CS 581: Theory of Computation
James Hook
Mid-term exam

This is a closed-notes, closed-book exam.

1. Characterizing Computational Problems as Languages

   Consider the following computational problems. For each problem give the corresponding language recognition problem. Without proof, attempt to classify the language as regular, context free but not regular, decidable but not context free, recognizable but not decidable, or not recognizable. For example, the problem of "equality of natural numbers" might be represented by $\{1^n\#1^n | n \geq 0\}$. This language is context free but not regular.

   (a) Multiplication of natural numbers

   (b) Equality modulo 3

   (c) Addition of natural numbers

   (d) The factorial function

   (e) Natural numbers that encode Turing machines that accept when given their own encoding as input.

2. We have explored 3 different ways to show that a language is *not* regular: apply the pumping lemma to contradict regularity, show the language has infinite index, and use closure properties to derive a contradiction. In this problem illustrate all three techniques to show that the language of balanced parentheses is not regular.

   (a) Use the pumping lemma to show the language of balanced parentheses is not a regular language.

   (b) Show the language of balanced parentheses is of infinite index.

   (c) Use closure properties to show that if the language of balanced parentheses is regular then so is $a^n b^n$.

3. Right Linear Grammars

A grammar G is *right linear* if every production is of one of the following forms: $S \to \epsilon$, $A \to aB$, or $A \to a$ (where S is the sentential symbol, A and B are arbitrary variables, and a is an arbitrary terminal).

For example the following grammar generates 0^+1^+:

$$\begin{array}{rcl} S & \to & 0A \\ & | & 0B \\ A & \to & 0A \\ & | & 0B \\ B & \to & 1B \\ & | & 1 \end{array}$$

Show that L is generated by a right linear grammar if and only if L is a regular language.

Suggestion: In my construction I show that if L is of finite index then L is generated by a right linear grammar and I show that if L is generated by a right linear grammar then L is recognized by an NFA. You are free to use other approaches.