HW 2 – Unification, Regular Expressions, Parse Trees and Ambiguous Grammars

CS 421 – Fall 2008

Revision 1.0

Assigned Tuesday, October 20, 2009
Due Tuesday, October 27, 2008, 2:00 PM - in class
Extension No extension, due to proximity to second midterm

1 Change Log

1.0 Initial Release.

2 Turn-In Procedure

Your answers to the following questions are to be hand-written, or printed, neatly on one or more sheets of paper, each with your name in the upper right corner. The homework is to be turned in in class at the start of class. Alternately, you may hand it to Prof. Elsa Gunter in person before the deadline.

3 Objectives and Background

The purpose of this HW is to test your understanding of

- How to unify a system of equations
- How to use regular expressions and regular grammars to formally express sets of strings (called *languages*) given by an English language description
- How to create a parse tree for a given string with a given grammar
- How to disambiguate a grammar

Another purpose of HW2 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. (18 points) Give a most general unifier for the following set of equations (unification problem). Capital letters (A, B, C, D, E) denote variables of unification. The lower-case letters (f, l, n, p) are constants or term constructors. (f and p have arity 2 - i.e., take 2 arguments, l has arity 1, and n has arity 0 - i.e. it is a constant.) Show all your work by listing the operations performed in each step of the unification and the result of that step.

$$\{(p(A, C) = p(B, D)); (l(C) = E); (l(f(n, A)) = E); (l(f(A, B)) = E)\}$$

Solution:

Rule	Resulting Equations / Substitution
Given	$\{(p(A,C) = p(B,D)); (l(C) = E); (l(f(n,A)) = E); (l(f(A,B)) = E)\}$with identity substitution
Decompose	$\{(A = B); (C = D); (l(C) = E); (l(f(n,A)) = E); (l(f(A,B)) = E)\}$with identity substitution
Elimiate	$\{(C = D); (l(C) = E); (l(f(n,B)) = E); (l(f(B,B)) = E)\}$with $\{A \mapsto B\}$
Elimiate	$\{(l(D) = E); (l(f(n,B)) = E); (l(f(B,B)) = E)\}$with $\{C \mapsto D; A \mapsto B\}$
Orient	$\{(E = l(D)); (l(f(n,B)) = E); (l(f(B,B)) = E)\}$with $\{C \mapsto D; A \mapsto B\}$
Elimiate	$\{(l(f(n,B)) = l(D)); (l(f(B,B)) = l(D))\}$with $\{E \mapsto l(D); C \mapsto D; A \mapsto B\}$
Decompose	$\{(f(n,B) = D); (l(f(B,B)) = l(D))\}$with $\{E \mapsto l(D); C \mapsto D; A \mapsto B\}$
Orient	$\{(D = f(n,B)); (l(f(B,B)) = l(D))\}$with $\{E \mapsto l(D); C \mapsto D; A \mapsto B\}$
Elimiate	$\{(l(f(B,B)) = l(f(n,B)))\}$with $\{D \mapsto f(n,B); E \mapsto l(f(n,B)); C \mapsto f(n,B); A \mapsto B\}$
Decompose	$\{(f(B,B) = f(n,B))\}$with $\{D \mapsto f(n,B); E \mapsto l(f(n,B)); C \mapsto f(n,B); A \mapsto B\}$
Decompose	$\{(B = n); (B = B)\}$with $\{D \mapsto f(n,B); E \mapsto l(f(n,B)); C \mapsto f(n,B); A \mapsto B\}$
Elimiate	$\{(n = n)\}$with $\{B \mapsto n; D \mapsto f(n,n); E \mapsto l(f(n,n)); C \mapsto f(n,n); A \mapsto n\}$
Discard	$\{\}$with $\{B \mapsto n; D \mapsto f(n,n); E \mapsto l(f(n,n)); C \mapsto f(n,n); A \mapsto n\}$

The final unifying substitution is $\{B \mapsto n; D \mapsto f(n,n); E \mapsto l(f(n,n)); C \mapsto f(n,n); A \mapsto n\}$.

2. (16 points) For each of the following languages (ie, sets of strings), write a regular expression and a regular grammar generating the set:

   a. (8 points) The set of all strings of `a`'s, `b`'s, and `c`'s such that for each string, between any `a` and a following `b` there is an intervening `c`.
   Solution: regular expression: $((\mathtt{b} \vee \mathtt{c}) * (\varepsilon \vee \mathtt{a} * \mathtt{c})) * \mathtt{a}*$
   regular grammar:

   $$\begin{array}{ll} \mathbf{S} & ::= \varepsilon | \mathtt{b}\mathbf{S} | \mathtt{c}\mathbf{S} | \mathtt{a}\mathbf{C} \\ \mathbf{C} & ::= \mathtt{a}\mathbf{C} | \mathtt{c}\mathbf{S} \end{array}$$

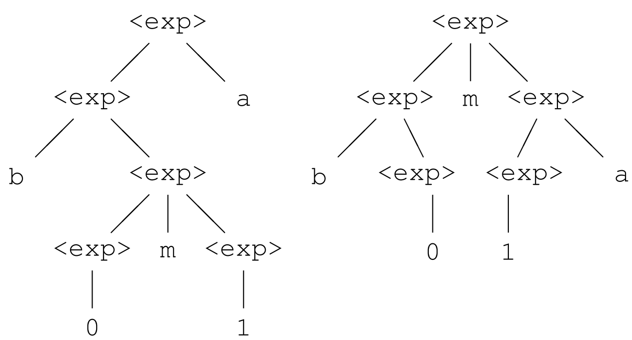
   b. (8 points) The set of all strings of `0`'s, and `1`'s, such that in each string any substring of consecutive `1`'s has length at most three.
   Solution: regular expression: $\mathtt{0} * ((\mathtt{1} \vee \mathtt{11} \vee \mathtt{111})\mathtt{00}*) * (\varepsilon \vee \mathtt{1} \vee \mathtt{11} \vee \mathtt{111})$
   regular grammar:

   $$\begin{array}{ll} \mathbf{S} & ::= \varepsilon | \mathtt{0}\mathbf{S} | \mathtt{1}\mathbf{O} \\ \mathbf{O} & ::= \varepsilon | \mathtt{0}\mathbf{S} | \mathtt{1}\mathbf{W} \\ \mathbf{W} & ::= \varepsilon | \mathtt{0}\mathbf{S} | \mathtt{1}\mathbf{R} \\ \mathbf{R} & ::= \varepsilon | \mathtt{0}\mathbf{S} \end{array}$$

3. (21 points) Consider the following grammar over the alphabet $0, 1, a, b, m$:

   $$< \mathtt{exp} > ::= \ \mathtt{0} \mid \mathtt{1} \mid \mathtt{b} < \mathtt{exp} > \ \mid \ < \mathtt{exp} > \mathtt{a} \mid \ < \mathtt{exp} > \mathtt{m} < \mathtt{exp} >$$

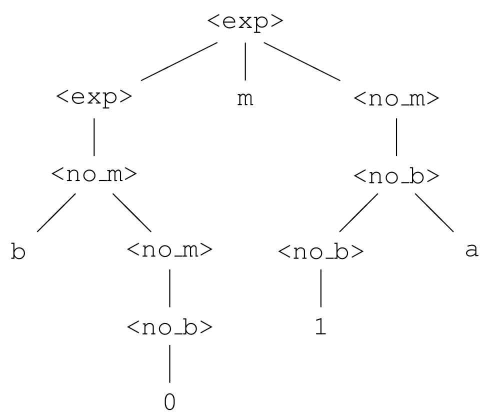
   a. (8 points) Show that the above grammar is ambiguous by showing to distinct parse trees for the string `b0m1a`.
   Solution:



b. (8 points) Write a new grammar accepting the same language that is unambiguous, and such that a has higher precedence than b, which in turn has higher precedence than m, and such that m associates to the left.
Solution:

$$\begin{array}{rcl} < \texttt{exp} > & ::= & < \texttt{exp} > \texttt{m} < \texttt{no\_m} > \mid < \texttt{no\_m} > \\ < \texttt{no\_m} > & ::= & \texttt{b} < \texttt{no\_m} > \mid < \texttt{no\_b} > \\ < \texttt{no\_b} > & ::= & < \texttt{no\_b} > \texttt{a} \mid \texttt{0} | \texttt{1} \end{array}$$

c. (5 points) Give the parse tree for b0m1a using the grammar you gave in the previous part of this problem.
Solution:



4. (Extra Credit) (6 points) Write a regular expression that generates the same language as the grammar given in problem 4.
Solution: $((\texttt{b}*)(\texttt{0} \vee \texttt{1})(\texttt{a}*)\texttt{m}) * (\texttt{b}*)(\texttt{0} \vee \texttt{1})(\texttt{a}*)$.