COMS22201: Language Engineering

Lab Exercises - Week 14

01/02/2016, csxor@bristol.ac.uk

This worksheet provides you with some practice on grammars, proofs, invariants and Haskell data types.

1. Consider the following C program, which uses the variable `y` to compute the the factorial of the value stored in the variable `x`:

   ```
   int y=x; while (x --> 1) y *= x;
   ```

   The syntax '`-->`' is intended to represent a 'down-to' operator that decrements the variable on its left (and returns `true`) until that variable falls below the value on the right (when the operator returns `false`).

   (a) First explain how this program actually works, given that the C language does not officially support such a 'down-to' operator.

   (b) Now use a loop invariant to prove this program does in fact compute `x`! for all $\texttt{x} > 0$.

2. Consider the language of signed decimal numerals $(\ldots, -1, 0, 1, 2, 3, \ldots)$

   (a) Write an EBNF grammar for this language which ensures there is exactly one numeral representing each and every integer.

   (b) Convert your grammar to BNF.

   (c) Explain how you can represent such numerals using Haskell data types.

   (d) Write a Haskell function that computes the integer associated with each such numeral.

3. Prove for all $n > 0$ there is a legal English sentence of the form

   $(\text{Buffalo})^n$.

   In other words prove there is an infinite sequence of grammatically correct sentences of the form

   Buffalo. Buffalo buffalo. Buffalo buffalo buffalo. ...

   Hint: the word 'buffalo' can be a noun (i.e. a bison-like animal), an adjectival noun (i.e. relating to the city of Buffalo in the state of New York), or a verb (meaning to bully or intimidate)!

4. Observe for all $n > 0$ there is a legal English sentence of the form

   A white male (whom a white male)^n (hired)^n hired a white male.

   Use this fact to prove that English is *not* a regular language.