CS 381 Homework 3 – Syntax

Submit a pdf for problems 1 – 4 and a Haskell *.hs file for problem 5.

1. Using the grammar below, show a parse tree and a leftmost derivation for the sentence

A = B * (C+A)

. <assign> → <id> = <expr>

<expr> → <expr> * <term>

| <term>

<term> → <factor> + <term> | <factor> - <term>

| <factor>

<factor> → (<expr>)

| <id>

<id> → A | B | C

2. Rewrite the following BNF to add the prefix ++ and **--** unary operators of Java.

. <assign> → <id> = <expr>

<expr> → <expr> * <term>

| <term>

<term> → <factor> + <term> | <factor> - <term>

| <factor>

<factor> → (<expr>)

| <id>

<id> → A | B | C

3. Show that the following grammar is ambiguous

. <compare> → <boolexpr> **==** <boolexpr>

<boolexpr> → <boolexpr> **AND** <boolexpr>

| <boolexpr> **OR** <boolexpr>

| <bool>

| **NOT** <bool>

<bool> → <boolvalue> | <boolvar>

<boolvalue> → **True** | **False | 0 | 1**

<boolvar> → **u** | **v** | **w**

CS 381 Homework 3 – Syntax

Submit a pdf for problems 1 – 4 and a Haskell *.hs file for problem 5.

4. Write a grammar G for the language L consisiting of strings of 0's and 1's that are the binary representation of odd integers greater that 4. For example 11 $\notin$L, 101$\in$L, 110 $\notin$L. Draw parse trees for the strings 1011 and 1101

5. Below is the EBNF grammar for the animal sentence language

```
<sentence> ->  <noun> <verb> [<noun>]
            |  <sentence> `and` <sentence>

<noun>      -> <adj> <noun> | <noun> `and` <noun>
            | `cats` | `dogs` | `ducks` | `bunnies`

<verb>      ->  `chase` | `cuddle` | `hug` | `scare`
<adj>       ->  `silly` | `small` | `old` | `happy`
```

Note: the nonterminals are in < > and the terminals are in ` `.

Using the animal.hs template provided.

a) Define the abstract syntax for the animal language as a Haskell data type.

b) Provide "pretty printing" functions for the sentences in the language.

c) Provide functions to build a sentence.

d) Write a function isNice to determine if a sentence only contains the verbs hug and cuddle.

e) Write a function to build a sentence that is a conjunction of other sentences.

f) Write a function wordCount that computes the number of words in a sentence

CS 381 Homework 3 – Syntax

Submit a pdf for problems 1 – 4 and a Haskell *.hs file for problem 5.

1. Using the grammar below, show a parse tree and a leftmost derivation for the sentence

A = B * (C+A)

.   <assign> → <id> = <expr>

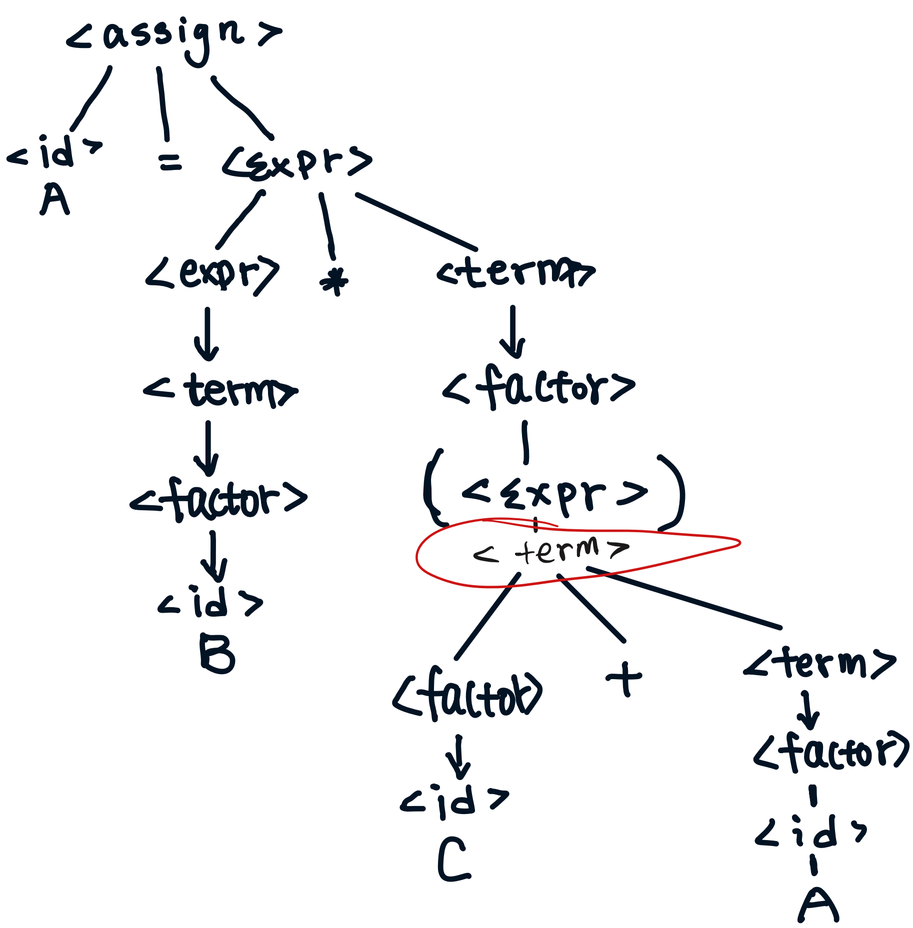
<expr> → <expr> * <term>
| <term>

<term> → <factor> + <term> | <factor> - <term>
| <factor>

<factor> → (<expr>)
| <id>

<id> → A | B | C

Parse tree:



Lefmost derivation:

assign ⇒ <id> = <expr>

⇒ A = <expr>

⇒ A = <expr> * <term>

⇒ A = <term> * <factor>

⇒ A = <factor> * (<expr>)

A = <factor * < term>

⇒ A = <id> * (<factor> + <term>)

⇒ A = B * (<id> + <factor>)

⇒ A = B * (C + <id>)

⇒ A = B * (C + A)

2. Rewrite the following BNF to add the prefix ++ and -- unary operators of Java.

<assign> → <id> = <expr>

<expr> → <expr> * <term>

| <term>

<term> → <factor> + <term> | <factor> - <term>

| <factor>

<factor> → (<expr>)

| <id>

<id> → A | B | C

added the prefix and -- unary operetors :

<assign> → <id> = <expr>

<expr> → <expr> * <term> | <term>

<term> → <factor> + <term> | <factor> – <term>

| <factor>

<factor> → (<expr>)

| ++ <id> | -- <id> | <id>

<id> → A | B | C

I add prefix ++ and -- in the front of <id>, Because acoding to the given grammar, <id> is terminals which means <id> are variables, accordig to Java, we can add "++" and "--" before a varible.

3. Show that the following grammar is ambiguous

.

<compare> → <boolexpr> **==** <boolexpr>

<boolexpr> → <boolexpr> **AND** <boolexpr>

| <boolexpr> **OR** <boolexpr>

| <bool>

| **NOT** <bool>

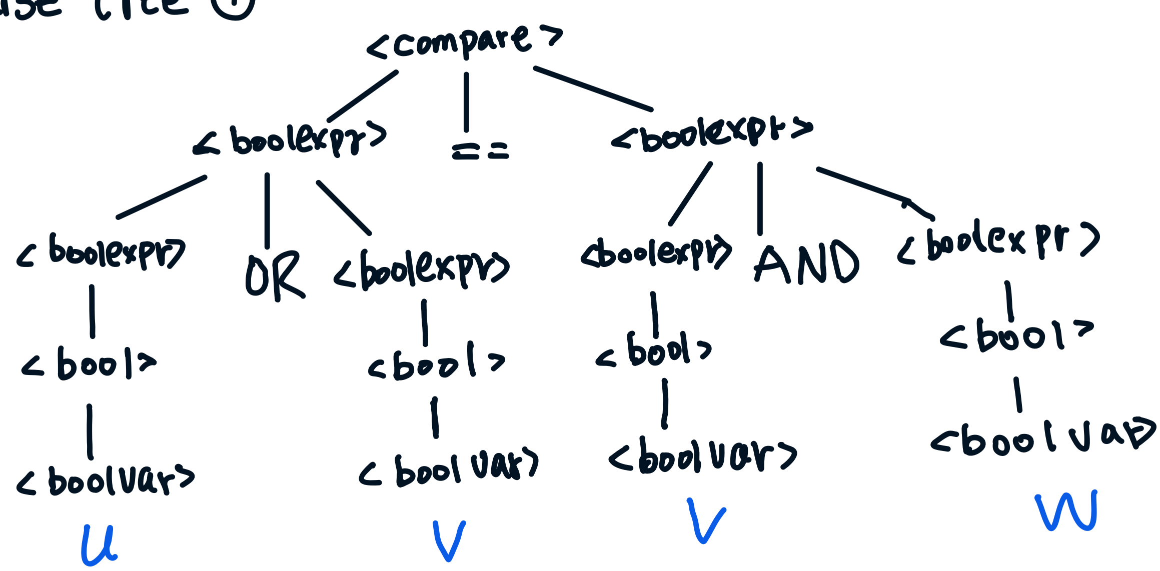
<bool> → <boolvalue> | <boolvar>

<boolvalue> → **True** | **False** | **0** | **1**

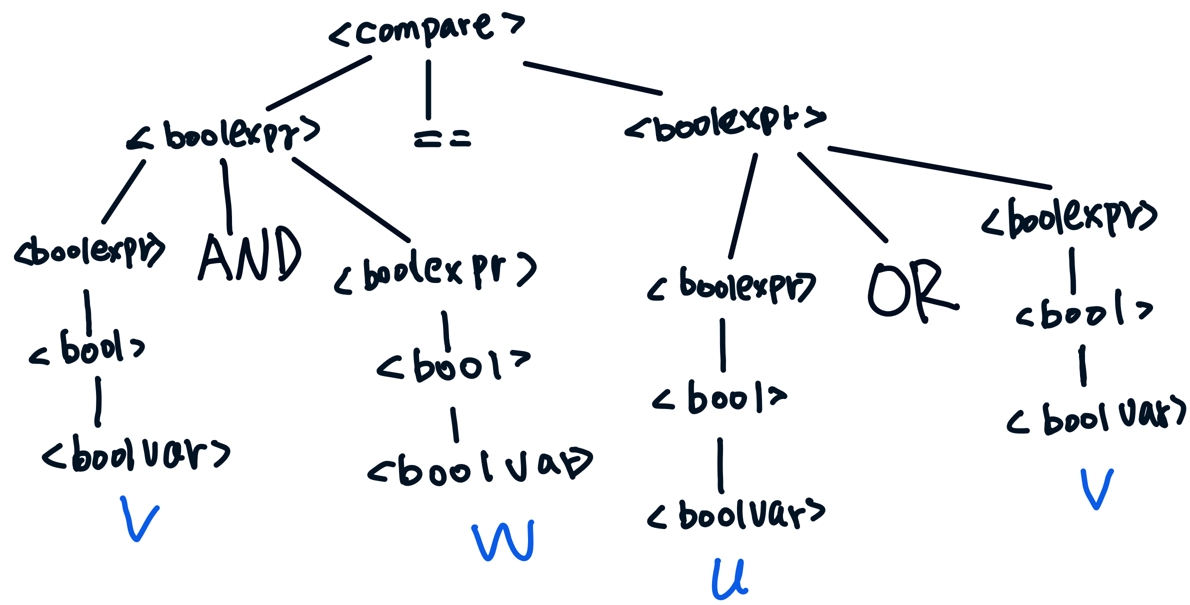
<boolvar> → **u** | **v** | **w**

Let me generate two parse trees for the sentence : u OR v == v AND w

parse tree ① :



Parse tree ②:



The first parse tree represents the sentence u OR v == v AND w and the second parse tree represents sentence v AND w == u OR v. Therefor, the grammer is ambiguous, Because the same iput sentence could be parsed in two different ways.

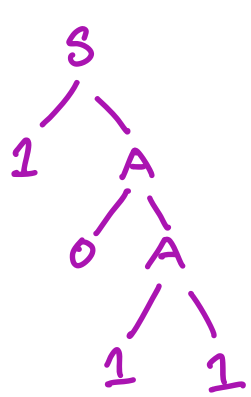
4. Write a grammar G for the language L consisiting of strings of 0's and 1's that are the binary representation of odd integers greater that 4. For example 11 $\notin$L, 101$\in$L, 110 $\notin$L. Draw parse trees for the strings 1011 and 1101

Let $G = (\{V\}, \{0,1\}, P, S)$, where S is the start symbol, $V=\{S,A\}$ is set variabe, $\{0,1\}$ is set set of terminals, and P is the set of production rules defined as:

$$S \to 1A$$
$$A \to 0A$$
$$A \to 1$$

This grammar genearates binary representation of odd integers greater than 4, where the first character is always 1, followed by zero or more 1's and 0's alternating, ending with a 0.

Parse tree for the string "1011"



Parse tree for the string "1101"