Instructions. This is a closed-book examination. You have 75 minutes. Answer five (only) of the following eight questions, for 20 points each. (Choose the five questions which you are able to answer best in the available time. Don’t do more. Only five answers will be graded.) Turn in this page along with your answers, and with the numbers of the questions you chose to answer circled.

1. Define each of the following:
   (a) Turing machine,
   \[ M = (Q, \Sigma, \Gamma, \delta, q_0, B, F), \] where \ldots, see textbook.
   (b) Church’s Thesis,
   Any “computable” problem can be identified as a partial recursive function.
   (c) undecidable problem,
   A problem whose language is not recursive.
   (d) partial recursive function,
   A function \( f(i_1, i_2, \ldots, i_k) \), with integer valued arguments, which can be evaluated by a TM (input: \( 0^m10^i \), \( m = f(i_1, i_2, \ldots, i_k) \))
   (e) Post’s correspondence problem.
   Given a pair of lists \( A = (w_1, w_2, \ldots, w_m), B = (x_1, x_2, \ldots, x_m) \), of words from a language \( \Sigma^* \), does there exist a sequence of integers \( i_1, i_2, \ldots, i_k \), \( k \geq 1 \), such that:
   \[ w_{i_1}, w_{i_2}, \ldots, w_{i_m} = x_{i_1}, x_{i_2}, \ldots, x_{i_m}. \]

2. State Rice’s Theorem for
   (a) recursive sets (undecidability),
   For any non-trivial property \( \mathcal{P} \) of r.e. languages, \( L_\mathcal{P} \) is undecidable.
   (b) r.e. sets.
   Let \( \mathcal{P} \) be a property of r.e. languages. \( L_\mathcal{P} \) is r.e. iff:
   - \( L \in \mathcal{P} \) & \( L \subseteq L' \) for some r.e. \( L' \Rightarrow L' \in \mathcal{P} \)
   - \( L \in \mathcal{P} \) is infinite \( \Rightarrow \) there exists a finite subset of \( L \in \mathcal{P} \).
   - The set of finite languages in \( \mathcal{P} \) is enumerable.

3. Explain the notion of reduction \( P_1 \rightarrow P_2 \) (\( P_1, P_2 \) problems). Use a diagram to illustrate the mapping and the Yes/No instances.
   A reduction \( P_1 \rightarrow P_2 \) is mapping from the instances of \( P_1 \) to those of \( P_2 \).
   - What particular property must the mapping have?
     It must be an algorithm.
   - How is the mapping related to the Yes/No instances?
     It must map Yes (No) instances of \( P_1 \) to Yes (No) instances of \( P_2 \).
   - Show (by contradiction) that if there is a reduction \( P_1 \rightarrow P_2 \), and if the problem \( P_1 \) is undecidable, then \( P_2 \) is also undecidable.
     Suppose that \( P_2 \) is decidable. We claim that then, \( P_1 \) must also be decidable – a contradiction. To see way, combine the algorithm of the mapping \( P_1 \rightarrow P_2 \) with the decidability algorithm of \( P_2 \) (the composition of algorithms is an algorithm).

4. Define each of the following sets, using set notation, and specify whether the set is (a) recursive, (b) r.e., but not recursive, (c) not r.e. but the complement is r.e., or (d) not r.e., and the complement is not r.e. either. For example: \( L_e = \{ <M> | L(M) = \emptyset \} \) is not r.e., but its complement \( \overline{L_e} \) is r.e.
   - \( L_{ne}, \)
     \( L_{ne} = \{ <M> | L(M) \neq \emptyset \} \).
• $L_d$,
  \[ L_d = \{ w = \langle M \rangle \mid M \text{ does not accept } w \} \].

• $L_u$,
  \[ L_u = \{ \langle M, w \rangle \mid M \text{ accepts } w \} \].

<table>
<thead>
<tr>
<th>Language</th>
<th>Recursive</th>
<th>r.e.</th>
<th>not recursive</th>
<th>not r.e. but complement r.e.</th>
<th>not r.e. and complement not r.e.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_{ne}$</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>$L_d$</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>$L_u$</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

5. Give a direct diagonal proof that $L_d$ is not recursive.

Suppose that $L_d$ is r.e. and let $M_d$ be a TM that decides $L_d = L(M_d)$. Let $w_d = \langle M_d \rangle$.

• $w_d \in L_d$. Then on one hand the diagonal entry of $w_d$ in the diagonalization array must be 0 (by definition of $L_d$), on the other it must be 1 (by definition of value of the array entries, since $M_d$ accepts $w_d$).

• $w_d \not\in L_d$. Then on one hand the diagonal entry of $w_d$ must be 1 ($w_d \not\in L_d$), on the other it must be 0 ($M_d$ does not accept $w_d$ since $w_d \not\in L_d$).

In both cases we have a contradiction.

6. Prove that $L_u$ is not recursive, by reduction.

We know that $\bar{L}_d$ is not recursive (but r.e.) The reduction is $\bar{L}_d \rightarrow L_u$, which maps $w = \langle M \rangle$ to $(M, w)$. Then $w = \langle M \rangle$ is in $\bar{L}_d$ iff $(\langle M \rangle, w)$ is in $L_u$.

7. Prove or disprove: \{ $\langle M \rangle \mid M$ is a TM that accepts a finite set \} is r.e. Any result used must be clearly stated.

[Hint: the “property” is: $|L|$ is finite.]

Use Rice’s second theorem (see Question 2b above). In this case Condition 1 fails, because there are infinite superset of finite r.e. languages. So the language is not r.e.

8. Design a Turing machine program that will accept the set \{ $a^ib^ic^i \mid i \geq 1$ \}.

See: The diagram in the Solutions for HA3, Question 3.