HW 9 – Parse Trees and Ambiguous Grammars

CS 421 – Fall 2015

Revision 1.0

Assigned Wednesday, October 28, 2015
Due Wednesday, November 4, 2015, 23:59 PM
Extension 48 hours (20% penalty)

1 Change Log

1.0 Initial Release.

2 Turn-In Procedure

For the main problem, answer the problem below, save your work as a PDF (either scanned if handwritten or converted from a program), add the PDF to the subversion repository (`svn add hw9-submission.pdf`) and commit it (`svn commit -m "submitting hw9"`). Your file should be named `hw9-submission.pdf` and committed in your `assignments/hw9` directory.

3 Objectives and Background

The purpose of this HW is to test your understanding of

- BNF grammars
- Grammar disambiguation
- Parse trees

Another purpose of HW9 is to provide you with experience answering non-programming written questions of the kind you may experience on the second midterm and final.
Caution: It is strongly advised that you know how to do these problems before the second midterm.

4 Problems

1. (23 points) Consider the following grammar over the terminal alphabet { `if`, `then`,`else`, **+**, `a`, `b`, `true`, `false`, (,)} and non-terminal alphabet {`<exp>`, `<bool>`, `<var>`}:

$$
\begin{array}{lcl}
<\texttt{exp}> & ::= & <\texttt{var}> \mid \texttt{if} <\texttt{bool}> \texttt{then} <\texttt{exp}> \texttt{else} <\texttt{exp}> \mid \\
 & & <\texttt{exp}> + <\texttt{exp}> \mid (<\texttt{exp}>) \\
<\texttt{bool}> & ::= & \texttt{true} \mid \texttt{false} \\
<\texttt{var}> & ::= & \texttt{a} \mid \texttt{b}
\end{array}
$$

(This is a grammar for a simple language of types similar to those in OCaml.)

a. (9 points) Show that the above grammar is ambiguous by showing at least two distinct parse trees for the string “ `if true then a else b + a`”

b. (9 points) Write a new grammar accepting the same language that is unambiguous, and such that the sum type constructor < `exp` > + < `exp` > binds more tightly than the if-then-else type constructor `if` < `bool` > `then` < `exp` > `else` < `exp` >, and such that + associates to the left.

c. (5 points) Give the parse tree for “ `if true then a else b + a` ” using the grammar you gave in the previous part of this problem.