Problem 1: Decidable Properties of Lesser Machines (40)

A. (10) Let $PALINDROME_{DFA}$ be the language of all encoded DFAs that accept s^R iff they accept s. Prove that $PALINDROME_{DFA}$ is a Turing-decidable language.
B. (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.
C. (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)

A. (10) Show that $PALINDROME_{TM}$ is undecidable.
B. (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)

A. (15) Show that the Post Correspondence Problem is decidable over the unary alphabet $\Sigma = \{1\}$.

Proof: Construction

PCP has an alphabet $\Sigma = \{1\}$ and $P = \{\frac{t_1}{b_1} \ldots \frac{t_k}{b_k}\}$ and a match is a sequence of $i_1 \ldots i_l$ where $t_{i1} \ldots t_{il} = b_{i1} \ldots b_{il}$ so

$$PCP = \{< P > | P \text{ an instance of Post Corr Problem with a match}\}$$

Assume PCP_{TM} where:

- WLOG, we receive string s and list of strings t and b
- We accept if s has a sequence of indices $i_1 \ldots i_l$ where $t_{i1} \ldots t_{il} = b_{i1} \ldots b_{il}$

- We reject if all strings of t have length greater than every string in b 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.

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

PCP has an alphabet Σ = {0, 1} and $P = \{\frac{t_1}{b_1} \ldots \frac{t_k}{b_k}\}$ where t and b are lists of strings, a match is a sequence of $i_1 \ldots i_l$ where $t_{i1} \ldots t_{il} = b_{i1} \ldots b_{il}$ so

$$PCP = \{< P > | P\ an\ \ instance\ of\ Post\ Corr\ Problem\ with\ a\ match\}$$

Proof: Assume BWOC that PCP_{TM} is decidable

By definition of decidability let M_{PCP} be its decider where:

- We receive string s and list of strings t and b
- We accept if s has a sequence of indices $i_1 \ldots i_l$ where $t_{i1} \ldots t_{il} = b_{i1} \ldots b_{il}$
- We reject if we can't find a match

Then we build a M_{accept} where:

- We mimic A_{TM} and we receive Turing machine M that receives string s
- We construct a new Turing machine $M_{Internal}$ that,
  - Receives string x and list of strings c and d
  - we reject if x has a sequence of indices $i_1 \ldots i_l$ where $c_{i1} \ldots c_{il} = d_{i1} \ldots d_{il}$
    - This insures that M always decides
  - Otherwise simulates M on s, c and d and mimics if M accepts, rejects or fails to halt

Interrogate $M_{Internal}$ for a match using M_{PCP} which causes M_{Accept} to decide A_{TM} and creates a contradiction. Therefore PCP_{TM} must be undecidable

C. (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 $P = \{\frac{t_1}{b_1} \dots \frac{t_k}{b_k}\}$ where t and b are lists of strings and all the strings are equal length, a match is a sequence of $i_1 \dots i_l$ and each string in t and b are equal so

$$SPCP = \{< P > | P\ an\ instance\ of\ Silly\ Post\ Corr\ Problem\ with\ a\ match\}$$

Assume $SPCP_{TM}$ where:

- WLOG, we receive string s and list of strings t and b
- We accept if s has a sequence of indices $i_1 \dots i_l$ where $t_{i1} \dots t_{il} = b_{i1} \dots b_{il}$ (so if we have a match)
- We reject if $|t_{ij}| \neq |b_{ij}|$ where $j = \{1,2 \dots |s|\}$ (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.