

CS536 Homework 5 Solution

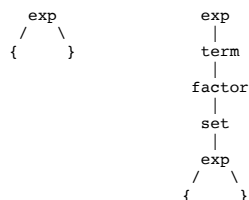
CFG 1:

This CFG doesn't allow empty list. Here is a string that is a legal set expression but is **not** in the language of the CFG:

{ }

CFG 2:

This CFG is ambiguous. Below are two different parse trees for the string { }. Different parse trees can also be built for expressions with operators so that one tree groups the subexpressions correctly (according to the precedence and associativity rules) and the other does not.



This CFG also allows a list of names to start with a comma. Here is a string that is **not** a legal set expression but is in the language of the CFG:

{ , NAME }

CFG 3:

This CFG doesn't allow parentheses around the right-hand operand of a union or intersection. Here is a legal set expression that is not in the language of the CFG:

{ } ∪ ({ })

CFG 4:

This CFG doesn't allow two or more unions in a row. Here is a legal set expression that is not in the language of the CFG:

{ } ∪ { } ∪ { }

It also doesn't allow a union to the right of an intersection. Here is another legal set expression that is not in the language of the CFG:

{ } ∩ { } ∪ { }

CFG 5:

This CFG is correct