Question 2: Parsing

T-by Chappell

1) Shift/Reduce Parser

(<7, int main() ...)
(int, main () ...)
(Type, main () ...)
(Type main, () ...)
(Type Id, () ...)
(Type Id (,) ...)
(Type Id ([Arg],) ...)
(Type Id ([Arg]), { int x, y; ...)
(... {, int x, y; ...)
(... { int, x, y; ...)
(... { Type, x, y; ...)
(... { Type x, , y; ...)
(... { Type Id, , y; ...)
(... Type Id, , y; ...)
(... Type Id, y, , ...)
(... Type Id, Id, , ...)
(... Type [Id], , ...)
(... Type [Id], { y=9; ...)
(... Stm, { y=9; ...)
(... {, y=9; ...)
(... { y, =9, ...)
(... { Id, =9, ...)
(... { Exp15, =9, ...)
(... { Exp3, =9, ...)
(... Exp3 =, 9, ...)
(... Exp3 = 9, , ...)
(... Exp3 = Integer, , ...)
(... Exp3 = Exp15, , ...)
(... Exp3 = Exp2, , ...)
(... Exp2, , ...)
(... Exp, , ...)
(... { Exp, , int y; ...)

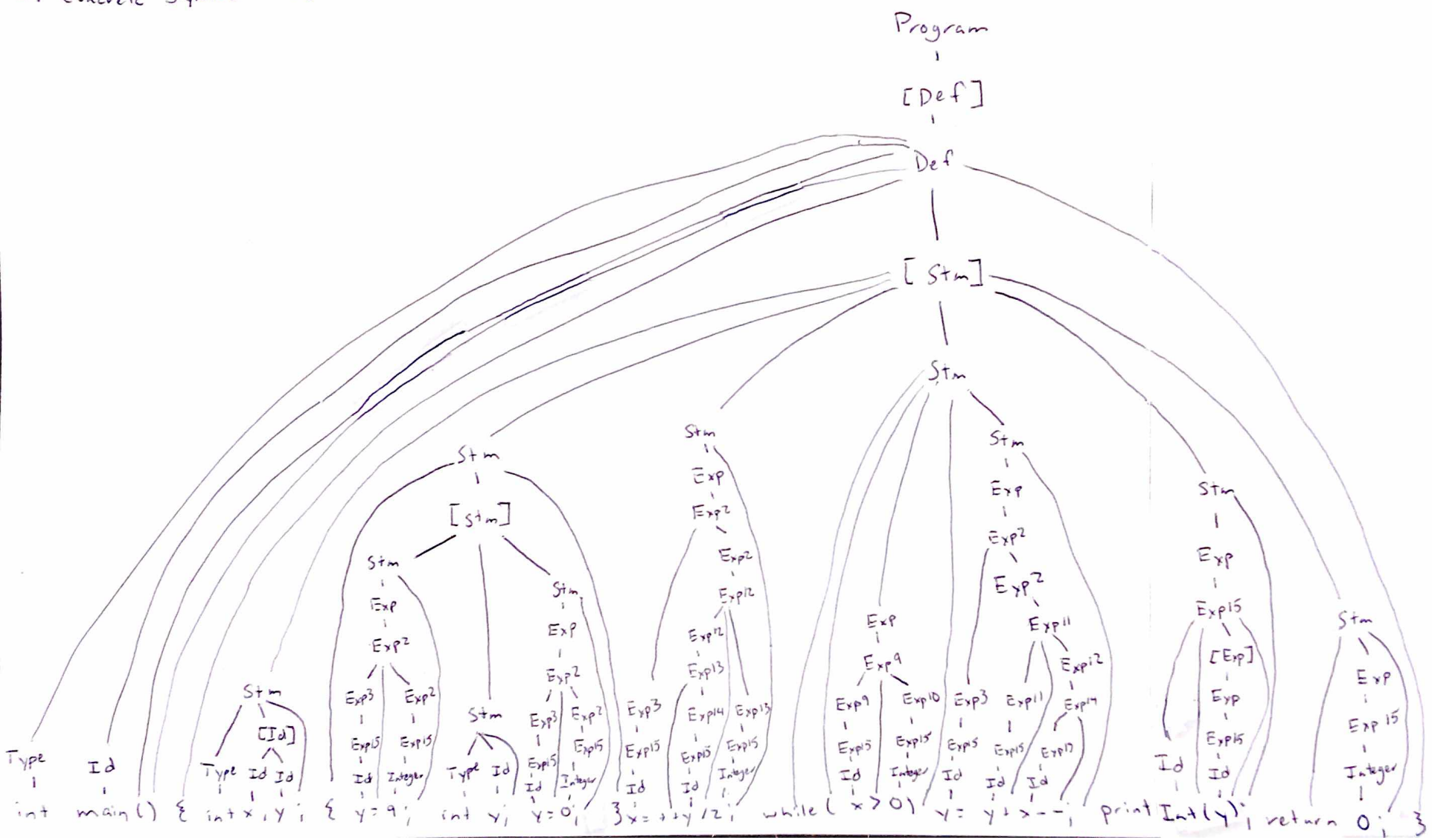
(... { Stm , int y; ...)
(... int, y; ...)
(... Type, y; ...)
(... Type y, ...)
(... Type Id, ...)
(... Type Id, y=0, ...)
(... { Stm Stm, y=0; ...)
(... y) =0, ...)
(... Id, =0 ...)
(... Exp15, =0, ...)
(... Exp3, =0, ...)
(... Exp3 =, 0, ...)
(... Exp3 = 0, , ...)
(... Exp3 = Integer, , ...)
(... Exp3 = Exp15, , ...)
(... Exp3 = Exp2, , ...)
(... Exp2, , 3 ...)
(... Exp1, , 3 ...)
(... Exp, , 3 ...)
(... Stm, 3 ...)
(... { [Stm], 3 ...)
(... { [Stm] 3, x=++y/2; ...)
(... Stm, x=++y/2; ...)
(... x, =++y/2; ...)
(... Id, =++y/2; ...)
(... Exp15, =++y/2; ...)
(... Exp3, =++y/2; ...)
(... Exp3 =, ++y/2 ...)
(... Exp3 = ++, y/2; ...)
(... Exp3 = ++y, /2; ...)
(... Exp3 = ++Id, /2; ...)
(... Exp3 = ++ Exp15, /2; ...)

(... Exp3 = +1 Exp14, /2; ...)
(... Exp3 = Exp13, /2; ...)
(... Exp3 = Exp12, /2; ...)
(... Exp3 = Exp12/, 2; ...)
(... Exp3 = Exp12/2, ; ...)
(... Exp3 = Exp12/Integer, ; ...)
(... Exp3 = Exp12/Exp15, ; ...)
(... Exp3 = Exp12/ Exp13, ; ...)
(... Exp3 = Exp12, ; ...)
(... Exp3 = Exp2
(... Exp2, ; ...)
(... Exp); ...)
(... Exp, while(x>0)...)
(... Stm, while(x>0)...)
(... while, (x>0)...)
(... while (, x>0)...)
(... while (x, >0)...)
(... while (Id, >0)...)
(... while (Exp15, >0)...)
(... while (Exp9, >0)...)
(... (Exp9 >, 0)...)
(... (Exp9 > 0,)...)
(... (Exp9 > Integer,)...)
(... (Exp9 > Exp15,)...)
(... (Exp9 > Exp10,)...)
(... (Exp9,)...)
(... (Exp,)...)
(... (Exp), y = y + x--, ...)
(... y, = y + x--, ...)
(... Id, = y + x--, ...)
(... Exp15, = y + x--, ...)
(... Exp3, = y + x--, ...)

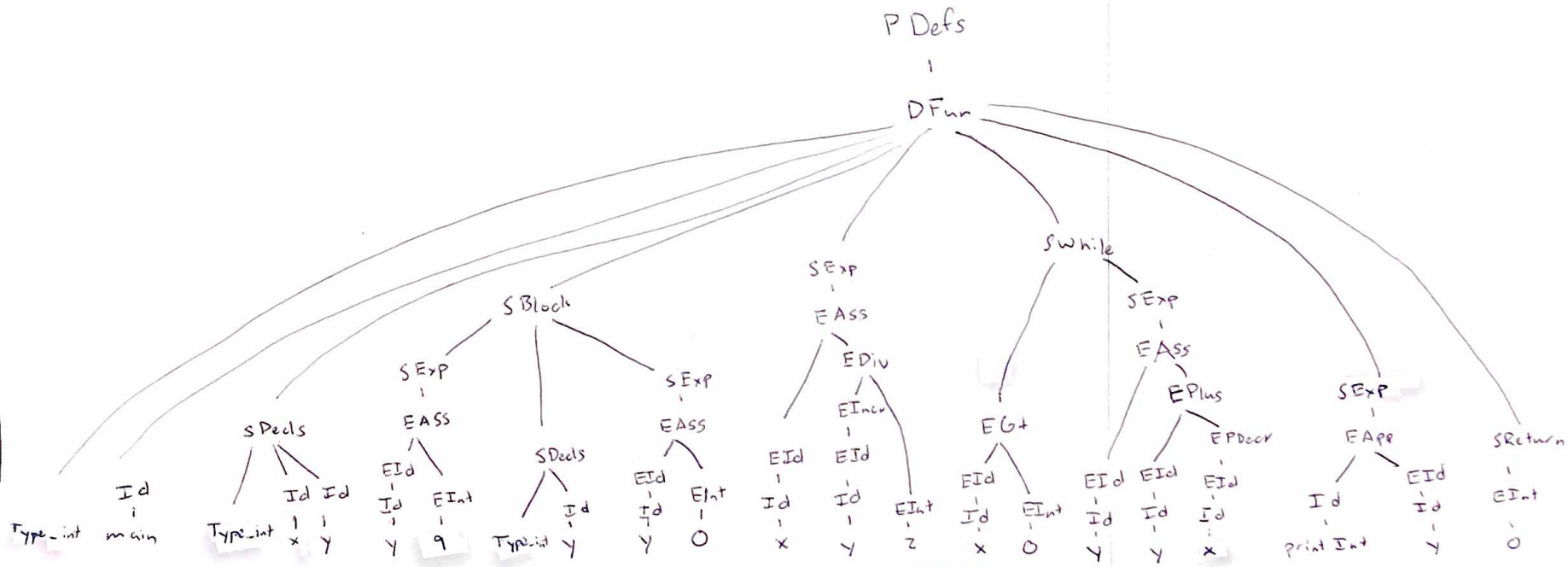
(... Exp 3 = , y + x --; ...)
(... Exp 3 = y , + x --; ...)
(... Id , + x --; ...)
(... Exp 15 , + x --; ...)
(... Exp 11 , + x --; ...)
(... Exp 11 + , x --; ...)
(... Exp 11 + x , --; ...)
(... Exp 11 + Id , --; ...)
(... Exp 11 + Exp 15 , --; ...)
(... Exp 11 + Exp 15 --, ; ...)
(... Exp 11 + Exp 14 , ; ...)
(... Exp 11 + Exp 12 , ; ...)
(... Exp 11 , ; ...)
(... Exp 3 = Exp 2 , ; ...)
(... Exp 2 , ; ...)
(... Exp , ; ...)
(... Exp , printInt(y); ...)
(... while, (Exp) Stm , printInt (y); ...)
(... Stm , printInt (y); ...)
(... printInt , (y); ...)
(... Id , (y); ...)
(... Id (, y) ; ...)
(... Id (y ,); ...)
(... Id (Id ,); ...)
(... Id (Exp 15 ,); ...)
(... Id (Exp ,); ...)
(... Id ([Exp] ,); ...)
(... Id ([Exp]), ; ...)
(... Exp 15 , ; ...)
(... Exp , ; ...)
(... Exp ; , return ...)
(... Stm , return 0 ; ...)

```
(... return, 0; })
(... return 0, ; })
(... return Integer, ; })
(... return Exp15, ; })
(... return Exp, ; })
(... return Exp; , })
(... { Stm Stm Stm Stm Stm Stm, })
(... { [Stm], })
( Type Id ([Arg]) { [Stm] }, <>)
( Def, <>)
( [Def], <>)
( Program, <>)
```

2) Concrete Syntax Tree



3) Abstract Syntax Tree



Question 3: Type Checking

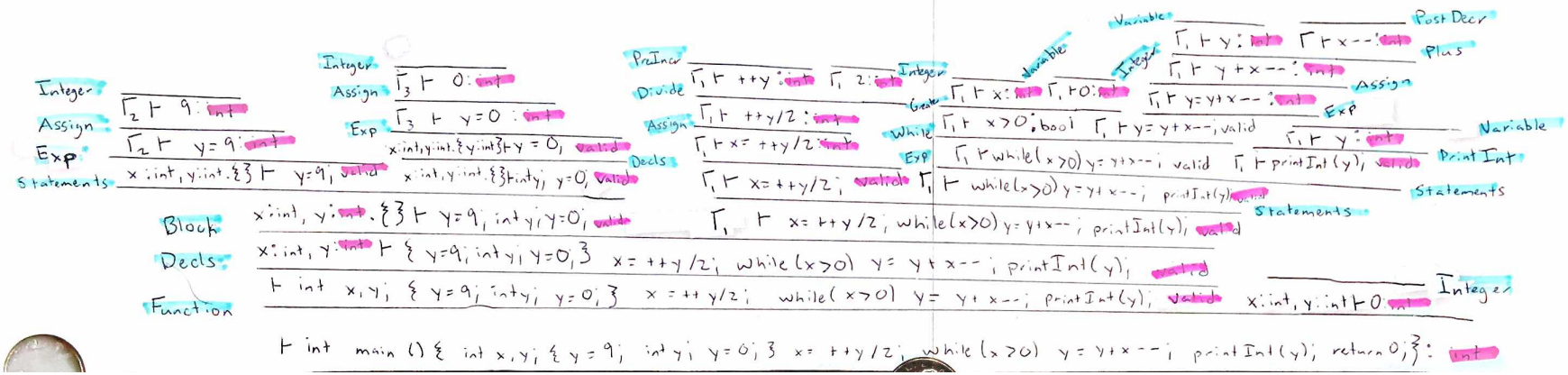
1) Proof Tree

2) Type Checking Rules at each Step

Γ_1 = x:int, y:int

Γ_2 = x:int, y:int. {}

Γ_3 = x:int, y:int. { y:int }



3) Type Checking Rules

$$\text{Plus} = \frac{\Gamma \vdash a : t \quad \Gamma \vdash b : t}{\Gamma \vdash a + b : t}, \; t \text{ is int/double}$$

$$\text{Divide} = \frac{\Gamma \vdash a : t \quad \Gamma \vdash b : t}{\Gamma \vdash a / b : t}, \; t \text{ is int/double}$$

$$\text{Greater} = \frac{\Gamma \vdash a : t \quad \Gamma \vdash b : t}{\Gamma \vdash a > b : \text{bool}}, \; t \text{ is int/double}$$

$$\text{Exp} = \frac{\Gamma \vdash e : t}{\Gamma \vdash e; \text{ valid}}$$

$$\text{While} = \frac{\Gamma \vdash e : \text{bool} \quad \Gamma \vdash s \text{ valid}}{\Gamma \vdash \text{while } (e)\ s \text{ valid}}$$

$$\text{Block} = \frac{\Gamma \vdash r_1 \cdots r_m \text{ valid} \quad \Gamma \vdash s_2 \cdots s_n \text{ valid}}{\Gamma \vdash \{ r_1 \cdots r_m \}\ s_2 \cdots s_n \text{ valid}}$$

$$\text{Decls} = \frac{\Gamma(x_1 : t \ldots x_n : t) \vdash s_2 \cdots s_n \text{ valid}}{\Gamma \vdash t\ x_1 \cdots x_n;\ s_2 \cdots s_n \text{ valid}}$$

$$\text{Assign} = \frac{\Gamma \vdash e : t}{\Gamma \vdash x = e : t}, \text{ if } t = \text{type of } x \text{ in } \Gamma$$

$$\text{Variable} = \frac{}{\Gamma \vdash x : t}, \text{ if } t = \text{type of } x \text{ in } \Gamma$$

$$\text{Integer} = \frac{}{\Gamma \vdash 3 : \text{int}}$$

$$\text{Staments} = \frac{\Gamma \vdash s_1 \text{ valid} \quad \Gamma \vdash s_2 \cdots s_n \text{ valid}}{\Gamma \vdash s_1 \cdots s_n \text{ valid}}$$

$$\text{PrintInt} = \frac{\Gamma \vdash x : \text{int}}{\Gamma \vdash \text{printInt}(x); \text{ valid}}$$

$$\text{Function} = \frac{\vdash S_1 \ldots S_n \text{ valid} \quad \vdash x : \text{int}}{\vdash \text{int } f() \{ S_1 \ldots S_n \text{ return } x; \} : \text{int}}$$

$$\text{PreIncr} = \frac{}{\Gamma \vdash ++x : t}, \text{ if } t \text{ is int/double} = \text{type of } x \text{ in } \Gamma$$

$$\text{PostDecr} = \frac{}{\Gamma \vdash x-- : t}, \text{ if } t \text{ is int/double} = \text{type of } x \text{ in } \Gamma$$

Question 4: Interpretation

1) Proof Tree

2) Rules for each Step

$\gamma_1 = \{ x := 0,\ y := 25 \}$

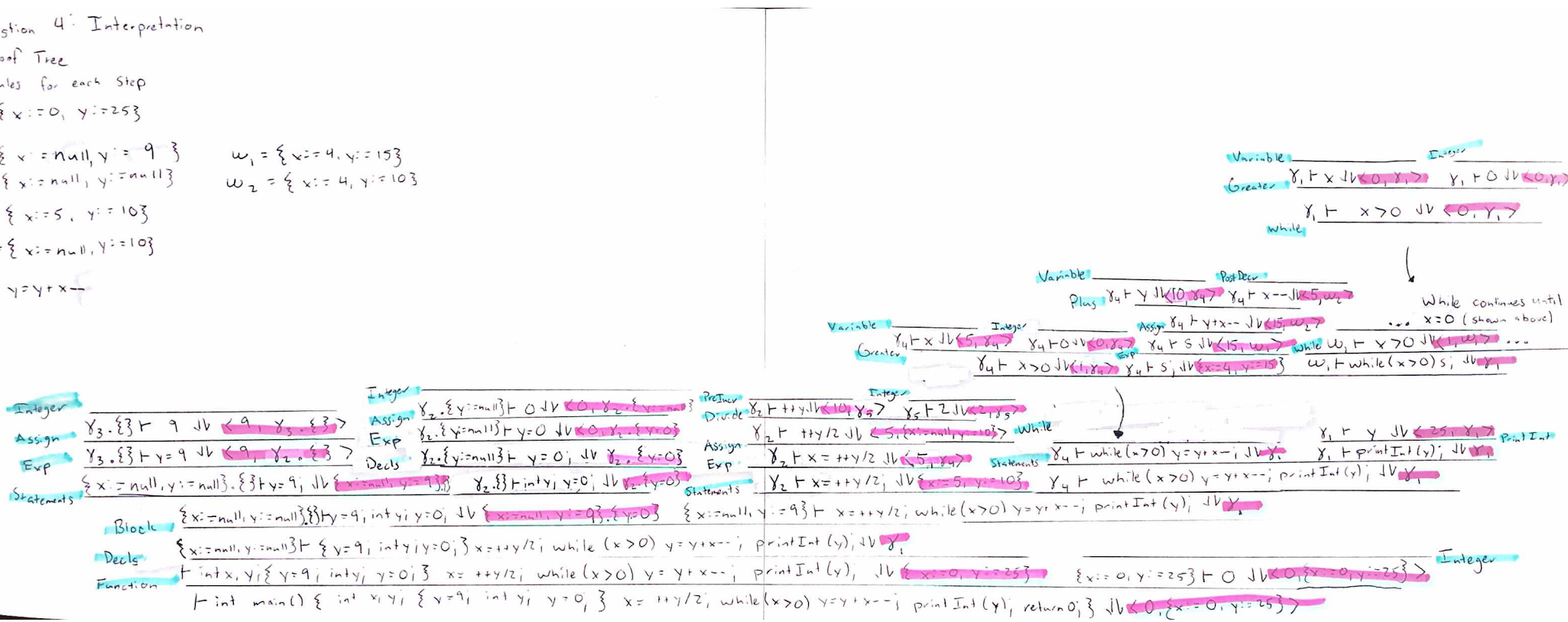
$\gamma_2 = \{ x := null,\ y := 9 \}$ $\quad \omega_1 = \{ x := 4,\ y := 15 \}$

$\gamma_3 = \{ x := null,\ y := null \}$ $\quad \omega_2 = \{ x := 4,\ y := 10 \}$

$\gamma_4 = \{ x := 5,\ y := 10 \}$

$\gamma_5 = \{ x := null,\ y := 10 \}$

$S = y = y + x--$



3) Interpreter Rules

$$\text{Plus} = \frac{\gamma \vdash a \Downarrow \langle u, \gamma' \rangle \quad \gamma' \vdash b \Downarrow \langle v, \gamma'' \rangle}{\gamma \vdash a + b \Downarrow \langle u + v, \gamma'' \rangle}$$

$$\text{Divide} = \frac{\gamma \vdash a \Downarrow \langle u, \gamma' \rangle \quad \gamma' \vdash b \Downarrow \langle v, \gamma'' \rangle}{\gamma \vdash a / b \Downarrow \langle u / v, \gamma'' \rangle}$$

$$\text{Greater} = \frac{\gamma \vdash a \Downarrow \langle u, \gamma' \rangle \quad \gamma' \vdash b \Downarrow \langle v, \gamma'' \rangle}{\gamma \vdash a > b \Downarrow \langle a > b, \gamma'' \rangle}$$

$$\text{Exp} = \frac{\gamma \vdash e \Downarrow \langle v, \gamma' \rangle}{\gamma \vdash e; \Downarrow \gamma'}$$

$$\text{While} = \frac{\gamma \vdash e \Downarrow \langle 1, \gamma' \rangle \quad \gamma' \vdash s \Downarrow \gamma'' \quad \gamma'' \vdash \text{while}(e) s \Downarrow \gamma'''}{\gamma \vdash \text{while}(e)\, s \Downarrow \gamma'''}$$

$$\frac{\gamma \vdash e \Downarrow \langle 0, \gamma' \rangle}{\gamma \vdash \text{while}(e) s \Downarrow \gamma'}$$

$$\text{Block} = \frac{\gamma_0 \vdash r_1 \ldots r_m \Downarrow \gamma' . \delta \quad \gamma' \vdash s_1 \ldots s_n \Downarrow \gamma''}{\gamma \vdash \{ r_1 \ldots r_m \}\, s_1 \ldots s_n \Downarrow \gamma''}$$

$$\text{Decls} = \frac{\gamma (x_1 := \text{null}, \ldots x_n := \text{null}) \vdash s_1 \ldots s_n \Downarrow \gamma'}{\gamma \rightleftharpoons \vdash x_1 \ldots x_n ; s_1 \ldots s_n \Downarrow \gamma'}$$

$$\text{Assign} = \frac{\gamma \vdash e \Downarrow \langle v, \gamma' \rangle}{\gamma \vdash x = e \Downarrow \langle v, \gamma' (x := v) \rangle}$$

$$\text{Variable} = \frac{}{\gamma \vdash x \Downarrow \langle v, \gamma \rangle} \text{ if value of } x = v \text{ in } \gamma$$

$$\text{Integer} = \frac{}{\gamma \vdash 3 \Downarrow \langle 3, \gamma \rangle}$$

$$\text{Statements} = \frac{\gamma \vdash s_1 \Downarrow \gamma' \quad \gamma' \vdash s_2 \cdots s_n \Downarrow \gamma''}{\gamma \vdash s_1 \cdots s_n \Downarrow \gamma''}$$

$$\text{PrintInt} = \frac{\gamma \vdash x \Downarrow \langle v, \gamma' \rangle}{\gamma \vdash \text{printInt}(x); \Downarrow \gamma'}$$

$$\text{Function} = \frac{\vdash s_1 \cdots s_n \Downarrow \gamma \quad \gamma \vdash x \Downarrow \langle v, \gamma' \rangle}{\vdash \text{int } f() \{ s_1 \cdots s_n \text{ return } x; \} \Downarrow \langle v, \gamma' \rangle}$$

$$\text{PreIncr} = \frac{}{\gamma \vdash ++x \Downarrow \langle v+1, \gamma(x := v+1) \rangle} \text{ if } x := v \text{ in } \gamma$$

$$\text{PostDecr} = \frac{}{\gamma \vdash x-- \Downarrow \langle v, \gamma(x := v-1) \rangle} \text{ if } x := v \text{ in } \gamma$$