Problem 1: Decidable Properties of Lesser Machines (40)

1. (10) Let PALINDROMEDFA be the language of all encoded DFAs that accept *s*R iff they accept *s*.  Prove that PALINDROMEDFA is a Turing-decidable language.
2. (15) Take the problem of the equivalence of the languages of two DFAs, formulate it *as*a language, and prove that language is Turing decidable by testing the two DFAs on all strings up to a certain size.  Calculate a size that works.
3. (15) A useless state in a machine is one that is never encountered on any input string.  Consider the problem of determining whether a pushdown automaton has any useless states.  Formulate this problem as a language, and show that it’s decidable.

Problem 2: Undecidable Properties of Turing Machines (25)

1. (10) Show that PALINDROMETM is undecidable.
2. (15) Consider the problem of determining whether a given state in a Turing machine is useless.  Formulate this problem as a language, and show that it's undecidable.

Problem 3: Variations are Important (35)

1. (15) Show that the Post Correspondence Problem is decidable over the unary alphabet Σ = {1}.

Proof: Construction

PCP has an alphabet Σ = {1} and and a match is a sequence of where so

Assume where:

* WLOG, we receive string and list of strings t and b
* We accept if s has a sequence of indices where
* We reject if all strings of have length greater than every string in or vice versa

Since the alphabet is only made of 1s we don’t need to worry about different characters and only the lengths. This means by following the above algorithm we will either find an accept state or reject without failing to halt.

1. (10) Show that the Post Correspondence Problem is undecidable over the binary alphabet Σ = {0, 1}.

PCP has an alphabet Σ = {0, 1} and where and are lists of strings, a match is a sequence of where so

Proof: Assume BWOC that is decidable

By definition of decidability let be its decider where:

* We receive string and list of strings t and b
* We accept if s has a sequence of indices where
* We reject if we can’t find a match

Then we build a where:

* We mimic and we receive Turing machine M that receives string s
* We construct a new Turing machine that,
  + Receives string and list of strings c and d
  + we reject if has a sequence of indices where
    - This insures that always decides
  + Otherwise simulates on and and mimics if accepts, rejects or fails to halt

Interrogate for a match using which causes to decide and creates a contradiction. Therefore must be undecidable

1. (10) In the Silly Post Correspondence Problem, the top string of each domino is equal in length to the bottom string of that same domino.  Show that SPCP is decidable.

Proof: Construction

SPCP has an any alphabet and where and are lists of strings and all the strings are equal length, a match is a sequence of and each string in and are equal so

Assume where:

* WLOG, we receive string and list of strings t and b
* We accept if s has a sequence of indices where (so if we have a match)
* We reject if where (so if the top and bottom of the dominos are not the same length)

Since the lengths of each pair (top and bottom of the domino) are the same, following the above algorithm we will either find an accept state or reject without failing to halt, since we only need to worry about whether the string matches.