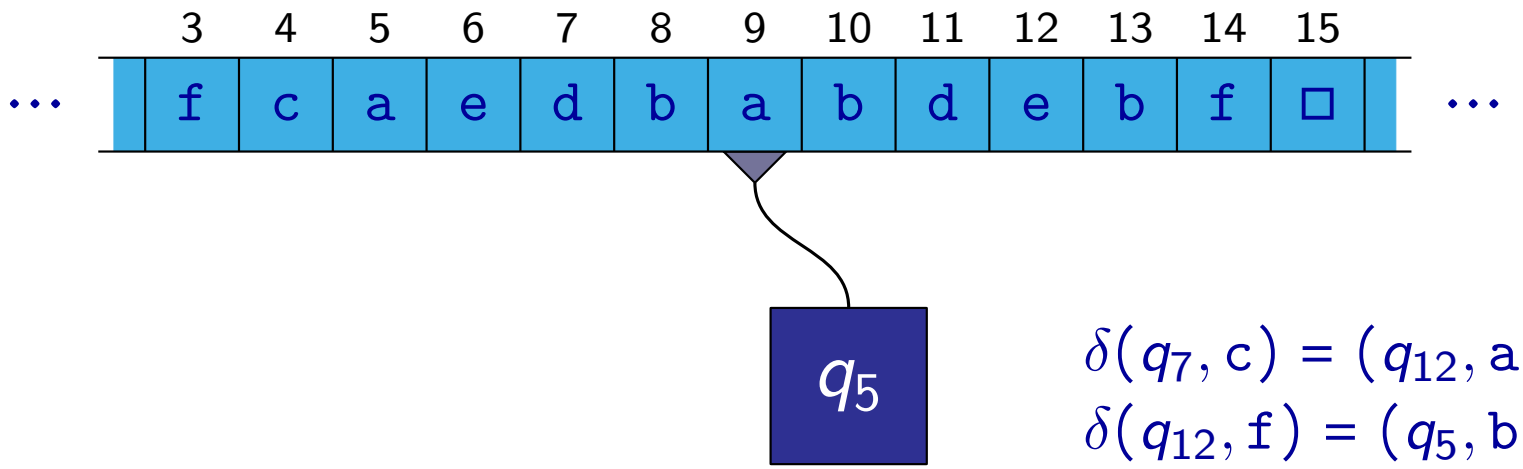
$\delta(q_7, \mathtt{c}) = (q_{12}, \mathtt{a}, +1)$
$\delta(q_{12}, \mathtt{f}) = (q_5, \mathtt{b}, -1)$

A machine with alphabet $\{0, 1\}$:



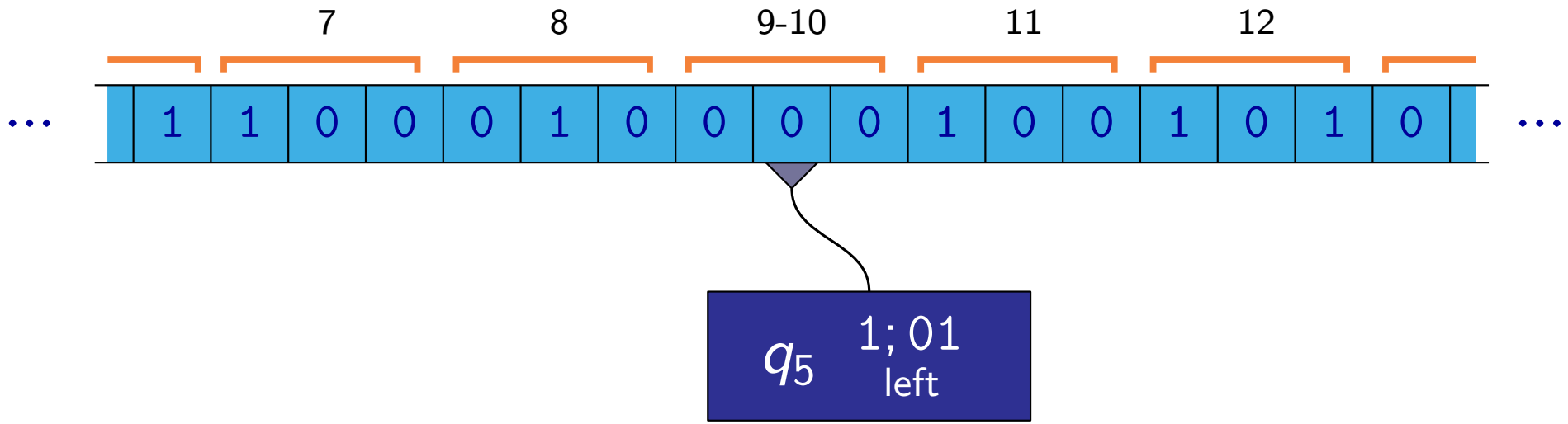
Alphabet $\{0, 1\}$

A machine with tape alphabet Γ:



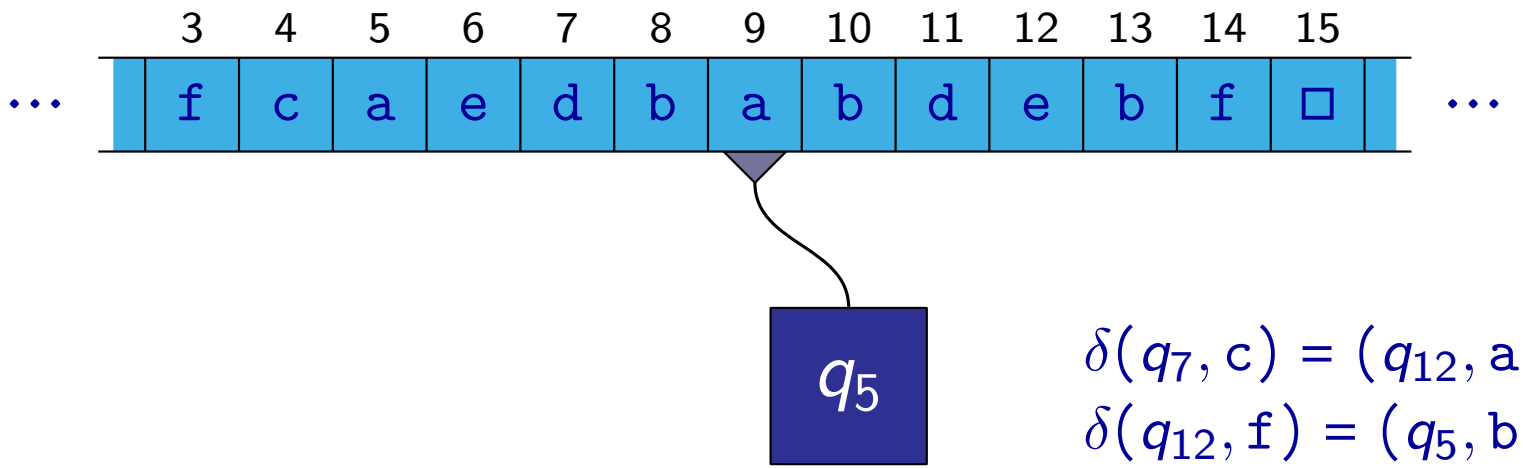
$$\delta(q_7, \mathtt{c}) = (q_{12}, \mathtt{a}, +1)$$
$$\delta(q_{12}, \mathtt{f}) = (q_5, \mathtt{b}, -1)$$

A machine with alphabet $\{0, 1\}$:



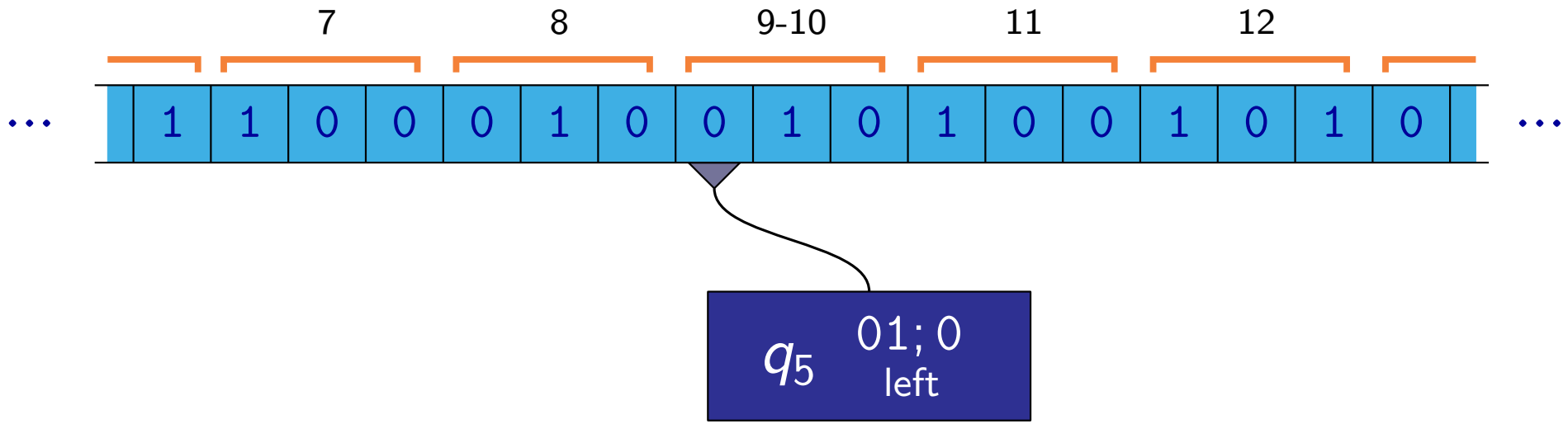
Alphabet $\{0, 1\}$

A machine with tape alphabet Γ:



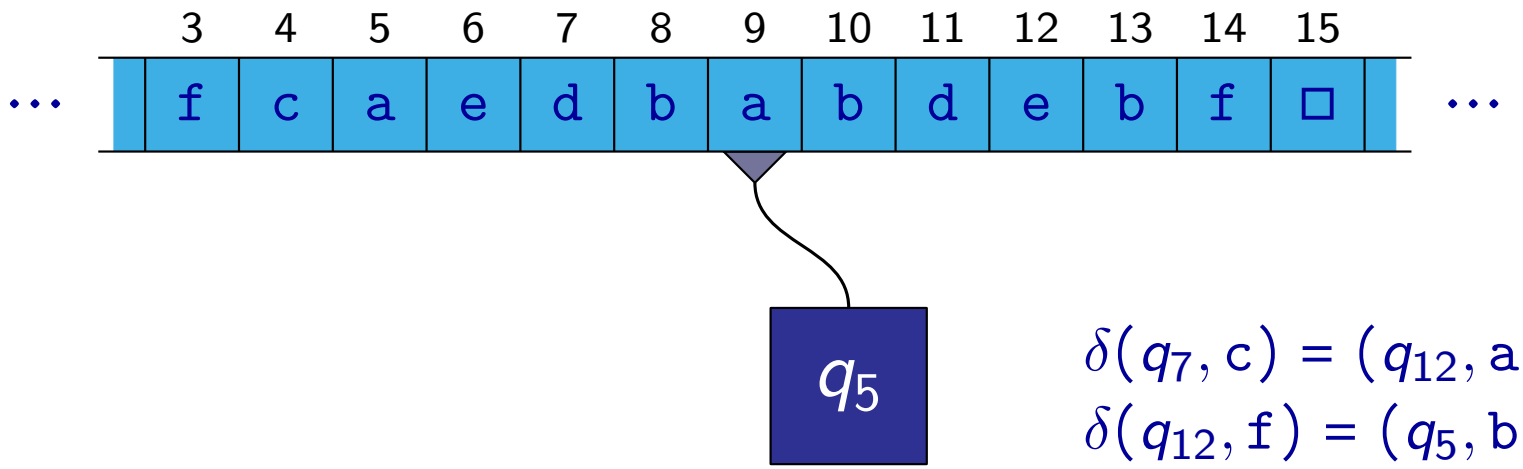
$$\delta(q_7, \mathtt{c}) = (q_{12}, \mathtt{a}, +1)$$
$$\delta(q_{12}, \mathtt{f}) = (q_5, \mathtt{b}, -1)$$

A machine with alphabet $\{0, 1\}$:



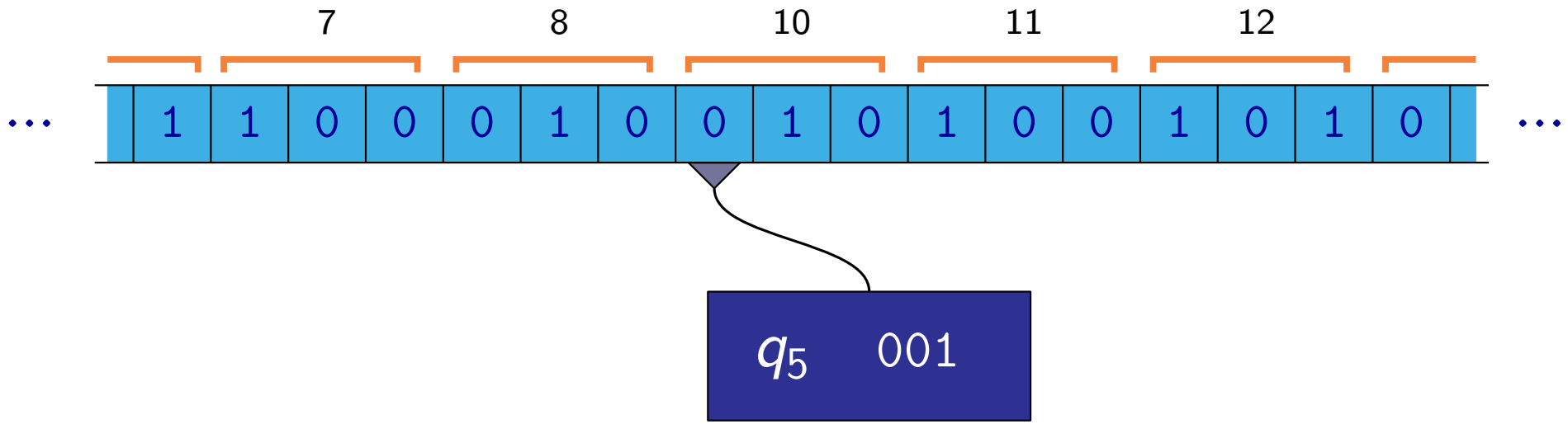
Alphabet $\{0, 1\}$

A machine with tape alphabet Γ:



$\delta(q_7, \mathtt{c}) = (q_{12}, \mathtt{a}, +1)$
$\delta(q_{12}, \mathtt{f}) = (q_5, \mathtt{b}, -1)$

A machine with alphabet $\{0, 1\}$:



Alphabet $\{0, 1\}$

In this simulation, one step of the original machine is simulated by $k + 1$ steps where k is the number of bits that encode one symbol of alphabet Γ.

So if the original machine performs in a computation t steps, the simulating machine performs $\mathcal{O}(t)$ steps.

Decreasing the number of states of the control unit

Remark: Similarly, as it is possible to decrease a tape alphabet to just two symbols with the increase of the number of the states:

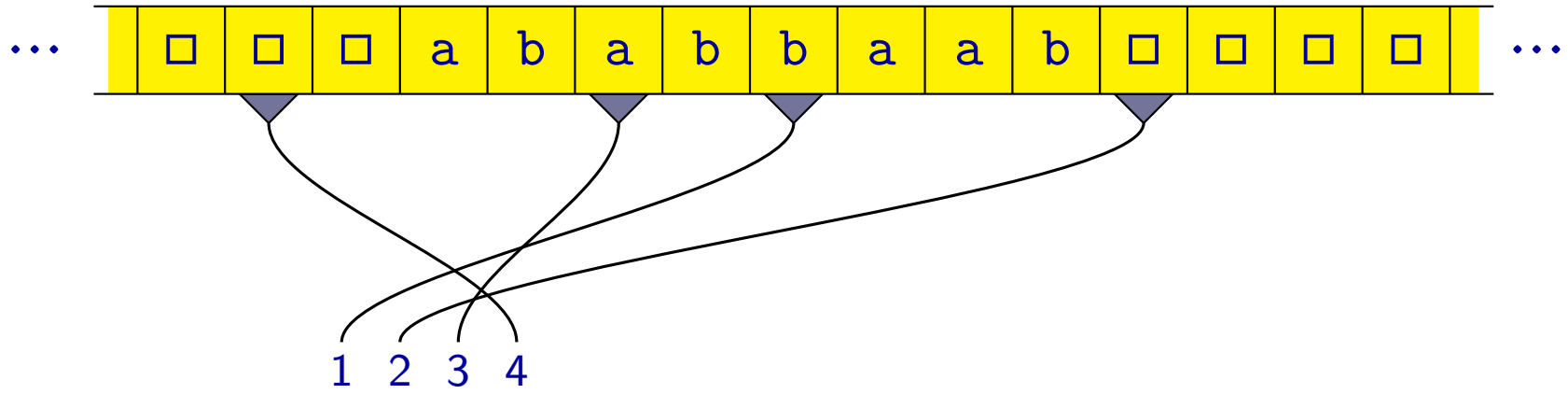
- A behaviour of an arbitrary Turing machine can be simulated by a Turing machine with only two non-final states of the control unit (and possibly with some final states), but with the increased size of its tape alphabet.

Similarly as in the previous case, one step of the original machine is simulated with s steps where s is a constant depending only on the number of states of the control unit of the original machine (i.e., on the size of Q).

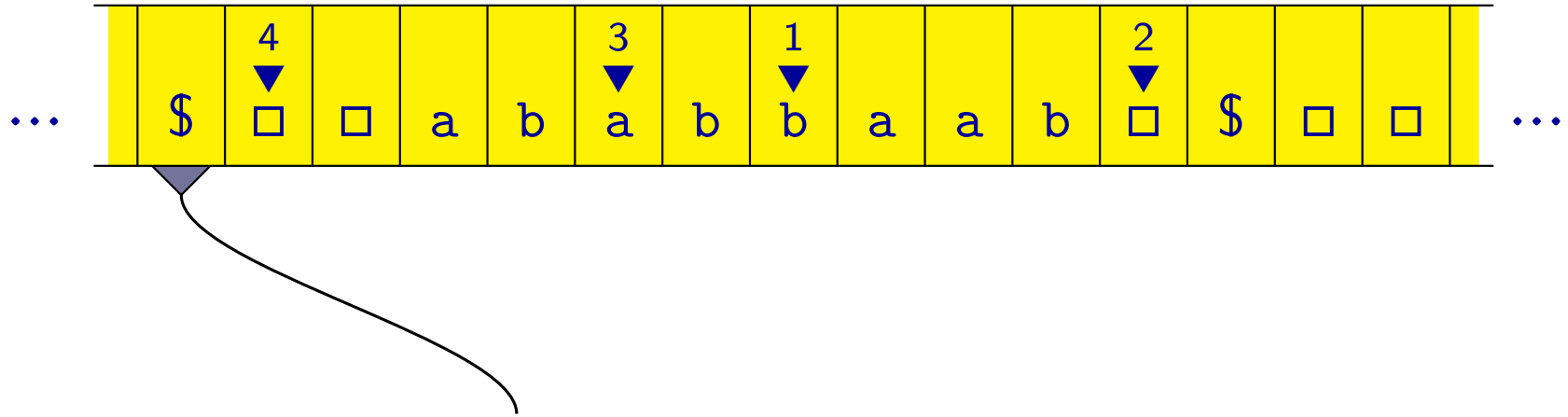
So also in this case it holds that if the original machine performs t steps in a computations, the simulating machine performs $\mathcal{O}(t)$ steps.

Simulation of multiple heads on a tape by one head

Several heads on a tape:

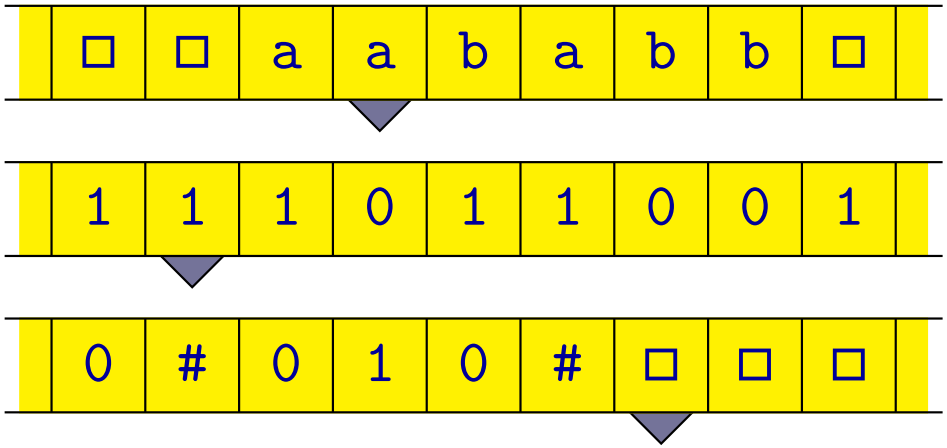


A tape with one head:

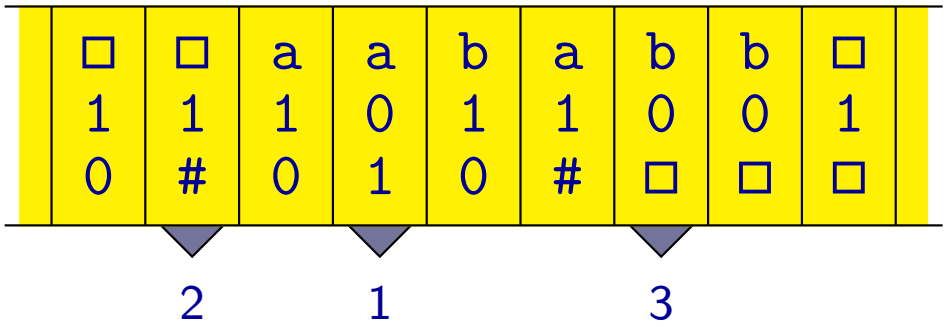


Simulation of several tapes with one tape

Several tapes:

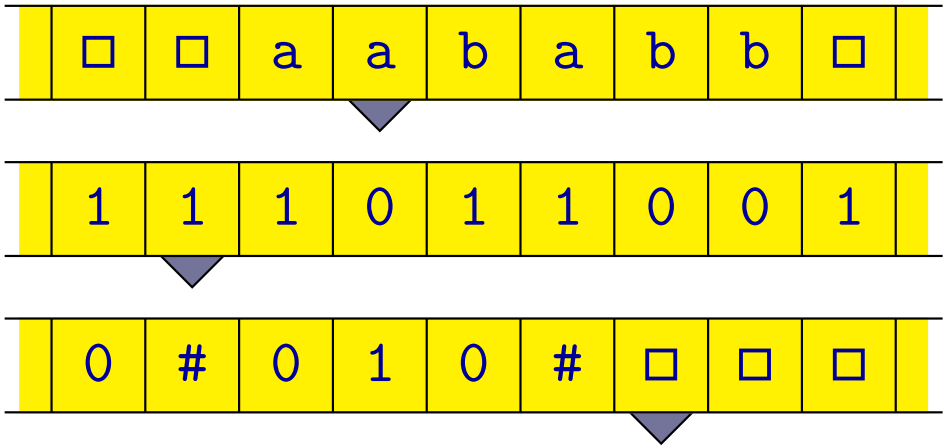


One tape with several heads:

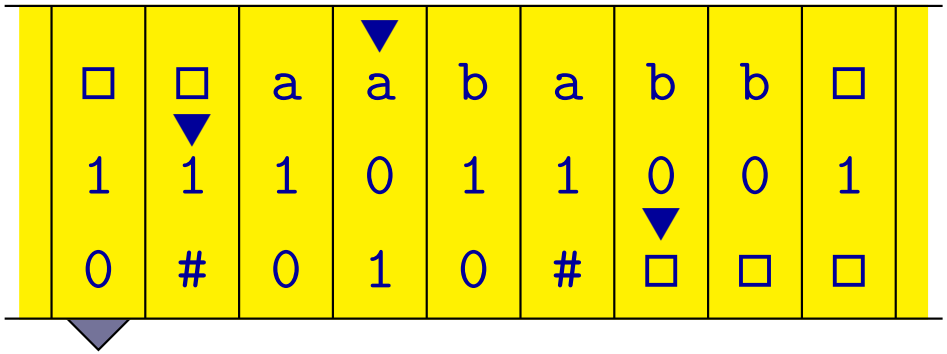


Simulation of several tapes with one tape

Several tapes:

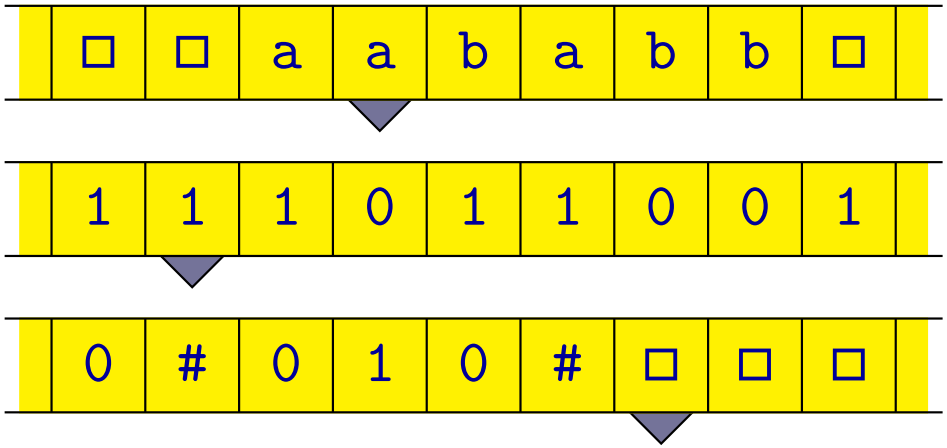


One tape with one head: the variant where where marks on the tape are moved

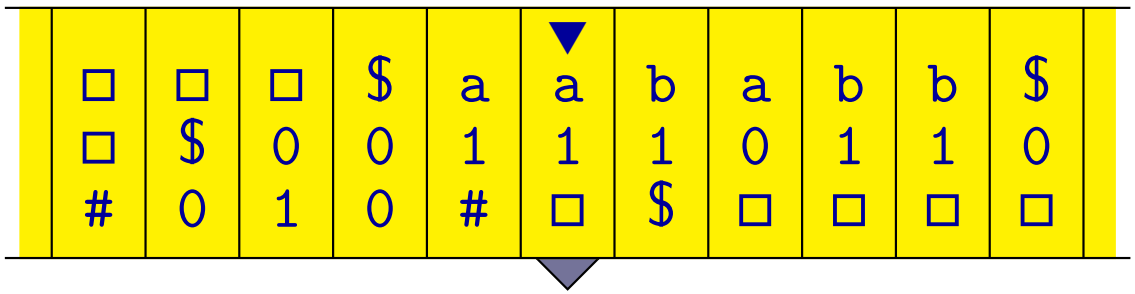


Simulation of several tapes with one tape

Several tapes:



One tape with one head: the variant where the content of tapes is moved



Simulation of multiple tapes or multiple heads

- All simulations of several tapes (or one tape with several heads) by one tape with one head that were described above require $\mathcal{O}(m)$ steps to simulate one step where m is the maximal number of cells that the original machine visits on some of its tapes.
- If the original machine performs t steps, it could have visited at most t cell on any of its tapes.
- So the simulation of a computation consisting of t steps can be done in at most $\mathcal{O}(t^2)$ steps.

Simulation of multiple tapes or multiple heads

Additional remarks:

- There are algorithmic problems that can be easily solved in a linear time (i.e., in $\mathcal{O}(n)$ steps where n is the size of the input) by a Turing machine with two tapes (or two heads on one tape) but for which it can be proved that every one-tape Turing machine that solves the problem must perform at least $\Omega(n^2)$ steps.

  Example:
  Recognizing palindroms, i.e., words in the language

$$\{ w \in \{a, b\}^* \mid w = w^R \}$$

Simulation of multiple tapes or multiple heads

- There exist (complicated and nontrivial) ways how a tape with several heads can be simulated with several tapes where every tape has a single head.
- There exists a (complicatedand nontrivial) way how to simulate k tapes by just two tapes (where both tapes have one head) in such a way that if a computation of the original machine consist of t steps then the simulating machines performs at most $\mathcal{O}(t \log t)$ steps.

Random Access Machine and Turing Machine

Any program in any programming language can be implemented as a program for a RAM.

It is not difficult (although a little bit tedious) to realize that every algorithm performed by a RAM can also be implemented by a Turing machine.

A Turing machine can implement any algorithm that can be written as a program in an arbitrary programming language.

Remark: And of course, it is possible to simulate a behaviour of a Turing machine by a RAM.

Turing Machine Simulating RAM

In the description of how a Turing machine can simulate a RAM, it is simpler to proceed by smaller steps:

- We will show how to simulate a varint of RAM described before by a variant of RAM with somewhat simpler instructions.
- We will show how to simulate the behaviour of this simpler variant of RAM by a multitape Turing machine.
- We have already seen before how a multitape Turing machine can be simulated by one-tape Turing machine.

A simpler variant of RAM

This simpler variant of RAM has, in addition to its working memory, also three **registers**:

- **register A** — almost all instructions work with this register, results of all operations are stored into this register

  Remark: This kind of register is often called an **accumulator**.

- **register B** — this register is used to store the second operand of arithmetic instructions (the first operand is always in the accumulator)

- **register C** — this register is used to store an address of a memory cell, to which a value is written by a store operation

A simpler variant of RAM

Overview of instructions:

$A := c$	– assinment of a constant
$B := A$	– assinment to register B
$C := A$	– assinment to register C
$A := [A]$	– load (reading from memory)
$[C] := A$	– store (writing to memory)
$A := A \; op \; B$	– arithmetic instructions, $op \in \{+, -, *, /\}$
if $(A \; rel \; 0)$ **goto** ℓ	– conditional jump, $rel \in \{=, \neq, \leq, \geq, <, >\}$
goto ℓ	– unconditional jump
$A :=$ READ $()$	– reading from input
WRITE (A)	– writing to output
halt	– program termination

A simpler variant of RAM

For example, instruction

$$R_5 := 42$$

can be replaced with a sequence of instructions:

$$A := 5$$
$$C := A$$
$$A := 42$$
$$[C] := A$$

A simpler variant of RAM

For example, instruction

$$R_{12} := R_3$$

can be replaced with a sequence of instructions:

$$A := 12$$
$$C := A$$
$$A := 3$$
$$A := [A]$$
$$[C] := A$$

A simpler variant of RAM

For example, instruction

$$R_8 := [R_2]$$

can be replaced with a sequence of instructions:

$$A := 8$$
$$C := A$$
$$A := 2$$
$$A := [A]$$
$$A := [A]$$
$$[C] := A$$

A simpler variant of RAM

For example, instruction

$$[R_{15}] := R_9$$

can be replaced with a sequence of instructions:

$$A := 15$$
$$A := [A]$$
$$C := A$$
$$A := 9$$
$$A := [A]$$
$$[C] := A$$

A simpler variant of RAM

For example, instruction

$$R_7 := R_3 + R_6$$

can be replaced with a sequence of instructions:

$$A := 7$$
$$C := A$$
$$A := 6$$
$$A := [A]$$
$$B := A$$
$$A := 3$$
$$A := [A]$$
$$A := A + B$$
$$[C] := A$$

A simpler variant of RAM

For example, instruction

$$\textbf{if } (R_4 \geq R_{11}) \textbf{ goto } \ell$$

can be replaced with a sequence of instructions:

$$\begin{aligned}
A &:= 11 \\
A &:= [A] \\
B &:= A \\
A &:= 4 \\
A &:= [A] \\
A &:= A - B \\
&\textbf{if } (A \geq 0) \textbf{ goto } \ell
\end{aligned}$$

A simpler variant of RAM

For example, instruction

$$R_{23} := \text{READ}\,()$$

can be replaced with a sequence of instructions:

$$A := 23$$
$$C := A$$
$$A := \text{READ}\,()$$
$$[C] := A$$

A simpler variant of RAM

For example, instruction

$$\text{WRITE}\,(R_{17})$$

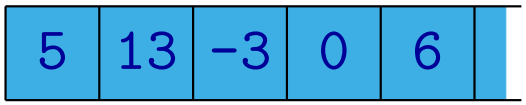
can be replaced with a sequence of instructions:

$$A := 17$$
$$A := [A]$$
$$\text{WRITE}\,(A)$$

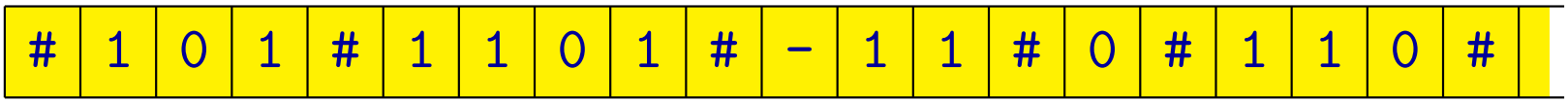
Turing Machine Simulating RAM

A Turing machine works with words over some alphabet, while a RAM works with numbers. But numbers can be written as sequences of symbols and conversely symbols of an alphabet can be written as numbers.

For example the following input of a RAM



can be represented for a Turing machine as



Turing Machine Simulating RAM

A Turing machine simulating a computation of a RAM has several tapes:

- A tape containing a content of the working memory of the RAM.
- Three tapes containing values of registers A, B, and C.

  (Values of registers A, B, and C will be written on these tapes in binary and delimited from the left and from the right by symbols #.)
- A tape representing the input tape of the RAM.
- A tape representing the output tape of the RAM.
- One auxiliary tape used for an implementation of the simulation of some instructions.

Turing Machine Simulating RAM

The Turing machine stores the information about the instruction of the RAM that is currently executed in its control unit.

Execution of most of instructions is not difficult:

- $A := c$

  it writes bits of the constant c to the tape of register A

- $B := A$ or $C := A$

  it will copy a content of the tape of register A to the tape of register B or C

- **goto** ℓ

  just changes the state of the control unit of the Turing machine

- **if** $(A \; rel \; 0)$ **goto** ℓ, kde $rel \in \{=, \neq, \leq, \geq, <, >\}$

  the content of the working register is tested and the state of the control unit is changed accordingly

Turing Machine Simulating RAM

- $A :=$ READ $()$

  copy the value (marked at the ends by symbols "#") from the input tape to the tape of register A

- WRITE (A)

  copy the value of register A to the output tape.

- **halt**

  the computation halts

Turing Machine Simulating RAM

Also arithmetic instructions are rather easy to implement, although the a little bit more complicated than the previous instructions:

- $A := A \; op \; B$, where $op \in \{+, -, *, /\}$

  The Turing machine performs the given operation (such as addition or subtraction) bit by bit, the result is stored to register A.

Remark: Multiplication and division can be done as a sequence of additions and bit shifts.

In the implementation of addition and division, it may be necessary to use an auxiliary tape to store intermediate results.

Turing Machine Simulating RAM

Probably the most complex is the implementation of the RAM memory.

One possibility is to store only values of those cells that were actually used so far in the computation of the RAM (we know that all other cells contain value 0).

Example: The RAM worked so far only with cells 2, 3 and 6:

- Cell 2 contains value 11.
- Cell 3 contains value -1.
- Cell 6 contains value 2.

The content of the tape of the Turing machine representing the content of the memory of the RAM will be as follows:

$	#	1	0	:	1	0	1	1	#	1	1	:	-	1	#	1	1	0	:	1	0	#	$	

Turing Machine Simulating RAM

Load instruction, i.e., $A := [A]$:

- The Turing machine will search the given address, stored in register A, on the tape containg the content of the memory of the RAM.
  (If it does not find it, it will appened it at the end with value 0.)
- The given value in the cell is copied to the tape of register A.

Turing Machine Simulating RAM

Store instruction, i.e., $[C] := A$:

- Similarly as before, the Turing machine will find the position of the tape representing a content of the memory, where the value in the given address, stored in register C, occurs.
- The rest of the memory tape is copied to an auxiliary tape.
- The content of the tape of register A is copied to the corresponding place.
- The rest of the tape, copied on the auxiliary tape, is copied back to the memory tape (after the newly written value).

Tapes, Stacks and Counters

Are there some Turing-complete machines that are even simpler than Turing machines? It turns out that this is indeed the case.

All the following models have a finite control unit equipped with some sort of memory of unbounded size.

Such memory can constist of one of more structures such as:

- **Tape** — reading and writing a symbol on a current position, movement of the head to the left and to the right

  Remark: The tape can be infinite of one side or on both sides.

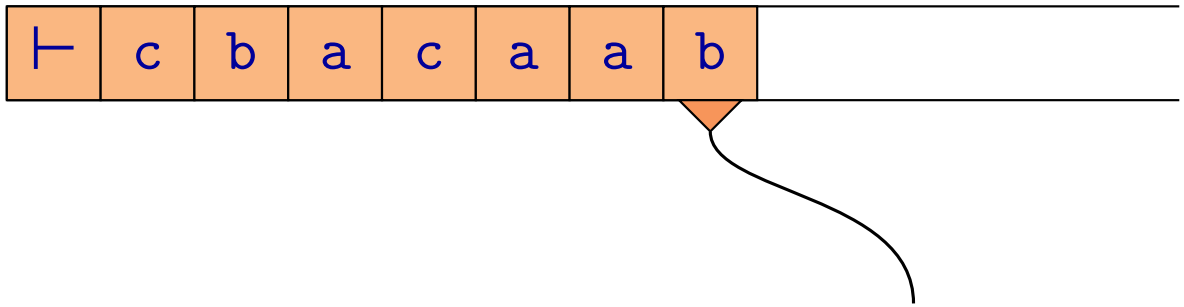
- **Stack** — push, pop, a test of emptiness of the stack

- **Counter** — a value is a natural number, operations of incrementing and decrementing by one, a test whether the value is equal to zero

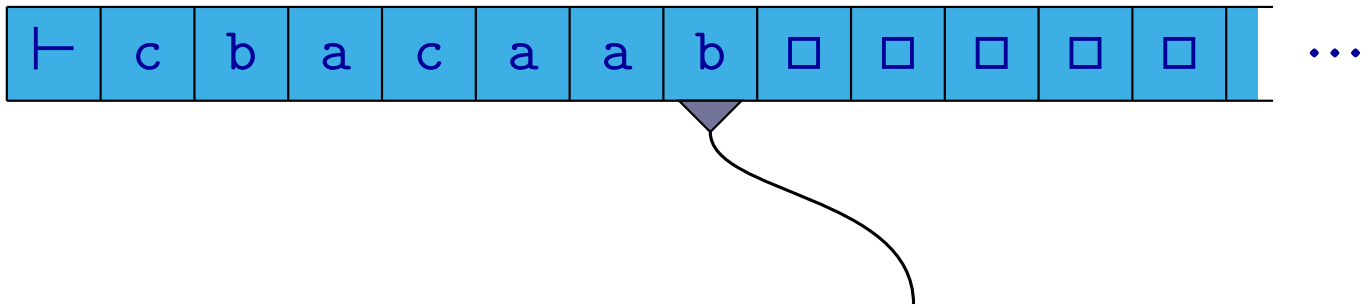
Stack

A stack can be viewed as a special case of a tape, which is infinite on one side.

Stack:



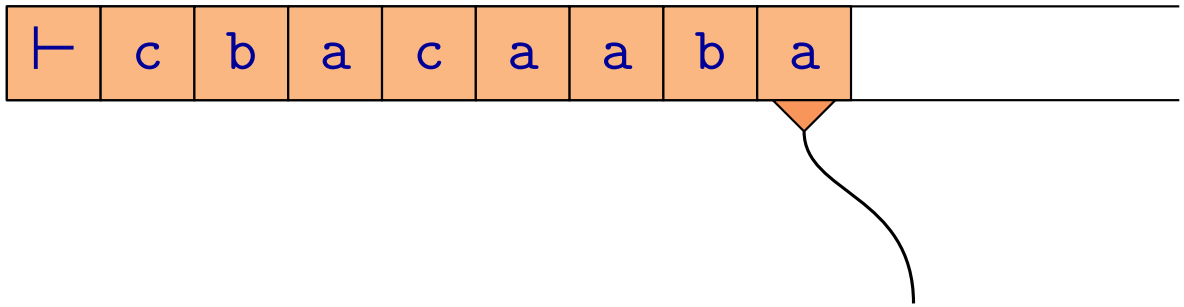
Tape:



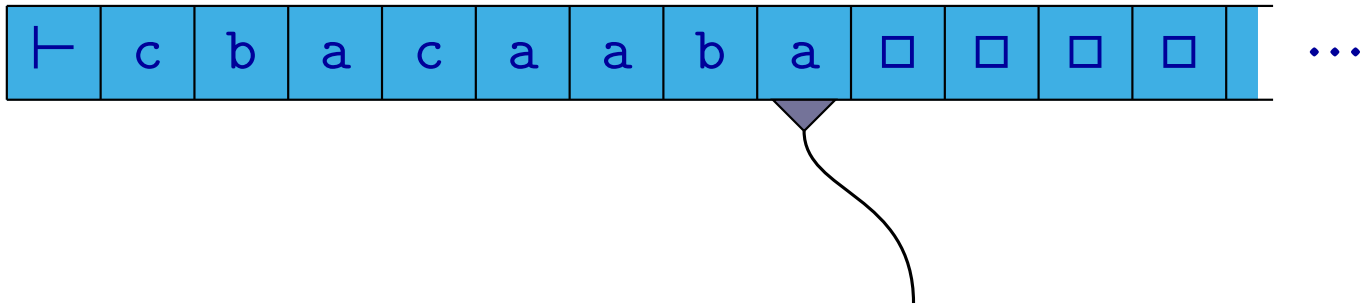
Stack

A stack can be viewed as a special case of a tape, which is infinite on one side.

Stack:



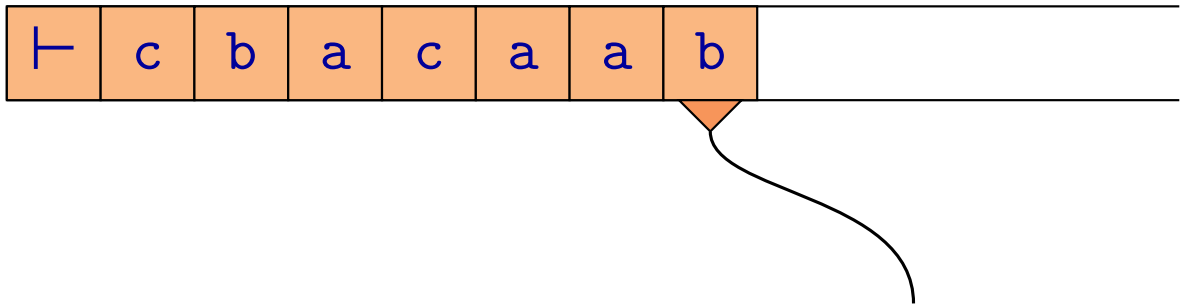
Tape:



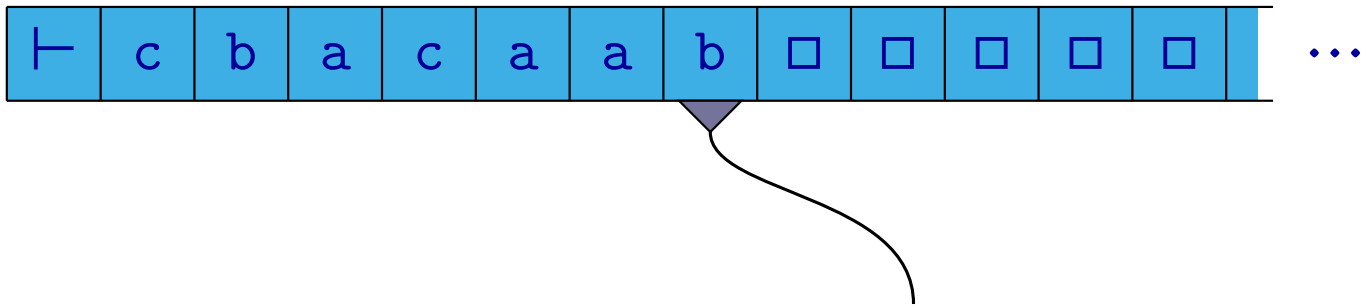
Stack

A stack can be viewed as a special case of a tape, which is infinite on one side.

Stack:



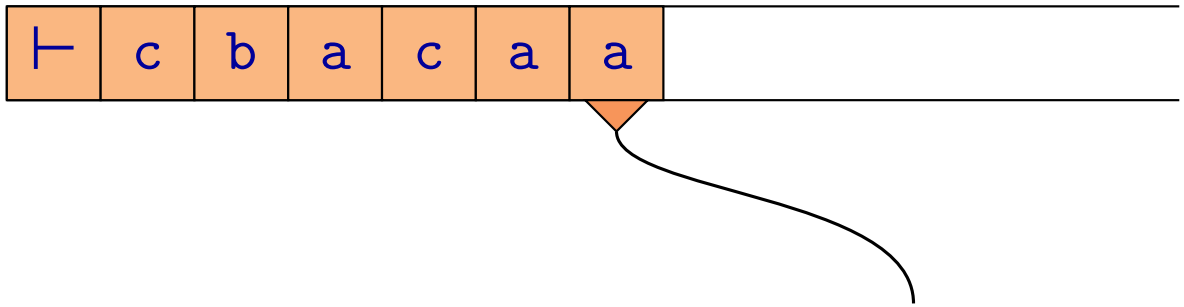
Tape:



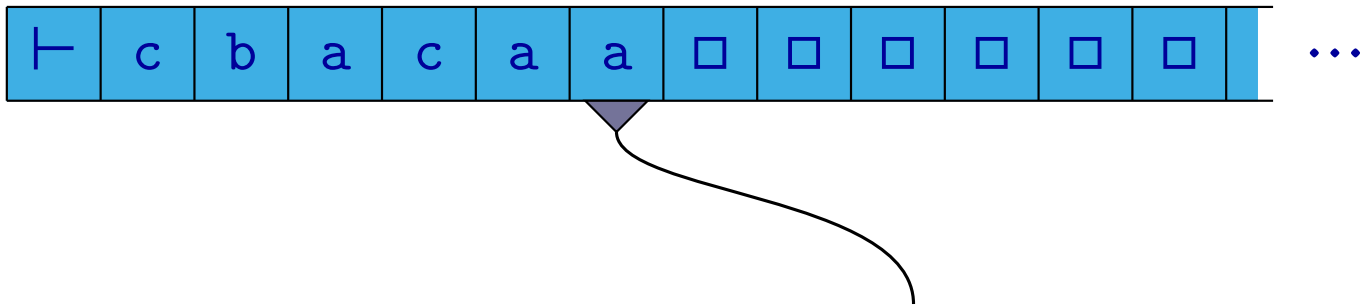
Stack

A stack can be viewed as a special case of a tape, which is infinite on one side.

Stack:



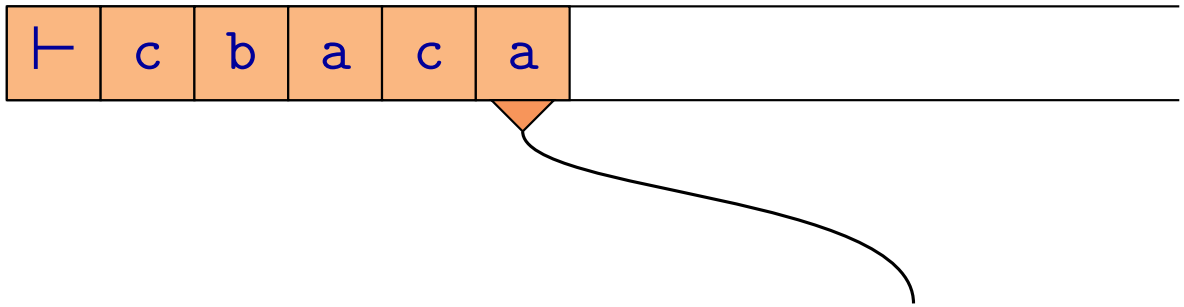
Tape:



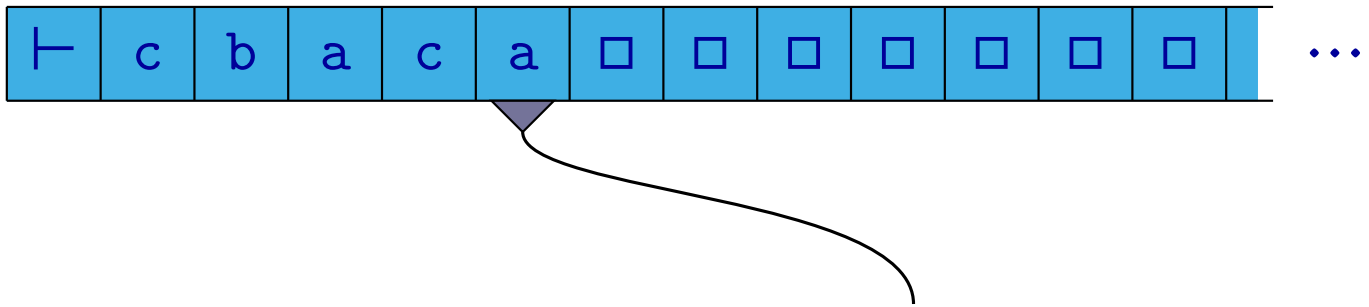
Stack

A stack can be viewed as a special case of a tape, which is infinite on one side.

Stack:

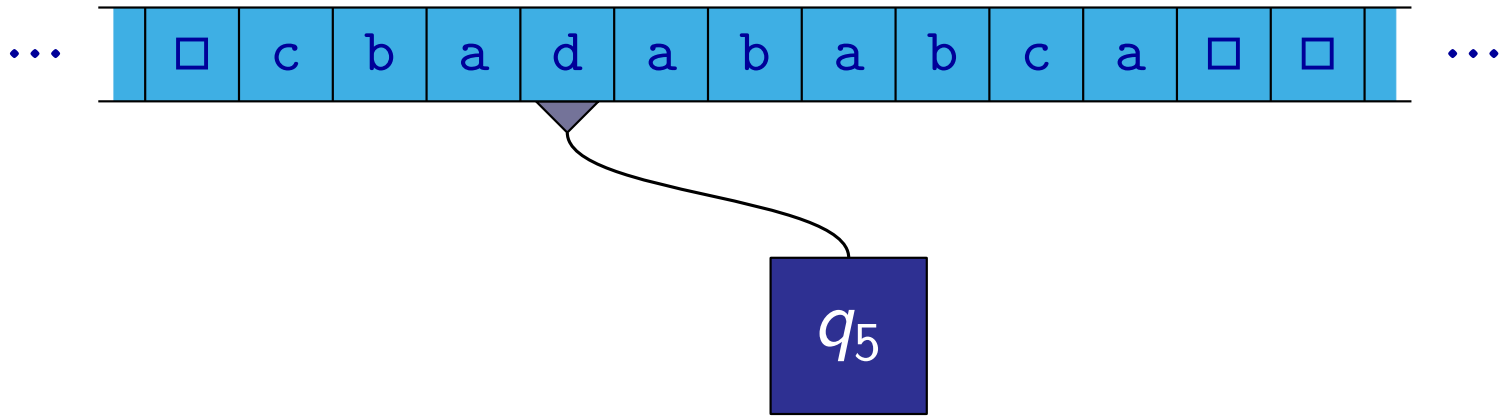


Tape:

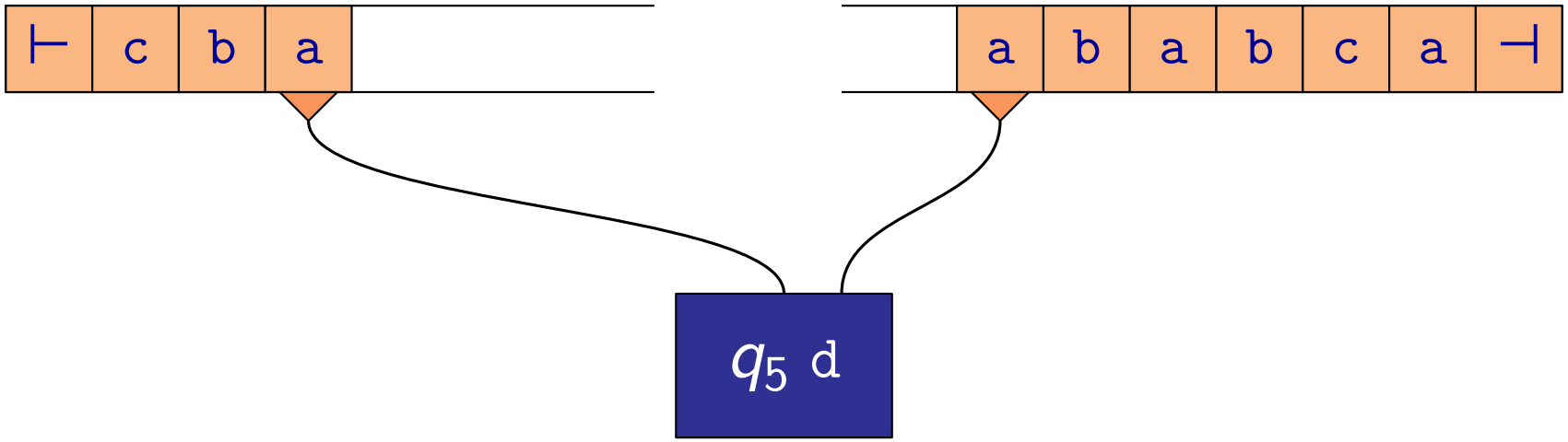


Stack

A tape, infinite on both sides, can be simulated by two stacks:

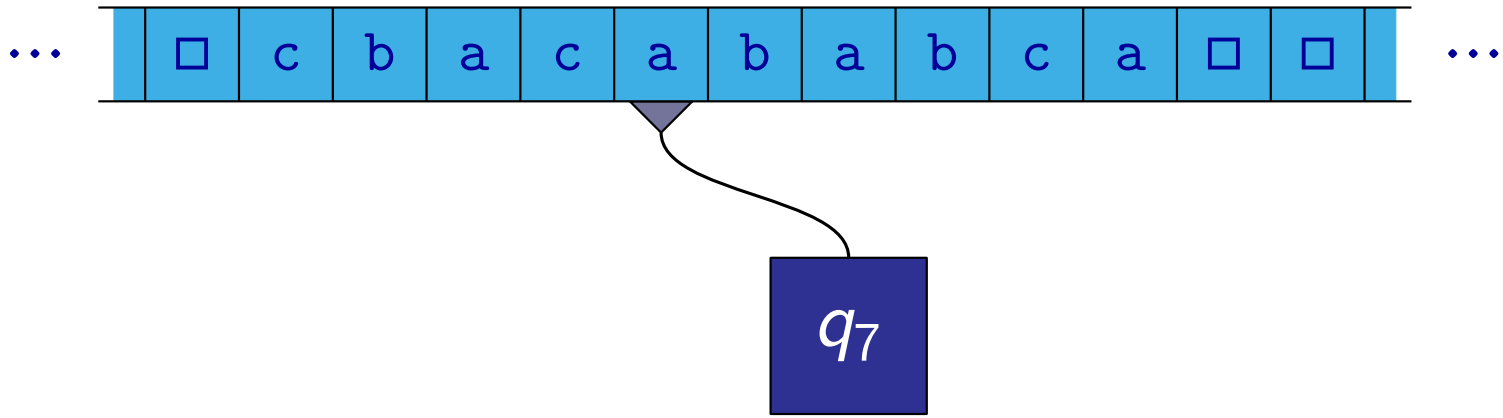


A machine with two stacks:

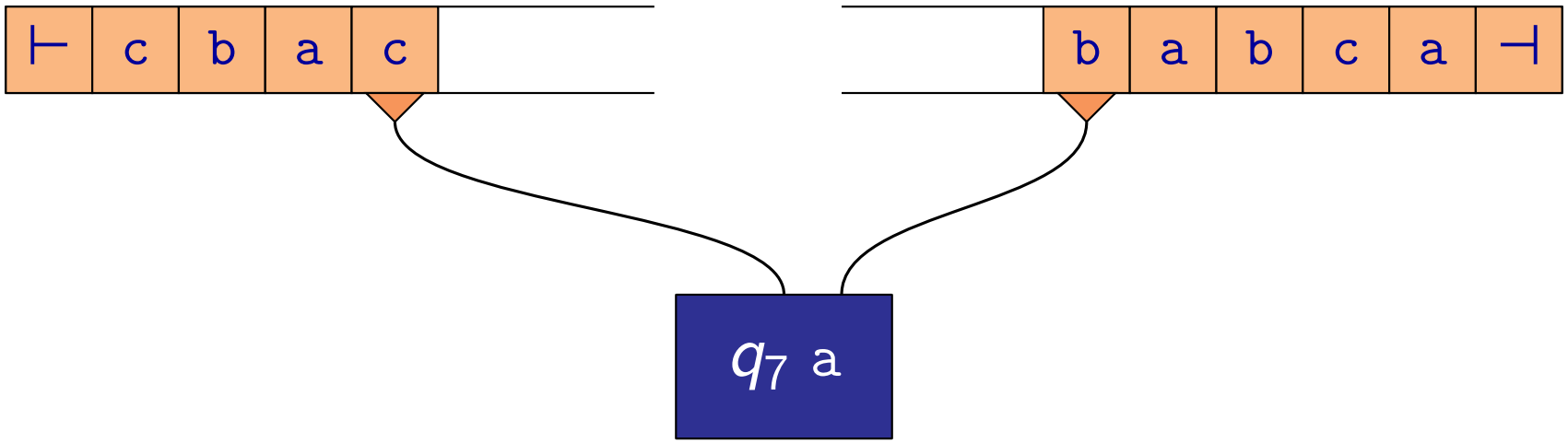


Stack

A tape, infinite on both sides, can be simulated by two stacks:



A machine with two stacks:



Counter

Counter — a value of a counter can be an arbitrarily big natural number, i.e., an element of the set $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$.

Basic operations:

- incrementing the value by one:

$$x := x + 1$$

- decrementing the value by one:

$$x := x - 1$$

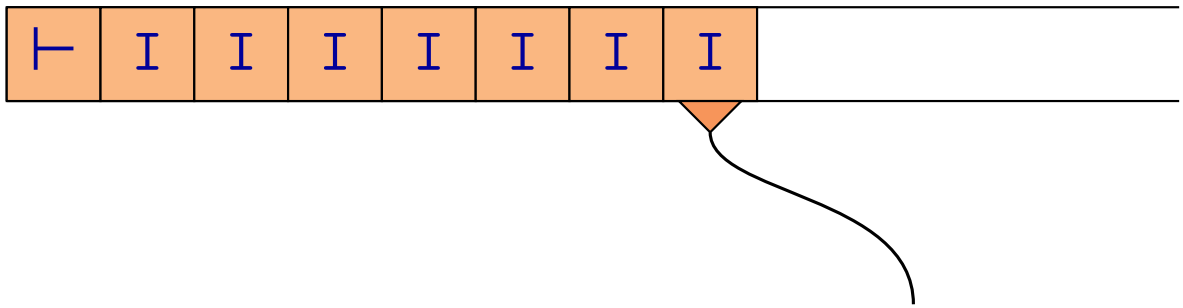
- test whether the value of the counter is zero:

$$\textbf{if } (x = 0) \textbf{ goto } \ell$$

Counter

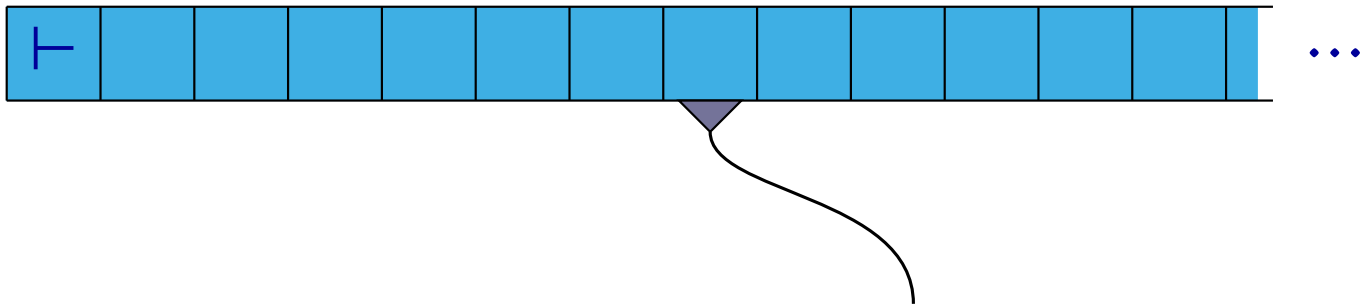
A counter can be viewed as a special case of a stack or of a tape.

Stack:



Counter:

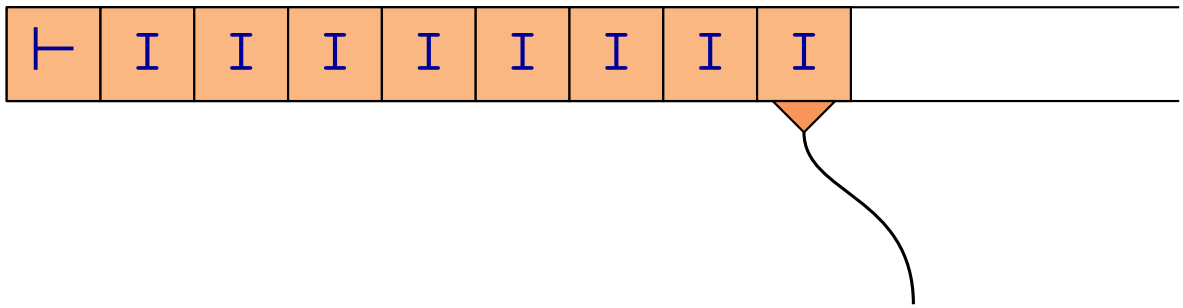
Tape:



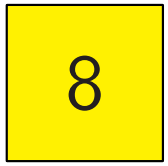
Counter

A counter can be viewed as a special case of a stack or of a tape.

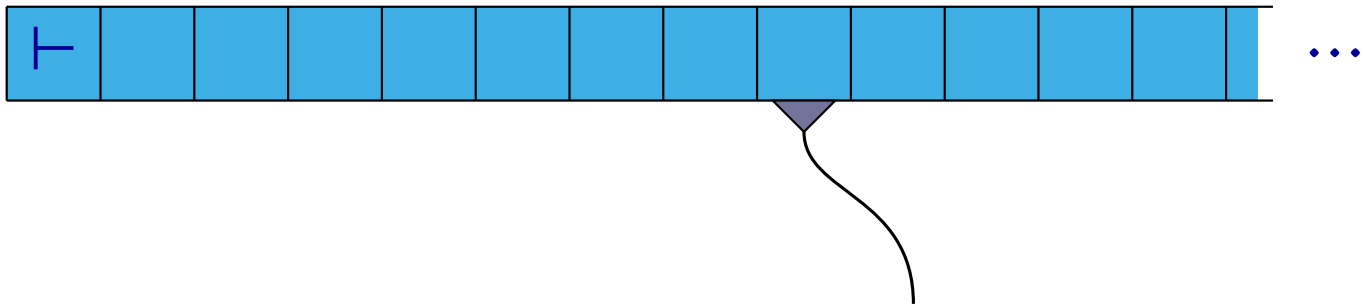
Stack:



Counter:



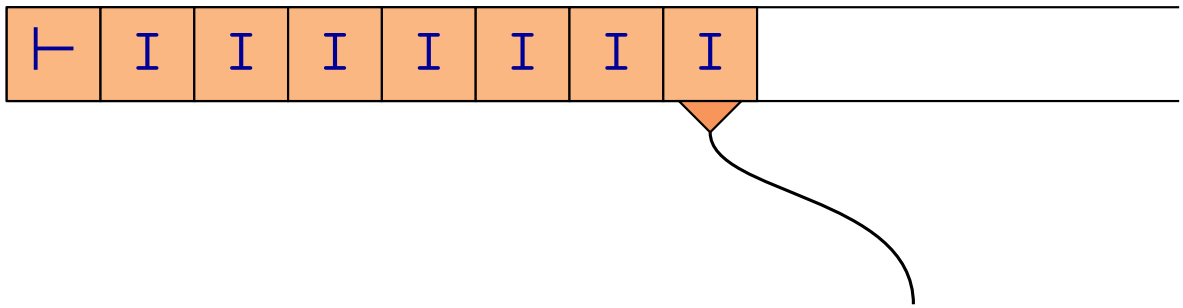
Tape:



Counter

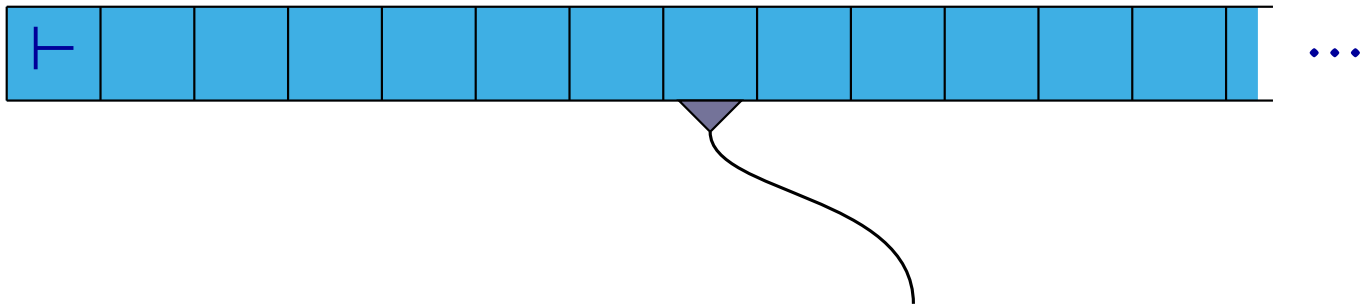
A counter can be viewed as a special case of a stack or of a tape.

Stack:



Counter:

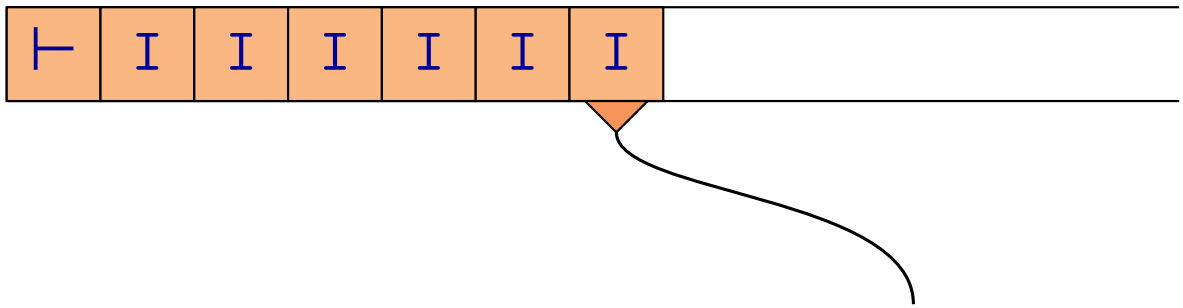
Tape:



Counter

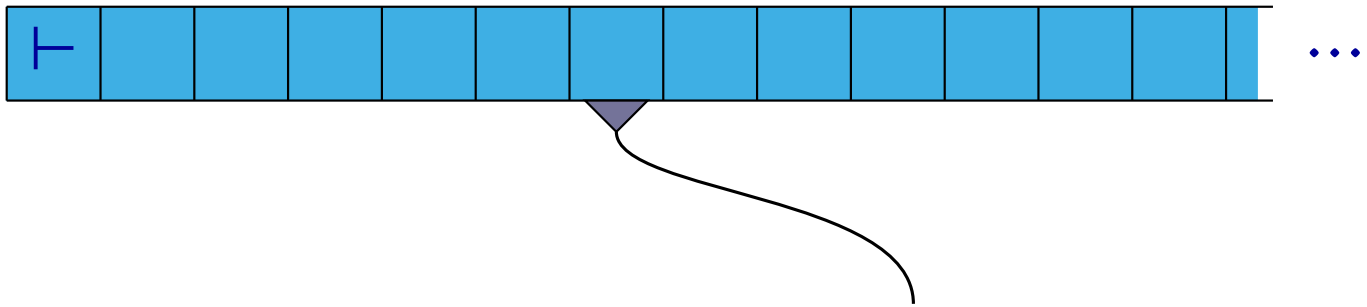
A counter can be viewed as a special case of a stack or of a tape.

Stack:



Counter:

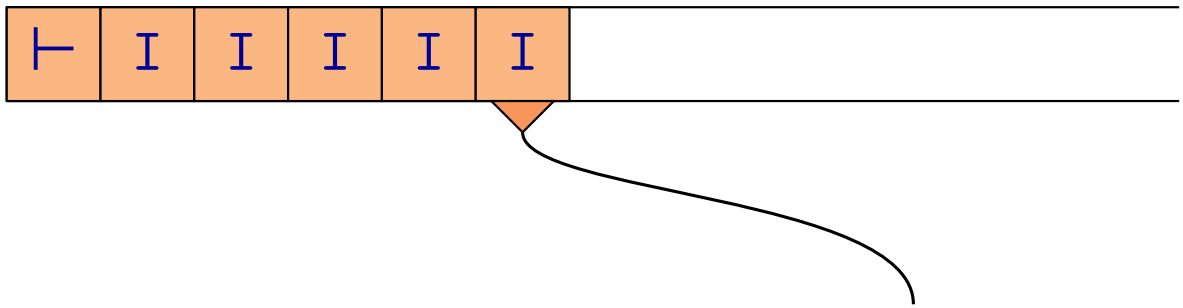
Tape:



Counter

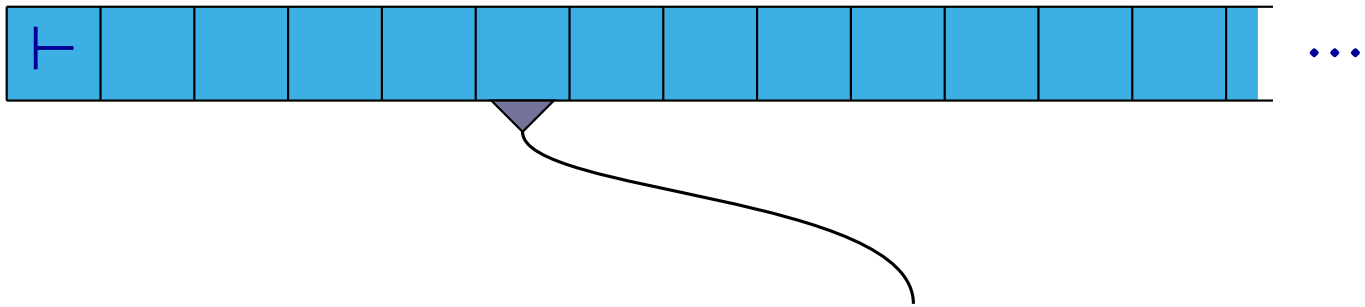
A counter can be viewed as a special case of a stack or of a tape.

Stack:



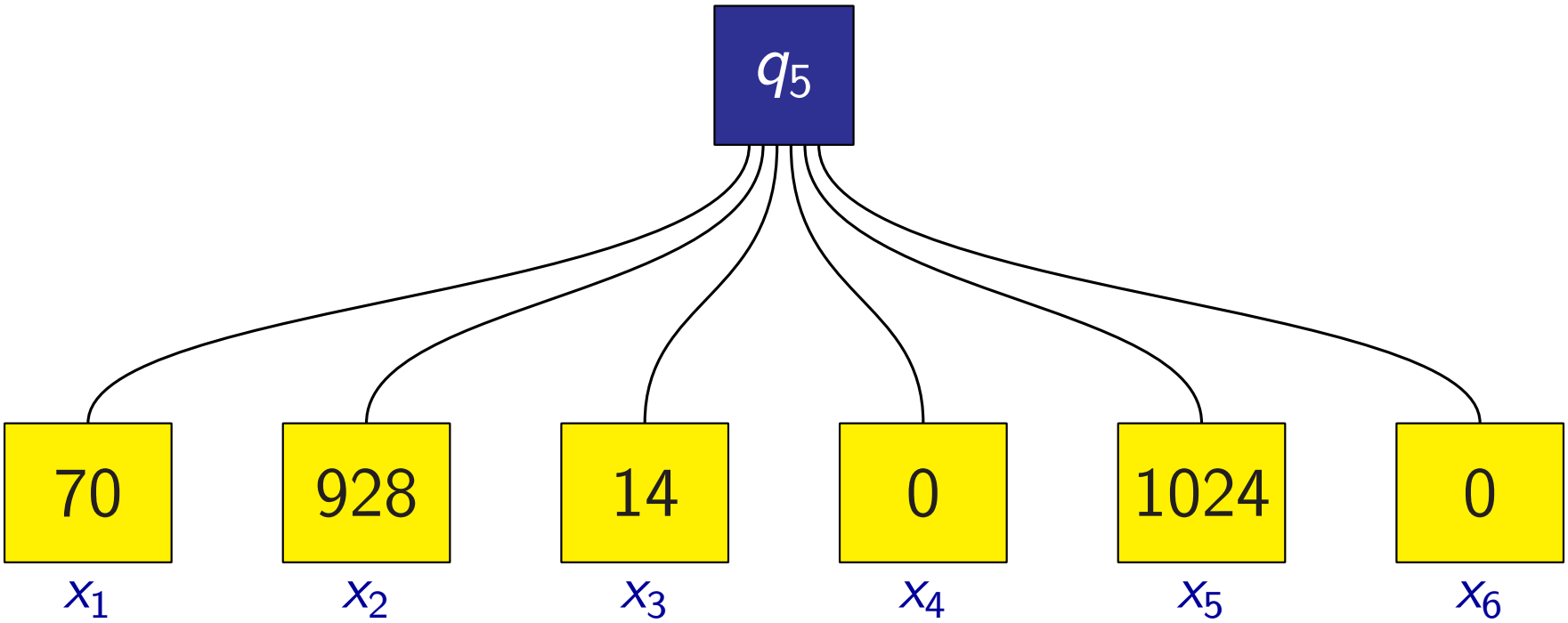
Counter:

Tape:



Minsky machine

Minsky machine — a machine with a finite control unit and a finite set of counters $x_1, x_2, \ldots, x_k$:



Remark: Instead of $x_1, x_2, \ldots$ to denote counters, we will use also symbols such as $x, y, z, \ldots$

Minsky machine

A Minsky machine can be viewed as a program consisting of a sequence of instructions, with the following five types of instructions:

- incrementing the value of a given counter by one:

$$x_i := x_i + 1$$

- decrementing the value of a given counter by one:

$$x_i := x_i - 1$$

- test whether the value of a given counter is zero:

$$\textbf{if } (x_i = 0) \textbf{ goto } \ell$$

- unconditional jump:

$$\textbf{goto } \ell$$

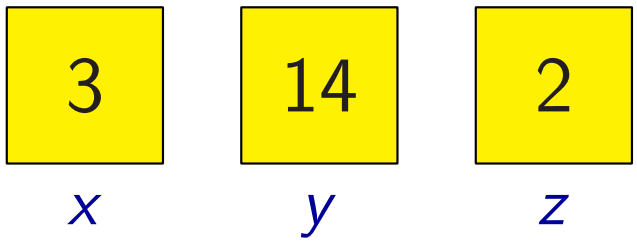
- halting of the computation of the program:

$$\textbf{halt}$$

Minsky machine

Reseting counter x to zero:

L_1 : **if** $(x = 0)$ **goto** L_2
$x := x - 1$
goto L_1
L_2 : ...



Minsky machine

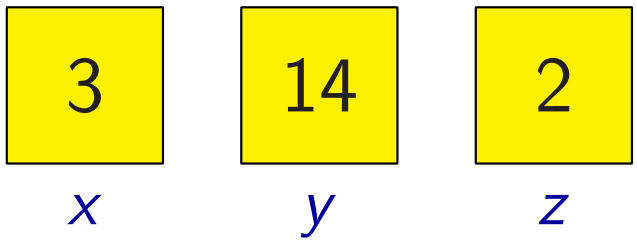
Reseting counter x to zero:

L_1 : **if** $(x = 0)$ **goto** L_2

→ $x := x - 1$

goto L_1

L_2 : ...



Minsky machine

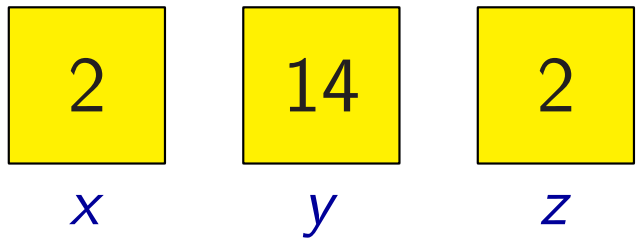
Reseting counter x to zero:

L_1 : **if** $(x = 0)$ **goto** L_2

$x := x - 1$

goto L_1

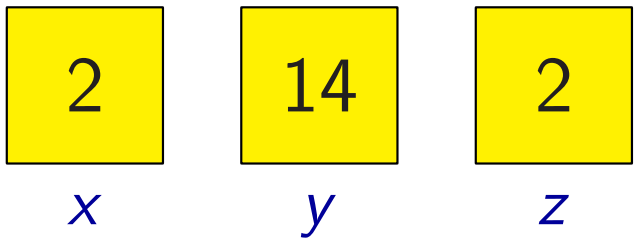
L_2 : ...



Minsky machine

Reseting counter x to zero:

L_1 : **if** $(x = 0)$ **goto** L_2
$x := x - 1$
goto L_1
L_2 : ...



Minsky machine

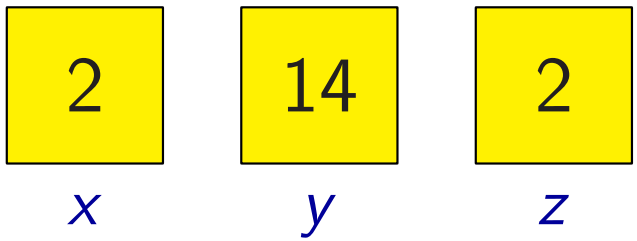
Reseting counter x to zero:

L_1 : **if** $(x = 0)$ **goto** L_2

→ $x := x - 1$

goto L_1

L_2 : ...



Minsky machine

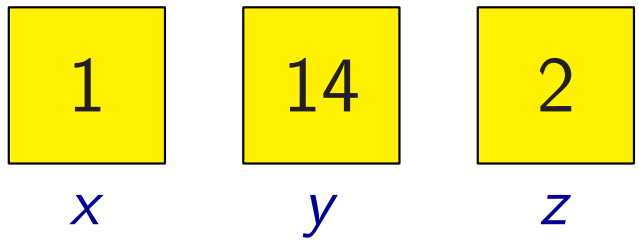
Reseting counter x to zero:

L_1 : **if** $(x = 0)$ **goto** L_2

$x := x - 1$

goto L_1

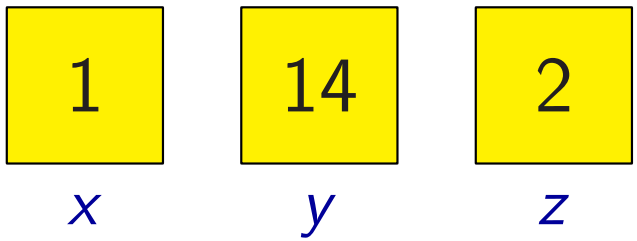
L_2 : ...



Minsky machine

Reseting counter x to zero:

L_1 : **if** $(x = 0)$ **goto** L_2
$x := x - 1$
goto L_1
L_2 : ...



Minsky machine

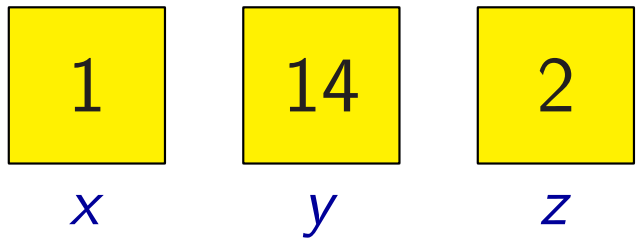
Reseting counter x to zero:

L_1 : **if** $(x = 0)$ **goto** L_2

$x := x - 1$

goto L_1

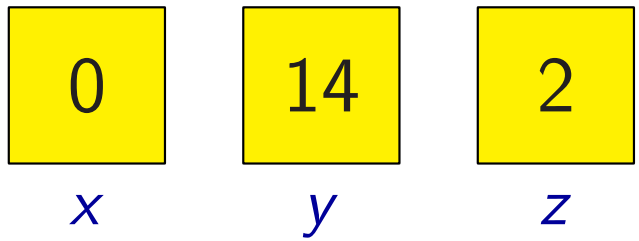
L_2 : ...



Minsky machine

Reseting counter x to zero:

L_1 : **if** $(x = 0)$ **goto** L_2
$\quad x := x - 1$
$\quad$ **goto** L_1
L_2 : $\quad \ldots$



Minsky machine

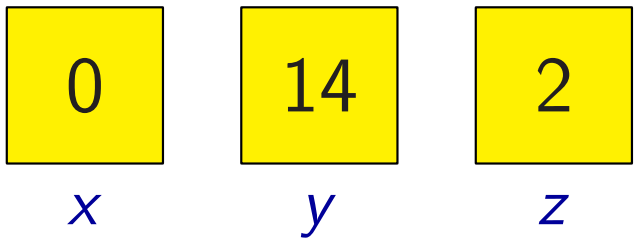
Reseting counter x to zero:

L_1 : **if** $(x = 0)$ **goto** L_2

$x := x - 1$

goto L_1

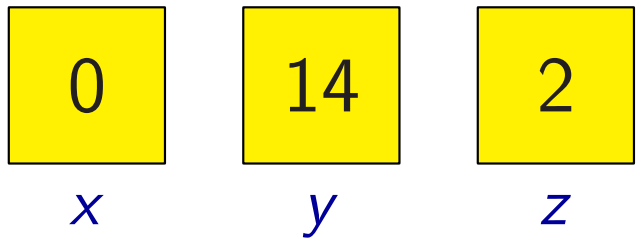
L_2 : ...



Minsky machine

Reseting counter x to zero:

L_1 : **if** $(x = 0)$ **goto** L_2
$x := x - 1$
goto L_1
L_2 : ...



Minsky machine

Adding the value of the counter z to the counter y
(and reseting counter x to zero):

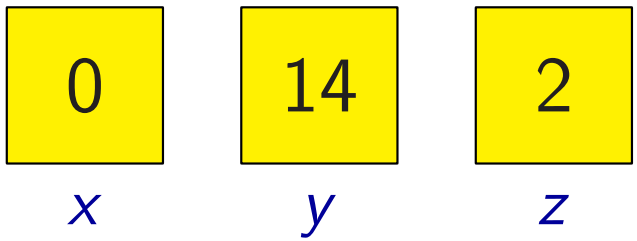
L_2 : **if** $(z = 0)$ **goto** L_3

$z := z - 1$

$y := y + 1$

goto L_1

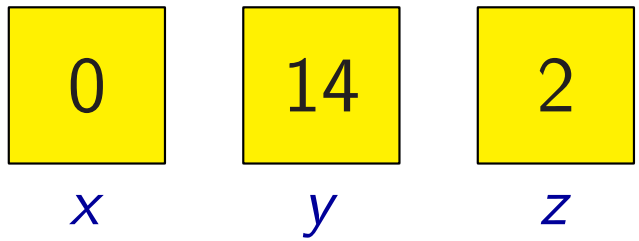
L_3 : ...



Minsky machine

Adding the value of the counter z to the counter y
(and reseting counter x to zero):

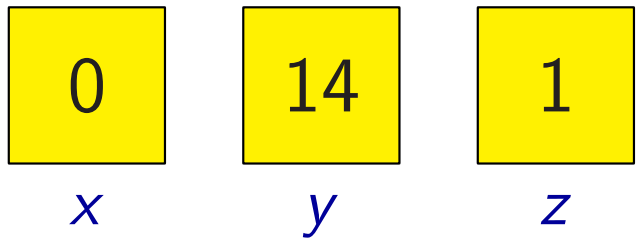
L_2 : **if** $(z = 0)$ **goto** L_3
$z := z - 1$
$y := y + 1$
goto L_1
L_3 : ...



Minsky machine

Adding the value of the counter z to the counter y
(and reseting counter x to zero):

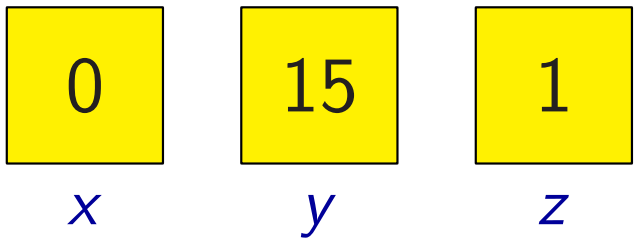
L_2 : **if** $(z = 0)$ **goto** L_3
$z := z - 1$
$y := y + 1$
goto L_1
L_3 : ...



Minsky machine

Adding the value of the counter z to the counter y
(and reseting counter x to zero):

L_2 : **if** $(z = 0)$ **goto** L_3
$z := z - 1$
$y := y + 1$
goto L_1
L_3 : ...



Minsky machine

Adding the value of the counter z to the counter y
(and reseting counter x to zero):

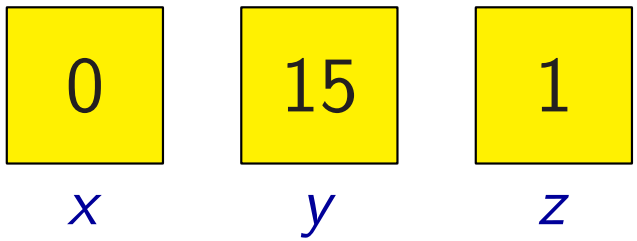
L_2 : **if** $(z = 0)$ **goto** L_3

$z := z - 1$

$y := y + 1$

goto L_1

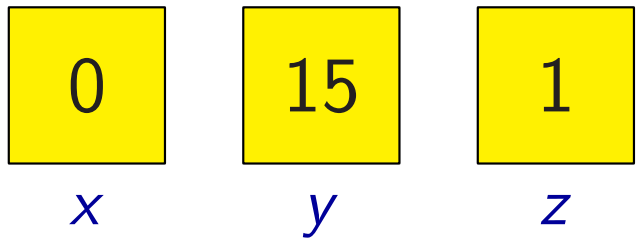
L_3 : ...



Minsky machine

Adding the value of the counter z to the counter y
(and reseting counter x to zero):

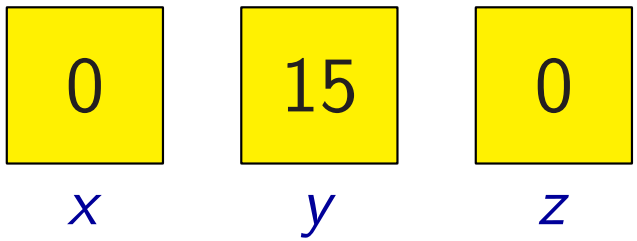
L_2 : **if** $(z = 0)$ **goto** L_3
$z := z - 1$
$y := y + 1$
goto L_1
L_3 : ...



Minsky machine

Adding the value of the counter z to the counter y
(and reseting counter x to zero):

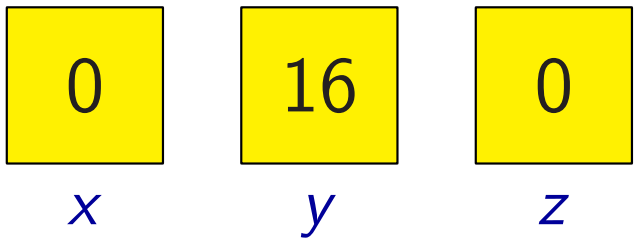
L_2 : **if** $(z = 0)$ **goto** L_3
$z := z - 1$
$y := y + 1$
goto L_1
L_3 : ...



Minsky machine

Adding the value of the counter z to the counter y
(and reseting counter x to zero):

L_2 : **if** $(z = 0)$ **goto** L_3
$z := z - 1$
$y := y + 1$
goto L_1
L_3 : ...



Minsky machine

Adding the value of the counter z to the counter y
(and reseting counter x to zero):

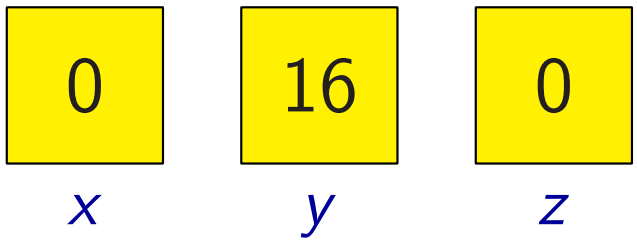
L_2 : **if** $(z = 0)$ **goto** L_3

$z := z - 1$

$y := y + 1$

goto L_1

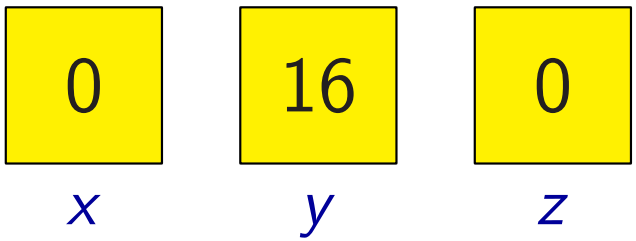
L_3 : ...



Minsky machine

Adding the value of the counter z to the counter y
(and reseting counter x to zero):

L_2 : **if** $(z = 0)$ **goto** L_3
$z := z - 1$
$y := y + 1$
goto L_1
L_3 : $\ldots$



Minsky machine

Multiplication of the value of a counter x by 5:

$$
\begin{aligned}
L_1 &: \textbf{if } (x = 0) \textbf{ goto } L_2 \\
&\quad x := x - 1 \\
&\quad y := y + 1 \\
&\quad y := y + 1 \\
&\quad y := y + 1 \\
&\quad y := y + 1 \\
&\quad y := y + 1 \\
&\quad \textbf{goto } L_1 \\
L_2 &: \textbf{if } (y = 0) \textbf{ goto } L_3 \\
&\quad y := y - 1 \\
&\quad x := x + 1 \\
&\quad \textbf{goto } L_2 \\
L_3 &: \quad \ldots
\end{aligned}
$$

Minsky machine

Division of value of the counter x by 5 and finding out the remainder of this division:

$$
\begin{aligned}
L_1 : \ & \textbf{if } (x = 0) \textbf{ goto } M_0 \\
& x := x - 1 \\
& \textbf{if } (x = 0) \textbf{ goto } M_1 \\
& x := x - 1 \\
& \textbf{if } (x = 0) \textbf{ goto } M_2 \\
& x := x - 1 \\
& \textbf{if } (x = 0) \textbf{ goto } M_3 \\
& x := x - 1 \\
& \textbf{if } (x = 0) \textbf{ goto } M_4 \\
& x := x - 1 \\
& y := y + 1 \\
& \textbf{goto } L_1
\end{aligned}
$$

Minsky machine

A stack can be simulated using a pair of counters — a value of the first counter represents the content of the stack as a number of base $k = |\Gamma| + 1$ (where Γ is a stack alphabet).

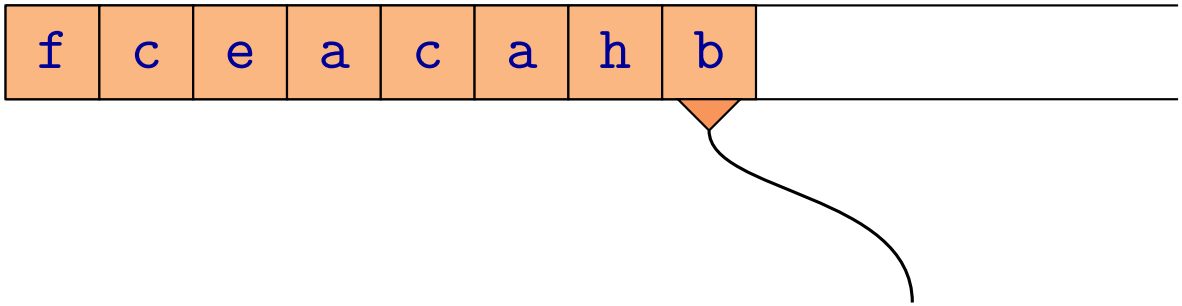
- A stack on the top of the stack — the remainder of division by k
- Pop — to divide by k
- Push — to multiply by k and to add the code of the given symbol

The second counter is used as an auxiliary variable in the implementation of above mentioned operations.

Minsky machine

Example:

a ↔ 1
b ↔ 2
c ↔ 3
d ↔ 4
e ↔ 5
f ↔ 6
g ↔ 7
h ↔ 8
i ↔ 9

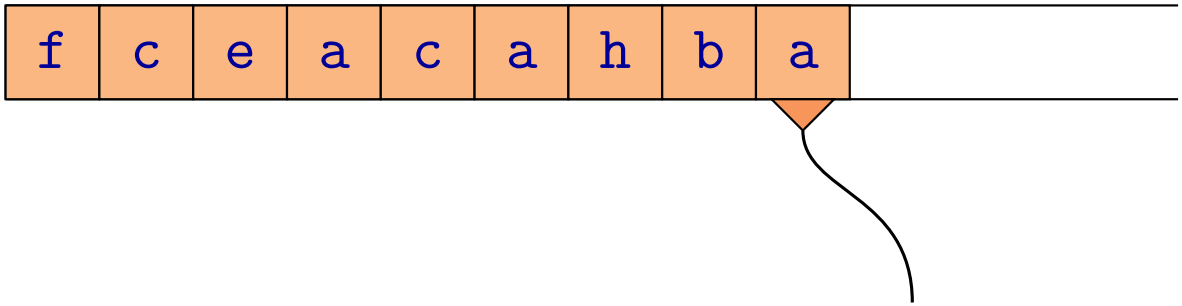


63513182

Minsky machine

Example:

a ↔ 1
b ↔ 2
c ↔ 3
d ↔ 4
e ↔ 5
f ↔ 6
g ↔ 7
h ↔ 8
i ↔ 9

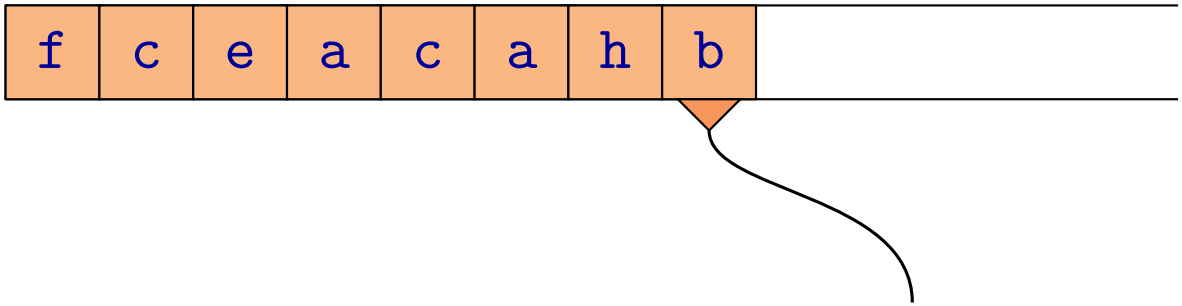


635131821

Minsky machine

Example:

a ↔ 1
b ↔ 2
c ↔ 3
d ↔ 4
e ↔ 5
f ↔ 6
g ↔ 7
h ↔ 8
i ↔ 9

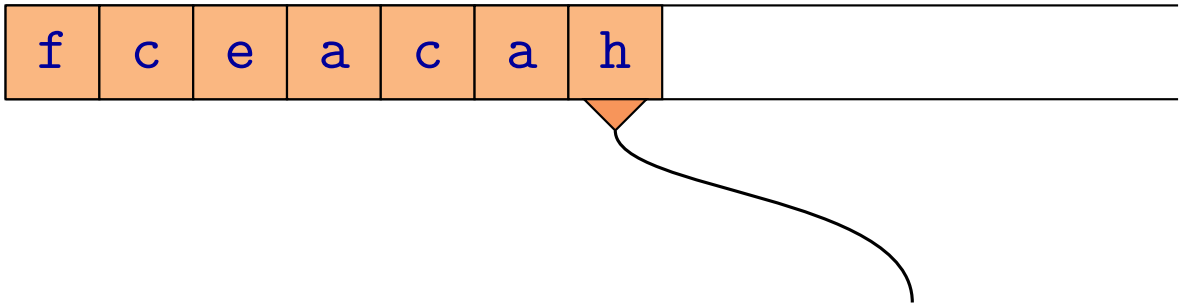


63513182

Minsky machine

Example:

a ↔ 1
b ↔ 2
c ↔ 3
d ↔ 4
e ↔ 5
f ↔ 6
g ↔ 7
h ↔ 8
i ↔ 9

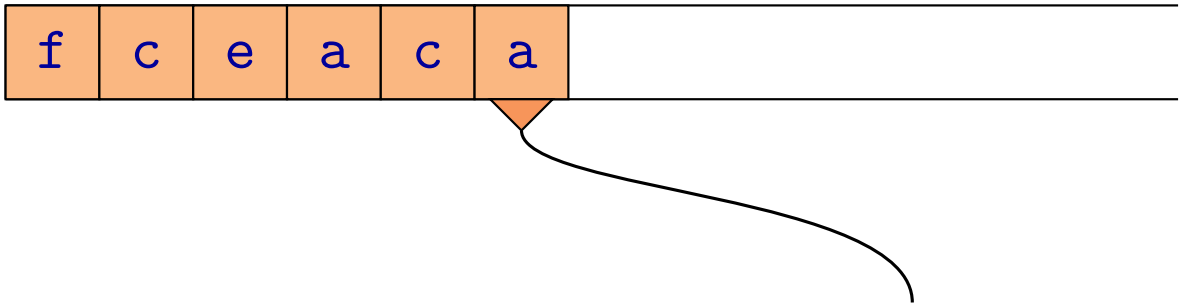


6351318

Minsky machine

Example:

a ↔ 1
b ↔ 2
c ↔ 3
d ↔ 4
e ↔ 5
f ↔ 6
g ↔ 7
h ↔ 8
i ↔ 9



635131

Minsky machine

Recall that two-side infinite tape can be simulated by a pair of stacks.

In Minsky machine, the content of each of these stacks can be represented by a corresponding counter.

Moreover, we need one additional auxiliary counter to allow implementation of operations addition and division on these counters representing contents of stacks.

We can see that a Turing machine with k tapes can be similated by Minsky machine with $2k + 1$ counters.

Minsky machine

Any finite number of counters can be simulated by two counters:

- One counter represents values of all counters — e.g., values of counters x, y, z can be represented number $2^x 3^y 5^z$.
- The second counter is used as auxiliary variable in operations of multiplication and division on counter C.
- Incrementing counter x by one is simulated as multiplying by 2, incrementing counter y by one is simulated as multiplying by 3, etc.
- In a similar way, decrementing by one is simulated by division by a corresponding constant.
- The test of condition $x = 0$ corresponds to the test that value C is not divisible by 2

Minsky machine

We can see that a computation of each Turing machine can be simulated by a Minsky machine with two counters.

However, this simulation is extremely inefficient:

- Already simulation of a tape of a Turing machine using three counters requires exponentially bigger number of steps than how many steps are performed by this Turing machine.
- Simulation of behaviour of these three counters by two counters further increments this number of steps exponentially.