Homework 2

Joe Baker, Brett Schreiber, Brian Knotten

January 17, 2018

3

Post's Correspondence Problem can be proved to be undecidable by reducing A_{TM} to PCP, where A_{TM} is the acceptance problem for a Turing Machine. This is done by transforming the Turing Machine and input string for A_{TM} into tiles as input for PCP as described in the Wikipedia page, and returning the value of PCP. However, there are a few details to work out, such as:

1. How to guarantee that the solution must start with a given block?
2. Dealing with boundaries between states.
3. What happens if PCP doesn't halt?
4. Can PCP have different results for the same input from A_{TM}?
5. Can the first move of the input Turing Machine be left?
6. What if the input string to the TM is ϵ?

Issue 1:
PCP needs the initial block to be first in its match

Solution:
Certain inputs to PCP can cause it to return a match with less tiles than what are necessary to create a valid TM computation history. To remedy this, create a new problem $MPCP$ which takes modified PCP parameters as input and outputs whether a match can be found with the tiles from PCP. $MPCP$ will work exactly the same as PCP except it will first modify each tile in the following ways:

- Transform the q_0 tile from $\frac{t_0}{b_0} \rightarrow \frac{*t_0}{*b_0*}$.
- Transform the rest of the tiles $q_1...q_N$ from $\frac{t_i}{b_i} \rightarrow \frac{*t_i}{b_i*}$.
- Add a new tile $q_{N+1} = \frac{*\square}{\square}$ to the input to $MPCP$ to consume the trailing new $*$ letter.

With this new configuration, $MPCP$ will need to match q_0 first since it is the only tile with $*$ to start, but any other tiles are not restricted further in matching. We now reduce A_{TM} to $MPCP$ to make sure the first tile comes first, and then reduce $MPCP$ to PCP to show that PCP is undecidable.

Issue 2:
Boundaries between states may not match up

Solution:
As described in the Wikipedia proof, the state strings are separated by a separator symbol (typically #), and there are both "copy blocks" for each symbol $a \in \Gamma$, and a "transition block" for each position transition the machine can make. Because each of these blocks either has an equal number of symbols on the top and bottom or more symbols on the bottom, two states on the top and bottom will never match in a way that cannot be understood. Additionally, as defined in the Wikipedia proof, no transition blocks contain symbols other than $*$ to the left of a state symbol

besides the q_f blocks that are not used until the top reaches an accept state and the bottom needs to catch up. Therefore the only way states can line up in an incorrect way is if the blocks are not a match according to PCP.

Issue 3:
What happens if PCP doesn't halt on a transformed input from A_{TM}

Solution:
If PCP doesn't halt on a transformed input from A_{TM} then A_{TM} won't halt and we will still have PCP if and only if A_{TM}.

Issue 4:
Can PCP have different results for the same input from A_{TM}?

Solution:
If PCP matches on an input from A_{TM} then it must match 1 or more copies of an acceptable computation history for A_{TM}. To show this, assume that PCP matches partially up to tile $\frac{t_i}{b_i}$. If the input is held constant for each time PCP is run, then PCP will always choose the same tile after $\frac{t_i}{b_i}$ so the results cannot differ. However, PCP can match the same correct computation history repeated more than one time. But in that case it will accept in any multiple of the computation history which is still the correct result and we still have A_{TM} accepting if and only if PCP accepts.

Issue 5:
Can the first move of the input Turing Machine be left?

Solution:
If the input to A_{TM} is $\langle M, w \rangle$ where M is a TM and w is the input string, and the first move of M is left past the end of the tape, the tiles for PCP could not simulate the machine M. However, in this case, we could replace M with an equivalent machine M' which is able to prevent this behavior but still have equivalent results to M with input w. So we can replace M with M' as input to PCP for the reduction.

Issue 6:
What if the input string to the TM is ϵ?

Solution:
If the input to A_{TM} is $\langle M, w \rangle$ where M is a TM and w is the input string, and the w is ϵ, then replace the input string in the first tile with $\square$.