CMSC 330 Quiz 3 Fall 2021 Solutions

Q1. Context-Free Grammars

Q1.1. Construct a CFG that matches the following regex: `a*m+n?`

```
S -> aS | M
M -> mM | mN
N -> n | ε
```

Q1.2. Prove that the following CFG is ambiguous:

```
S -> S + T | T
T -> 1 + T | 1
```

We can show there are two possible derivations that yield the same result

```
S -> S + T -> T + T -> 1 + T -> 1 + 1
S -> T -> 1 + T -> 1 + 1
```

OR

```
S -> S + T -> T + T -> 1 + T + T -> 1 + 1 + 1
S -> T -> 1 + T -> 1 + 1 + T -> 1 + 1 + 1
```

Q2. Parsing

Q2.1. Rewrite the following context-free grammar so that it can be parsed through recursive descent without creating an infinite loop.

```
S -> S or S | S and S | B
B -> not B | V
V -> true | false
```

Note: The rewritten grammar should accept the same strings as the one provided above.

```
S -> B or S | B and S | B
B -> not B | V
V -> true | false
```

Q2.2. Consider the following:

```
type token =
| Tok_Char of char
| Tok_Plus
| Tok_Comma

(* NOTE: This is an imperative implementation! *)

let lookahead () =
  match !tok_list with
  | [] -> raise (ParseError "no tokens")
  | (h::t) -> h

let match_tok a =
  match !tok_list with
  | (h::t) when a = h -> tok_list := t
```

```
  | _ -> raise (ParseError "bad match")
```

Complete the context-free grammar that is parsed by the code below.

```
let rec parse_S () =
  parse_T ();
  match lookahead () with
  | Tok_Plus -> (match_tok Tok_Plus; parse_S ())
  | Tok_Comma -> (match_tok Tok_Comma; parse_T (); match_tok Tok_Comma; parse_S ())
  | _ -> ()

and parse_T () =
  parse_A ();
  match lookahead () with
  | Tok_Char 'b' -> (match_tok (Tok_Char 'b'))
  | Tok_Char 'c' -> (match_tok (Tok_Char 'c'))
  | _ -> ()

and parse_A () =
  match lookahead () with
  | Tok_Char 'a' -> (match_tok (Tok_Char 'a'))
  | _ -> ()
```

Note: You can use E or e to denote an epsilon

```
S -> T + S | T, T, S | T
T -> A b | A c | A
A -> a | ε
```

Q3. Operational Semantics

$$\frac{}{A; n \to n} \qquad \frac{A(x) = v}{A; x \to v} \qquad \frac{A; e_1 \to v_1 \qquad A, x : v_1; e_2 \to v_2}{A;\ \textbf{let } x = e_1 \textbf{ in } e_2 \to v_2}$$

$$\frac{A; e_1 \to v_1 \quad A;\ e_2 \to v_2 \quad v_3 \textit{ is } v_1 + v_2}{A;\ e_1 + e_2 \to v_3}$$

$$\frac{A; e_1 \to n_1 \qquad A; e_2 \to n_2 \qquad n_1 > n_2}{A; e_1 > e_2 \to \textbf{true}} \qquad \frac{A; e_1 \to n_1 \qquad A; e_2 \to n_2 \qquad n_1 \leq n_2}{A; e_1 > e_2 \to \textbf{false}}$$

$$\frac{A; e_1 \to \textbf{true} \qquad A; e_2 \to v}{A;\ \textbf{if } e_1 \textbf{ then } e_2 \textbf{ else } e_3 \to v} \qquad \frac{A; e_1 \to \textbf{false} \qquad A; e_3 \to v}{A;\ \textbf{if } e_1 \textbf{ then } e_2 \textbf{ else } e_3 \to v}$$

Using the above rules, fill in the blank for the derivation below:

$$\dfrac{\dfrac{}{A;\ 5\to 5} \qquad \dfrac{\dfrac{}{\boxed{2}} \qquad \dfrac{\dfrac{A,x{:}5,y{:}2;\ 1\to 1 \quad A,x{:}5,y{:}2;\ 0\to 0 \quad 1>0}{A,x{:}5,y{:}2;\ \boxed{3}} \qquad \dfrac{\boxed{4}}{A,x{:}5,y{:}2;\ y\to 2}}{A,x{:}5,y{:}2;\ if\ 1>0\ then\ y\ else\ x\ \to\ 2}}{A,x{:}5;\ let\ y=2\ in\ if\ 1>0\ then\ y\ else\ x\ \to\ 2}}{A;\ \boxed{1}\ let\ y=2\ in\ if\ 1>0\ then\ y\ else\ x\ \to\ 2}$$

#1: `let x = 5 in`

#2: `A,x:5; 2 -> 2`

#3: `1 > 0 -> true`

#4: `A,x:5,y:2(y) = 2`