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
```

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.

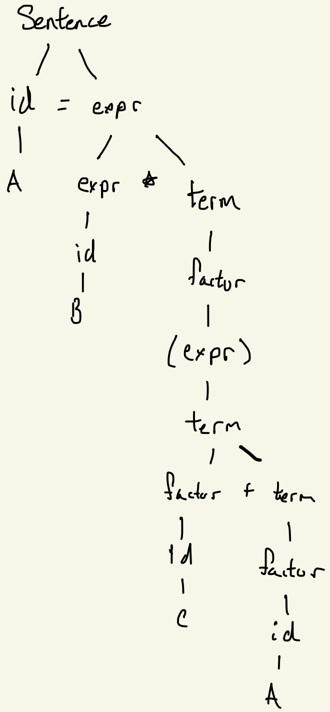
c) Provide functions to build a sentence.

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

1: Parse tree



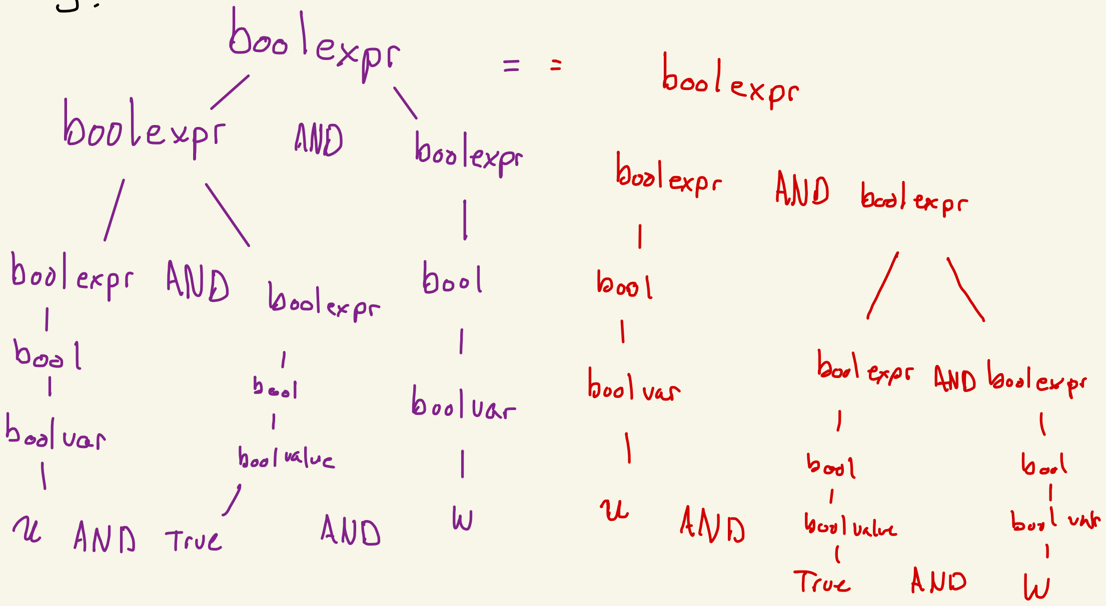
Leftmost deviation

Program => Sentence
=> id = expr
=> A = expr
=> A = expr * term
=> A = id * term
=> A = B * term
=> A = B * factor
=> A = B * (expr)
=> A = B * (term)
=> A = B * (factor + term)
=> A = B * (id + term)
=> A = B * (C + term)
=> A = B * (C + factor)
=> A = B * (C + id)
=> A = B * (C + A)

2. BNF

assign -> id = expr
expr -> expr * term | term
term -> factor + term | factor - term | factor
id -> A | B | C
factors -> (expr) | id | ++id | --id

3:


boolexpr
= =
boolexpr
boolexpr
AND
boolexpr
boolexpr
AND
boolexpr
bool
bool
bool
boolvar
bool
boolvalue
boolvar
u
AND
True
AND
w
boolexpr
AND
boolexpr
bool
boolvar
boolexpr
AND
boolexpr
bool
bool
boolvalue
bool var
u
AND
True
AND
w

4.

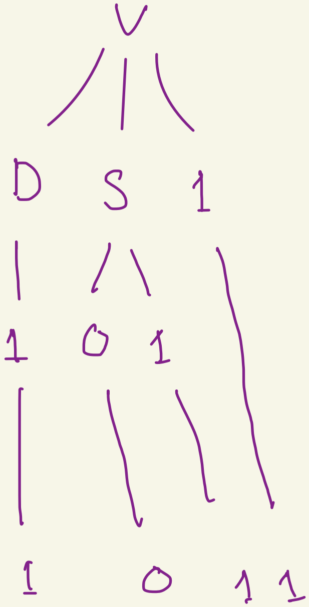
$N = \{S, D, V\}$

$\Sigma = \{0, 1\}$

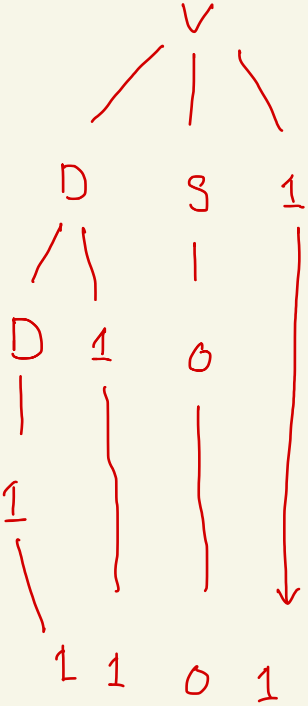
$P: V \to DS1$

$S \to SD \mid 01 \mid 0$

$D \to D1 \mid 10 \mid 1$



V
D S 1
1 0 1
1 0 1 1



V
D S 1
D 1 0
1
1 1 0 1