CS 154 Introduction to Automata and Complexity Theory	Handout 9
Winter 1996 R. van Glabbeek	February 7, 1996

Problem Set 5 Date Due: February 14, 1996

Reading: Floyd and Beigel, Chapter 0-6 to be read by Wednesday February 14'th.

Homework: Do the following exercises (FB refers to Floyd and Beigel *The Language of Machines*). Each problem must be submitted on a separate sheet. The homeworks must be solved without any collaboration.

Remember: The midterm is 2- 5 pm Friday February 16'th at the Annenberg Auditorium in the Cummings Art Building. It will cover Chapters 1-6.

1. (10 points) 5.1-1 page 321.
2. (5 points) 5.2-8 page 328.
3. (10 points) Prove that the following are not CFLs:
   (a) $\{a^i b^j c^k \ : \ i < j < k\}$
   (b) $\{a^{i^3} \ : \ i \geq 0\}$.
4. (10 points) Give a context free grammar for the *complement* of the following language:
$$\{a^n b^n \ : \ n \geq 0\}$$
5. (10 points) 5.9-4 page 374.

Extra credit problem (optional)

Consider the simultaneous regular equations over the alphabet Σ:

$$\begin{aligned} X &= f(X,Y) \\ Y &= g(X,Y) \end{aligned}$$

Let L_X and L_Y denote the context free languages that are obtained as the simultaneous least solution to the equations above. Thus, they satisfy:

$$(L_X, L_Y) = \bigcap\{(X,Y) \ : \ f(X,Y) \subseteq X, g(X,Y) \subseteq Y\},$$

where $\bigcap$ is defined as on page 326. Consider on the other hand the languages

$$L_A = \bigcap\{X \subseteq (\Sigma \cup \{y\})^* \ : \ f(X,y) \subseteq X\}$$

where y is a fresh character not in Σ. And define

$$L_A(L) = \{w \in \Sigma^* \ : \ \begin{array}{l}(\exists v \in L_A, u_1, \ldots, u_i, \ldots \in L) \\ w \text{ is obtained from } v \text{ by inserting} \\ u_1, \ldots, u_i, \ldots \text{ at the positions where } y \text{ occurs } \}\end{array}$$

and define

$$L_B = \bigcap\{Y \ : \ g(L_A(Y), Y) \subseteq Y\}$$

Show that

$$L_Y = L_B$$

$$L_X = L_A(L_B)$$

The properties you are asked to show establish that nested fixpoints can replace simultaneous fixpoints. Hence to solve a system of least fixpoints it suffices to solve one fixpoint treating the other variables as constants, and solve the other equations using the partial solution. The above correspondence applies equally well to context free grammars as programming languages with constructs for recursion.