CSCI 4325
Practice for Quiz/Exam 5

Problem 1: In the following instance of the Post Correspondence Problem, is there a match? Describe your approach to the problem. Knowing that the problem is undecidable, try to explain where your approach might fail.

$$\left[\frac{b}{ab}\right],\left[\frac{a}{ab}\right],\left[\frac{aba}{ba}\right],\left[\frac{aba}{a}\right]$$

Problem 2: Which of the following problems about Turing machines are solvable, and which are undecidable? Explain your answers carefully.

(a) To determine, given a Turing machine M, a state q, and a string w, whether M ever reaches state q when started with input w from its initial state.

(b) To determine, given a Turing machine M and a symbol a, whether M ever writes the symbol a when started with the empty tape.

Problem 3: LBAs have many useful properties, such as A_{LBA} being decidable. It would be nice to show a TM is an LBA, however, this is not possible. Prove that it's undecidable to decide whether a TM is an LBA by a reduction from A_{TM} or $Halt_{TM}$. The LBA definition is on page 221. *LBA* = {<*M*>| *M is a TM that never uses the tape outside the input*}. Does some *TM M* ever write a character after the input?

Problem 4: Given some language L_4 with $\sum = \{a, b\}$ and $|Q| = 27$. If I claimed that an LBA M can not decide input "abbba", how can you **verify** for this specific instance (the numbers) that the input does cause an infinite loop?

Problem 5: Prove that the following language ($KHALT_{TM}$) is undecidable with a reduction from the halting problem: $HALT_{TM}$ = {<*M*, *w*> | *M* is a TM and *M* halts on input *w*}.

$KHALT_{TM}$ = {<*M*> | *M* is a TM that halts after at most *k* steps for some input *w* of length *k*}

Problem 6: Define the following problem as a language and prove that it is undecidable with a reduction from the halting problem. Knowing whether a Turing machine will ever output your name on the tape. The language is the set of all TMs that print your name. Reduce from $HALT_{TM}$.

Problem 7: Prove the following problems are decidable.

(a) Hampath = {<*G, s ,t*>: Graph *G* has a Hamiltonian path from vertex *s* to vertex *t*}.

(b) Vertex-Cover = {<*G, k*>: Graph *G* has a vertex cover of size *k*}.

(c) No-Vertex-Cover = {<*G, k*>: Graph *G* does not have a vertex cover of size *k*}.

(d) Path = {<*G, s, t*>: Graph *G* has a path from vertex *s* to vertex *t*}

(e) NoPath = {<*G, s, t*>: Graph *G* does not has a path from vertex *s* to vertex *t*}