Turing Machine*

Xiaofeng Gao

Department of Computer Science and Engineering
Shanghai Jiao Tong University, P.R.China

Algorithm Class @ Shanghai Jiao Tong University

* Special thanks is given to Prof. Yuxi Fu for sharing his teaching materials.

Outline

1. Effective Procedures
   - Basic Concepts
   - Computable Function
2. Turing Machine
   - Introduction
   - One-Tape Turing Machine
   - Multi-Tape Turing Machine
3. TM Variation and TM-Computability
   - TM Variations
   - Computable and Decidable

Outline

What is Effective Procedure

- Methods for addition, multiplication $\cdots$
  - Given n, finding the nth prime number.
  - Differentiating a polynomial.
  - Finding the highest common factor of two numbers $HCF(x, y) \to$ Euclidean algorithm
  - Given two numbers x, y, deciding whether x is a multiple of y.
- Their implementation requires no ingenuity, intelligence, inventiveness.

Intuitive Definition

An *algorithm* or *effective procedure* is a mechanical rule, or automatic method, or programme for performing some mathematical operations.

Blackbox: input $\longrightarrow \blacksquare \longrightarrow$ output

What is “effective procedure”?

An Example: Consider the function $g(n)$ defined as follows:

$$g(n) = \begin{cases} 1, & \text{if there is a run of exactly } n \text{ consecutive } 7'\text{s} \\ & \text{in the decimal expansion of } \pi, \\ 0, & \text{otherwise}. \end{cases}$$

Question: Is $g(n)$ effective?

What is “effective procedure”?

An Example: Consider the function $g(n)$ defined as follows:

$$g(n) = \begin{cases} 1, & \text{if there is a run of exactly } n \text{ consecutive } 7'\text{s} \\ & \text{in the decimal expansion of } \pi, \\ 0, & \text{otherwise.} \end{cases}$$

Question: Is $g(n)$ effective?

▷ The answer is unknown $\neq$ the answer is negative.

What is “effective procedure”?

An Example: Consider the function $g(n)$ defined as follows:

$$g(n) = \begin{cases} 1, & \text{if there is a run of exactly } n \text{ consecutive } 7's \\ & \text{in the decimal expansion of } \pi, \\ 0, & \text{otherwise.} \end{cases}$$

Question: Is $g(n)$ effective?

▷ The answer is unknown $\neq$ the answer is negative.

Other Examples:

- *Theorem Proving* is in general not effective/algorithmic.
- *Proof Verification* is effective/algorithmic.

Algorithm

An algorithm is a procedure that consists of a finite set of *instructions* which, given an *input* from some set of possible inputs, enables us to obtain an *output* through a systematic execution of the instructions that *terminates* in a finite number of steps.

Outline

Computable Function

When an algorithm or effective procedure is used to calculate the value of a numerical function then the function in question is effectively calculable (or algorithmically computable, effectively computable, computable).

Computable Function

When an algorithm or effective procedure is used to calculate the value of a numerical function then the function in question is effectively calculable (or algorithmically computable, effectively computable, computable).

Examples:

- $HCF(x, y)$ is computable;
- $g(n)$ is non-computable.

Development of Computation Models

1. Gödel-Kleene (1936): Partial recursive functions.
2. Turing (1936): Turing machines.
3. Church (1936): λ-terms.
4. Post (1943): Post systems.
5. Markov (1951): Variants of the Post systems.
6. Shepherdson-Sturgis (1963): URM-computable functions.

Development of Computation Models

1. Gödel-Kleene (1936): Partial recursive functions.
2. Turing (1936): Turing machines.
3. Church (1936): λ-terms.
4. Post (1943): Post systems.
5. Markov (1951): Variants of the Post systems.
6. Shepherdson-Sturgis (1963): URM-computable functions.

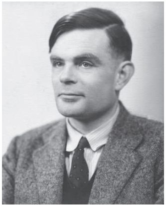
Church-Turing Thesis: Each of the above proposals for a characterization of the notion of effective computability gives rise to the same class of functions.

Outline

1. Effective Procedures
   - Basic Concepts
   - Computable Function
2. Turing Machine
   - Introduction
   - One-Tape Turing Machine
   - Multi-Tape Turing Machine
3. TM Variation and TM-Computability
   - TM Variations
   - Computable and Decidable

Alan Turing (23 Jun. 1912 - 7 Jun. 1954)

- An English student of Church
- Introduced a machine model for effective calculation in "On Computable Numbers, with an Application to the Entscheidungsproblem", Proc. of the London Mathematical Society, 42:230-265, 1936.
- Turing Machine, Halting Problem, Turing Test



Motivation

What are necessary for a machine to calculate a function?

Motivation

What are necessary for a machine to calculate a function?

- The machine should be able to interpret numbers
- The machine must be able to operate and manipulate numbers according to a set of predefined instructions

Motivation

What are necessary for a machine to calculate a function?

- The machine should be able to interpret numbers
- The machine must be able to operate and manipulate numbers according to a set of predefined instructions

and

- The input number has to be stored in an accessible place
- The output number has to be put in an accessible place
- There should be an accessible place for the machine to store intermediate results

Outline

1. Effective Procedures
   - Basic Concepts
   - Computable Function
2. Turing Machine
   - Introduction
   - One-Tape Turing Machine
   - Multi-Tape Turing Machine
3. TM Variation and TM-Computability
   - TM Variations
   - Computable and Decidable

One-Tape Turing Machine

A Turing machine has five components:

1. A finite set $\{s_1, \ldots, s_n\} \cup \{\triangleright, \triangleleft\} \cup \{\Box\}$ of symbols.

2. A tape consists of an infinite number of cells, each cell may store a symbol.

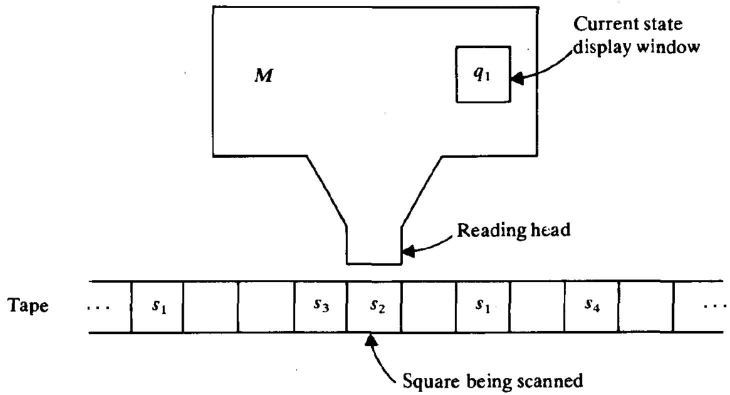
$$\cdots \Box\Box\Box\Box\Box\Box\Box\Box\Box\Box\Box\Box\Box\Box\Box\Box\Box\Box \cdots$$

3. A reading head that scans and writes on the cells.

4. A finite set $\{q_S, q_1, \ldots, q_m, q_H\}$ of states.

5. A finite set of instructions (specification).

One-Tape Turing Machine



One-Tape Turing Machine

The input data

$$\triangleright s_1^1 \ldots s_{i_1}^1 \Box \ldots \Box s_1^k \ldots s_{i_k}^k \triangleleft \Box \cdots$$

The reading head may write a symbol, move left, move right.

An instruction is of the form:

$$\langle q_i, s_j \rangle \rightarrow \langle q_l, s_k, L/R/S \rangle,$$

which means when reads s_j with state q_i, the machine will turn to state q_l, replace s_j with s_k, and turn one cell to the left.

The direction can be L, R, or S, meaning move to left, right, or stay at the current position.

An Example

Given a Turing machine M with the alphabet $\{0, 1\} \cup \{\triangleright, \Box, \triangleleft\}$.

$$
\begin{aligned}
\langle q_S, \triangleright\rangle &\rightarrow \langle q_1, \triangleright, R\rangle\\
\langle q_1, 0\rangle &\rightarrow \langle q_1, 0, R\rangle\\
\langle q_1, 1\rangle &\rightarrow \langle q_2, 0, S\rangle\\
\langle q_2, 0\rangle &\rightarrow \langle q_2, 0, R\rangle\\
\langle q_2, 1\rangle &\rightarrow \langle q_1, 1, R\rangle\\
\langle q_1, \triangleleft\rangle &\rightarrow \langle q_3, \triangleleft, L\rangle\\
\langle q_2, \triangleleft\rangle &\rightarrow \langle q_3, \triangleleft, L\rangle\\
\langle q_3, 0\rangle &\rightarrow \langle q_3, 0, L\rangle\\
\langle q_3, 1\rangle &\rightarrow \langle q_3, 1, L\rangle\\
\langle q_3, \triangleright\rangle &\rightarrow \langle q_H, \triangleright, R\rangle
\end{aligned}
$$

An Example

Given a Turing machine M with the alphabet $\{0, 1\} \cup \{\triangleright, \Box, \triangleleft\}$.

$\triangleright$	1	1	0	1	$\triangleleft$

$\widehat{q_S}$

$$\langle q_S, \triangleright\rangle \rightarrow \langle q_1, \triangleright, R\rangle$$
$$\langle q_1, 0\rangle \rightarrow \langle q_1, 0, R\rangle$$
$$\langle q_1, 1\rangle \rightarrow \langle q_2, 0, S\rangle$$
$$\langle q_2, 0\rangle \rightarrow \langle q_2, 0, R\rangle$$
$$\langle q_2, 1\rangle \rightarrow \langle q_1, 1, R\rangle$$
$$\langle q_1, \triangleleft\rangle \rightarrow \langle q_3, \triangleleft, L\rangle$$
$$\langle q_2, \triangleleft\rangle \rightarrow \langle q_3, \triangleleft, L\rangle$$
$$\langle q_3, 0\rangle \rightarrow \langle q_3, 0, L\rangle$$
$$\langle q_3, 1\rangle \rightarrow \langle q_3, 1, L\rangle$$
$$\langle q_3, \triangleright\rangle \rightarrow \langle q_H, \triangleright, R\rangle$$

An Example

Given a Turing machine M with the alphabet $\{0, 1\} \cup \{\triangleright, \Box, \triangleleft\}$.

$\triangleright$	1	1	0	1	$\triangleleft$
$\widehat{q_S}$					

$\triangleright$	**1**	1	0	1	$\triangleleft$
	$\widehat{q_1}$				

$$
\begin{aligned}
\langle q_S, \triangleright\rangle &\rightarrow \langle q_1, \triangleright, R\rangle \\
\langle q_1, 0\rangle &\rightarrow \langle q_1, 0, R\rangle \\
\langle q_1, 1\rangle &\rightarrow \langle q_2, 0, S\rangle \\
\langle q_2, 0\rangle &\rightarrow \langle q_2, 0, R\rangle \\
\langle q_2, 1\rangle &\rightarrow \langle q_1, 1, R\rangle \\
\langle q_1, \triangleleft\rangle &\rightarrow \langle q_3, \triangleleft, L\rangle \\
\langle q_2, \triangleleft\rangle &\rightarrow \langle q_3, \triangleleft, L\rangle \\
\langle q_3, 0\rangle &\rightarrow \langle q_3, 0, L\rangle \\
\langle q_3, 1\rangle &\rightarrow \langle q_3, 1, L\rangle \\
\langle q_3, \triangleright\rangle &\rightarrow \langle q_H, \triangleright, R\rangle
\end{aligned}
$$

An Example

Given a Turing machine M with the alphabet $\{0, 1\} \cup \{\triangleright, \Box, \triangleleft\}$.

$$\langle q_S, \triangleright\rangle \rightarrow \langle q_1, \triangleright, R\rangle$$
$$\langle q_1, 0\rangle \rightarrow \langle q_1, 0, R\rangle$$
$$\langle q_1, 1\rangle \rightarrow \langle q_2, 0, S\rangle$$
$$\langle q_2, 0\rangle \rightarrow \langle q_2, 0, R\rangle$$
$$\langle q_2, 1\rangle \rightarrow \langle q_1, 1, R\rangle$$
$$\langle q_1, \triangleleft\rangle \rightarrow \langle q_3, \triangleleft, L\rangle$$
$$\langle q_2, \triangleleft\rangle \rightarrow \langle q_3, \triangleleft, L\rangle$$
$$\langle q_3, 0\rangle \rightarrow \langle q_3, 0, L\rangle$$
$$\langle q_3, 1\rangle \rightarrow \langle q_3, 1, L\rangle$$
$$\langle q_3, \triangleright\rangle \rightarrow \langle q_H, \triangleright, R\rangle$$

$\triangleright$	1	1	0	1	$\triangleleft$
$\widehat{q_S}$					

$\triangleright$	**1**	1	0	1	$\triangleleft$
	$\widehat{q_1}$				

$\triangleright$	**0**	1	0	1	$\triangleleft$
	$\widehat{q_2}$				

An Example

Given a Turing machine M with the alphabet $\{0,1\} \cup \{\rhd, \Box, \lhd\}$.

$$\langle q_S, \rhd \rangle \rightarrow \langle q_1, \rhd, R \rangle$$
$$\langle q_1, 0 \rangle \rightarrow \langle q_1, 0, R \rangle$$
$$\langle q_1, 1 \rangle \rightarrow \langle q_2, 0, S \rangle$$
$$\langle q_2, 0 \rangle \rightarrow \langle q_2, 0, R \rangle$$
$$\langle q_2, 1 \rangle \rightarrow \langle q_1, 1, R \rangle$$
$$\langle q_1, \lhd \rangle \rightarrow \langle q_3, \lhd, L \rangle$$
$$\langle q_2, \lhd \rangle \rightarrow \langle q_3, \lhd, L \rangle$$
$$\langle q_3, 0 \rangle \rightarrow \langle q_3, 0, L \rangle$$
$$\langle q_3, 1 \rangle \rightarrow \langle q_3, 1, L \rangle$$
$$\langle q_3, \rhd \rangle \rightarrow \langle q_H, \rhd, R \rangle$$

$\rhd$	1	1	0	1	$\lhd$
$\widehat{q_S}$					

$\rhd$	**1**	1	0	1	$\lhd$
	$\widehat{q_1}$				

$\rhd$	**0**	1	0	1	$\lhd$
	$\widehat{q_2}$				

$\rhd$	0	**1**	0	1	$\lhd$
		$\widehat{q_2}$			

An Example

Given a Turing machine M with the alphabet $\{0,1\} \cup \{\rhd, \Box, \lhd\}$.

$$\langle q_S, \rhd\rangle \to \langle q_1, \rhd, R\rangle$$
$$\langle q_1, 0\rangle \to \langle q_1, 0, R\rangle$$
$$\langle q_1, 1\rangle \to \langle q_2, 0, S\rangle$$
$$\langle q_2, 0\rangle \to \langle q_2, 0, R\rangle$$
$$\langle q_2, 1\rangle \to \langle q_1, 1, R\rangle$$
$$\langle q_1, \lhd\rangle \to \langle q_3, \lhd, L\rangle$$
$$\langle q_2, \lhd\rangle \to \langle q_3, \lhd, L\rangle$$
$$\langle q_3, 0\rangle \to \langle q_3, 0, L\rangle$$
$$\langle q_3, 1\rangle \to \langle q_3, 1, L\rangle$$
$$\langle q_3, \rhd\rangle \to \langle q_H, \rhd, R\rangle$$

$\rhd$	1	1	0	1	$\lhd$
$\widehat{q_S}$					

$\rhd$	**1**	1	0	1	$\lhd$
	$\widehat{q_1}$				

$\rhd$	**0**	1	0	1	$\lhd$
	$\widehat{q_2}$				

$\rhd$	0	**1**	0	1	$\lhd$
		$\widehat{q_2}$			

$\rhd$	0	1	**0**	1	$\lhd$
			$\widehat{q_1}$		

An Example

Given a Turing machine M with the alphabet $\{0,1\} \cup \{\triangleright, \Box, \triangleleft\}$.

$$
\begin{aligned}
\langle q_S, \triangleright\rangle &\rightarrow \langle q_1, \triangleright, R\rangle \\
\langle q_1, 0\rangle &\rightarrow \langle q_1, 0, R\rangle \\
\langle q_1, 1\rangle &\rightarrow \langle q_2, 0, S\rangle \\
\langle q_2, 0\rangle &\rightarrow \langle q_2, 0, R\rangle \\
\langle q_2, 1\rangle &\rightarrow \langle q_1, 1, R\rangle \\
\langle q_1, \triangleleft\rangle &\rightarrow \langle q_3, \triangleleft, L\rangle \\
\langle q_2, \triangleleft\rangle &\rightarrow \langle q_3, \triangleleft, L\rangle \\
\langle q_3, 0\rangle &\rightarrow \langle q_3, 0, L\rangle \\
\langle q_3, 1\rangle &\rightarrow \langle q_3, 1, L\rangle \\
\langle q_3, \triangleright\rangle &\rightarrow \langle q_H, \triangleright, R\rangle
\end{aligned}
$$

$\triangleright$	1	1	0	1	$\triangleleft$
$\widehat{q_S}$					

$\triangleright$	**1**	1	0	1	$\triangleleft$
	$\widehat{q_1}$				

$\triangleright$	**0**	1	0	1	$\triangleleft$
	$\widehat{q_2}$				

$\triangleright$	0	**1**	0	1	$\triangleleft$
		$\widehat{q_2}$			

$\triangleright$	0	1	**0**	1	$\triangleleft$
			$\widehat{q_1}$		

$\triangleright$	0	1	0	**1**	$\triangleleft$
				$\widehat{q_1}$	

An Example

Given a Turing machine M with the alphabet $\{0,1\} \cup \{\triangleright, \square, \triangleleft\}$.

$$\langle q_S, \triangleright\rangle \rightarrow \langle q_1, \triangleright, R\rangle$$
$$\langle q_1, 0\rangle \rightarrow \langle q_1, 0, R\rangle$$
$$\langle q_1, 1\rangle \rightarrow \langle q_2, 0, S\rangle$$
$$\langle q_2, 0\rangle \rightarrow \langle q_2, 0, R\rangle$$
$$\langle q_2, 1\rangle \rightarrow \langle q_1, 1, R\rangle$$
$$\langle q_1, \triangleleft\rangle \rightarrow \langle q_3, \triangleleft, L\rangle$$
$$\langle q_2, \triangleleft\rangle \rightarrow \langle q_3, \triangleleft, L\rangle$$
$$\langle q_3, 0\rangle \rightarrow \langle q_3, 0, L\rangle$$
$$\langle q_3, 1\rangle \rightarrow \langle q_3, 1, L\rangle$$
$$\langle q_3, \triangleright\rangle \rightarrow \langle q_H, \triangleright, R\rangle$$

$\triangleright$	1	1	0	1	$\triangleleft$
$\widehat{q_S}$					

$\triangleright$	**1**	1	0	1	$\triangleleft$
	$\widehat{q_1}$				

$\triangleright$	**0**	1	0	1	$\triangleleft$
	$\widehat{q_2}$				

$\triangleright$	0	**1**	0	1	$\triangleleft$
		$\widehat{q_2}$			

$\triangleright$	0	1	**0**	1	$\triangleleft$
			$\widehat{q_1}$		

$\triangleright$	0	1	0	**1**	$\triangleleft$
				$\widehat{q_1}$	

$\triangleright$	0	1	0	**0**	$\triangleleft$
				$\widehat{q_2}$	

An Example

Given a Turing machine M with the alphabet $\{0,1\} \cup \{\triangleright, \Box, \triangleleft\}$.

$$\langle q_S, \triangleright\rangle \rightarrow \langle q_1, \triangleright, R\rangle$$
$$\langle q_1, 0\rangle \rightarrow \langle q_1, 0, R\rangle$$
$$\langle q_1, 1\rangle \rightarrow \langle q_2, 0, S\rangle$$
$$\langle q_2, 0\rangle \rightarrow \langle q_2, 0, R\rangle$$
$$\langle q_2, 1\rangle \rightarrow \langle q_1, 1, R\rangle$$
$$\langle q_1, \triangleleft\rangle \rightarrow \langle q_3, \triangleleft, L\rangle$$
$$\langle q_2, \triangleleft\rangle \rightarrow \langle q_3, \triangleleft, L\rangle$$
$$\langle q_3, 0\rangle \rightarrow \langle q_3, 0, L\rangle$$
$$\langle q_3, 1\rangle \rightarrow \langle q_3, 1, L\rangle$$
$$\langle q_3, \triangleright\rangle \rightarrow \langle q_H, \triangleright, R\rangle$$

$\triangleright$	1	1	0	1	$\triangleleft$	$\widehat{q_S}$
$\triangleright$	**1**	1	0	1	$\triangleleft$	$\widehat{q_1}$
$\triangleright$	**0**	1	0	1	$\triangleleft$	$\widehat{q_2}$
$\triangleright$	0	**1**	0	1	$\triangleleft$	$\widehat{q_2}$
$\triangleright$	0	1	**0**	1	$\triangleleft$	$\widehat{q_1}$
$\triangleright$	0	1	0	**1**	$\triangleleft$	$\widehat{q_1}$
$\triangleright$	0	1	0	**0**	$\triangleleft$	$\widehat{q_2}$
$\triangleright$	0	1	0	0	**$\triangleleft$**	$\widehat{q_2}$

An Example

Given a Turing machine M with the alphabet $\{0,1\} \cup \{\rhd, \Box, \lhd\}$.

$$\begin{aligned}
\langle q_S, \rhd\rangle &\to \langle q_1, \rhd, R\rangle \\
\langle q_1, 0\rangle &\to \langle q_1, 0, R\rangle \\
\langle q_1, 1\rangle &\to \langle q_2, 0, S\rangle \\
\langle q_2, 0\rangle &\to \langle q_2, 0, R\rangle \\
\langle q_2, 1\rangle &\to \langle q_1, 1, R\rangle \\
\langle q_1, \lhd\rangle &\to \langle q_3, \lhd, L\rangle \\
\langle q_2, \lhd\rangle &\to \langle q_3, \lhd, L\rangle \\
\langle q_3, 0\rangle &\to \langle q_3, 0, L\rangle \\
\langle q_3, 1\rangle &\to \langle q_3, 1, L\rangle \\
\langle q_3, \rhd\rangle &\to \langle q_H, \rhd, R\rangle
\end{aligned}$$

$\rhd$	1	1	0	1	$\lhd$
$\widehat{q_S}$					

$\rhd$	**1**	1	0	1	$\lhd$
	$\widehat{q_1}$				

$\rhd$	**0**	1	0	1	$\lhd$
	$\widehat{q_2}$				

$\rhd$	0	**1**	0	1	$\lhd$
		$\widehat{q_2}$			

$\rhd$	0	1	**0**	1	$\lhd$
			$\widehat{q_1}$		

$\rhd$	0	1	0	**1**	$\lhd$
				$\widehat{q_1}$	

$\rhd$	0	1	0	**0**	$\lhd$
				$\widehat{q_2}$	

$\rhd$	0	1	0	0	**$\lhd$**
					$\widehat{q_2}$

$\rhd$	0	1	0	**0**	$\lhd$
				$\widehat{q_3}$	

An Example

Given a Turing machine M with the alphabet $\{0, 1\} \cup \{\triangleright, \square, \triangleleft\}$.

$$
\begin{aligned}
\langle q_S, \triangleright \rangle &\rightarrow \langle q_1, \triangleright, R \rangle \\
\langle q_1, 0 \rangle &\rightarrow \langle q_1, 0, R \rangle \\
\langle q_1, 1 \rangle &\rightarrow \langle q_2, 0, S \rangle \\
\langle q_2, 0 \rangle &\rightarrow \langle q_2, 0, R \rangle \\
\langle q_2, 1 \rangle &\rightarrow \langle q_1, 1, R \rangle \\
\langle q_1, \triangleleft \rangle &\rightarrow \langle q_3, \triangleleft, L \rangle \\
\langle q_2, \triangleleft \rangle &\rightarrow \langle q_3, \triangleleft, L \rangle \\
\langle q_3, 0 \rangle &\rightarrow \langle q_3, 0, L \rangle \\
\langle q_3, 1 \rangle &\rightarrow \langle q_3, 1, L \rangle \\
\langle q_3, \triangleright \rangle &\rightarrow \langle q_H, \triangleright, R \rangle
\end{aligned}
$$

$\triangleright$	1	1	0	1	$\triangleleft$
$\widehat{q_S}$					

$\triangleright$	**1**	1	0	1	$\triangleleft$
	$\widehat{q_1}$				

$\triangleright$	**0**	1	0	1	$\triangleleft$
	$\widehat{q_2}$				

$\triangleright$	0	**1**	0	1	$\triangleleft$
		$\widehat{q_2}$			

$\triangleright$	0	1	**0**	1	$\triangleleft$
			$\widehat{q_1}$		

$\triangleright$	0	1	0	**1**	$\triangleleft$
				$\widehat{q_1}$	

$\triangleright$	0	1	0	**0**	$\triangleleft$
				$\widehat{q_2}$	

$\triangleright$	0	1	0	0	**$\triangleleft$**
					$\widehat{q_2}$

$\triangleright$	0	1	0	**0**	$\triangleleft$
				$\widehat{q_3}$	

$\triangleright$	0	1	**0**	0	$\triangleleft$
			$\widehat{q_3}$		

An Example

Given a Turing machine M with the alphabet $\{0,1\} \cup \{\triangleright, \Box, \triangleleft\}$.

$$\langle q_S, \triangleright \rangle \rightarrow \langle q_1, \triangleright, R \rangle$$
$$\langle q_1, 0 \rangle \rightarrow \langle q_1, 0, R \rangle$$
$$\langle q_1, 1 \rangle \rightarrow \langle q_2, 0, S \rangle$$
$$\langle q_2, 0 \rangle \rightarrow \langle q_2, 0, R \rangle$$
$$\langle q_2, 1 \rangle \rightarrow \langle q_1, 1, R \rangle$$
$$\langle q_1, \triangleleft \rangle \rightarrow \langle q_3, \triangleleft, L \rangle$$
$$\langle q_2, \triangleleft \rangle \rightarrow \langle q_3, \triangleleft, L \rangle$$
$$\langle q_3, 0 \rangle \rightarrow \langle q_3, 0, L \rangle$$
$$\langle q_3, 1 \rangle \rightarrow \langle q_3, 1, L \rangle$$
$$\langle q_3, \triangleright \rangle \rightarrow \langle q_H, \triangleright, R \rangle$$

$\triangleright$	1	1	0	1	$\triangleleft$
$\widehat{q_S}$					

$\triangleright$	**1**	1	0	1	$\triangleleft$
	$\widehat{q_1}$				

$\triangleright$	**0**	1	0	1	$\triangleleft$
	$\widehat{q_2}$				

$\triangleright$	0	**1**	0	1	$\triangleleft$
		$\widehat{q_2}$			

$\triangleright$	0	1	**0**	1	$\triangleleft$
			$\widehat{q_1}$		

$\triangleright$	0	1	0	**1**	$\triangleleft$
				$\widehat{q_1}$	

$\triangleright$	0	1	0	**0**	$\triangleleft$
				$\widehat{q_2}$	

$\triangleright$	0	1	0	0	**$\triangleleft$**
					$\widehat{q_2}$

$\triangleright$	0	1	0	**0**	$\triangleleft$
				$\widehat{q_3}$	

$\triangleright$	0	1	**0**	0	$\triangleleft$
			$\widehat{q_3}$		

$\triangleright$	0	1	0	0	$\triangleleft$
$\widehat{q_3}$					

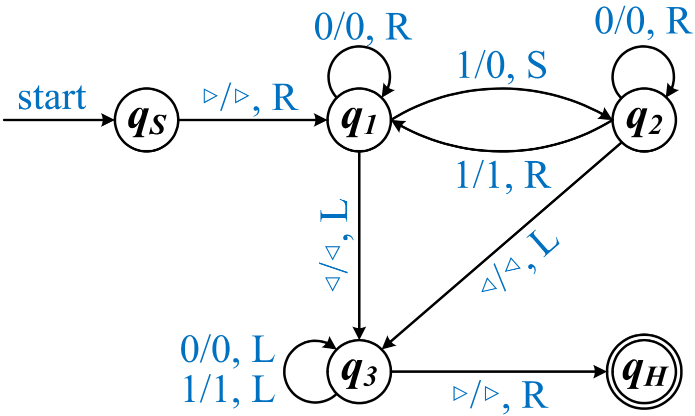
An Example

Given a Turing machine M with the alphabet $\{0, 1\} \cup \{\triangleright, \Box, \triangleleft\}$.

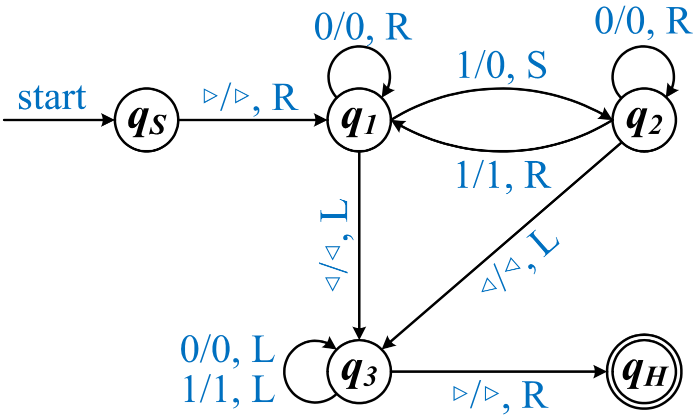
$$\langle q_S, \triangleright\rangle \to \langle q_1, \triangleright, R\rangle$$
$$\langle q_1, 0\rangle \to \langle q_1, 0, R\rangle$$
$$\langle q_1, 1\rangle \to \langle q_2, 0, S\rangle$$
$$\langle q_2, 0\rangle \to \langle q_2, 0, R\rangle$$
$$\langle q_2, 1\rangle \to \langle q_1, 1, R\rangle$$
$$\langle q_1, \triangleleft\rangle \to \langle q_3, \triangleleft, L\rangle$$
$$\langle q_2, \triangleleft\rangle \to \langle q_3, \triangleleft, L\rangle$$
$$\langle q_3, 0\rangle \to \langle q_3, 0, L\rangle$$
$$\langle q_3, 1\rangle \to \langle q_3, 1, L\rangle$$
$$\langle q_3, \triangleright\rangle \to \langle q_H, \triangleright, R\rangle$$

$$\underset{\widehat{q_S}}{\boldsymbol{\triangleright}}\ 1\ 1\ 0\ 1\ \triangleleft$$
$$\triangleright\ \underset{\widehat{q_1}}{\mathbf{1}}\ 1\ 0\ 1\ \triangleleft$$
$$\triangleright\ \underset{\widehat{q_2}}{\mathbf{0}}\ 1\ 0\ 1\ \triangleleft$$
$$\triangleright\ 0\ \underset{\widehat{q_2}}{\mathbf{1}}\ 0\ 1\ \triangleleft$$
$$\triangleright\ 0\ 1\ \underset{\widehat{q_1}}{\mathbf{0}}\ 1\ \triangleleft$$
$$\triangleright\ 0\ 1\ 0\ \underset{\widehat{q_1}}{\mathbf{1}}\ \triangleleft$$
$$\triangleright\ 0\ 1\ 0\ \underset{\widehat{q_2}}{\mathbf{0}}\ \triangleleft$$
$$\triangleright\ 0\ 1\ 0\ 0\ \underset{\widehat{q_2}}{\boldsymbol{\triangleleft}}$$
$$\triangleright\ 0\ 1\ 0\ \underset{\widehat{q_3}}{\mathbf{0}}\ \triangleleft$$
$$\triangleright\ 0\ 1\ \underset{\widehat{q_3}}{\mathbf{0}}\ 0\ \triangleleft$$
$$\underset{\widehat{q_3}}{\boldsymbol{\triangleright}}\ 0\ 1\ 0\ 0\ \triangleleft$$
$$\triangleright\ \underset{\widehat{q_H}}{\mathbf{0}}\ 1\ 0\ 0\ \triangleleft$$

State Transition Diagram



State Transition Diagram



M's action is to work from left to right along the tape, replacing alternate 1's by the symbol 0's.

Outline

1. Effective Procedures
   - Basic Concepts
   - Computable Function
2. Turing Machine
   - Introduction
   - One-Tape Turing Machine
   - Multi-Tape Turing Machine
3. TM Variation and TM-Computability
   - TM Variations
   - Computable and Decidable

Multi-Tape Turing Machine

A multi-tape TM is described by a tuple (Γ, Q, δ) containing

- A finite set Γ called alphabet, of symbols. It contains a blank symbol $\Box$, a start symbol $\triangleright$, and the digits 0 and 1.
- A finite set Q of states. It contains a start state q_{start} and a halting state q_{halt}.
- A transition function $\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^{k-1} \times L, S, R^k$, describing the rules of each computation step.

Multi-Tape Turing Machine

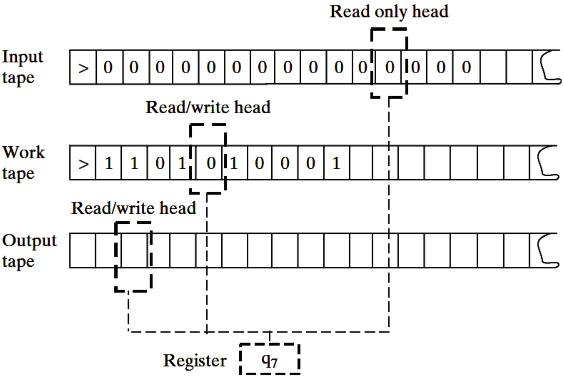
A multi-tape TM is described by a tuple (Γ, Q, δ) containing

- A finite set Γ called alphabet, of symbols. It contains a blank symbol $\square$, a start symbol $\triangleright$, and the digits 0 and 1.
- A finite set Q of states. It contains a start state q_{start} and a halting state q_{halt}.
- A transition function $\delta : Q \times \Gamma^k \to Q \times \Gamma^{k-1} \times L, S, R^k$, describing the rules of each computation step.

Example: A 2-Tape TM will have transition function (also named as specification) like follows:

$$\begin{array}{lcl} \langle q_s, \triangleright, \triangleright \rangle & \to & \langle q_1, \triangleright, R, R \rangle \\ \langle q_1, 0, 1 \rangle & \to & \langle q_2, 0, S, L \rangle \end{array}$$

Computation and Configuration



Computation, configuration, initial/final configuration

A 3-Tape TM for the Palindrome Problem

A palindrome is a word that reads the same both forwards and backwards. For instance:

ada, anna, madam, and nitalarbralatin.

A 3-Tape TM for the Palindrome Problem

A palindrome is a word that reads the same both forwards and backwards. For instance:

ada, anna, madam, and nitalarbralatin.

Requirement: Give the specification of M with $k = 3$ to recognize palindromes on symbol set $\{0, 1, \triangleright, \triangleleft, \Box\}$.

Preparation

To recognize palindrome we need to check the input string, output 1 if the string is a palindrome, and 0 otherwise.

Preparation

To recognize palindrome we need to check the input string, output 1 if the string is a palindrome, and 0 otherwise.

Initially the input string is located on the first tape like "$\triangleright$ 0110001 $\triangleleft \square\square\square \cdots$", strings on all other tapes are "$\triangleright\square\square\square \cdots$".

Preparation

To recognize palindrome we need to check the input string, output 1 if the string is a palindrome, and 0 otherwise.

Initially the input string is located on the first tape like "$\triangleright$ 0110001 $\triangleleft \Box\Box\Box \cdots$", strings on all other tapes are "$\triangleright\Box\Box\Box \cdots$".

The head on each tape points the first symbol "$\triangleright$" as the starting state, with state mark q_S.

Preparation

To recognize palindrome we need to check the input string, output 1 if the string is a palindrome, and 0 otherwise.

Initially the input string is located on the first tape like "$\triangleright$ 0110001 $\triangleleft \square\square\square \cdots$", strings on all other tapes are "$\triangleright\square\square\square \cdots$".

The head on each tape points the first symbol "$\triangleright$" as the starting state, with state mark q_S.

In the final state q_F, the output of the k^{th} tape should be "$\triangleright$ 1 $\triangleleft \square$" if the input is a palindrome, and "$\triangleright$ 0 $\triangleleft \square$" otherwise.

A 3-Tape TM for the Palindrome Problem

$Q = \{q_s, q_h, q_c, q_l, q_t, q_r\}$; $\Gamma = \{\Box, \rhd, \lhd, 0, 1\}$; two work tapes.

A 3-Tape TM for the Palindrome Problem

$Q = \{q_s, q_h, q_c, q_l, q_t, q_r\}$; $\Gamma = \{\Box, \rhd, \lhd, 0, 1\}$; two work tapes.

Start State:

$\langle q_s, \rhd, \rhd, \rhd \rangle \rightarrow \langle q_c, \rhd, \rhd, R, R, R \rangle$

A 3-Tape TM for the Palindrome Problem

$$Q = \{q_s, q_h, q_c, q_l, q_t, q_r\}; \Gamma = \{\Box, \rhd, \lhd, 0, 1\}; \text{two work tapes.}$$

Start State:

$\langle q_s, \rhd, \rhd, \rhd \rangle \rightarrow \langle q_c, \rhd, \rhd, R, R, R \rangle$

Begin to copy:

$\langle q_c, 0, \Box, \Box \rangle \rightarrow \langle q_c, 0, \Box, R, R, S \rangle$
$\langle q_c, 1, \Box, \Box \rangle \rightarrow \langle q_c, 1, \Box, R, R, S \rangle$
$\langle q_c, \lhd, \Box, \Box \rangle \rightarrow \langle q_l, \Box, \Box, L, S, S \rangle$

A 3-Tape TM for the Palindrome Problem

$$Q = \{q_s, q_h, q_c, q_l, q_t, q_r\}; \Gamma = \{\Box, \rhd, \lhd, 0, 1\};\text{ two work tapes.}$$

Start State:

$\langle q_s, \rhd, \rhd, \rhd \rangle \rightarrow \langle q_c, \rhd, \rhd, R, R, R \rangle$

Begin to copy:

$\langle q_c, 0, \Box, \Box \rangle \rightarrow \langle q_c, 0, \Box, R, R, S \rangle$
$\langle q_c, 1, \Box, \Box \rangle \rightarrow \langle q_c, 1, \Box, R, R, S \rangle$
$\langle q_c, \lhd, \Box, \Box \rangle \rightarrow \langle q_l, \Box, \Box, L, S, S \rangle$

Return back to the leftmost:

$\langle q_l, 0, \Box, \Box \rangle \rightarrow \langle q_l, \Box, \Box, L, S, S \rangle$
$\langle q_l, 1, \Box, \Box \rangle \rightarrow \langle q_l, \Box, \Box, L, S, S \rangle$
$\langle q_l, \rhd, \Box, \Box \rangle \rightarrow \langle q_t, \Box, \Box, R, L, S \rangle$

A 3-Tape TM for the Palindrome Problem

$Q = \{q_s, q_h, q_c, q_l, q_t, q_r\}$; $\Gamma = \{\Box, \rhd, \lhd, 0, 1\}$; two work tapes.

Start State:

$\langle q_s, \rhd, \rhd, \rhd \rangle \to \langle q_c, \rhd, \rhd, R, R, R \rangle$

Begin to copy:

$\langle q_c, 0, \Box, \Box \rangle \to \langle q_c, 0, \Box, R, R, S \rangle$
$\langle q_c, 1, \Box, \Box \rangle \to \langle q_c, 1, \Box, R, R, S \rangle$
$\langle q_c, \lhd, \Box, \Box \rangle \to \langle q_l, \Box, \Box, L, S, S \rangle$

Return back to the leftmost:

$\langle q_l, 0, \Box, \Box \rangle \to \langle q_l, \Box, \Box, L, S, S \rangle$
$\langle q_l, 1, \Box, \Box \rangle \to \langle q_l, \Box, \Box, L, S, S \rangle$
$\langle q_l, \rhd, \Box, \Box \rangle \to \langle q_t, \Box, \Box, R, L, S \rangle$

Begin to compare:

$\langle q_t, \lhd, \rhd, \Box \rangle \to \langle q_r, \rhd, 1, S, S, R \rangle$
$\langle q_t, 0, 1, \Box \rangle \to \langle q_r, 1, 0, S, S, R \rangle$
$\langle q_t, 1, 0, \Box \rangle \to \langle q_r, 0, 0, S, S, R \rangle$
$\langle q_t, 0, 0, \Box \rangle \to \langle q_t, 0, \Box, R, L, S \rangle$
$\langle q_t, 1, 1, \Box \rangle \to \langle q_t, 1, \Box, R, L, S \rangle$

A 3-Tape TM for the Palindrome Problem

$Q = \{q_s, q_h, q_c, q_l, q_t, q_r\}$; $\Gamma = \{\Box, \rhd, \lhd, 0, 1\}$; two work tapes.

Start State:

$\langle q_s, \rhd, \rhd, \rhd \rangle \rightarrow \langle q_c, \rhd, \rhd, R, R, R \rangle$

Begin to copy:

$\langle q_c, 0, \Box, \Box \rangle \rightarrow \langle q_c, 0, \Box, R, R, S \rangle$
$\langle q_c, 1, \Box, \Box \rangle \rightarrow \langle q_c, 1, \Box, R, R, S \rangle$
$\langle q_c, \lhd, \Box, \Box \rangle \rightarrow \langle q_l, \Box, \Box, L, S, S \rangle$

Return back to the leftmost:

$\langle q_l, 0, \Box, \Box \rangle \rightarrow \langle q_l, \Box, \Box, L, S, S \rangle$
$\langle q_l, 1, \Box, \Box \rangle \rightarrow \langle q_l, \Box, \Box, L, S, S \rangle$
$\langle q_l, \rhd, \Box, \Box \rangle \rightarrow \langle q_t, \Box, \Box, R, L, S \rangle$

Begin to compare:

$\langle q_t, \lhd, \rhd, \Box \rangle \rightarrow \langle q_r, \rhd, 1, S, S, R \rangle$
$\langle q_t, 0, 1, \Box \rangle \rightarrow \langle q_r, 1, 0, S, S, R \rangle$
$\langle q_t, 1, 0, \Box \rangle \rightarrow \langle q_r, 0, 0, S, S, R \rangle$
$\langle q_t, 0, 0, \Box \rangle \rightarrow \langle q_t, 0, \Box, R, L, S \rangle$
$\langle q_t, 1, 1, \Box \rangle \rightarrow \langle q_t, 1, \Box, R, L, S \rangle$

Ready to terminate:

$\langle q_r, \lhd, \rhd, \Box \rangle \rightarrow \langle q_h, \rhd, \lhd, S, S, S \rangle$
$\langle q_r, 0, 1, \Box \rangle \rightarrow \langle q_h, 1, \lhd, S, S, S \rangle$
$\langle q_r, 1, 0, \Box \rangle \rightarrow \langle q_r, 0, \lhd, S, S, S \rangle$

Outline

Language System

Let $\Sigma = \{a_1, \ldots, a_k\}$ be the set of symbols, called alphabet.

Language System

Let $\Sigma = \{a_1, \ldots, a_k\}$ be the set of symbols, called alphabet.

A string (word) from Σ is a sequence $a_{i_1}, \cdots, a_{i_n}$ of symbols from Σ.

Language System

Let $\Sigma = \{a_1, \ldots, a_k\}$ be the set of symbols, called alphabet.

A string (word) from Σ is a sequence $a_{i_1}, \cdots, a_{i_n}$ of symbols from Σ.

Σ^* is the set of all words/strings from Σ. (Kleene Star)

Language System

Let $\Sigma = \{a_1, \ldots, a_k\}$ be the set of symbols, called alphabet.

A string (word) from Σ is a sequence $a_{i_1}, \cdots, a_{i_n}$ of symbols from Σ.

Σ^* is the set of all words/strings from Σ. (Kleene Star)

For example, if $\Sigma = \{a, b\}$, we have

$$\Sigma^* = \{a, b\}^* = \{\Lambda, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, \ldots\}.$$

Language System

Let $\Sigma = \{a_1, \ldots, a_k\}$ be the set of symbols, called alphabet.

A string (word) from Σ is a sequence $a_{i_1}, \cdots, a_{i_n}$ of symbols from Σ.

Σ^* is the set of all words/strings from Σ. (Kleene Star)

For example, if $\Sigma = \{a, b\}$, we have

$$\Sigma^* = \{a, b\}^* = \{\Lambda, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, \ldots\}.$$

Λ is the empty string, that has no symbols. (ε)

$\{0, 1, \Box, \rhd\}$ vs. Larger Alphabets

Fact: If $f : \{0,1\}^* \to \{0,1\}^*$ is computable in time $T(n)$ by a TM M using the alphabet set Γ, then it is computable in time $4 \log |\Gamma| T(n)$ by a TM $\widetilde{M}$ using the alphabet $\{0, 1, \Box, \rhd\}$.

$\{0, 1, \square, \triangleright\}$ vs. Larger Alphabets

Suppose M has k tapes with the alphabet Γ.

A symbol of M is encoded in $\widetilde{M}$ by a string $\sigma \in \{0, 1\}^*$ of length $\log |\Gamma|$.

A state q in M is turned into a number of states in $\widetilde{M}$

- q,
- $\langle q, \sigma_1^1, \ldots, \sigma_1^k \rangle$ where $|\sigma_1^1| = \ldots = |\sigma_1^k| = 1$,
- $\cdots$,
- $\langle q, \sigma_{\log |\Gamma|}^1, \ldots, \sigma_{\log |\Gamma|}^k \rangle$, the size of $\sigma_{\log |\Gamma|}^1, \ldots, \sigma_{\log |\Gamma|}^k$ is $\log |\Gamma|$.

$\{0, 1, \square, \triangleright\}$ vs. Larger Alphabets

To simulate one step of M, the machine $\widetilde{M}$ will

1. use $\log|\Gamma|$ steps to read from each tape the $\log|\Gamma|$ bits encoding a symbol of Γ,
2. use its state register to store the symbols read,
3. use M's transition function to compute the symbols M writes and M's new state given this information,
4. store this information in its state register, and
5. use $\log|\Gamma|$ steps to write the encodings of these symbols on its tapes.

$\{0, 1, \Box, \triangleright\}$ vs. Larger Alphabets

Example: $\{0, 1, \Box, \triangleright\}$ vs. English Alphabets

M's tape:	>	m	a	c	h	i	n	e										
$\tilde{M}$'s tape:	>	0	1	1	0	1	0	0	0	0	1	0	0	0	1	1		

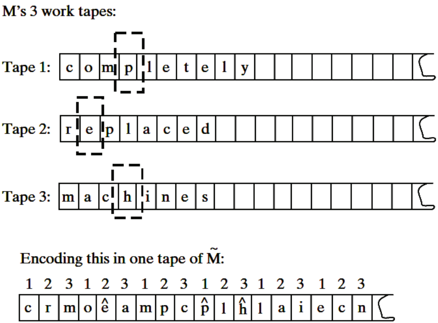
Single-Tape vs. Multi-Tape

Define a single-tape TM to be a TM that has one read-write tape.

Fact: If $f : \{0,1\}^* \to \{0,1\}^*$ is computable in time $T(n)$ by a TM M using k tapes, then it is computable in time $5kT(n)^2$ by a single-tape TM $\widetilde{M}$.

Single-Tape vs. Multi-Tape

- The basic idea is to interleave k tapes into one tape.
- The first $n+1$ cells are reserved for the input.



- Every symbol a of M is turned into two symbols $a, \hat{a}$ in $\tilde{M}$, with $\hat{a}$ used to indicate head position.

Single-Tape vs. Multi-Tape

The outline of the algorithm:

The machine $\widetilde{M}$ places $\triangleright$ after the input string and then starts copying the input bits to the imaginary input tape. During this process whenever an input symbol is copied it is overwritten by $\triangleright$.

$\widetilde{M}$ marks the $n+2$-cell, . . ., the $n+k$-cell to indicate the initial head positions.

$\widetilde{M}$ Sweeps $kT(n)$ cells from the $(n+1)$-th cell to right, recording in the register the k symbols marked with the hat $\hat{\_}$.

$\widetilde{M}$ Sweeps $kT(n)$ cells from right to left to update using the transitions of M. Whenever it comes across a symbol with hat, it moves right k cells, and then moves left to update.

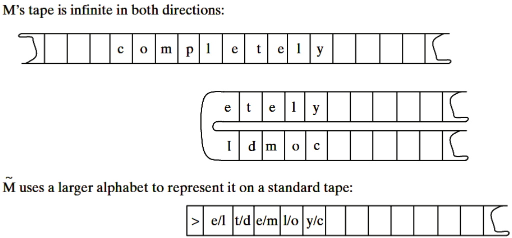
Unidirectional Tape vs. Bidirectional Tape

Define a bidirectional Turing Machine to be a TM whose tapes are infinite in both directions.

Fact: If $f : \{0,1\}^* \to \{0,1\}^*$ is computable in time $T(n)$ by a bidirectional TM M, then it is computable in time $4T(n)$ by a TM $\widetilde{M}$ with one-directional tape.

Unidirectional Tape vs. Bidirectional Tape

- The idea is that $\widetilde{M}$ makes use of the alphabet $\Gamma \times \Gamma$.



- Every state q of M is turned into $\bar{q}$ and $\underline{q}$.

Unidirectional Tape vs. Bidirectional Tape

Let H range over $\{L, S, R\}$ and let $-H$ be defined by

$$-H = \begin{cases} R, & \text{if} \quad H = L, \\ S, & \text{if} \quad H = S, \\ L, & \text{if} \quad H = R. \end{cases}$$

$\widetilde{M}$ contains the following transitions:

$$\langle \overline{q}, (\rhd, \rhd) \rangle \rightarrow \langle \underline{q}, (\rhd, \rhd), R \rangle$$
$$\langle \underline{q}, (\rhd, \rhd) \rangle \rightarrow \langle \overline{q}, (\rhd, \rhd), R \rangle$$

$$\langle \overline{q}, (a, b) \rangle \rightarrow \langle \overline{q'}, (a', b), H \rangle \text{ if } \langle q, a \rangle \rightarrow \langle q', a', H \rangle$$
$$\langle \underline{q}, (a, b) \rangle \rightarrow \langle \underline{q'}, (a, b'), -H \rangle \text{ if } \langle q, b \rangle \rightarrow \langle q', b', H \rangle$$

Outline

TM-Computable Function

What does it mean that a TM computes a (partial) n-ary function f?

TM-Computable Function

What does it mean that a TM computes a (partial) n-ary function f?

Let M be a TM and $a_1, \ldots, a_n, b \in \mathbb{N}$. When computation $M(a_1, \ldots, a_n)$ converges to b if $M(a_1, \ldots, a_n) \downarrow$ and $r_1 = b$ in the final configuration. We write $M(a_1, \ldots, a_n) \downarrow b$.

- M TM-computes f if, for all $a_1, \ldots, a_n, b \in \mathbb{N}$,

$$M(a_1, \ldots, a_n) \downarrow b \text{ iff } f(a_1, \ldots, a_n) = b$$

- Function f is TM-computable if there is a Turing Machine that TM-computes f.
- (We abbreviate "TM-computable" to "computable")

Function Defined by Program

Given any program P and $n \geq 1$, by thinking of the effect of P on initial configurations of the form $a_1, \cdots, a_n, 0, 0, \cdots$, there is a unique n-ary function that P computes, denoted by $f_P^{(n)}$.

$$f_P^{(n)}(a_1, \ldots, a_n) = \begin{cases} b, & \text{if } P(a_1, \ldots, a_n) \downarrow b, \\ \text{undefined}, & \text{if } P(a_1, \ldots, a_n) \uparrow. \end{cases}$$

Predicate and Decision Problem

The value of a predicate is either ‘true’ or ‘false’.

The answer of a *decision problem* is either ‘yes’ or ‘no’.

Example: Given two numbers x, y, check whether x is a multiple of y.
Input: x, y;
Output: ‘Yes’ or ‘No’.

The operation amounts to calculation of the function

$$f(x,y) = \begin{cases} 1, & \text{if } x \text{ is a multiple of } y, \\ 0, & \text{if otherwise.} \end{cases}$$

Thus the property or predicate ‘x is a multiple of y’ is algorithmically or effectively decidable, or just decidable if function f is computable.

Decidable Predicate and Decidable Problem

Suppose that $P(x_1, \ldots, x_n)$ is an n-ary predicate of natural numbers. The characteristic function $c_P(\mathbf{x})$,

$$f_M^{(n)}(a_1, \ldots, a_n) = \begin{cases} 1, & \text{if } P(\mathbf{x}) \text{ holds}, \\ 0, & \text{if otherwise}. \end{cases}$$

The predicate $P(\mathbf{x})$ is decidable if c_P is computable; it is undecidable otherwise.

Computability on other Domains

Suppose D is an object domain. A coding of D is an explicit and effective injection $\alpha : D \to \mathbb{N}$. We say that an object $d \in D$ is coded by the natural number $\alpha(d)$.

A function $f : D \to D$ extends to a numeric function $f^* : \mathbb{N} \to \mathbb{N}$. We say that f is computable if f^* is computable.

$$f^* = \alpha \circ f \circ \alpha^{-1}$$

Example

Consider the domain $\mathbb{Z}$. An explicit coding is given by the function α where

$$\alpha(n) = \begin{cases} 2n, & \text{if } n \geq 0, \\ -2n-1, & \text{if } n < 0. \end{cases}$$

Then α^{-1} is given by

$$\alpha^{-1}(m) = \begin{cases} \frac{1}{2}m, & \text{if } m \text{ is even}, \\ -\frac{1}{2}(m+1), & \text{if } m \text{ is odd}. \end{cases}$$

Example (Continued)

Consider the function $f(x) = x - 1$ on $\mathbb{Z}$, then $f^* : \mathbb{N} \to \mathbb{N}$ is given by

$$f^*(x) = \begin{cases} 1 & \text{if } x = 0 \text{ (i.e. } x = \alpha(0)), \\ x - 2 & \text{if } x > 0 \text{ and } x \text{ is even } (\text{i.e. } x = \alpha(n), n > 0), \\ x + 2 & \text{if } x \text{ is odd } (\text{i.e. } x = \alpha(n), n < 0). \end{cases}$$

It is a routine exercise to write a program that computes f^*, hence $x - 1$ is a computable function on $\mathbb{Z}$.