Formal methods, Fall 2008
Final
Assigned: 12/6, Due: 12/15 at 5:00pm, in my office or my box

1 Regular and context free grammars

For each of the following regular languages over $\Sigma = \{0, 1\}$, give the smallest right linear grammar you can construct that generates it (where we measure size in terms of number of rules). Then, give the smallest context-free grammar you can construct that generates each one.

[**a**] $\{w | w$ contains at least 4 1s$\}$

[**b**] $\{w | w$ starts and ends with the same symbol$\}$

What do you notice about the relative size of these grammars (in terms of the number of rules and non-terminals they require)? What does this tell us about the difference between RLGs and CFGs apart from the sets of languages they can generate?

2 Turing machines

Let $L = \{w \mid w$ contains twice as many 0s as it does 1s$\}$.

[**a**] Is L context free? If it is, give a PDA that recognizes it. If it isn't, show (using the pumping lemma for CFLs) that it isn't.

[**b**] Design a (deterministic) Turing machine (including both a sketch of the algorithm, and implementation details) that decides L.

[**c**] Give a type 0 grammar that generates L (see the corrected version of the previous homework for the definition of a type 0 grammar).

3 More on the halting problem

An oracle is a (potentially magical) device that makes a yes or no decision on whether a string is in a particular language – this device is a kind of black box. For instance, an oracle for A_{TM} could tell you whether a string is a member of that language. An oracle machine is a Turing machine with the additional ability to consult a particular oracle as a step of the computation. This is a mathematical abstraction and could not exist. It is easy to see that an oracle machine with the oracle for A_{TM} could decide A_{TM}, hence this is an example of a magical oracle. (One way of construing the proof that A_{TM} is undecidable is that it says that no such oracle could actually exist. Whereas, a decider for any given language basically can be viewed an oracle for that language.)

[**a**] Show that an oracle machine with an oracle for A_{TM} could decide $\overline{A_{TM}}$. (Consequently, oracle machines can decide even some languages that are not Turing recognizable.)

[**b**] Show that an oracle machine with an oracle for $HALT_{TM}$ could decide A_{TM}.

[**c**] (Grads, but I encourage everyone to think about this) Let a TMA machine be an oracle machine that can consult an oracle for A_{TM}. Show that a TMA machine could not decide

$$A_{TMA} = \{\langle M, w\rangle | M \text{ is a TMA machine that accepts on input w}\}.$$

The proof is **very** similar to the proof in class that A_{TM} is undecidable.

(This result continues all the way up, resulting in an infinite hierarchy; adding oracles to any given machine can allow you to decide some undecidable languages, but not all. It may be of interest to think about consequences for arguments, due to e.g. Lucas, Penrose, that the human mind (/cognition) is more powerful than a Turing machine – that humans can decide undecidable problems. It is not obvious that even if this could conceivably be true, a mind can have unfettered power. That is, even giving a computing device magical powers provably does not give it unlimited computing ability.)

The end!