UNDECIDABILITY PROBLEMS

* (1) Playing with PCP
(Adapted from John Martin, *Introduction to Languages and the Theory of Computation*, 20.3)

Say you are given the following five PCP dominos:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ab</td>
<td>ba</td>
</tr>
<tr>
<td>aba</td>
<td>abb</td>
</tr>
<tr>
<td>#1</td>
<td>#2</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>a</td>
</tr>
<tr>
<td>#3</td>
<td>#5</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ab</td>
<td>b</td>
</tr>
<tr>
<td>#4</td>
<td></td>
</tr>
</tbody>
</table>

a) Which domino(s) could be used first in a PCP solution? Why?

b) Which domino(s) could be used last in a PCP solution? Why?

c) Find a PCP solution with these dominos.
(Don’t spend too much time on this part if it’s taking too long!)

** (2) Silly PCP, Tricks are for Kids!

* a) In a variation of PCP, each domino the top string has the same length as the bottom string. Show that this variation of PCP is decidable.

** b) Prove that PCP is decidable over the unary* alphabet \( \sum = \{a\} \).

* Compare the word “unary” to “binary,” and note that the root for one is “un” (e.g. “unit,” “universal,”) while the root for two is “bi” (“biweekly”).
** (2) TMs can feel useless, too

(Adapted from Michael Sipser, *Introduction to the Theory of Computation, 2nd ed.*, Problem 5.13.)

A useless state in a Turing machine is one that is never entered on any input string.

Consider the problem of determining whether a Turing machine has any useless states.

a) Formulate this problem as a language:

\[ \text{USELESS}_\text{TM} = \]

b) Fill in the steps of the following proof that \( \text{USELESS}_\text{TM} \) is undecidable:

For contradiction, assume that \( \text{USELESS}_\text{TM} \) is decidable by TM R.

Construct a new TM S that uses R to decide \( A_{\text{TM}} \).

\( S \) creates a new TM \( T \) that has a useless state when \( M \) doesn’t accept \( w \), and does not have a useless state when \( M \) does accept \( w \).

\( S = \) “On input ________________ :

1. Construct a new TM \( T = \) “On input \( x \):

   a. Replace \( x \) on the input by the string \( <M, w> \)

   b. Run the universal TM \( U \) to simulate. (Note that \( U \) was designed to use all its states.)

   c. If \( U \) accepts, enter a special state \( q_A \) and accept.

2. Run \( R \) on ____________ to determine whether \( T \) has any useless states.

3. If \( R \) rejects, then \( M \) _______ (accepts/rejects) \( w \), so \( S \) _______ (accepts/rejects).

   Otherwise, \( S \) _______ (accepts/rejects).

If \( M \) accepts \( w \), then \( T \) enters all states, but if \( M \) doesn’t accept \( w \) then \( T \) avoids \( q_A \).

So \( T \) has a useless state, \( q_A \), if and only if \( M \) doesn’t accept \( w \).

Thus \( S \) decides \( A_{\text{TM}} \). Because \( A_{\text{TM}} \) is ____________, we have reached a ____________ and conclude that ____________.