CS331 Autumn 2001 End-term Examination

Time: 3 hrs | Total marks: 80

---

- ***The exam is open book and notes.***
- ***Results/proofs covered in class/problem sessions/assignments may simply be cited, unless specifically asked for.***
- ***Unnecessarily lengthy solutions will be penalized.***
- ***If you need to make any assumptions, state them clearly.***

1. *[15 marks]* Let $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \phi)$ denote a 1-tape Turing Machine that accepts by halting, $w \in \Sigma^*$ denote an input string, and $k$ denote a natural number greater than 0. We wish to know if there exists an algorithm that takes $(M, w, k)$ as input and returns YES if $M$ on input $w$ scans *at least* $k$ distinct tape squares, and returns NO otherwise. Note that $M$ may or may not halt on $w$.

   If you think such an algorithm exists, informally describe the pseudocode (alternatively, a Turing Machine) and argue why your algorithm (or Turing Machine) computes the right answer and halts for every $(M, w, k)$. If you believe such an algorithm does not exist, give a proof of undecidability.

2. *[7.5 + 7.5 marks]* Given languages $L_1, L_2$ and $L_3$, each with alphabet $\Sigma$, define $L_1/L_2/L_3$ as $\{w | w \in \Sigma^* \text{ and } \exists u \in L_2, \exists v \in L_3, \text{ such that } wuv \in L_1\}$.

   (a) If $L_1, L_2$ and $L_3$ are recursively enumerable, is $L_1/L_2/L_3$ necessarily r.e.?

   (b) If $L_1, L_2$ and $L_3$ are recursive, is $L_1/L_2/L_3$ necessarily recursive?

   In each case, you must either give a counterexample, or give a proof of your claim.

3. *[15 marks]* Consider the same definition of $L_1/L_2/L_3$ as above. Show that if $L_1$ is context-free, and $L_2$ and $L_3$ are regular, then $L_1/L_2/L_3$ is context-free.

4. *[10 + 10 marks]* Let $\Sigma$ be an alphabet. Suppose strings in $\Sigma^*$ are encoded using a Godel numbering scheme $gn$. Define $pos(n, i)$ as the Godel number of the $i^{th}$ symbol in the string $gn^{-1}(n)$ if $i \leq |gn^{-1}(n)|$, and 0 otherwise.

   (a) Show that $pos$ is a primitive recursive function.

   (b) Given $n$ and $m$, define the predicate $anagram(n, m)$, which evaluates to true if $gn^{-1}(m)$ is obtained by permuting the symbols of $gn^{-1}(n)$, and evaluates to false otherwise. Is $anagram$ primitive recursive?

5. *[15 marks]* Given context-free languages $L_1$ and $L_2$, show that it is undecidable whether $L_1 - L_2 = L_1 \cap L_2^c$ is a regular language.