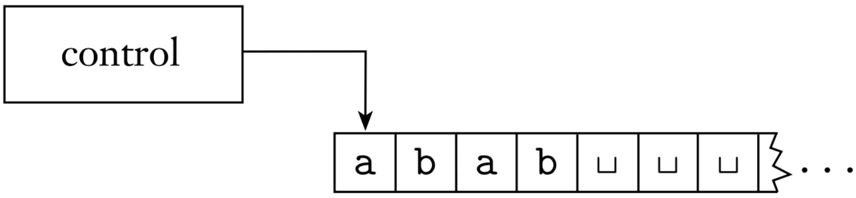
4. The Church-Turing Thesis

Introduction of Turing machine

- Turing Machine M uses an infinite tape as its unlimited memory, with a tape head reading and writing symbols and moving around on the tape.
- The tape initially contains only the input string and is blank everywhere else.
- If M needs to store information, it may write this information on the tape. To read the information that it has written, M can move its head back over it.
- M continues computing until it decides to produce an output. The output accept and reject are obtained by enterring designated accepting and rejecting states.
- If M doesn't enter an accepting or rejecting state, it will go on forever, never halting.



Turing Machine (TM): A Turing Machine is a 7-tuple, $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, where Q, Σ, Γ are all finite and

- Q is a set of states;
- Σ is the input alphabet not containing the **blank symbol** $\sqcup$;
- Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$;
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ is the transition function;
- $q_0 \in Q$ is the start state;
- $q_{accept} \in Q$ is the accept state, and
- $q_{reject} \in Q$ is the reject state, where $q_{reject} \neq q_{accept}$.

Configuration: A configuration of a Turing machine consists of:

- the current state;
- the current tape contents, and
- the current head location.

by $u\ q\ v$ we mean the configuration where

- the current state is q;
- the current tape contents is uv, and
- the current head location is the first symbol of v;
- the tape contains only blanks following the last symbol of v.

Special Configurations

- **Initial Configuration (start configuration)** of M on input w is the configuartion $q_0\ w$;
- **Accepting Configuration** of M is the configuration $u\ q_{accept} v$;

- **Rejecting Configuration** of M is the configuration $u\ q_{reject}\ v$;
- **Halting Configurations**: accepting configurations and rejecting configurations (do not yield further configurations).

Configurations in transitions: Let $a, b, c \in \Gamma, u, v \in \Gamma^*$, and $q_i, q_j \in Q$.

1. If $\delta(q_i, b) = (q_j, c, L)$, then $ua\ q_i\ bv$ yields $u\ q_j\ acv$;
2. If $\delta(q_i, b) = (q_j, c, R)$, then $ua\ q_i\ bv$ yields $uac\ q_j\ v$.

Special cases occur when the head is at one of the ends of the configuration:

- For the left-hand end, the configuration $q_i\ bv$ yields $q_j\ cv$ if the transition is left moving; because we prevent the machine from going off the left-hand end of the tape, and it yields $c\ q_j\ v$ for the right-moving transition.
- For the right-hand end, the configuration $ua\ q_i$ is equivalent to $ua\ q_i\ \sqcup$ because we assume that blanks follow the part of the tape represented in the configuration.

Computation by TM: M accepts w if there are sequence of configurations $C_1, C_2, \cdots, C_k$ such that

- C_1 is the start configuration of M on w;
- Each C_i yields C_{i+1}, and
- C_k is an accepting configuration.

[Note] $C_1, C_2, \cdots, C_{k-1}$ can not be halting configurations. TM stop immediately when in a halting configuration.

The collection of strings that M accepts is the language of M, or the language recognized by M, denoted $L(M)$.

Turing-recognizable: A language is Turing-recognizable, if some Turing machine recognizes it.

Turing-decidable: A language is Turing-decidable or simply decidable if some Turing machine decides it.

On an input, the machine M may accept, reject or *loop*. By loop we mean that the machine does not halt. If M always halt, then it is a <u>decider</u>. A decider that recognizes some language is said to <u>decide</u> that language.

Multitape Turing Machines: A multitape Turing Machine M has several tapes:

- Each tape has its own head for reading and writing;
- The input is initially on tape 1, with all the other tapes being blank;
- The transition function is

$$\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R, S\}^k$$

  Where k is the number of tapes.

$$\delta(q_i, a_1, \cdots, a_k) = (q_j, b_1, \cdots, b_k, L, R, \cdots, L)$$

means that if M is in state q_i and head 1 through k are reading symbols a_1 through a_k, the machine goes to state q_j, writes symbols b_1 through b_k, and directs each head to move left or right, or to *stay put*, as specified.

Theorem. Every multitape Turing machine has an equivalent single-tape Turing machine.

Corollary. A language is Turing-recognizable if and only if some multitape Turing machine recognizes it.

Nondeterministic Turing machines (NTM): The transition function for a nondeterministic Turing machine has the form

$$\delta : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$$

and the computation of a nondeterministic Turing machine is a tree whose branches correspond to different possibilities for the machine. If *some* branch of the computation leads to the accept state, the machine accept itts input.

Theorem. Every NTM has an equivalent DTM.

Corollary. A language is Turing-recognizable iff some nondeterministic Turing machine recognizes it.

Corollary. A language is decidable iff some nondeterministic Turing machine decides it.

Enumerator: A enumerator is a Turing machine with an attached printer. The Turing machine can use that printer as an output device to print strings. Every time the Turing machine wants to add a string to the list, it sends the string to the printer.

Theorem. A language is Turing-recognizable iff some enumerator enumerates it.

Church-Turing Thesis: Intuitive notion of algorithms = Turing machine algorithms.

Hilbert's Ten Problem

$$D = \{p \mid p \text{ is a polynomial with integer coefficients and with an integral root}\}$$

- **Theorem**. D is not decidable.

$$D_1 = \{p \mid p \text{ is a polynomial on a single variable } x \text{ with integer coefficients and with an integral root}\}$$

- **Lemma**. Both D and D_1 are Turing-recognizable.
- **Lemma**. D_1 is Turing-decidable.