

CSE105 (Winter 2014): Homework 5

Instructor: Daniele Micciancio

Due on Wednesday, February 19, 2014.

Solutions to this homework assignment should consist of a writeup with your answers to problems 1 and 5 (to be submitted in class on the due date) and jflap files `HW5p1.jff`, `HW5p2a.jff`, `HW5p2b.jff`, `HW5p3.jff`, `HW5p4.jff`, `HW5p5b.jff`, `HW5p5a.jff`. Zip the files together and submit the zipped file using turnin.

1 CFG design

In this problem you are asked to design a context free grammar for a fragment of a typical programming language. Below we give an informal description of the programming language. Your task is to formalize the description into a context free grammar. As part of this assignment you should submit a jflap file `HW5p1.jff` containing the grammar, and a brief English description of the grammar, and how you came up with it.

The language: we consider a subset of a typical programming language with (integer valued) functions and multidimensional arrays, in which you can write expressions of the form $f(1, a[g(0)][b[1]])$.

- The syntax for functions is $f(x_1, \dots, x_n)$, where f is the function name, and $x_1, \dots, x_n$ the arguments. Functions can take any number of inputs. Calls to functions with no inputs are written as $f()$.
- The syntax for multidimensional arrays is $a[x_1][x_2] \dots [x_n]$. An array can take any number $n \geq 1$ of indexes.
- Functions calls and arrays can be arbitrarily nested, i.e., the result of a function call or array lookup can be used as a function input or array index.
- For simplicity, assume f, g are the only possible function names, a, b the only possible array names, and $0, 1$ the only possible numerical constants.
- Function and array names can be “overloaded”, i.e., you can use the same function name or array with different number of arguments. For example, $f(0, 1, f(0))$ and $a[a[1]][f(f(0))]$ are valid expressions.

You should give a grammar for the language of all expressions of the form $f(1, a[g(0)][b[1]])$, $f(0, 1, f(0))$ and $a[a[1]][f(f(0))]$, following the above rules. Examples of invalid expressions (that should not be generated by the grammar) are: $f(a)[0]$, $a[0, 1]$, $a[]$, $f(b(0, 1))$, etc.

2 Push Down Automata

For each of the following languages design a corresponding push down automaton, and submit your answer as jflap files `HW5p2a.jff`, `HW5p2b.jff`.

(a) The set of all strings over the alphabet $\{a, b\}$ that contain an unequal number of a 's and b 's (in any order). For example, $abbab, a, bba$ are in L , while $\epsilon, ab, baab$ are not.

(b) The *complement* of the following language (over the alphabet $\{a, b\}$):

$$L = \{a^k b^n a^m : 0 \leq k \leq n < m\}.$$

Examples of strings in L are $aabbaaaa$, $abbaaa$, baa , aa . Examples of strings not in L are ϵ , $abab$, $abbaa$, $aabbb$. *Notice:* you are asked to design a PDA for the **complement** of L , so your PDA should accept the strings not in L , and reject the strings in L !

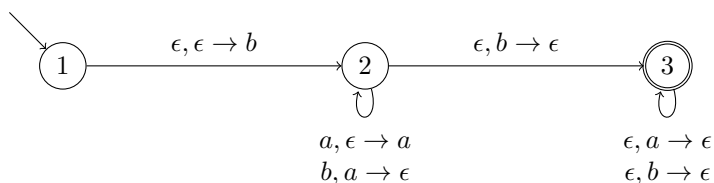
3 From CFG to PDA

Convert the following CFG into equivalent PDA using the procedure in the textbook. Submit your solution as jflap file HW5p3.jff. As an exercise, you should perform the transformation by hand, and then draw the automaton in jflap, but you are allowed to use jflap extended PDA operations, including the possibility of pushing several symbols on the stack in a single step.

$$\begin{aligned} S &\rightarrow aSaX \mid XbSb \mid \epsilon \\ X &\rightarrow c \mid Xc \end{aligned}$$

4 From PDA to CFG

Transform the following PDA into an equivalent CFG using the procedure in the textbook, and submit it as a jflap file HW5p4.jff. Notice that the automaton already has the properties that every step is either a push or a pop (but not both), and it empties the stack upon accepting the input.



5 Closure properties

Let the *reverse* of a language be $L^R = \{w^R \mid w \in L\}$, where the reverse of a word w^R is the word w written backward. In this problem you will give two different proofs that the reverse of a context free language is context free.

(a) First prove the result using context free grammars, i.e., show how to transform any context free grammar G into another grammar G' such that the language generated by G' is the reverse of the language generated by G . Include the description of a general transformation procedure in your writeup, and exemplify it on the grammar of problem 3. Include your example grammar both as part of your writeup, and as jflap file HW5p5a.jff.

(b) Next prove the result using push down automata, i.e., show how to transform any PDA P into another PDA P' such that the language generated by P' is the reverse of the language generated by P . Include the description of a general transformation procedure in your writeup, and exemplify it on the PDA of problem 4. Include your example PDA both as part of your writeup (as a state diagram), and as jflap file HW5p5b.jff.