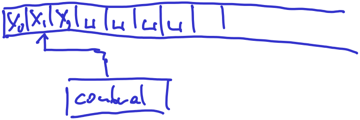
Turing machines

Peter Mayr

Computability Theory, January 27, 2021

Turing machines (TM) were proposed by Turing (1937) as computational model to formalize the intuitive notion of algorithms.



A TM consists of

- **tape:** consists of cells that can contain a symbol, has a leftmost end but is infinite to the right, initial part holds input, else filled with blanks ␣
- **control:** in one of finitely many states, contains instructions
- **head:** points to a cell on tape

In one step a TM

- changes state according to its current state and the symbol at the location of the head,
- writes a new symbol at the location of the head,
- moves the head one position left or right.

TM runs until it reaches its **accept** or **reject** state (may never stop).

DTM specification

Definition

A **deterministic Turing machine (DTM)** M is a 7-tuple $(Q, \Sigma, \Gamma, s, t, r, \delta)$ with

- Q a finite set of **states**,
- Σ a finite **input alphabet**,
- Γ a finite **tape alphabet** with $\sqcup \in \Gamma, \Sigma \subseteq \Gamma \setminus \{\sqcup\}$,
- $s \in Q$ the **start state**,
- $t \in Q$ the **accept state**,
- $r \in Q$ the **reject state** with $r \neq t$,
- $\delta \colon (Q \setminus \{t, r\}) \times \Gamma \to Q \times \Gamma \times \{-1, +1\}$ the **transition function**.

Example

DTM with $Q = \{s, t, r\}, \Sigma = \{0, 1\}, \Gamma = \Sigma \cup \{\textvisiblespace\}$ and

δ	0	1	$\textvisiblespace$
s	$(s, 0, +1)$	$(s, 0, +1)$	$(s, 0, +1)$

fills tape with 0, never stops

Differences DTM and DFA

- DFA can be interpreted as DTM but
- DTM has infinite tape to read and write (memory!)
- DTM computation not bound by input length (DTM may not halt at all)

Example

DTM to accept $L = \{0^n 1^n \ : \ n \in \mathbb{N}\}$.

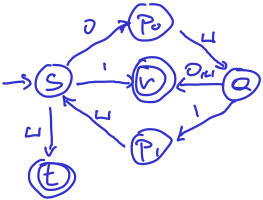
Idea: Alternatingly delete 0, 1 from ends of the input string. Accept if only blanks remain.

	δ	0	1	$\sqcup$
start	s	$(p_0, \sqcup, +1)$	$(r, \sqcup, +1)$	$(t, \sqcup, +1)$
deleted 0	p_0	$(p_0, 0, +1)$	$(p_0, 1, +1)$	$(a, \sqcup, -1)$
return	a	$(r, \ldots)$	$(p_1, \sqcup, -1)$	$(r, \ldots)$
deleted 1	p_1	$(p_1, 0, -1)$	$(p_1, 1, -1)$	$(s, \sqcup, +1)$

computation on input 0011

state	tape	position
s	$0011\sqcup\ldots$	0
p_0	$\sqcup 011 \sqcup$	1
p_0	$\sqcup 011 \sqcup$	2
p_0	$\sqcup 011 \sqcup$	3
p_0	$\sqcup 011 \sqcup$	4
a	$\sqcup 011 \sqcup$	3
p_1	$\sqcup 01 \sqcup \sqcup$	2
p_1	$\sqcup 01 \sqcup \sqcup$	1
p_1	$\sqcup 01 \sqcup$	0
s	$\sqcup 01 \sqcup$	1
	$\vdots$	

State transitions



Describing the computation of a DTM

Definition

Let $M = (Q, \Sigma, \Gamma, s, t, r, \delta)$ be a DTM.

- Enumerate the cells on the tape of M by $\mathbb{N} = \{0, 1, 2, \dots\}$.
- A **configuration** of M is a triple $(q, \alpha, k) \in Q \times \Gamma^{\mathbb{N}} \times \mathbb{N}$ of state q, tape content α and head in position k of the tape.
- The **successor configuration** of (q, α, k) for $q \in Q \setminus \{t, r\}$ and $\delta(q, \alpha(k)) = (p, a, d)$ is

$$(p, \alpha[k \to a], \max(k + d, 0)).$$

  (The function $\alpha[k \to a]$ is equal to α except it maps k to a.)
- Write $c \vdash_M d$ if d is the successor configuration of c. Let $\vdash_M^*$ denote the transitive closure of $\vdash_M$ on $Q \times \Gamma^{\mathbb{N}} \times \mathbb{N}$. Then $c \vdash_M^* d$ ("c yields d") if d is obtained from c in finitely many steps of M.

Definition

- M **accepts** $w \in \Sigma^*$ if $(s, w\sqcup\ldots, 0) \vdash_M^* (t, ., .)$.
  M **rejects** w if $(s, w\sqcup\ldots, 0) \vdash_M^* (r, ., .)$.
  M **halts** on w if it either accepts or rejects w; else M **loops** on w.
- The **language** of M is

$$L(M) := \{w \in \Sigma^* \ : \ M \text{ accepts } w\}.$$