Man Tik Li
CS 360 - 002
Krzysztof Nowak
August 19, 2020

Quiz 5

1. 1) gcd(33, 24, X) ← not zero(24), gcd(24, 33 mod 24, X), gcd(33, 24, X).

gcd(24, 9, X) ← not zero(9), gcd(9, 24 mod 9, X), gcd(24, 9, X).

gcd(9, 6, X) ← not zero(6), gcd(6, 9 mod 6, X), gcd(9, 6, X).

gcd(6, 3, X) ← not zero(3), gcd(3, 6 mod 3, X), gcd(6, 3, X).

gcd(3, 0, X)

∴ Setting X to 3 gives the empty statement and matches the gcd(u, 0, u) statement.

2. 2)

1.	$p \to q$	Premise
2.	$p \to r$	Premise
3.	$\bar{p} + q$	1(a)[Subsumption. $(p \to q) \equiv (\bar{p} + q)$]
4.	$\bar{p} + r$	2(a)[Subsumption. $(p \to q) \equiv (\bar{p} + q)$]
5.	$(\bar{p}+q)(\bar{p}+r)$	3,4(c)
6.	$\bar{p}+qr$	5(a)[Subsumption. $p+qr \equiv (p+q)(p+r)$]
7.	$p \to qr$	6(a)[Subsumption. $p \to q \equiv \bar{p} + q$]

3. 1.$E_1 \rightarrow E_2 + T$ (The sum of the values of E2 and T will be assigned to E1)

   2.$E_1 \rightarrow E_2 - T$ (The difference of the values of E2 and T will be assigned to E1)

   3.$E \rightarrow T$ (The value of T will be assigned to E)

   4.$T_1 \rightarrow T_2 * F$ (The product of T2 and F will be assigned to T1)

   5.$T_1 \rightarrow T_2/F$ The division of T2 by F will be assigned to T1)

   6.$T \rightarrow F$ (The value of F will be assigned to T)

   7.$F_1 \rightarrow -F_2$ (The value of F1 will be equal to the additive inverse of the value of F2)

   8.$F \rightarrow (E)$ (The value of E will be assigned to F)

   9.$F \rightarrow const$ (A constant will be assigned to F)

   1.$E \rightarrow T$ TT (TT is the value of the T pointer and E is the value of the TT pointer)

   2.$TT_1 \rightarrow +T\ TT_2$ (The sum of T and TT_2 will be assigned to TT_1)

   3.$TT_1 \rightarrow -T\ TT_2$ (The difference between T and TT_2 will be assigned toTT_1)

   4.$TT \rightarrow \epsilon$ (The value of TT is stored)

   5.$T \rightarrow F\ FT$ (FT is the value of the F pointer and T is the value of the FT pointer)

   6.$FT_1 \rightarrow *F\ FT_2$ (The product of F and FT_2 will be assigned to FT_1)

   7.$FT_1 \rightarrow /F\ FT_2$ (The division of F by FT_2 will be assigned to FT_1)

   8.$FT \rightarrow \epsilon$ (The value of FT will be stored)

   9.$F_1 \rightarrow -F_2$ (The value of F1 will be equal to the additive inverse of the value of F2)

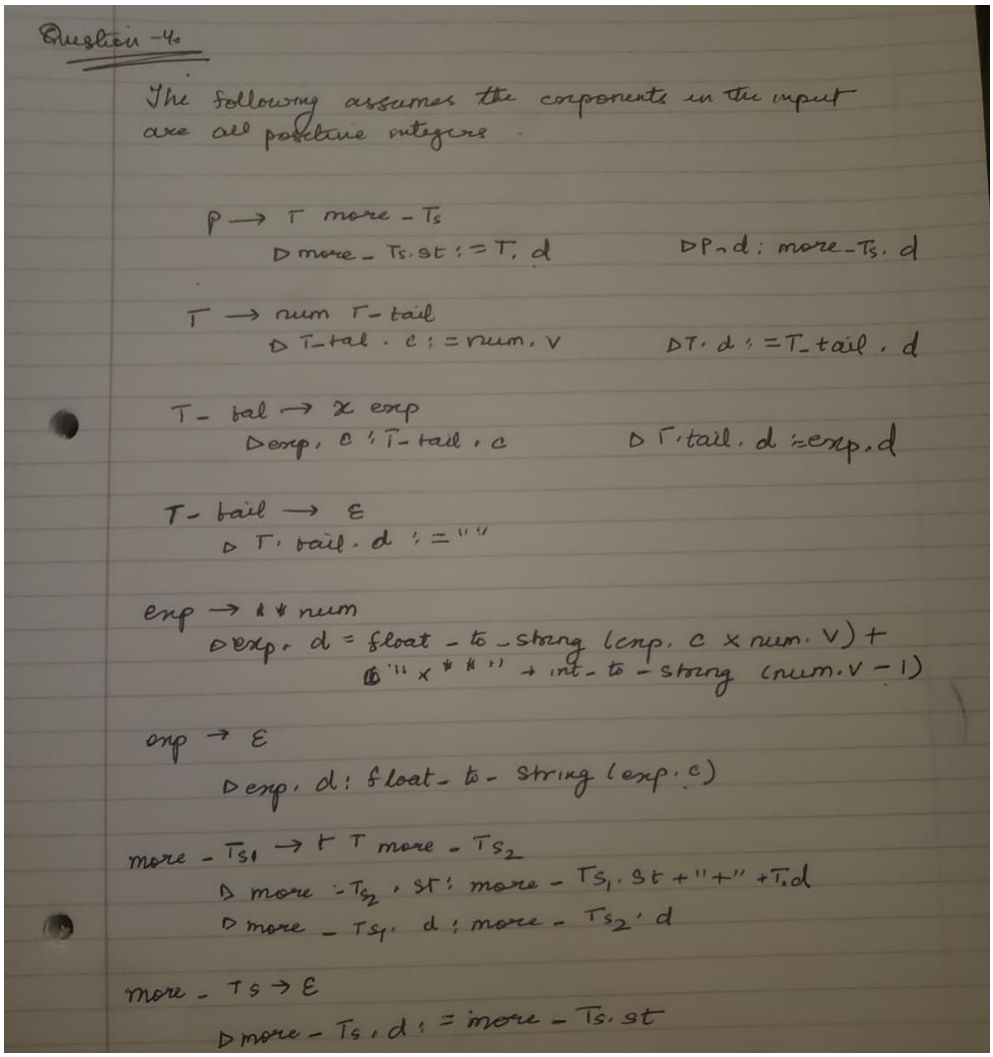
   10.$F \rightarrow (E)$ (The value of E will be assigned to F)

   11.$F \rightarrow const$ (A constant will be assigned to F)

   An action routine is a semantic function that we tell the compiler to execute at a particular point in the parse. When it predicts a production, the parser pushes all of the right-hand side onto the stack, including terminals to be matched, nonterminals (to drive future predictions), and pointers to action routines. When a pointer to an action routine at the top of the parse stack is found, the parser calls it and passes the appropriate attributes as arguments. In Fig 6 and Fig 7, the action routines would be embedded among the symbols of the right hand sides before performing the same task as in the syntax tree.

3.

4.



Question -4

The following assumes the components in the input are all positive integers

P → T more - Ts
▷ more - Ts.st := T.d ▷ P.d := more - Ts.d

T → num T - tail
▷ T - tail.c := num.V ▷ T.d := T - tail.d

T - tail → x exp
▷ exp.c := T - tail.c ▷ T.tail.d := exp.d

T - tail → ε
▷ T.tail.d := ""

exp → ** num
▷ exp.d = float - to - string (exp.c x num.V) +
▷ "x**" + int - to - string (num.V - 1)

exp → ε
▷ exp.d := float - to - string (exp.c)

more - Ts1 → + T more - Ts2
▷ more - Ts2.st := more - Ts1.st + "+" + T.d
▷ more - Ts1.d := more - Ts2.d

more - Ts → ε
▷ more - Ts.d := more - Ts.st

Practice Problems:

4.7 – 3
4.8 – 3
4.9 – 2
4.10 – 2
4.13 – 2
4.16 – 3
4.17 - 2