CSL 705 - Major, Wednesday, April 30, 2008

The exam is open notes: you may use results from the course or from exercises solved in class or in tutorial. Of course, you need to be specifie about which known result you use.

Read the whole paper before you get started. To solve Problem i, you may also use the result of Problem j ($j < i$), even if you did not solve it.

Point assignment: 20 - 15 - 10 - 10 - 10 - 10 - 25 - 15 (sum = 115)

This set of problems is about *linear* languages, a class of context-free languages. A context-free grammar (CFG) $G = (N, \Sigma, P, S)$ is called *linear* if for every production $A \to \alpha$ in P, α has at most one occurrence of a non-terminal.

For instanee, the first and fourth of the CFGs below are linear, the second and the third are not.

1. $S \to aSb + ab$
2. $S \to SS + aSb + ab$
3. $S \to ASB + ab$; $A \to a$; $B \to b$
4. $S \to aT + ab$; $T \to Sb$.

A context-free language (CFL) L is *linear* if there exists a linear CFG G such that $L = L(G)$. For instance, the language generated by the third grammar above is linear, even though that grammar is not, because it is also generated by the first grammar (and by the fourth as well).

Caution: We discussed *right linear* and *left linear* grammars in class and we showed that a language is generated by a right linear grammar (respectively by a left linear grammar) if and only if that language is regular. Right linear and left linear grammars are linear, but the converse is not true.

Problem 1 The first problem is actually not about linear languages[1], but about an undecidability result that is reminiscent of Rice's theorem. Rice's theorem is about properties of recursively enumerable languages and states that almost all of them are undecidable. This result is about properties of context-free languages instead, and shows that many are undecidable.

[1] however there is a link, as you will see

Let P be a non-trivial property of context-free languages, such that every regular language has property P and such that if L has property P, then so does La^{-1} for each $a \in \Sigma$ (recall that $La^{-1} = \{u \in \Sigma^* \mid ua \in L\}$).

Show that it is undecidable, given a CFG G, whether $L(G)$ has property P.

[Hint: You may use (a) the map $L \longmapsto L_0\#\Sigma^* \cup \Sigma^*\#L$, where L_0 is a CFL that does not have property P (there exists such a language since P is non-trivial), and (b) the fact that it is undecidable whether $L(G) = \Sigma^*$ (seen in class).]

Problem 2 Show the following variant of the pumping lemma for linear languages: *If L is a linear language, there exists an integer k such that, for all $w \in L$ with $|w| \geq k$, there exist strings x, u, v, y, z such that $w = xuyvz$ with $|xuvz| \leq k$, $|uv| > 0$ and $xu^iyv^iz \in L$ for each $i \geq 0$.*

Problem 3 Show that $L = \{a^nb^nc^md^m \mid n, m \geq 0\}$ is context-free but not linear.

Problem 4 Show that if L is linear and $a \in \Sigma$, then La^{-1} is linear. (It is enough to give a correct grammar, you don't need to justify your answer.)

Problem 5 Show that it is undecidable, given a CFG G, whether $L(G)$ is linear.

Problem 6 Show that if L is linear and R is regular, then LR is linear. (It is enough to give a correct grammar, you don't need to justify your answer.)

Problem 7 Show that it is undecidable, given a linear CFG G, whether $L(G) = \Sigma^*$.

[Hint: you may use (a) a reduction from EMPTY (to decide whether a given Turing machine accepts any string at all) and (b) the idea of encoding invalid computation histories of a Turing machine, under the following form:

$$\#\alpha_0\#\alpha_1\#\cdots\#\alpha_N\#\#\widetilde{\alpha_N}\#\cdots\#\widetilde{\alpha_1}\#\widetilde{\alpha_0}\#$$

where α_0 is the initial configuration, α_N an accepting configuration, α_{i+1} is the next configuration from α_i and $\widetilde{u}$ is the mirror image of the string u.]

Problem 8 State and sketch a proof of an analogue of the undecidability result in the first question, for non-trivial properties of linear languages (instead of context-free languages).