15-354: Midterm

Thursday, October 9, 2025

Instructions

- You have until Friday, 10/10, 24:00, to work on this test.
- When you are done, email your pdf to me at `sutner@cs.cmu.edu`.
- You can consult all course materials, but no other sources.
- Do not talk to anyone about this exam.
- Post questions to ed, but make your posts private (unless it's about a typo or the like).
- Each question is 40 points, but their difficulty varies a bit. Pick off the easier ones first.
- Good luck.

Problem 1: Mutual Recursion (40 pts.)

We can define arithmetic functions f_0 and f_1 by mutual recursion as follows:

$$\begin{aligned} f_0(0, \boldsymbol{y}) &= g_0(\boldsymbol{y}) & f_1(0, \boldsymbol{y}) &= g_1(\boldsymbol{y}) \\ f_0(x^+, \boldsymbol{y}) &= h_0(x, f_1(x, \boldsymbol{y}), \boldsymbol{y}) & f_1(x^+, \boldsymbol{y}) &= h_1(x, f_0(x, \boldsymbol{y}), \boldsymbol{y}) \end{aligned}$$

A. Explain how to define the characteristic functions for the even and odd numbers this way.

B. Show that the f_i are primitive recursive whenever the g_i and h_i are.

Problem 2: Trim Turing Machines (40 pts.)

Let $\mathcal{M}$ be a Turing machine. Following terminology from finite state machines, call a state p of $\mathcal{M}$ useful iff it appears on at least one computation from an initial configuration to a halting configuration.

Similarly, call a tape cell $c \in \mathbb{Z}$ useful iff there is at least one computation from an initial configuration to a halting configuration that touches cell c.

A. Is it decidable whether a state p is useful?

B. Is it decidable whether cell $c = -42$ is useful?

In both cases, give a proof for your claim. Recall that by our input convention, the tape head is initially in position 0 and the input string is in positions 1 through n.

Problem 3: Tally Languages (40 pts.)

Languages $L \subseteq \{a\}^\star$ are referred to as tally languages. Hence, a tally language is essentially a subset of $\mathbb{N}$, dressed up as a language. Let $\gamma_L(n) = |L \cap \{\varepsilon, a, \ldots, a^{n-1}\}|$ be the census function for L. Recall that L is semidecidable iff there is a Turing machine that halts exactly on all $x \in L$.

A. Show that γ_L is computable whenever L is decidable.

B. Show that L is decidable whenever γ_L is computable.

C. For this part, **ignore** the result in (A). Assume that L is semidecidable and γ_L is computable. Show how to construct the sets $L \cap \{\varepsilon, a, \ldots, a^{n-1}\}$ directly.

Incidentally, the last part is not frivolous, it's important for Kolmogorov-Chaitin complexity.

Problem 4: More Tally (40 pts.)

We use the same terminology as in the last problem, but this time we consider recognizable L. Define the density of L to be $\Delta(L) = \lim_{n\to\infty} \gamma_L(n)/n$. For example, $\Delta\Big((aa)^\star\Big) = 1/2$.

A. Characterize the minimal DFAs that are associated with recognizable tally languages.

B. Explain how to compute the density $\Delta(L)$ for any regular recognizable language L.

C. Show that every infinite recognizable language L can be split into two infinite recognizable parts L^+ and L^-.

There are different approaches to part (C), you might want to use (A).

Problem 5: Interleaving Words (40 pts.)

The strict shuffle of two words x, y of the same length n over alphabet $\Sigma = \{a, b\}$ is defined as

$$x \parallel y = x_1 y_1 x_2 y_2 \dots x_n y_n$$

For words of different lengths, first right-pad the shorter one by a special marker $\# \notin \Sigma$, then use strict shuffle. E.g., $aab \parallel bba = ababba$ and $aa \parallel bbab = abab\#a\#b$. Lift to languages as usual: $K \parallel L = \{ x \parallel y \mid x \in K, y \in L \}$.

For a relation $R \subseteq \Sigma^* \times \Sigma^*$, define the language

$$R^{\parallel} = \{ x \parallel y \mid R(x, y) \} \subseteq (\Sigma \cup \#)^\star$$

A. Let R be the relation "x is a prefix of y". Show that $R^{\parallel}$ is recognizable.

B. How about the relation "x is a suffix of y". Justify your answer.

C. Show that $K \parallel L$ is recognizable whenever both K and L are.

D. Prove that your construction works as advertised.

For part (C), if you decide to construct a finite state machine that recognizes $K \parallel L$, do the following. First explain in words what the idea behind your construction is, then carefully specify

- the state set,
- the transitions,
- the initial/final states.

Do it in this order.