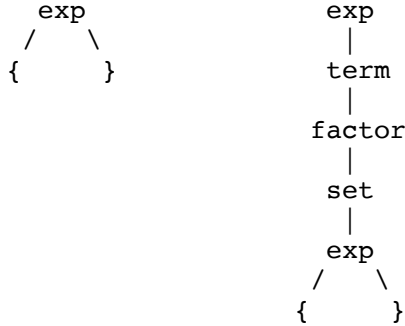


CS536 Homework 3 Solution

Question 1:

CFG 1:

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.



CFG 2:

This CFG only allows parentheses around set literals (lists of countries inside curly braces). Here are two legal set expressions that are not in the language of the CFG:

$(\{ \} \cap \{ \})$

$\{ \} \cup (\{ \} + \{ \})$

CFG 3:

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

$\{ \# \text{ COUNTRY} \}$

CFG 4:

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:

$\{ \} \cup (\{ \})$

CFG 5:

This CFG only allows parentheses around the very outside of the expression. Here are two legal set expressions that are not in the language of the CFG:

$\{ \} + (\{ \})$

$(\{ \}) \cup (\{ \})$

CFG 6:

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

$\{ \} \cap \{ \} \cap \{ \}$

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

$\{ \} \cup \{ \} \cap \{ \}$ **Question 2:**

Nonterminal X	FIRST(X)	FOLLOW(X)
program	{	EOF
stmts	ID, if, ϵ	}
stmt	ID, if	}, ID, if
exp	ID	;,)
tail	+, -, ϵ	;,)