Review for Quiz 3 (Scribe: Anuja Chaudhary)
Chapters 4 & 5

Topics

- Turing machines
  - Definition
    1. Formal definition – transition function
    2. High-level description – English + Math notation
    3. Middle-level description – using modules (L, R, links to generate loops)
  - Configurations (q, uav)
    1. Computation path
  - Languages accepted by Turing machines
  - Functions computed by Turing machines

- Variants of Turing machines
  - K-tape Turing machines
    1. Definition
    2. Given a k-tape Turing machine, we can use a one-tape Turing machine to simulate its computation
  - RAM Turing machines
    1. Definition
    2. Given a RAM Turing machine with k registers, we can simulate its computation using a Turing machine with k+3 tapes
  - Nondeterministic Turing machines (NTM)
    1. Definition
    2. Acceptance notion
    3. Given a one-tape NTM, we can simulate its computation using a DTM with 3-tapes. Similarly, it is straightforward to extend the proof of this result to show that any given k-tape NTM can be simulated by a (2+k)-tape DTM. Then, from the previous results, we know that k-tape NTM can be simulated by a one-tape DTM.

- Decidability and semidecidability (the latter is also called Turing-acceptability or recursive enumerability)
  - Differences between decidability and semidecidability
    A language L is decidable iff L and \( \overline{L} \) are semidecidable.
• Universal Turing machines and Halting problems
  ➢ Encoding of Turing machines: \(<M>\)
  ➢ Halting problems
    1. \(H_0\) (also denoted by \(K\)), \(H\) (also denoted by \(HALT\) in textbook)
    2. \(H_0\) and \(H\) are not decidable, but they are Turing-acceptable.

• Reductions and properties related to decidability and semidecidability
  ➢ \(A \leq_m B\) via \(f\)
    \(f\): Turing computable
    \(\forall x: x \in A \iff f(x) \in B\)
  ➢ If \(A \leq_m B\) and \(B\) is decidable (Turing-acceptable), then \(A\) is decidable (Turing-acceptable).
    Corollary: If \(A \leq_m B\) and \(A\) is not decidable (Turing-acceptable), then \(B\) is not decidable (Turing-acceptable).
  ➢ Applications:
    1. \(H_0 \leq_m H \rightarrow H\) is not decidable.
    2. \(\overline{H} \leq_m L = \{<M_1><M_2> | M_1\text{ and }M_2\text{ are Turing machines and }L(M_1) = L(M_2)\}\)
       Similarly, we can show \(\overline{H} \leq_m \overline{L}\).
       Hence, both \(L\) and \(\overline{L}\) are not Turing-acceptable.

• Rice’s theorem
  ➢ Statement
  ➢ Applications

Rice’s theorem concerns about decidability of properties of r.e. sets.

Property – a set of r.e. sets
  e.g, \(\{\phi\}\), \(\{\Sigma^*\}\), \{context free languages\}

Let \(P\) denote a property. Then we consider
\[L_P = \{<M> | M \text{ is a Turing machine and } L(M) \in P\}\]

*Rice’s theorem*: If \(P\) is non-trivial (i.e., \(P \neq \phi\) and \(P \neq\) the set of all r.e. sets), then \(L_P\) is undecidable.

*Applications:*
Let \(P_1 = \{\phi\}\). Then \(P_1\) is not trivial. Thus,
\[L_{P_1} = E = \{<M> | M \text{ is a Turing machine and } L(M) = \phi\}\] is not decidable.
Similarly, we know by Rice’s theorem that
\[\{<M> | M \text{ is a Turing machine and } L(M) = \Sigma^*\}\]
is not decidable, and
\[\{<M> | M \text{ is a Turing machine and } L(M) \text{ is context free}\}\]
is not decidable.

*Note:* Rice’s theorem applies on set of all Turing machines $M$ with $L(M) \in P$. If we only look at a proper subset of these machines, then Rice’s theorem is no longer valid; e.g., the following set

$$\{<M>| M \text{ is a DFA and } L(M) = \phi\}$$

is decidable.

- Church-Turing thesis
  Any reasonable computation model can be simulated by Turing machines.

**Sample Quiz # 3**

**Problem # 7.** Let $H = \{<M, w>| M \text{ is a TM and } M \text{ accepts } w\}$ and $R = \{<M>| M \text{ is a TM and } L(M) \text{ is a regular language}\}$. Construct a reduction from $A_{\text{TM}}$ to $R_{\text{TM}}$.

**Solution.** Construct a reduction as follows:

$$f(<M, x>) = M_1,$$

where on input $w$, if $w$ is not in the form of $0^n1^n$, then $M_1$ simulates $M$ on $x$, and $M_1$ halts if $M$ on $w$ halts; if $w$ is in the form of $0^n1^n$, then $M_1$ halts. We have:

If $<M, x> \in H$, then $M$ on $x$ halts, then $M_1$ on $w$ halts (regardless whether $w$ is in the form of $0^n1^n$); this implies that $L(M_1) = \Sigma^* \in R$ (since $\Sigma^*$ is regular).

If $<M, x> \notin H$, then $M$ on $x$ does not halt. This implies that $M_1$ only halts on inputs of the form $0^n1^n$, and so $L(M_1) = \{0^n1^n | n \geq 0\} \notin R$.

Hence, $f$ is the reduction we are looking for.