Harvard CS 121 and CSCI E-207
Lecture 16: Decidability & The Universal TM

Salil Vadhan

October 30, 2012

• Reading: Sipser §4.1, §4.2.
A “Universal” algorithm for deciding regular languages

**Proposition:** $A_{DFA} = \{ \langle D, w \rangle : D \text{ a DFA that accepts } w \}$ is decidable.

**Proof sketch:**

- First check that input is of proper form.
- Then simulate $D$ on $w$. Implementation on a multitape TM:
  - Tape 2: String $w$ with head at current position (or to be precise, its representation).
  - Tape 3: Current state $q$ of $D$ (i.e., its representation).
- Could work with other encodings, e.g. transition function as a matrix rather than list of triples.
Representation independence

**General point:** Notions of computability (e.g. decidability and recognizability) are independent of data representations.

- A TM can convert any reasonable encoding to any other reasonable encoding.

- We will use $\langle \cdot \rangle$ to mean “any reasonable encoding”.

- We’ll need to revisit representation issues again when we discuss computational *speed*.

- For the moment when we are interested only in whether problems are decidable, undecidable, recognizable, etc., so we can be content knowing that there is *some* representation on which an algorithm could work.
High-Level Algorithm Descriptions

Given the C–T Thesis and representation independence, we no longer need to refer to a specific computing model or data representation when describing an algorithm. Instead:

- Describe it as a sequence of steps operating on higher-level data types (e.g. numbers, graphs, automata, grammars).
- Each step: simple enough that it is clear it can be implemented on a reasonable model (such as a TM) using a reasonable data representation.
- Freely make use of algorithms we have seen (or are well-known, such as elementary arithmetic) as subroutines.
- Freely make use of control-flow primitives, such as loops, if-then-else, gotos, etc.
More Decidable Problems

- \{ \langle R, w \rangle : R \text{ is a regular expression that generates } w \}\}.

- \{ \langle X \rangle : X \text{ is an DFA/NFA/RE such that } L(X) = \emptyset \}\}.

- \{ \langle X \rangle : X \text{ is a DFA/NFA/RE such that } |L(X)| = \infty \}\}.
More Decidable Problems

- \{\langle M, w \rangle : M \text{ is a PDA that accepts } w \}\.

- Any given context-free language (what does this question mean?)
A Universal Turing Machine

Theorem: There is a Turing machine $U$, such that when $U$ is given $\langle M, w \rangle$ for any TM $M$ and $w$, $U$ produces the same result (accept/reject/loop) as running $M$ on $w$.

Proof: Initially,

• First tape contains $\langle M \rangle$, including in particular its transition function $\delta_M$.

• Second tape contains $\langle w \rangle$.

• Third tape contains $\langle q_{\text{start}} \rangle$.

• Simulate steps of $M$ by multiple steps of $U$.

(Brief return to implementation description.)

$\Rightarrow$ Turing machines can be “programmed”.
From “On Computable Numbers” (1936)

6. The universal computing machine. It is possible to invent a single machine which can be used to compute any computable sequence. If this machine I is supplied with a tape on the beginning of which is written the S.D of some computing machine M, then I will compute the same sequence as M. In this section I explain in outline the behavior of the machine. The next section is devoted to giving the complete table for I.
The Mark I (1944): “Harvard Architecture”
The Institute for Advanced Study Machine (1946-51): “Von Neumann Architecture
Technological Consequences of Universal TMs

General-purpose, programmable computers:

- Single hardware can support all computing tasks.
- Arbitrary hardware can be represented as software programs (cf. virtual machines).
- Programs can be treated like data (von Neumann architecture).
Theoretical Consequences of Universal TMs

- \( A_{\text{TM}} = \{ \langle M, w \rangle : M \text{ accepts } w \} \) is Turing-recognizable.

- \( \text{HALT}_{\text{TM}} = \{ \langle M, w \rangle : M \text{ eventually halts on } w \} \) ("The Halting Problem") is Turing-recognizable.

- **Q:** Are these sets decidable?

- **Q:** Are there undecidable languages?