Theory of Computer Science

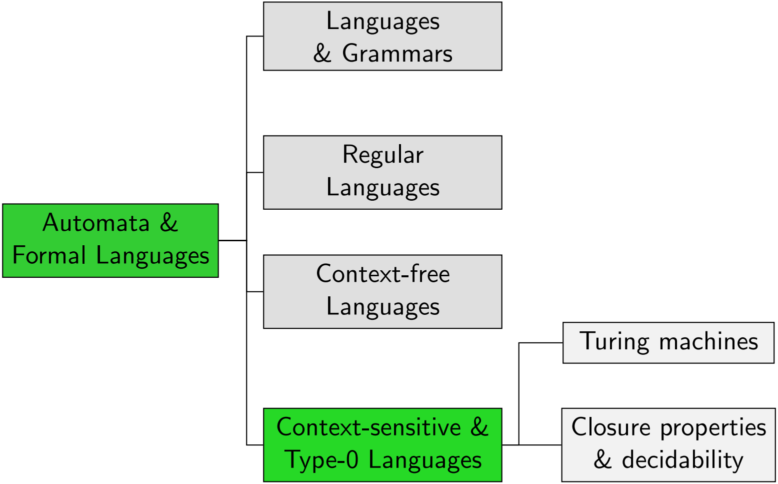
C7. Context-Sensitive and Type-0 Languages: Turing Machines

Gabriele Röger

University of Basel

April 8, 2020

Overview



Languages & Grammars
Regular Languages
Automata & Formal Languages
Context-free Languages
Turing machines
Context-sensitive & Type-0 Languages
Closure properties & decidability

C7.1 Context-Sensitive and General Grammars

Repetition: (Context-Sensitive) Grammars

Definition (Grammar)

A grammar is a 4-tuple $\langle \Sigma, V, P, S \rangle$ with:

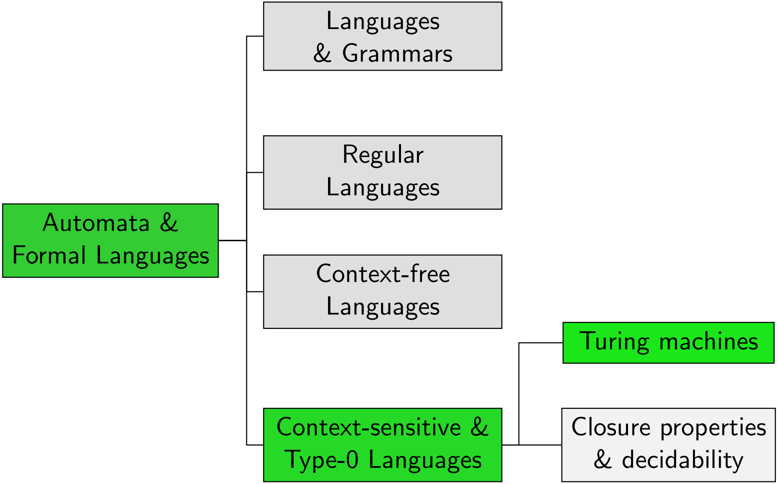
- Σ finite alphabet of terminal symbols
- V finite set of variables (with $V \cap \Sigma = \emptyset$)
- $P \subseteq (V \cup \Sigma)^+ \times (V \cup \Sigma)^*$ finite set of rules
- $S \in V$ start variable

Type 0 and type 1:

- Every grammar is type 0.
- A grammar is context-sensitive (type-1) if all rules $w_1 \to w_2$ satisfy $|w_1| \leq |w_2|$.
  - only exception: $S \to \varepsilon$ is allowed for the start symbol S if S never occurs on a right-hand side.

C7.2 Turing Machines

Overview


Languages & Grammars
Regular Languages
Automata & Formal Languages
Context-free Languages
Turing machines
Context-sensitive & Type-0 Languages
Closure properties & decidability

Automata for Type-1 and Type-0 Languages?

Finite automata accept exactly the regular languages, push-down automata exactly the context-free languages. Are there automata models for context-sensitive and type-0 languages?

Yes! ⇝ Turing machines

German: Turingmaschinen

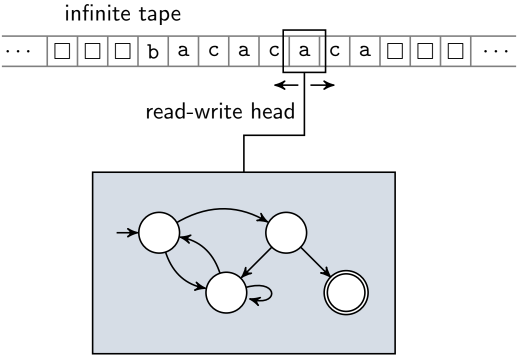
Picture courtesy of imagerymajestic / FreeDigitalPhotos.net

Alan Turing (1912–1954)

Picture courtesy of Jon Callas / wikimedia commons

- British logician, mathematician, cryptanalyst and computer scientist
- most important work (for us): On Computable Numbers, with an Application to the Entscheidungsproblem ⇝ Turing machines
- collaboration on Enigma decryption
- conviction due to homosexuality; pardoned by Elizabeth II in Dec. 2013
- Turing award most important science award in computer science

Turing Machines: Conceptually



Nondeterministic Turing Machine: Definition

Definition (Nondeterministic Turing Machine)

A nondeterministic Turing machine (NTM) is given by a 7-tuple $M = \langle Q, \Sigma, \Gamma, \delta, q_0, \square, E\rangle$ with:

- Q finite non-empty set of states
- $\Sigma \neq \emptyset$ finite input alphabet
- $\Gamma \supset \Sigma$ finite tape alphabet
- $\delta : (Q \setminus E) \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{\mathsf{L}, \mathsf{R}, \mathsf{N}\})$ transition function
- $q_0 \in Q$ start state
- $\square \in \Gamma \setminus \Sigma$ blank symbol
- $E \subseteq Q$ end states

German: Turingmaschine, Zustände, Eingabealphabet, Bandalphabet, Übergangsfunktion, Startzustand, Blank-Symbol, Endzustände

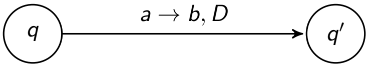
Turing Machine: Transition Function

Let $M = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, E\rangle$ be an NTM.

What is the Intuitive Meaning of the Transition Function δ?

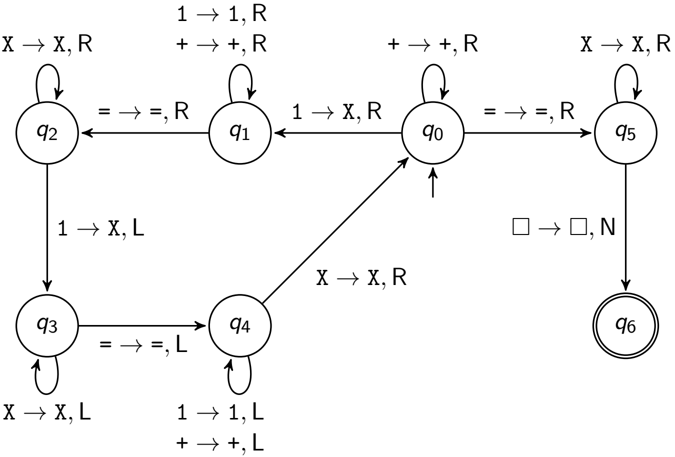
$\langle q', b, D\rangle \in \delta(q, a)$:

- If M is in state q and reads a, then
- M can transition to state q' in the next step,
- replacing a with b,
- and moving the head in direction $D \in \{\text{L}, \text{R}, \text{N}\}$, where:
  - L: one step to the left,
  - R: one step to the right,
  - N: neutral (no) movement.



Nondeterministic Turing Machine: Example

$$M = \langle \{q_0, q_1, \ldots, q_6\}, \{\mathtt{1}, \mathtt{+}, \mathtt{=}\}, \{\mathtt{1}, \mathtt{+}, \mathtt{=}, \mathtt{X}, \square\}, \delta, q_0, \square, \{q_6\} \rangle$$



Turing Machine: Configuration

Definition (Configuration of a Turing Machine)

A configuration of a Turing machine $M = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, E \rangle$ is given by a triple $c \in \Gamma^* \times Q \times \Gamma^+$.

German: Konfiguration

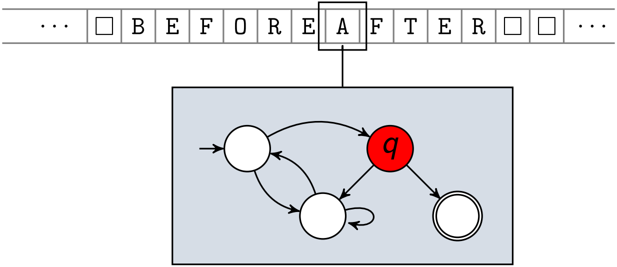
Configuration $\langle w_1, q, w_2 \rangle$ intuitively means that

- the non-empty or already visited part of the tape contains the word $w_1 w_2$,
- the read-write head is on the first symbol of w_2, and
- the TM is in state q.

Turing Machine Configurations: Example

Example

configuration $\langle \square\mathtt{BEFORE}, q, \mathtt{AFTER}\square\square \rangle$.



Turing Machine: Step

Definition (Transition/Step of a Turing Machine)

An NTM $M = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, E \rangle$ can transition from configuration c to configuration c' in one step ($c \vdash_M c'$) according to the following rules:

- $\langle a_1 \dots a_m, q, b_1 \dots b_n \rangle \vdash_M \langle a_1 \dots a_m, q', cb_2 \dots b_n \rangle$
  if $\langle q', c, \mathrm{N} \rangle \in \delta(q, b_1)$, $m \geq 0$, $n \geq 1$
- $\langle a_1 \dots a_m, q, b_1 \dots b_n \rangle \vdash_M \langle a_1 \dots a_{m-1}, q', a_m cb_2 \dots b_n \rangle$
  if $\langle q', c, \mathrm{L} \rangle \in \delta(q, b_1)$, $m \geq 1$, $n \geq 1$
- $\langle \varepsilon, q, b_1 \dots b_n \rangle \vdash_M \langle \varepsilon, q', \Box cb_2 \dots b_n \rangle$
  if $\langle q', c, \mathrm{L} \rangle \in \delta(q, b_1)$, $n \geq 1$
- $\langle a_1 \dots a_m, q, b_1 \dots b_n \rangle \vdash_M \langle a_1 \dots a_m c, q', b_2 \dots b_n \rangle$
  if $\langle q', c, \mathrm{R} \rangle \in \delta(q, b_1)$, $m \geq 0$, $n \geq 2$
- $\langle a_1 \dots a_m, q, b_1 \rangle \vdash_M \langle a_1 \dots a_m c, q', \Box \rangle$
  if $\langle q', c, \mathrm{R} \rangle \in \delta(q, b_1)$, $m \geq 0$

Turing Machines: Reachability of Configurations

Definition (Reachable Configuration)

Configuration c' is reachable from configuration c in NTM M ($c \vdash_M^* c'$) if there are configurations $c_0, \dots, c_n$ ($n \geq 0$) where

- $c_0 = c$,
- $c_i \vdash_M c_{i+1}$ for all $i \in \{0, \dots, n-1\}$, and
- $c_n = c'$.

German: c' ist in M von c erreichbar

Turing Machines: Recognized Words

Definition (Recognized Word of a Turing Machine)

NTM $M = \langle Q, \Sigma, \Gamma, \delta, q_0, \square, E \rangle$ recognizes the word $w = a_1 \dots a_n$ iff an accepting configuration (where M is in an end state) is reachable from the start configuration:

$$M \text{ recognizes } w \text{ iff } \langle \varepsilon, q_0, w \rangle \vdash_M^* \langle w_1, q, w_2 \rangle$$
$$\text{for some } q \in E, w_1 \in \Gamma^*, w_2 \in \Gamma^+$$

special case: for $w = \varepsilon$ the start configuration is $\langle \varepsilon, q_0, \square \rangle$ rather than $\langle \varepsilon, q_0, \varepsilon \rangle$

German: M erkennt w, akzeptierende Konfiguration, Startkonfiguration

example: blackboard

Turing Machines: Accepted Language

Definition (Accepted Language of an NTM)

Let M be an NTM with input alphabet Σ.

The language accepted by M is defined as

$$\mathcal{L}(M) = \{w \in \Sigma^* \mid M \text{ recognizes } w\}.$$

German: erkannte Sprache

example: blackboard

Exercise

Specify the state diagram of an NTM that accepts language

$$L = \{\mathtt{a}^n \mathtt{b}^n \mathtt{c}^n \mid n \geq 1\}.$$

C7.3 Summary

Summary

- Turing machines only have finitely many states but an unbounded tape as "memory".
- Alan Turing proposed them as a mathematical model for arbitrary algorithmic computations.
- In this role, we will revisit them in the parts on computability and complexity theory.