CSCI 335 Theory of Computing

Lecture 14 – Turing Machines IV: Church-Turing Thesis

DATE : December 05 2022

Learning Outcomes

Learning Outcomes

By the end of this section you will be able to:

- Understand Church-Turing thesis
- Get acquainted with an undecidable problem among Hilbert's problems
- Get familiar with string encoding of different objects for TM simulations

Outline

- Church-Turing thesis
- Hilbert’s problems
- String representation of objects

Church–Turing thesis

Church–Turing thesis

Alonzo Church
1903–1995

Algorithm	=	Turing machine
Intuitive		Formal

Alan Turing
1912–1954

Instead of Turing machines,
can use any other "reasonable" model
of unrestricted computation:
λ-calculus, random access machine,
your favorite programming language, …

Big impact on mathematics, computer science, and artificial intelligence.

Church–Turing thesis

Alonzo Church
1903–1995

Algorithm	=	Turing machine
Intuitive		Formal

- This thesis states that **all possible models of computation, if they are sufficiently broad, must be equivalent.**
- It also implies that **there is an inherent limitation in this and that there are functions that cannot be expressed in any way that gives an explicit method for their computation.**
- A general principle for algorithmic computation and, while not provable, gives strong evidence that no more powerful models can be found.

Alan Turing
1912–1954

Hilbert's problems

Hilbert's 10^{th} problem

In 1900 David Hilbert posed 23 problems

#1) Problem of the continuum (Does set A exist where $|\mathbb{N}| < |A| < |\mathbb{R}|$?).
#2) Prove that the axioms of mathematics are consistent.
#10) Give an algorithm for solving *Diophantine equations.*

David Hilbert
1862—1943

Diophantine equations:

Equations of polynomials where <u>solutions must be integers</u>.
Example: $3x^2 - 2xy - y^2z = 7$ solution: $x = 1,\ y = 2,\ z = -2$

Let $D = \{p|$ polynomial $p(x_1, x_2, \dots, x_k) = 0$ has a <u>solution in integers</u>)

Hilbert's 10th problem: Give an algorithm to decide D.

Matiyasevich proved in 1970: D is not decidable.

Note: D is T-recognizable.

String representation of objects

String representation of objects

Notation for encoding objects into strings

- If O is some object (e.g., polynomial, automaton, graph, etc.), we write $\langle O \rangle$ to be an encoding of that object into a string.

- If $O_1, O_2, \dots, O_k$ is a list of objects then we write $\langle O_1, O_2, \dots, O_k \rangle$ to be an encoding of them together into a single string.

Notation for writing Turing machines

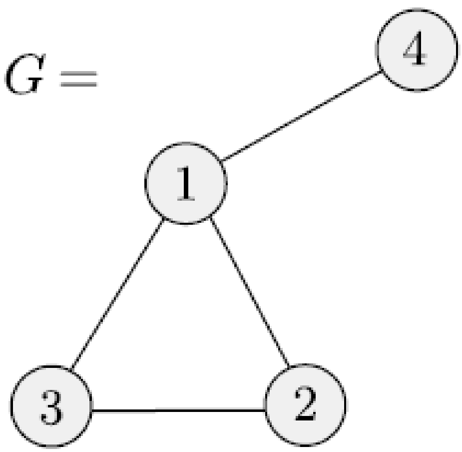
We will use high-level English descriptions of algorithms when we describe TMs, knowing that we could (in principle) convert those descriptions into states, transition function, etc. Our notation for writing a TM M is

$M =$ "On input w

[English description of the algorithm]"

Example: String representation of graphs

$G =$

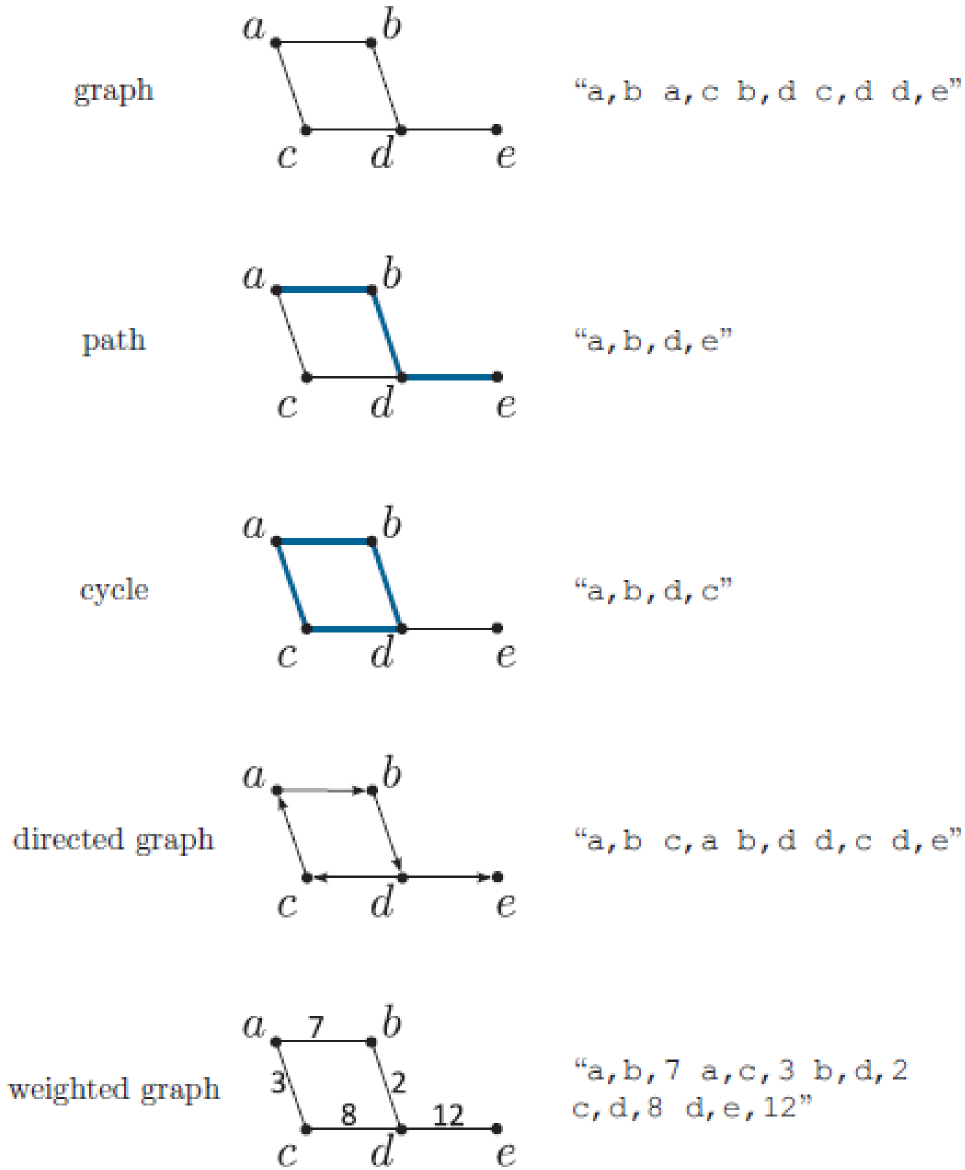


$\langle G \rangle =$

`(1,2,3,4)((1,2),(2,3),(3,1),(1,4))`

FIGURE 3.24
A graph G and its encoding $\langle G \rangle$

Example: String representation of graphs



The right-hand column shows possible ASCII string representations of the corresponding examples of graph concepts.

Page 48, John MacCormick: What Can Be Computed?: A Practical Guide to the Theory of Computation. Princeton University Press, 2018

Summary

Summary

- Church-Turing thesis
- Hilbert's problems
- String representation of objects

Readings

- Church-Turing thesis: Section 3.3 (Sipser 2013)