
Universal Turing Machine

— Ertug Umsur & Joshua Espinoza —

Introduction to Turing Machine

Key Concepts:

- **Transition Function:** Dictates what happens based on the current state and symbol
- **Tape:** Infinite memory tape divided into cells
- **Head:** Reads and writes symbols on the tape
- **State:** The Machine's current status

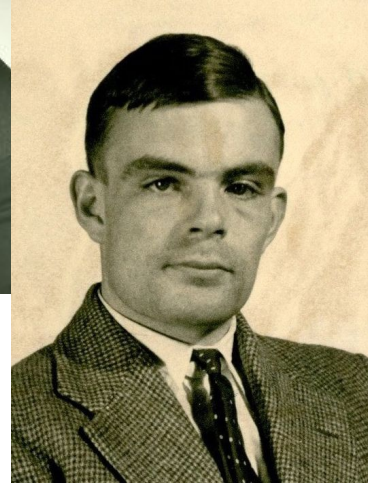


Konrad Zuse's Z3

History of the Turing Machine

First Mentioned In: Turing's 1936 paper "On Computable Numbers, with an Application to the Entscheidungsproblem"

Church-Turing Thesis: Postulates that anything computable by an algorithm can be computable by Turing Machines

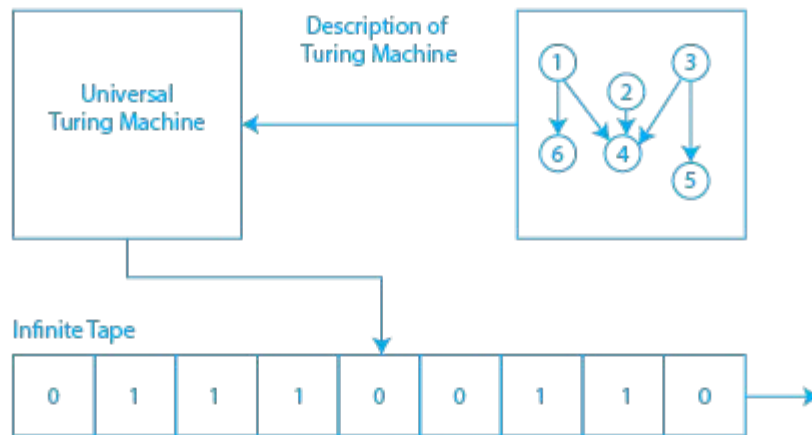


**Alan Turing &
Alonzo Church**

The Universal Turing Machine

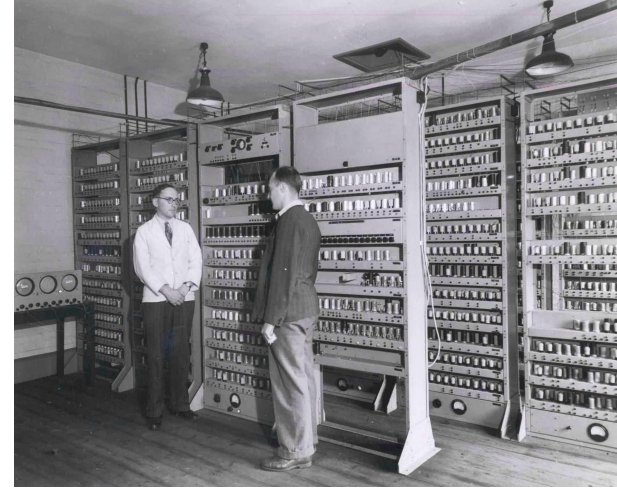
A **Universal Turing Machine** is a special type of Turing Machine that can simulate any other Turing Machine.

Normal Turing Machines are designed to perform one specific task but **UTMs** can perform any Turing Machine's tasks



The First Universal Turing Machine

- Mentioned in the same 1936 paper by Turing
- The first physical copies were made in 1950s and 60s
- In US, John von Neumann's contributions created ENIAC and EDVAC. US's first programmable computers.



Importance of Universal Turing Machines

Universal Turing Machines demonstrate the universality of the Turing Machines and it lays the foundation of **modern computers**.

Introduces the idea that all computations can be **reduced to same basic operations**, regardless of the specific algorithms.

Introduced the concept of undecidability and the **Halting Problem**

Our Universal Turing Machine

— Part I: The Encoding Scheme —

The Encoding Scheme

Our encoding $[M]$ for a TM M with input string w

$$[M] = \langle \text{current state / Buffer} \rangle 0000 \langle M \rangle 0000 \langle w \rangle$$

$$M = \{Q, \Sigma, \Gamma, \delta, q_{\text{start}}, q_{\text{acc}}, q_{\text{rej}}\}$$

$$Q = \{1, 2, 3, \dots\}$$

↑ accept state
↑ start state ↑ reject state

$$\langle Q \rangle = 101101110 \dots$$

1011 0111 10 ...
q_s q_acc q_rej

← ends with the final state (no final 0)

$$\Sigma = \{a, b, c, \dots\}$$

1 2 3
a b c

$$\langle \Sigma \rangle = 101101110 \dots$$

1011 0111 10 ...
a b c

$$\Gamma = \{\epsilon, x, y, \dots\}$$

$$\langle \Gamma \rangle = \langle \Sigma \rangle 0111 \dots 1011 \dots 10 \dots$$

0111 1011 10 ...
x y

Machine alphabet first, then the letters of the tape alphabet

$$\delta = \{p, a, q, b, \pm 1\} \dots \dots \dots$$

d₁ d₂ d₃ d₄ d₅ d₆ d₇ d₈ d₉ d₁₀ d₁₁ d₁₂ d₁₃ d₁₄ d₁₅ d₁₆ d₁₇ d₁₈ d₁₉ d₂₀ d₂₁ d₂₂ d₂₃ d₂₄ d₂₅ d₂₆ d₂₇ d₂₈ d₂₉ d₃₀ d₃₁ d₃₂ d₃₃ d₃₄ d₃₅ d₃₆ d₃₇ d₃₈ d₃₉ d₄₀ d₄₁ d₄₂ d₄₃ d₄₄ d₄₅ d₄₆ d₄₇ d₄₈ d₄₉ d₅₀ d₅₁ d₅₂ d₅₃ d₅₄ d₅₅ d₅₆ d₅₇ d₅₈ d₅₉ d₆₀ d₆₁ d₆₂ d₆₃ d₆₄ d₆₅ d₆₆ d₆₇ d₆₈ d₆₉ d₇₀ d₇₁ d₇₂ d₇₃ d₇₄ d₇₅ d₇₆ d₇₇ d₇₈ d₇₉ d₈₀ d₈₁ d₈₂ d₈₃ d₈₄ d₈₅ d₈₆ d₈₇ d₈₈ d₈₉ d₉₀ d₉₁ d₉₂ d₉₃ d₉₄ d₉₅ d₉₆ d₉₇ d₉₈ d₉₉ d₁₀₀ d₁₀₁ d₁₀₂ d₁₀₃ d₁₀₄ d₁₀₅ d₁₀₆ d₁₀₇ d₁₀₈ d₁₀₉ d₁₁₀ d₁₁₁ d₁₁₂ d₁₁₃ d₁₁₄ d₁₁₅ d₁₁₆ d₁₁₇ d₁₁₈ d₁₁₉ d₁₂₀ d₁₂₁ d₁₂₂ d₁₂₃ d₁₂₄ d₁₂₅ d₁₂₆ d₁₂₇ d₁₂₈ d₁₂₉ d₁₃₀ d₁₃₁ d₁₃₂ d₁₃₃ d₁₃₄ d₁₃₅ d₁₃₆ d₁₃₇ d₁₃₈ d₁₃₉ d₁₄₀ d₁₄₁ d₁₄₂ d₁₄₃ d₁₄₄ d₁₄₅ d₁₄₆ d₁₄₇ d₁₄₈ d₁₄₉ d₁₅₀ d₁₅₁ d₁₅₂ d₁₅₃ d₁₅₄ d₁₅₅ d₁₅₆ d₁₅₇ d₁₅₈ d₁₅₉ d₁₆₀ d₁₆₁ d₁₆₂ d₁₆₃ d₁₆₄ d₁₆₅ d₁₆₆ d₁₆₇ d₁₆₈ d₁₆₉ d₁₇₀ d₁₇₁ d₁₇₂ d₁₇₃ d₁₇₄ d₁₇₅ d₁₇₆ d₁₇₇ d₁₇₈ d₁₇₉ d₁₈₀ d₁₈₁ d₁₈₂ d₁₈₃ d₁₈₄ d₁₈₅ d₁₈₆ d₁₈₇ d₁₈₈ d₁₈₉ d₁₉₀ d₁₉₁ d₁₉₂ d₁₉₃ d₁₉₄ d₁₉₅ d₁₉₆ d₁₉₇ d₁₉₈ d₁₉₉ d₂₀₀ d₂₀₁ d₂₀₂ d₂₀₃ d₂₀₄ d₂₀₅ d₂₀₆ d₂₀₇ d₂₀₈ d₂₀₉ d₂₁₀ d₂₁₁ d₂₁₂ d₂₁₃ d₂₁₄ d₂₁₅ d₂₁₆ d₂₁₇ d₂₁₈ d₂₁₉ d₂₂₀ d₂₂₁ d₂₂₂ d₂₂₃ d₂₂₄ d₂₂₅ d₂₂₆ d₂₂₇ d₂₂₈ d₂₂₉ d₂₃₀ d₂₃₁ d₂₃₂ d₂₃₃ d₂₃₄ d₂₃₅ d₂₃₆ d₂₃₇ d₂₃₈ d₂₃₉ d₂₄₀ d₂₄₁ d₂₄₂ d₂₄₃ d₂₄₄ d₂₄₅ d₂₄₆ d₂₄₇ d₂₄₈ d₂₄₉ d₂₅₀ d₂₅₁ d₂₅₂ d₂₅₃ d₂₅₄ d₂₅₅ d₂₅₆ d₂₅₇ d₂₅₈ d₂₅₉ d₂₆₀ d₂₆₁ d₂₆₂ d₂₆₃ d₂₆₄ d₂₆₅ d₂₆₆ d₂₆₇ d₂₆₈ d₂₆₉ d₂₇₀ d₂₇₁ d₂₇₂ d₂₇₃ d₂₇₄ d₂₇₅ d₂₇₆ d₂₇₇ d₂₇₈ d₂₇₉ d₂₈₀ d₂₈₁ d₂₈₂ d₂₈₃ d₂₈₄ d₂₈₅ d₂₈₆ d₂₈₇ d₂₈₈ d₂₈₉ d₂₉₀ d₂₉₁ d₂₉₂ d₂₉₃ d₂₉₄ d₂₉₅ d₂₉₆ d₂₉₇ d₂₉₈ d₂₉₉ d₃₀₀ d₃₀₁ d₃₀₂ d₃₀₃ d₃₀₄ d₃₀₅ d₃₀₆ d₃₀₇ d₃₀₈ d₃₀₉ d₃₁₀ d₃₁₁ d₃₁₂ d₃₁₃ d₃₁₄ d₃₁₅ d₃₁₆ d₃₁₇ d₃₁₈ d₃₁₉ d₃₂₀ d₃₂₁ d₃₂₂ d₃₂₃ d₃₂₄ d₃₂₅ d₃₂₆ d₃₂₇ d₃₂₈ d₃₂₉ d₃₃₀ d₃₃₁ d₃₃₂ d₃₃₃ d₃₃₄ d₃₃₅ d₃₃₆ d₃₃₇ d₃₃₈ d₃₃₉ d₃₄₀ d₃₄₁ d₃₄₂ d₃₄₃ d₃₄₄ d₃₄₅ d₃₄₆ d₃₄₇ d₃₄₈ d₃₄₉ d₃₅₀ d₃₅₁ d₃₅₂ d₃₅₃ d₃₅₄ d₃₅₅ d₃₅₆ d₃₅₇ d₃₅₈ d₃₅₉ d₃₆₀ d₃₆₁ d₃₆₂ d₃₆₃ d₃₆₄ d₃₆₅ d₃₆₆ d₃₆₇ d₃₆₈ d₃₆₉ d₃₇₀ d₃₇₁ d₃₇₂ d₃₇₃ d₃₇₄ d₃₇₅ d₃₇₆ d₃₇₇ d₃₇₈ d₃₇₉ d₃₈₀ d₃₈₁ d₃₈₂ d₃₈₃ d₃₈₄ d₃₈₅ d₃₈₆ d₃₈₇ d₃₈₈ d₃₈₉ d₃₉₀ d₃₉₁ d₃₉₂ d₃₉₃ d₃₉₄ d₃₉₅ d₃₉₆ d₃₉₇ d₃₉₈ d₃₉₉ d₄₀₀ d₄₀₁ d₄₀₂ d₄₀₃ d₄₀₄ d₄₀₅ d₄₀₆ d₄₀₇ d₄₀₈ d₄₀₉ d₄₁₀ d₄₁₁ d₄₁₂ d₄₁₃ d₄₁₄ d₄₁₅ d₄₁₆ d₄₁₇ d₄₁₈ d₄₁₉ d₄₂₀ d₄₂₁ d₄₂₂ d₄₂₃ d₄₂₄ d₄₂₅ d₄₂₆ d₄₂₇ d₄₂₈ d₄₂₉ d₄₃₀ d₄₃₁ d₄₃₂ d₄₃₃ d₄₃₄ d₄₃₅ d₄₃₆ d₄₃₇ d₄₃₈ d₄₃₉ d₄₄₀ d₄₄₁ d₄₄₂ d₄₄₃ d₄₄₄ d₄₄₅ d₄₄₆ d₄₄₇ d₄₄₈ d₄₄₉ d₄₅₀ d₄₅₁ d₄₅₂ d₄₅₃ d₄₅₄ d₄₅₅ d₄₅₆ d₄₅₇ d₄₅₈ d₄₅₉ d₄₆₀ d₄₆₁ d₄₆₂ d₄₆₃ d₄₆₄ d₄₆₅ d₄₆₆ d₄₆₇ d₄₆₈ d₄₆₉ d₄₇₀ d₄₇₁ d₄₇₂ d₄₇₃ d₄₇₄ d₄₇₅ d₄₇₆ d₄₇₇ d₄₇₈ d₄₇₉ d₄₈₀ d₄₈₁ d₄₈₂ d₄₈₃ d₄₈₄ d₄₈₅ d₄₈₆ d₄₈₇ d₄₈₈ d₄₈₉ d₄₉₀ d₄₉₁ d₄₉₂ d₄₉₃ d₄₉₄ d₄₉₅ d₄₉₆ d₄₉₇ d₄₉₈ d₄₉₉ d₅₀₀ d₅₀₁ d₅₀₂ d₅₀₃ d₅₀₄ d₅₀₅ d₅₀₆ d₅₀₇ d₅₀₈ d₅₀₉ d₅₁₀ d₅₁₁ d₅₁₂ d₅₁₃ d₅₁₄ d₅₁₅ d₅₁₆ d₅₁₇ d₅₁₈ d₅₁₉ d₅₂₀ d₅₂₁ d₅₂₂ d₅₂₃ d₅₂₄ d₅₂₅ d₅₂₆ d₅₂₇ d₅₂₈ d₅₂₉ d₅₃₀ d₅₃₁ d₅₃₂ d₅₃₃ d₅₃₄ d₅₃₅ d₅₃₆ d₅₃₇ d₅₃₈ d₅₃₉ d₅₄₀ d₅₄₁ d₅₄₂ d₅₄₃ d₅₄₄ d₅₄₅ d₅₄₆ d₅₄₇ d₅₄₈ d₅₄₉ d₅₅₀ d₅₅₁ d₅₅₂ d₅₅₃ d₅₅₄ d₅₅₅ d₅₅₆ d₅₅₇ d₅₅₈ d₅₅₉ d₅₆₀ d₅₆₁ d₅₆₂ d₅₆₃ d₅₆₄ d₅₆₅ d₅₆₆ d₅₆₇ d₅₆₈ d₅₆₉

Demonstration

How Does It Work?

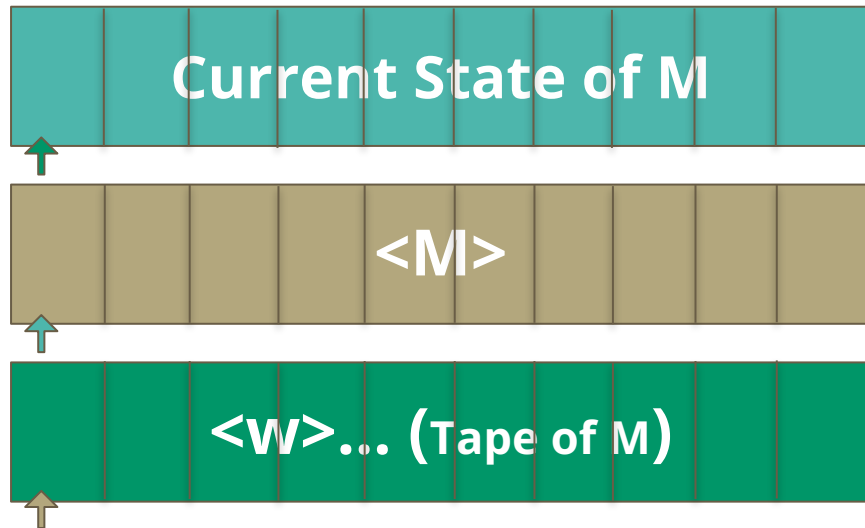
UTM Overview

Base scenario: 3-Tape UTM.

Given a Turing Machine M , develop a UTM to simulate it using 3 tapes.

In a 3-Tape UTM, **tape 1** is allocated for the **current state of M** . **Tape 2** is for the TM **M encoding** ($\langle M \rangle$).

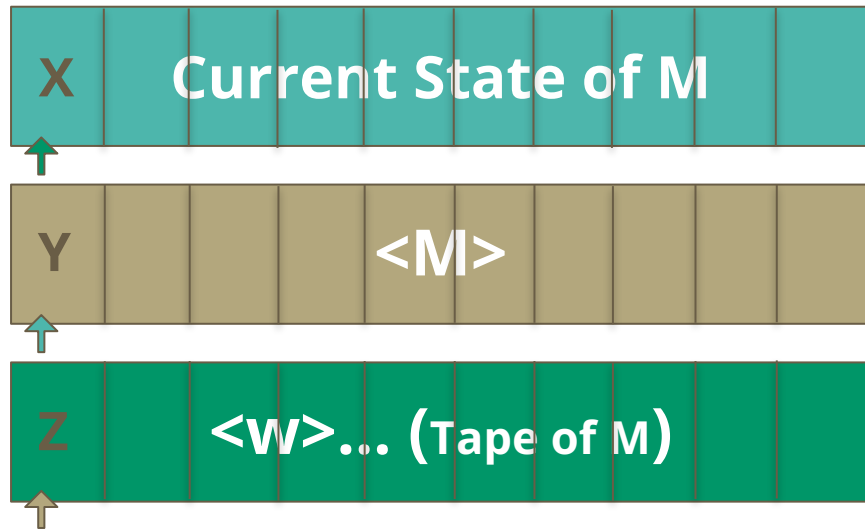
Tape 3 is for the string w (also the **tape of M**).



3-Tape to 1 Tape UTM transformation

- a) The **tape heads** will be **replaced** with special tape symbols

The symbols **X**, **Y** and **Z** of the tape alphabet for the 1 tape UTM will serve as the tape head of the three tapes.

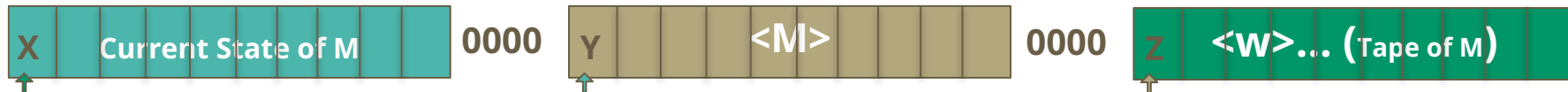


* We will use the word pointer to refer to these symbols.

3-Tape to 1 Tape UTM transformation

- b) The three tapes will be **concatenated** and a series of 0000 will be added to separate each tape.

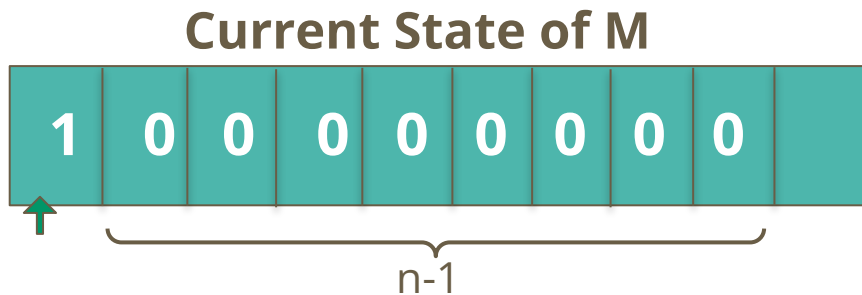
This is a process was performed in the encoding of M



3-Tape to 1 Tape UTM transformation

c) For simplicity, the tape 1 will start with a 1 and a sequence of $n-1$ 0's.

Where n is the state with the greatest value for the encoding. This will allow the tape 1 to hold any state without requiring to move the contents of the UTM tape.



* For this reason, we will call tape 1 also as the buffer to hold the current state of M.

Gadgets to build the UTM

We developed small portions of a TM to simplify the process of implementing the 1-Tape UTM

Definitions:

A,B: Pointers (the actions apply to the sequence of 1's next to them).

a,b: Symbols (the actions apply to the symbol itself).

P: Placeholder symbol

C: Any other symbol that is not A, B, P, 1 or 0

Symbology:



End state of a gadget that can be connected to the rest of the TM



Normal connections with a different color to differentiate when multiple lines cross

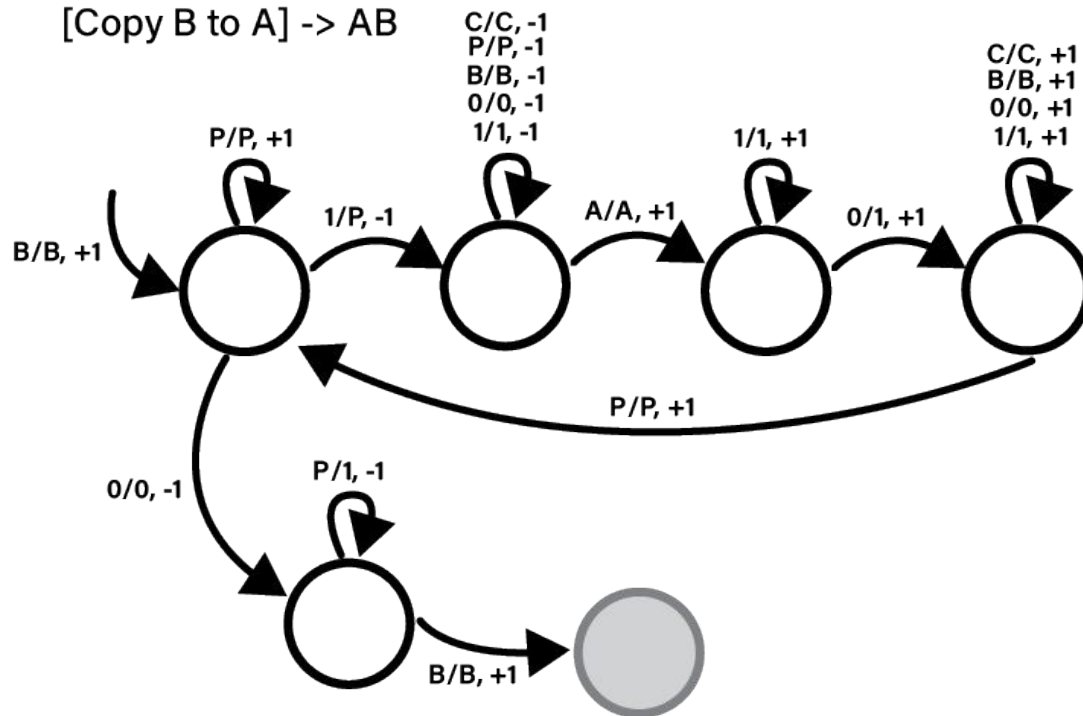


Expanded form of the gadget

Gadgets

Copy Machine

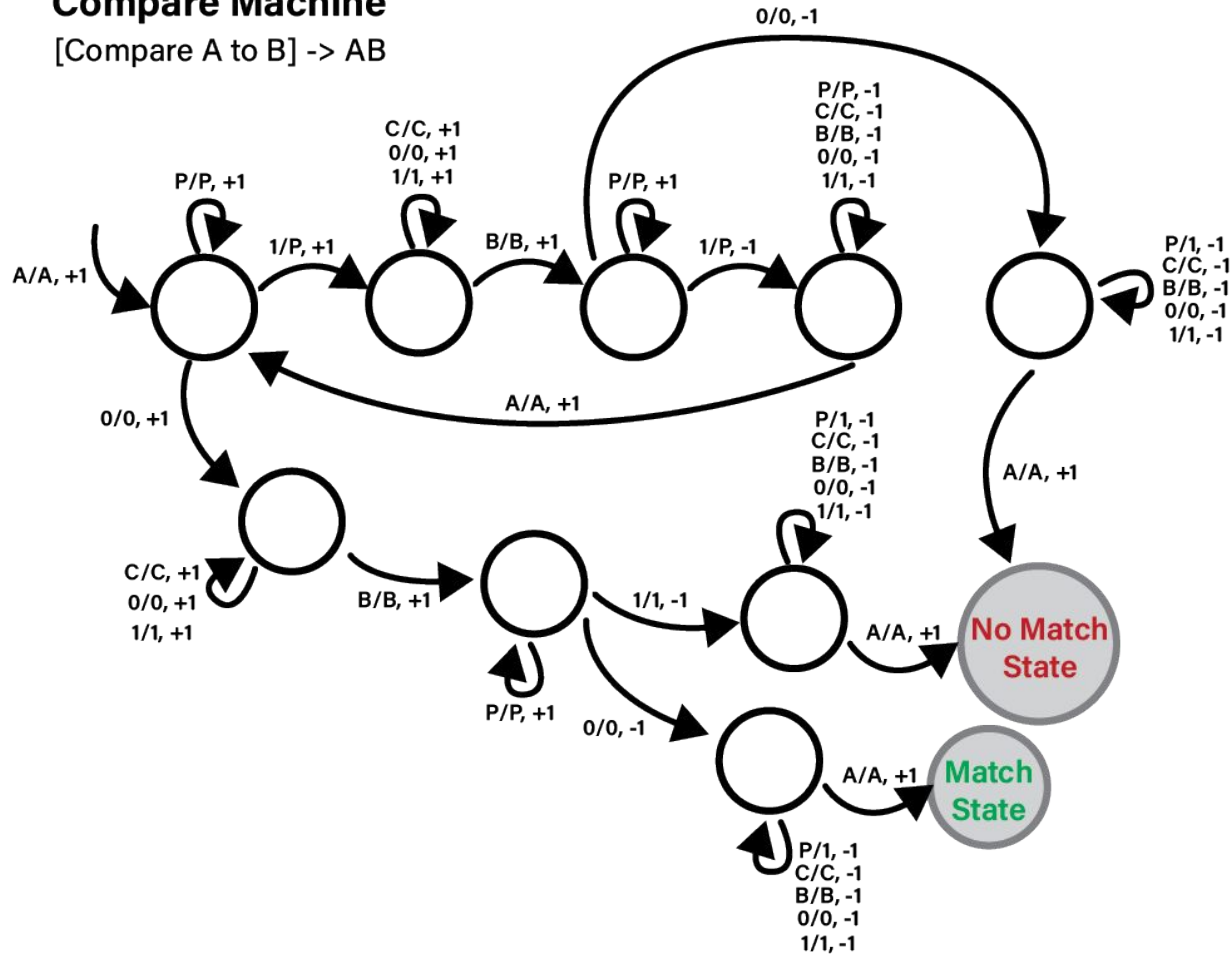
[Copy B to A] -> AB



* Used to copy the elements of a pointer B to the pointer A (Assumes that A has enough 0's to allocate B)

Compare Machine

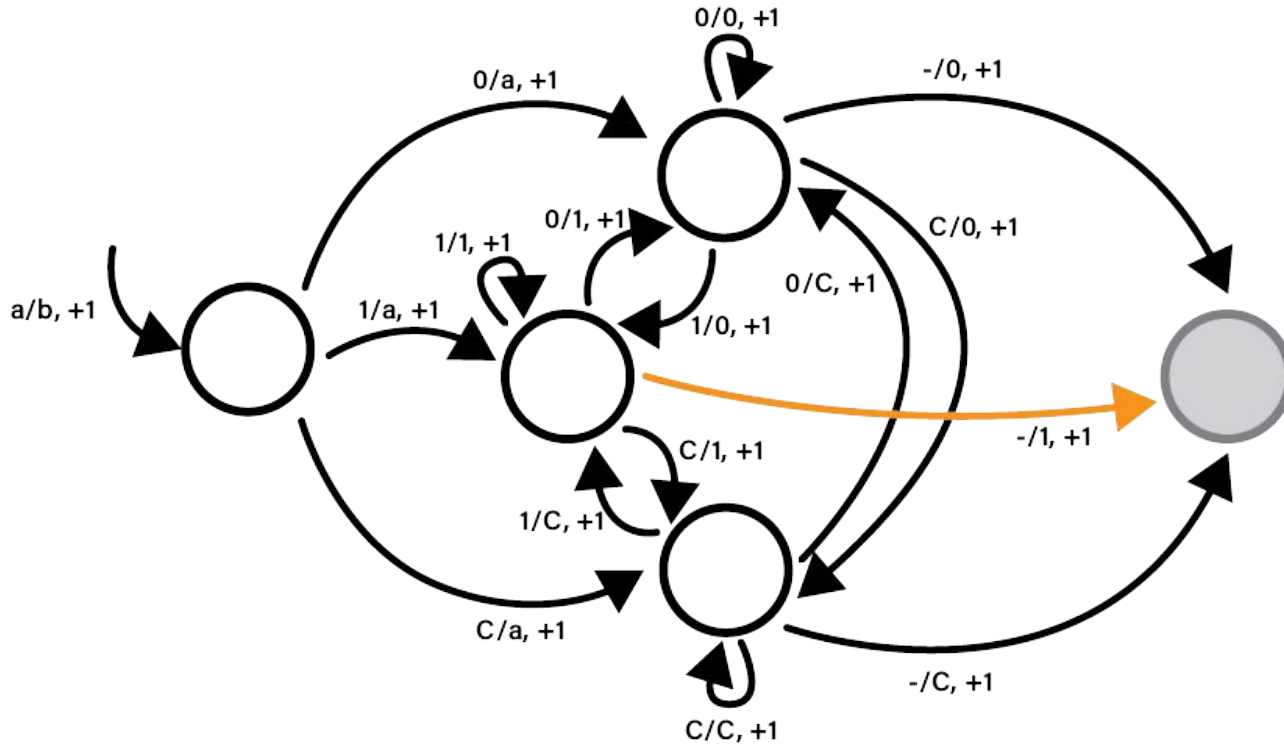
[Compare A to B] -> AB



Gadgets

* Compare the content of pointer A with pointer B

[Insert symbol b next to a] -> ba, pointer starts on a



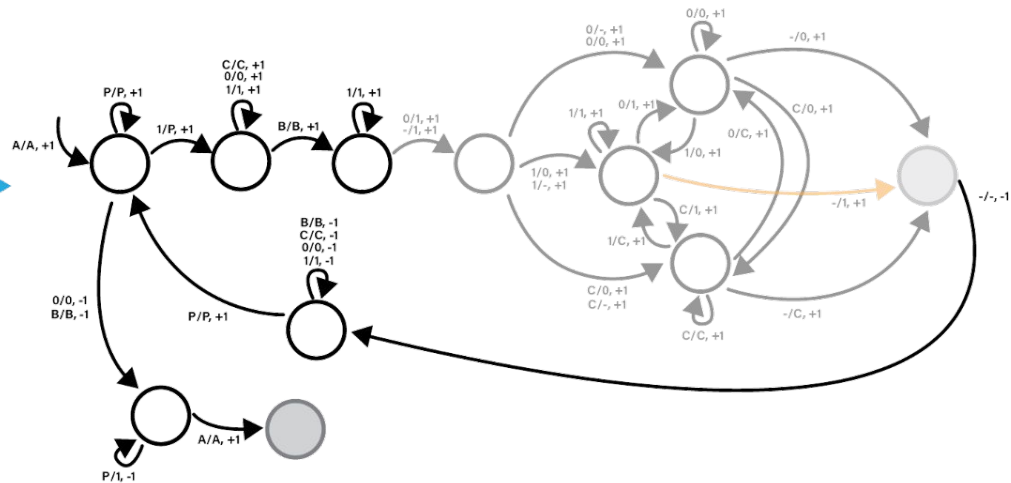
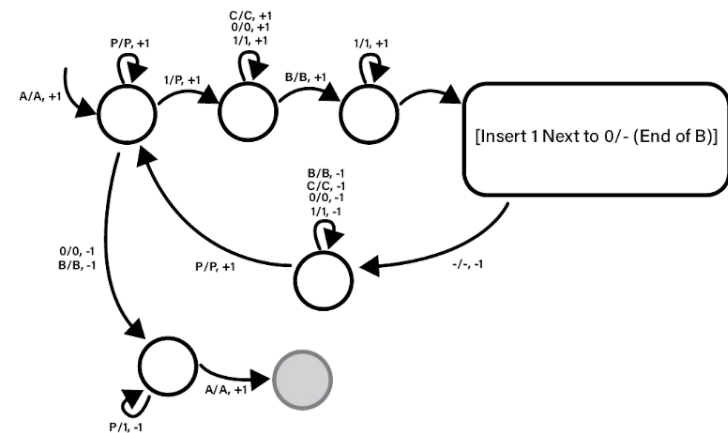
Gadgets

* Inserts symbol b into the tape at the position a, then moves the rest of the tape 1 cell to the right.

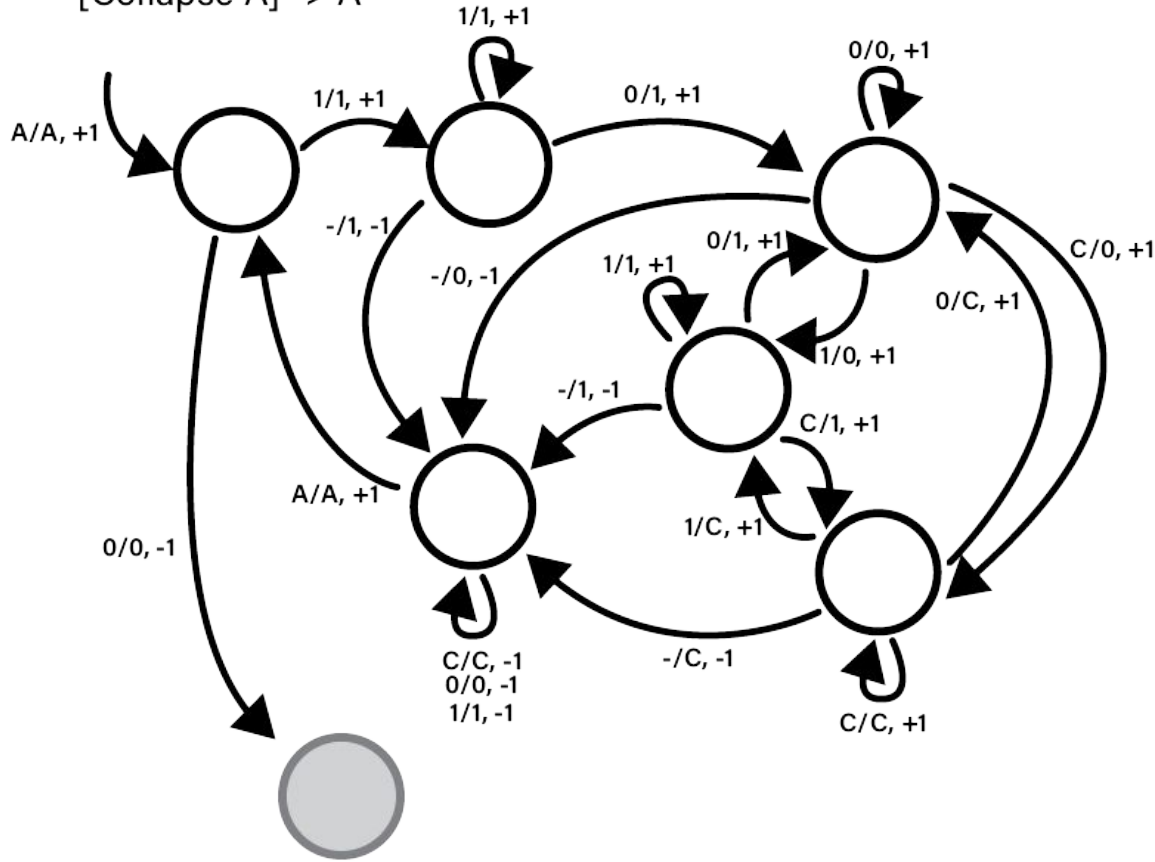
Gadgets

Inserts the contents of pointer A into pointer B

[Insert A to B] -> AB



[Collapse A] \rightarrow A



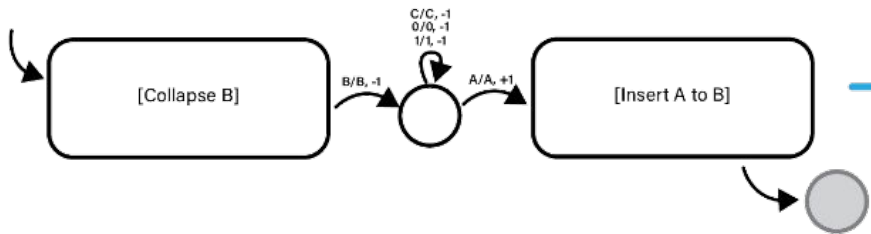
Gadgets

* Deletes the string of 1's next to pointer A (If there is one)

Gadgets

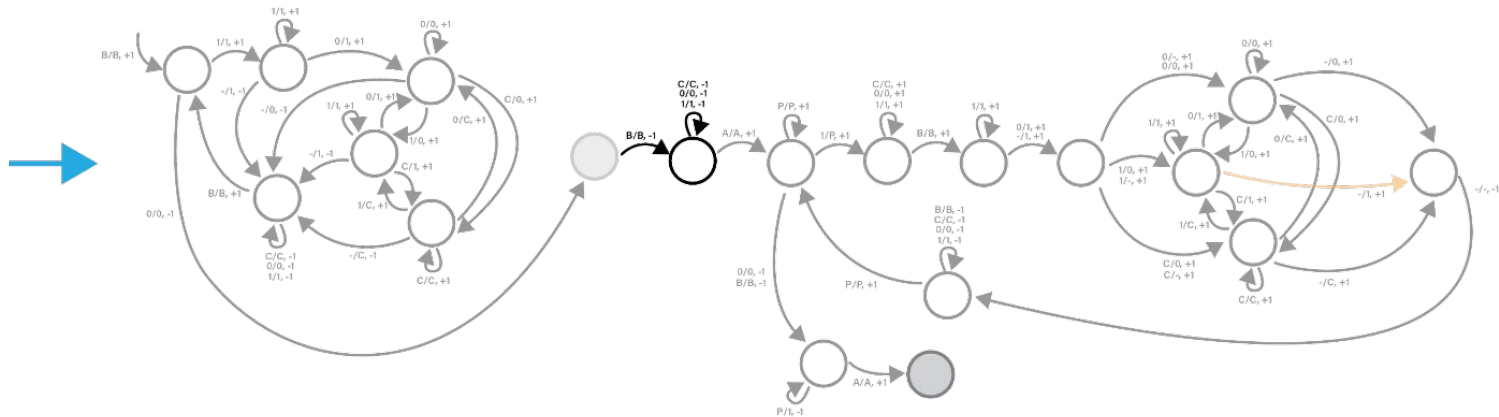
Replace Machine

[Replace B for A] -> AB, pointer starts on B



*** Replaces the content of pointer B with the content in pointer A.**

First it collapses pointer B and then inserts the content of pointer A into pointer B

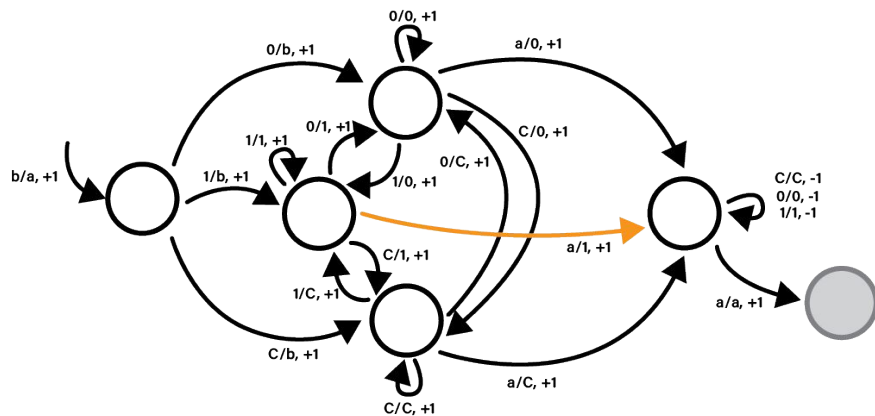


Gadgets

Moves the pointer (symbol) to a designated position

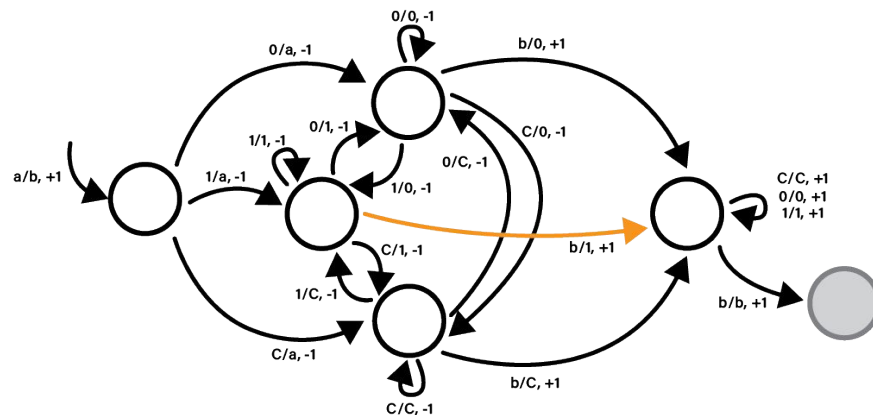
Move Pointer A to the Left

[Move a to position b] \rightarrow ba



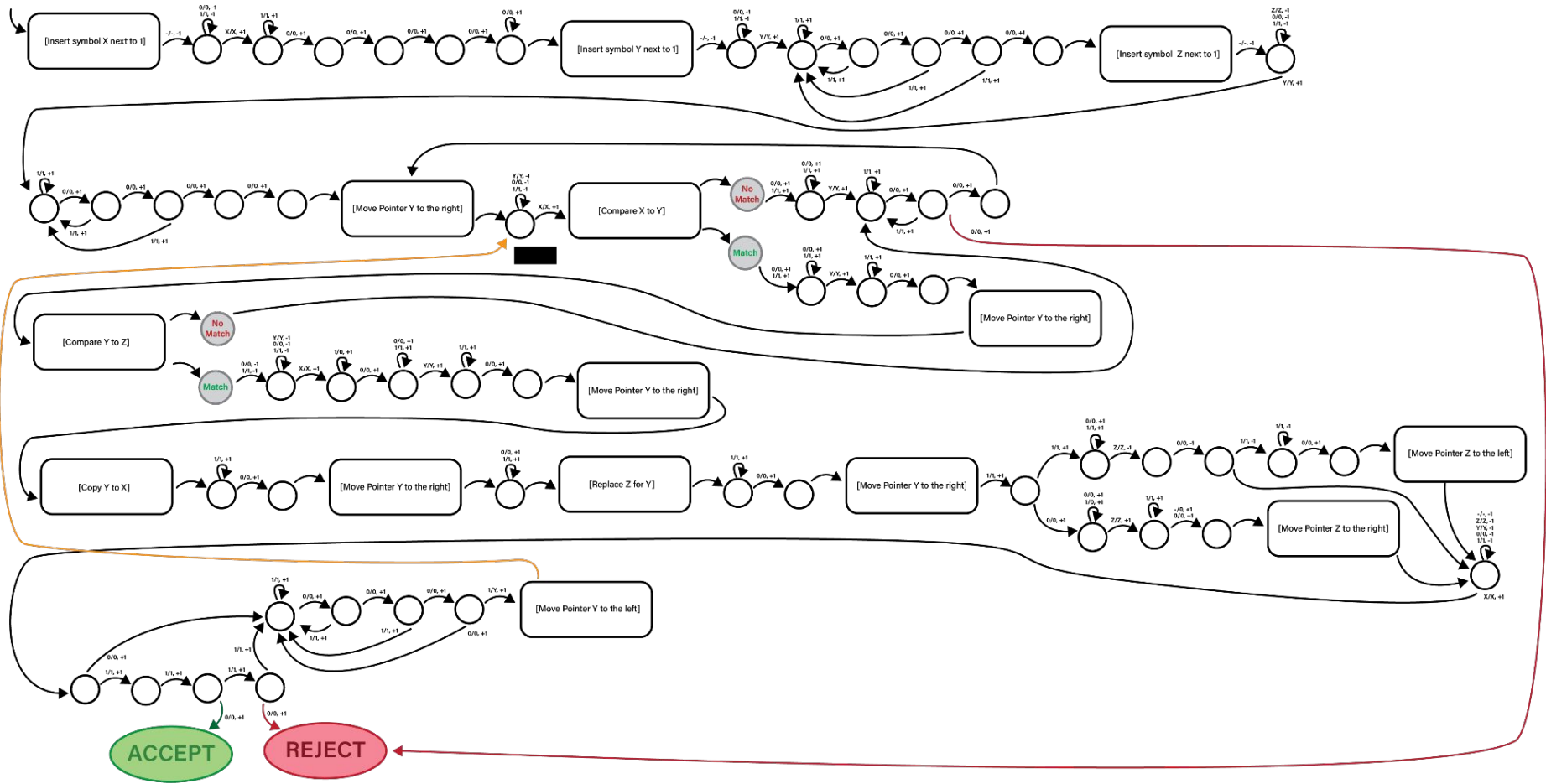
Move Pointer B to the Right

[Move b to position a] \rightarrow ba

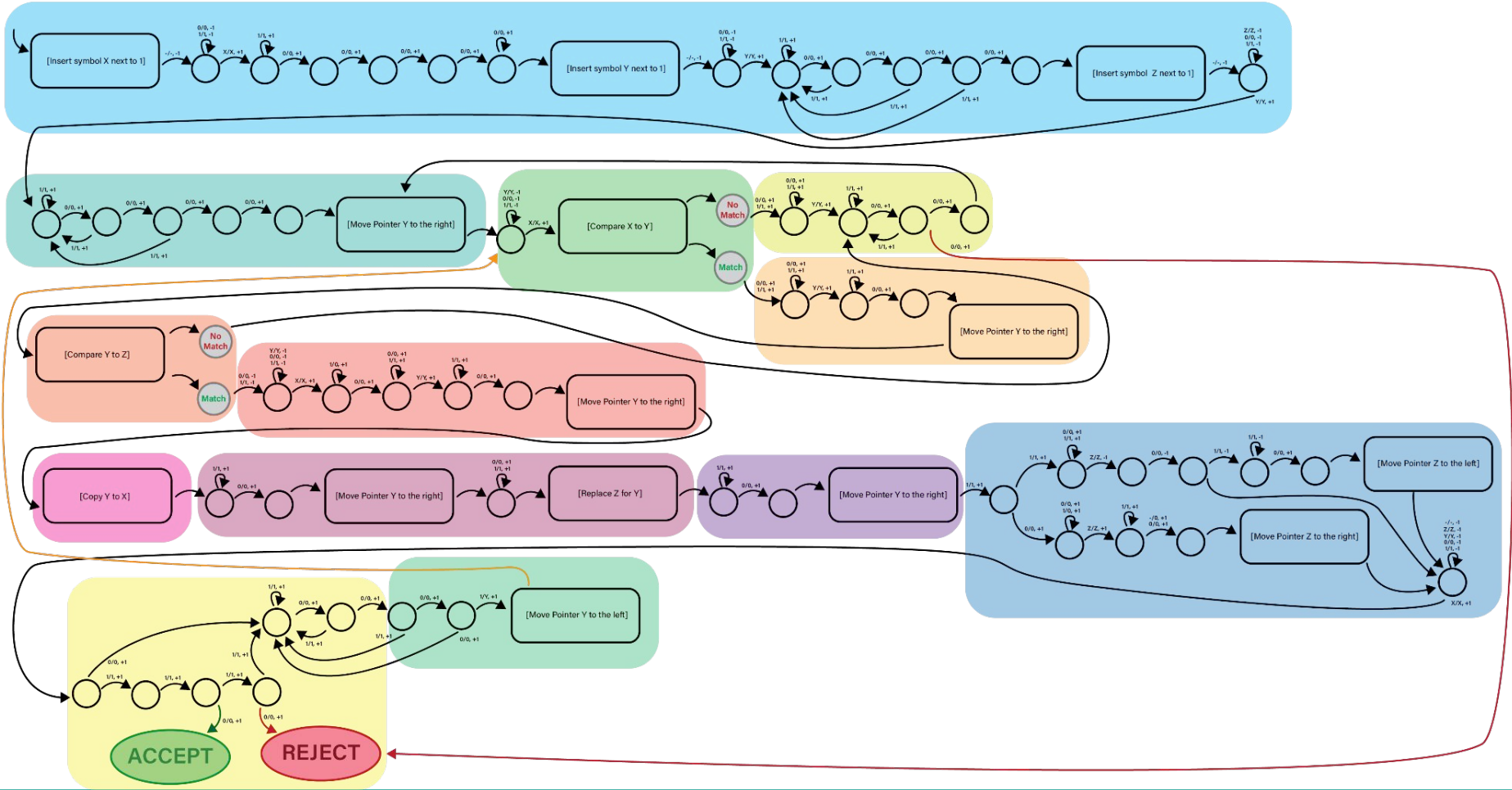


1-Tape UTM Implementation

Single Tape Universal Turing Machine

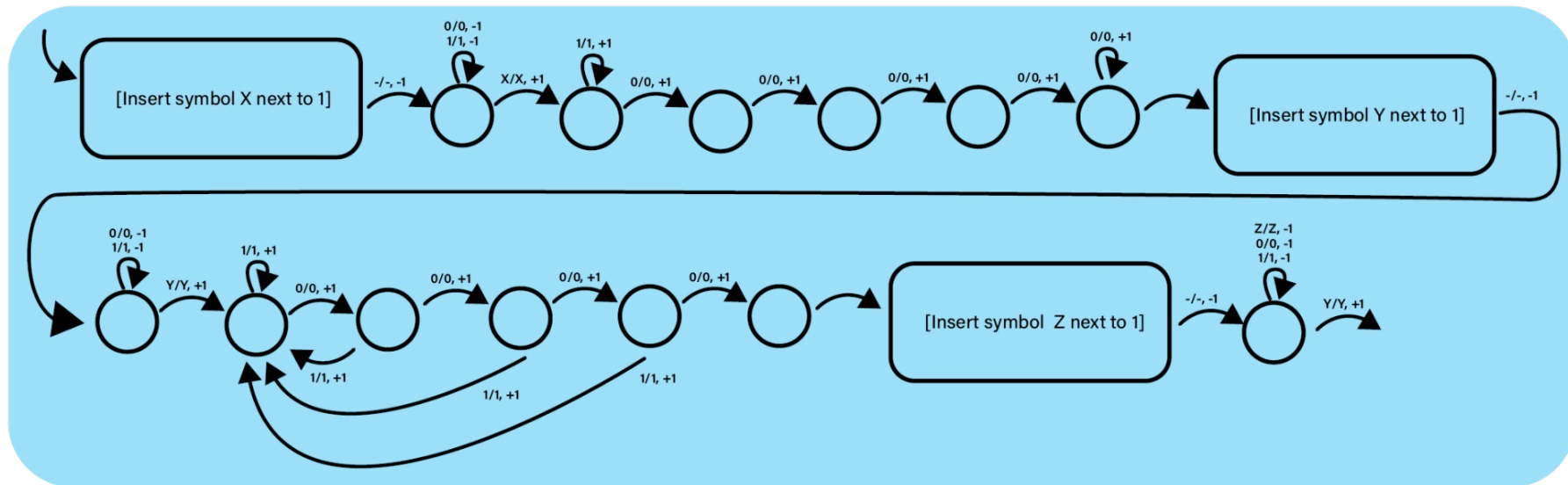


Single Tape Universal Turing Machine



UTM Algorithm

1. Add the pointers to the encoding of the Turing Machine M.

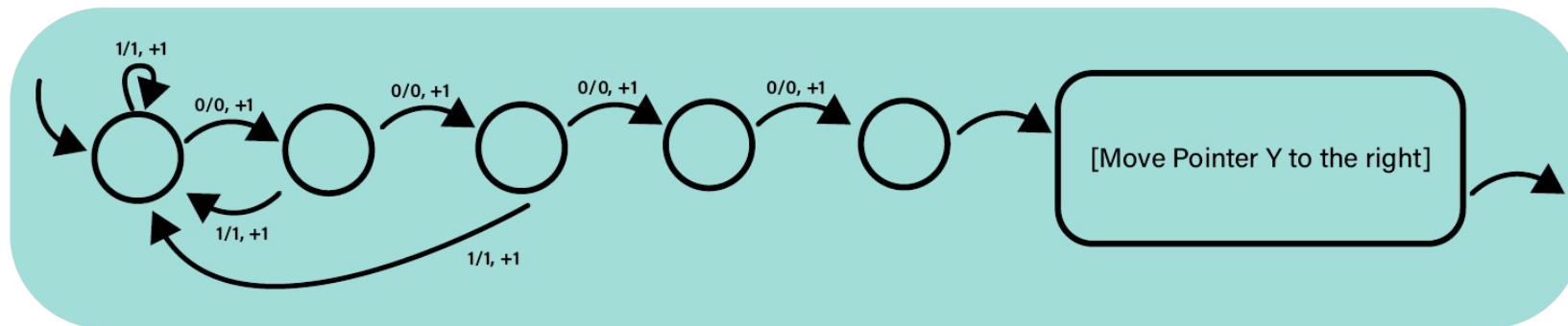


Example tape:

\leq buffer> 0000 <M> 0000 <w>
 \underline{X} <buffer> 0000 <M> 0000 <w>
X<buffer> 0000 \underline{Y} <M> 0000 <w>
X<buffer> 0000 Y<M> 0000 \underline{Z} <w>

UTM Algorithm

2. Move the Y pointer to the first transition in the transition function "delta."



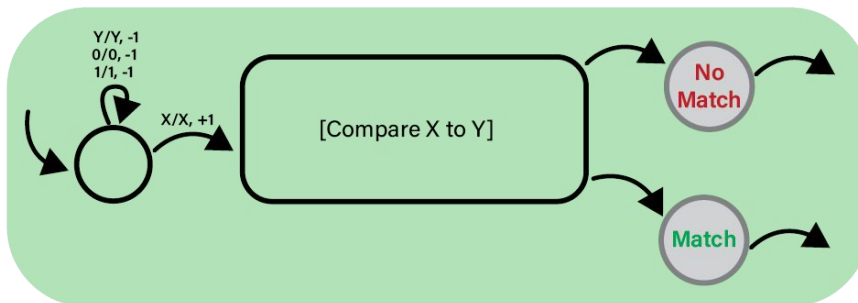
Example tape:

X<buffer> 0000 Y<Q> 00 < Σ > 00 < Γ > 000 < δ > 000 <q_s> 00 <q_{acc}> 00 <q_{rej}> 0000 Z<w>
X<buffer> 0000 Y<Q> < Σ > < Γ > 000 p 0 <a> 0 <q> 0 0 < ± 1 > 00 <p> 0 <a> 0 <q> 0 0 < ± 1 > ...
X<buffer> 0000 <Q> < Σ > < Γ > 000 <p> 0 <a> 0 <q> 0 0 < ± 1 > 00 <p> 0 <a> 0 <q> 0 0 < ± 1 > ...
X<buffer> 0000 <Q> < Σ > < Γ > 000 Y<p> 0 <a> 0 <q> 0 0 < ± 1 > 00 <p> 0 <a> 0 <q> 0 0 < ± 1 > ...

UTM Algorithm

3. Return to X and compare X and Y. If they are the same go to step 5 If they are not, go to step 4.

Compare the current state of M with the state p in the first transition of "delta."

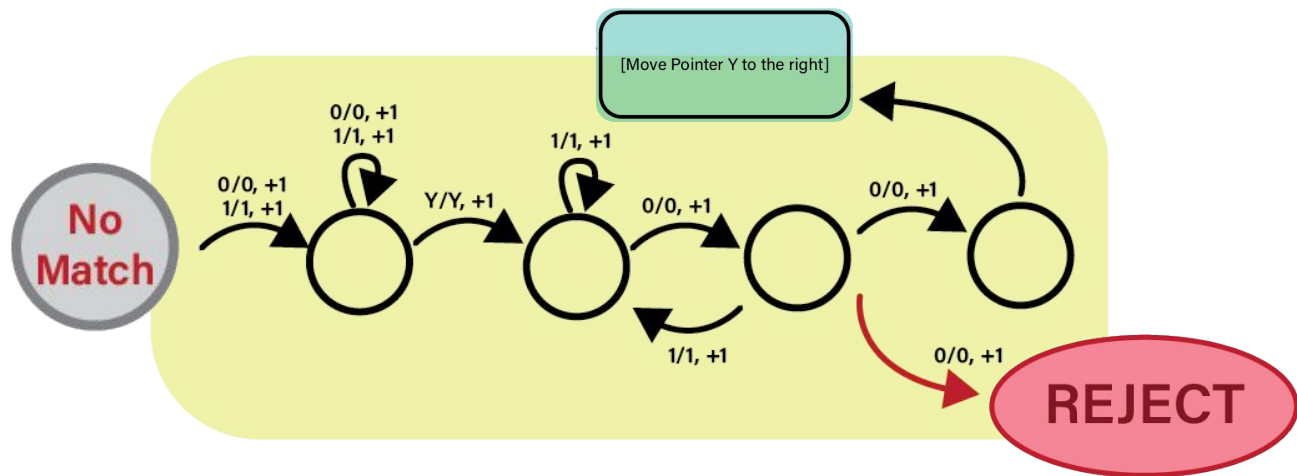


Example tape:

X<buffer> 0000 <Q> < Σ > < Γ > 000 Y<p> 0 <a> 0 <q> 0 0 < ± 1 > ...
X<buffer> 0000 <Q> < Σ > < Γ > 000 Y<p> 0 <a> 0 <q> 0 0 < ± 1 > ...
X<buffer> 0000 <Q> < Σ > < Γ > 000 Y<p> 0 <a> 0 <q> 0 0 < ± 1 > ...
X<buffer> 0000 <Q> < Σ > < Γ > 000 Y<p> 0 <a> 0 <q> 0 0 < ± 1 > ...
X<buffer> 0000 <Q> < Σ > < Γ > 000 Y<p> 0 <a> 0 <q> 0 0 < ± 1 > ...

UTM Algorithm

4. Move the Y pointer to the next transition of delta and return to step 3. If there are no transitions left, reject the string.



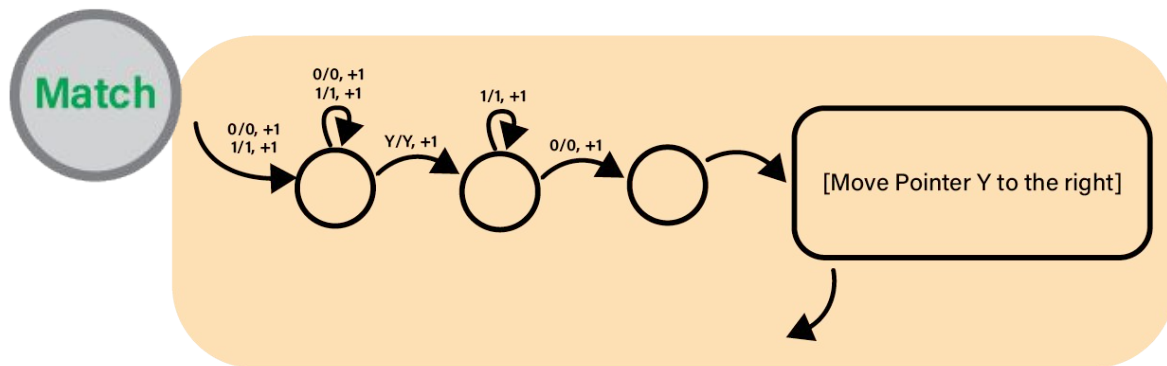
Example tape:

X<buffer> 0000 <Q> <Σ> <Γ> 000 Y<p> 0 <a> 0 <q> 0 0 <±1> 00 <p> 0 <a> 0 <q> 0 0 <±1> ...

X<buffer> 0000 <Q> <Σ> <Γ> 000 Y<p> 0 <a> 0 <q> 0 0 <±1> 00 ±p> 0 <a> 0 <q> 0 0 <±1> ...

UTM Algorithm

5. Move the Y pointer to the next item in the current transition (symbol a).



Example tape:

X<buffer> 0000 <Q> <Σ> <Γ> 000 Y<p> 0 <a> 0 <q> 0 0 <±1> 00 <p> 0 <a> 0 <q> 0 0 <±1> ...

X<buffer> 0000 <Q> <Σ> <Γ> 000 Y<p> 0 <a> 0 <q> 0 0 <±1> 00 <p> 0 <a> 0 <q> 0 0 <±1> ...

X<buffer> 0000 <Q> <Σ> <Γ> 000 Y<p> Y<a> 0 <q> 0 0 <±1> 00 <p> 0 <a> 0 <q> 0 0 <±1> ...

X<buffer> 0000 <Q> <Σ> <Γ> 000 Y<p> 0 Y<a> 0 <q> 0 0 <±1> 00 <p> 0 <a> 0 <q> 0 0 <±1> ...

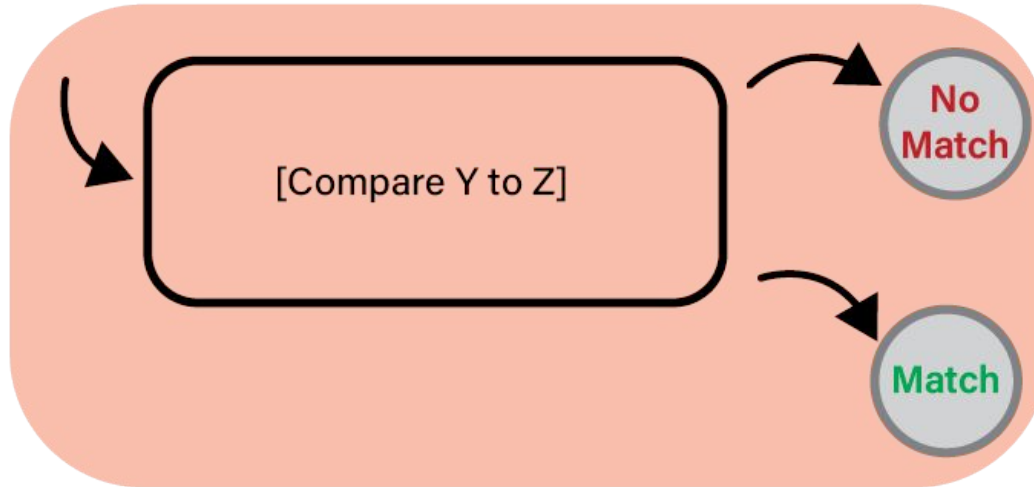
...

X<buffer> 0000 <Q> <Σ> <Γ> 000 <p> 0 Y<a> 0 <q> 0 0 <±1> 00 <p> 0 <a> 0 <q> 0 0 <±1> ...

UTM Algorithm

6. Compare Y and Z. If they are the same, go to step 7, if they are not, return to step 4.

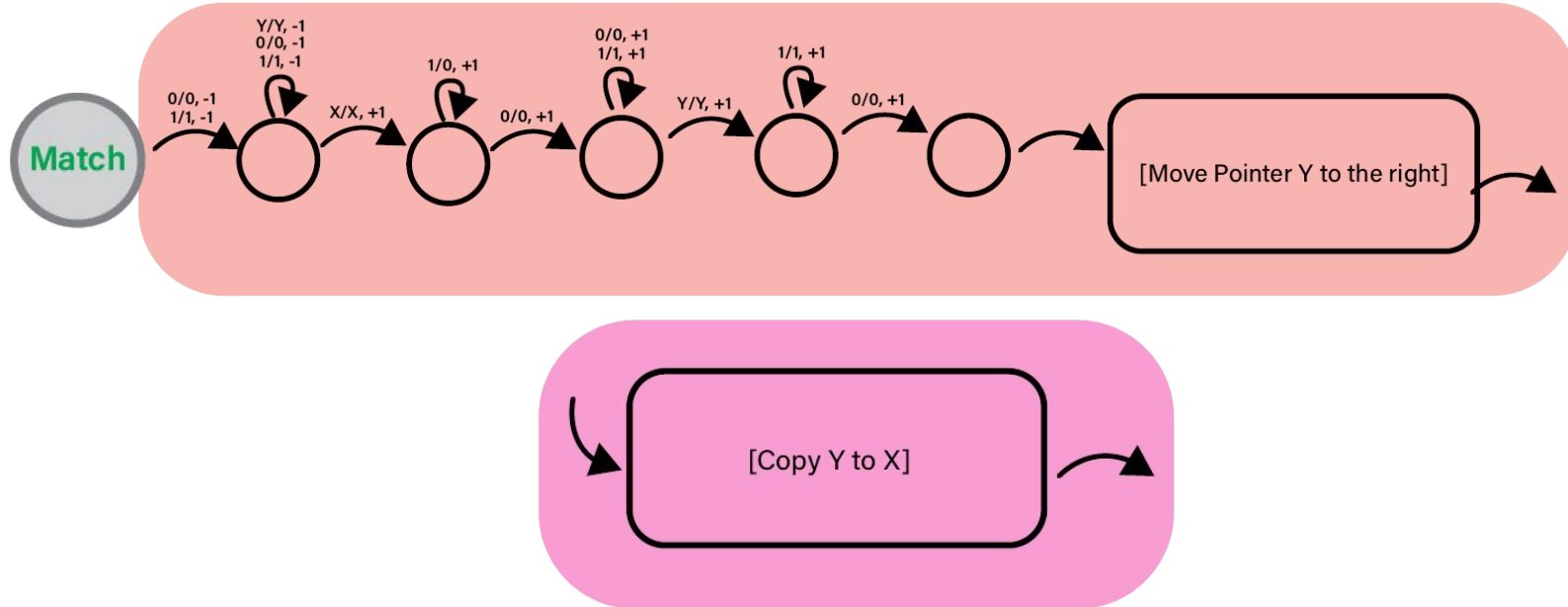
Compare the current symbol in the tape of M with the symbol in the transition



UTM Algorithm

7. Empty the buffer in X (turn the current state to a string of 0's) and move the Y pointer to the next item in the current transition (state q). Then, copy Y to X

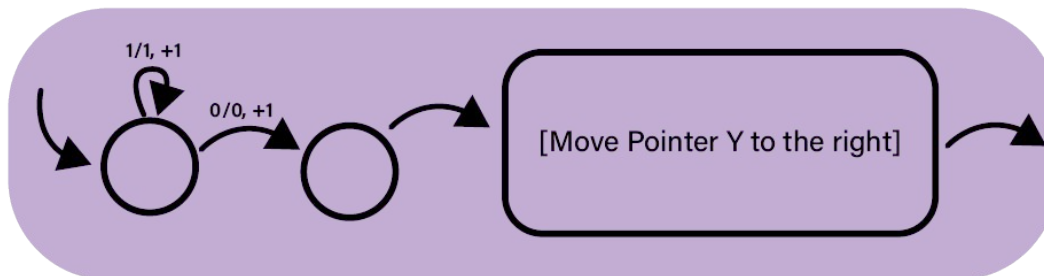
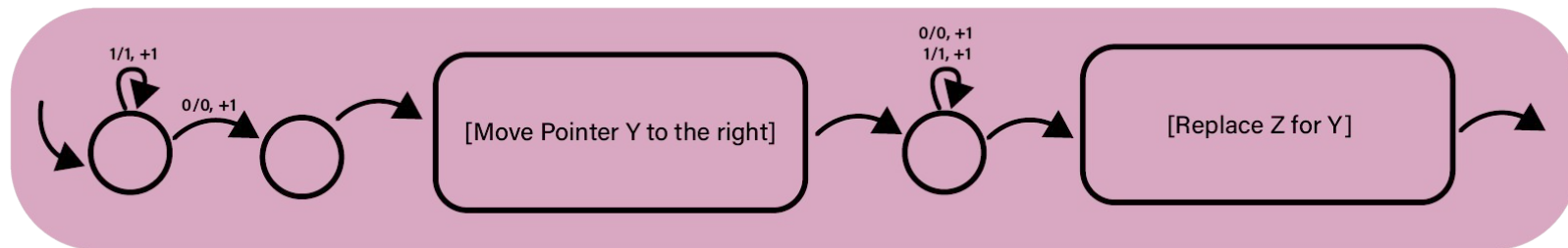
Copy the state q to the buffer to set it as the next state.



UTM Algorithm

8. Move the Y pointer to the next item in the current transition (symbol **b**) and replace Z with Y. Then move the Y pointer to the next item of the current transition (Movement of the tape head in M).

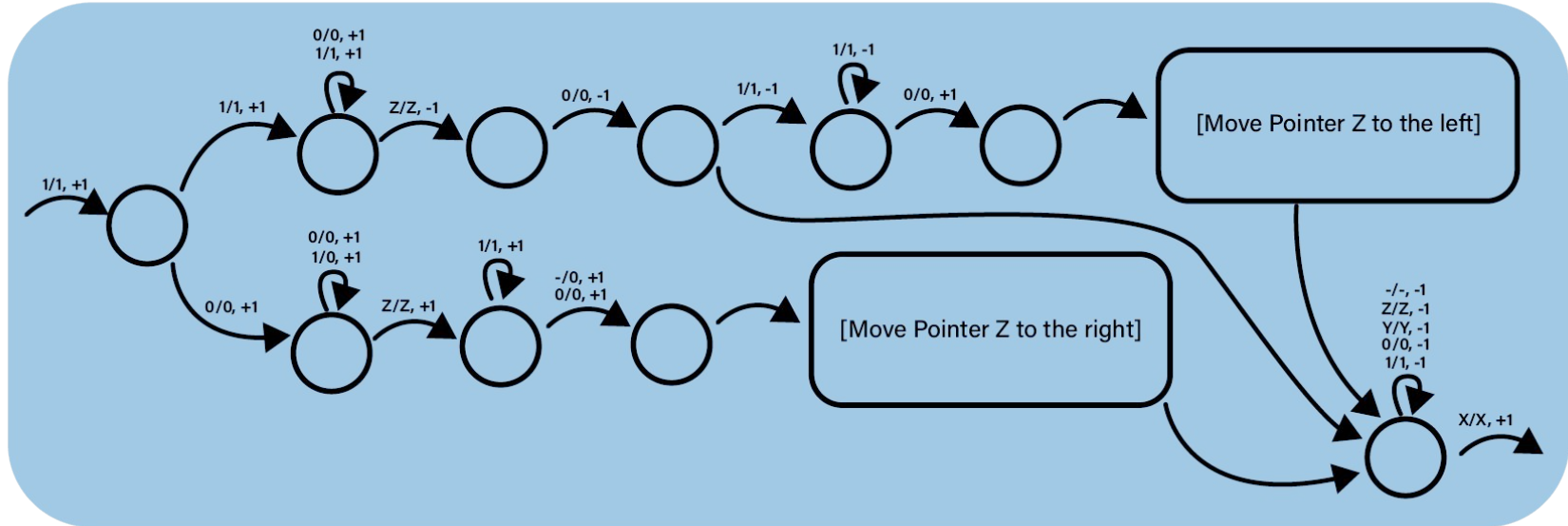
Replace the current symbol **a** in the tape for M with the new symbol **b** from the transition



UTM Algorithm

9. In the next item of the current transition (Movement of the tape head in M).

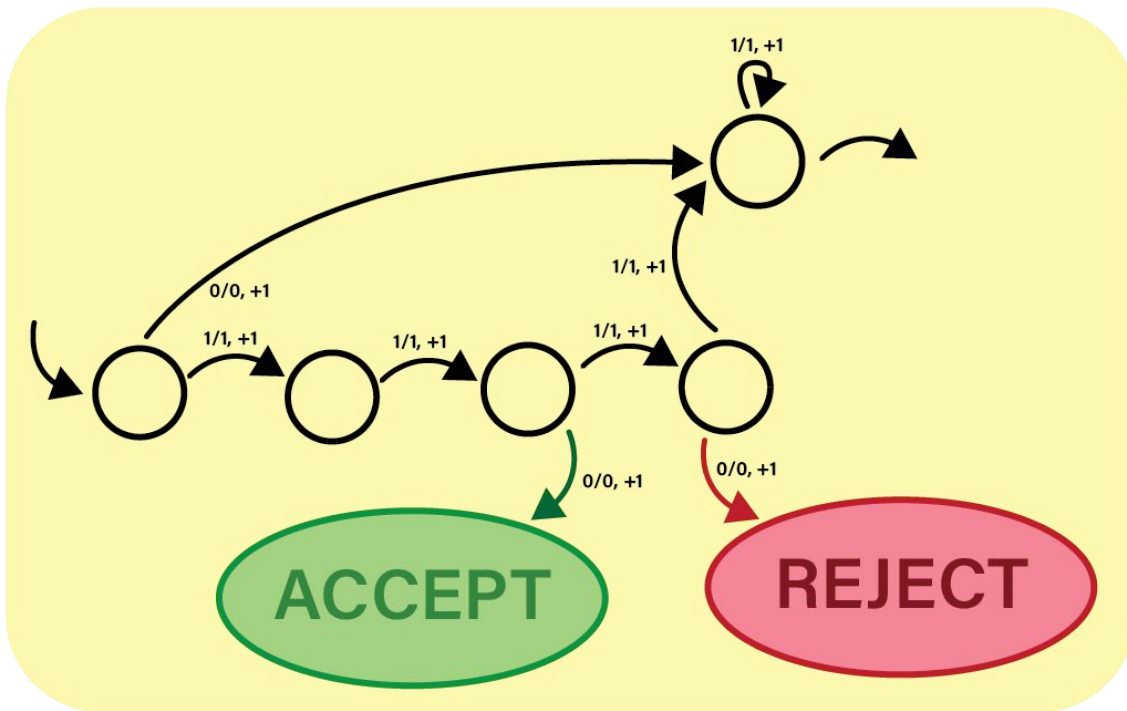
- If the transition indicates that the tape head has to move to the right, move the Z pointer to the right.
- If the transition indicates a movement to the left, move the Z pointer to the left.
- If the tape head is at the beginning of the tape, do not move it.



UTM Algorithm

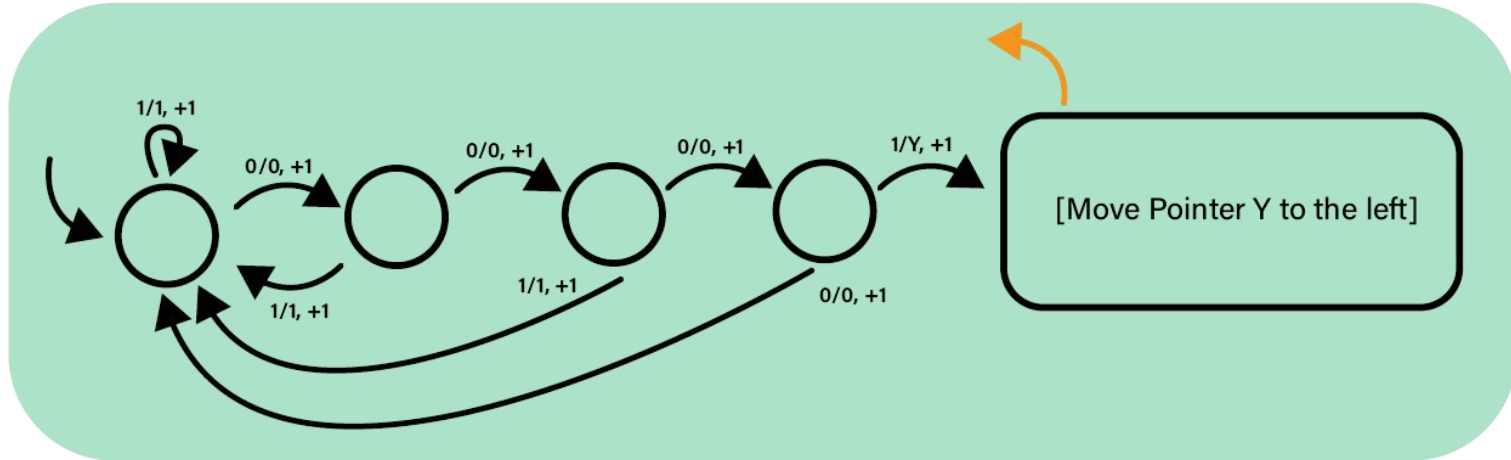
10. Return to X and read the current state of M.

- If the state is the accepting state, accept the string.
- If the state is in the rejecting state, reject the string.
- If the state is any other state, go to step 11.

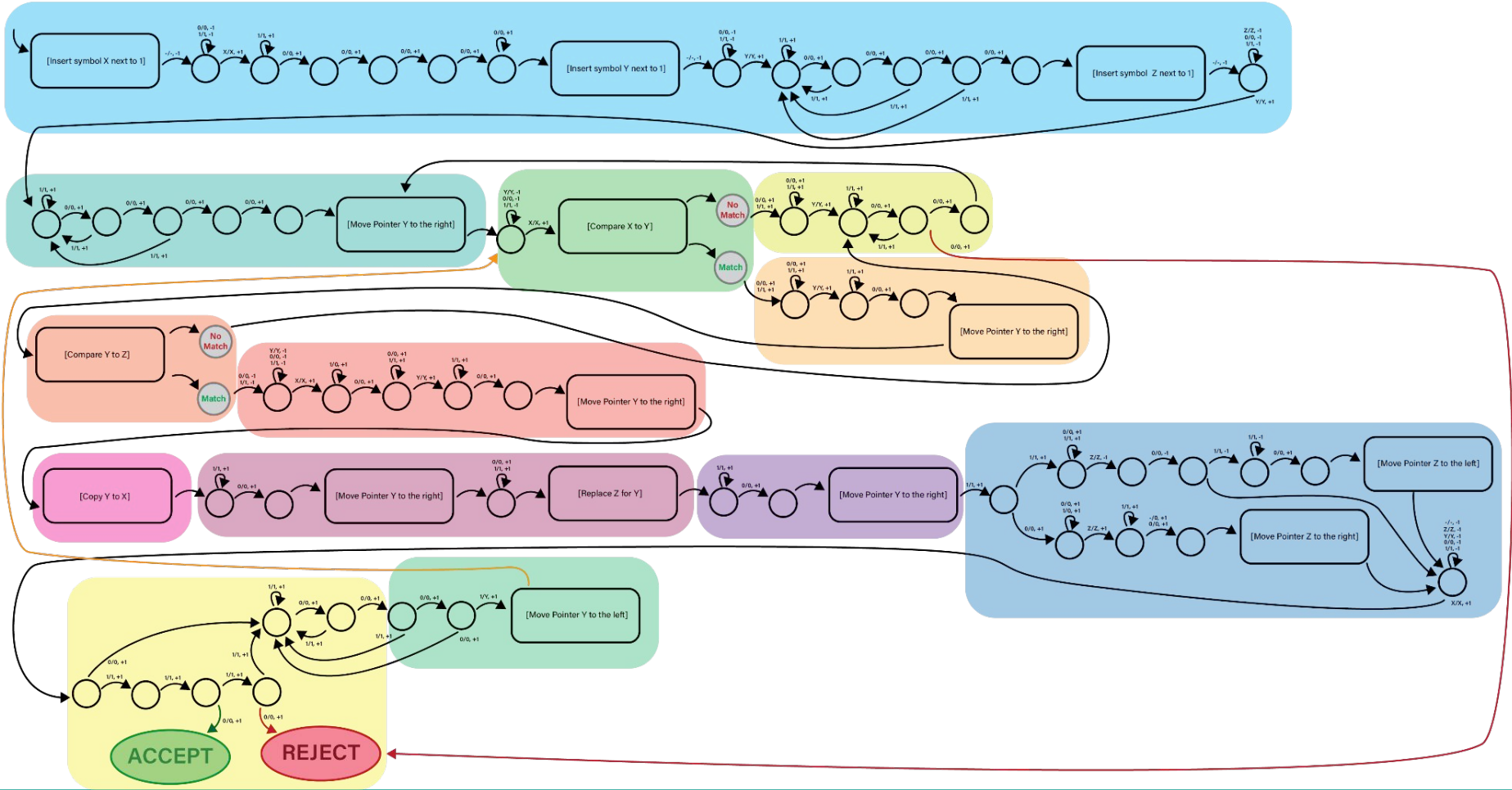


UTM Algorithm

11. Move the Y pointer to the first term of the transition function "delta" and loop back to step 2 to continue simulating the machine M.



Single Tape Universal Turing Machine



Thank you!
