POPL Assignment 6

Itish Agarwal
18CS30021

Q1. (a) We have,

((six equals (two plus one)) → one [] (three minus one)) plus two

= (six equals three) → one [] (three minus one)) plus two

= (false → one [] (three minus one)) plus two

= (three minus one) plus two

= two plus two

= four

(b) We have,

(two equals (true → one [] two)) and true

= (two equals one) and true

= false and two

= false

(c) We have,

not (false) → not (true) [] not (true)

= true → not (true) [] not (true)

= not (true)

= false

Q 2. Dynamic Array Algebra with upper and lower bounds:

Domain:

Array = $(Nat \rightarrow A) \times Nat \times Nat + Error$, where A is the domain with error element

Error is a Unit domain used to return error during an update and contains only 1 value: errorarray

Operations:

newarray = $Nat \times Nat \rightarrow Array$

newarray = $\lambda l. \lambda u. (\lambda n. error, l, u)$

newarray represents an empty array

and it maps all index arguments between lower bound and upper bound u, to error.

access: Nat X Array → A

access = $\lambda n. \lambda(r, l, u). n$ greater than u

→ error [] n less than l →

error [] r(n)

access check if index lies between lower and upper bounds if it doesn't lie b/w upper and lower bound then returns error element.

update: Nat X A X Array → Array

update = $\lambda n. \lambda u. \lambda(r, l, u). n$ greater than u

→ errorarray [] n less than l

→ errorarray [] ([n → v] r, l, u)

update checks if index lies between upperbound and lowerbound. If yes, it performs normal array update otherwise it returns errorarray.

Itish Agarwal
18CS30021

Q3. Abstract syntactic domains:

Program, operator, numeral, expr sequence, expression, answer, digit

Abstract production rules:

Program ::= Expr. sequence

Expr. sequence ::= Expression | Expression Expr. sequence

Expression ::= Numeral | m^R | clear | Expression Answer

| Expression operator expression

operator ::= + | - | x

Answer ::= m^+ | = | ±

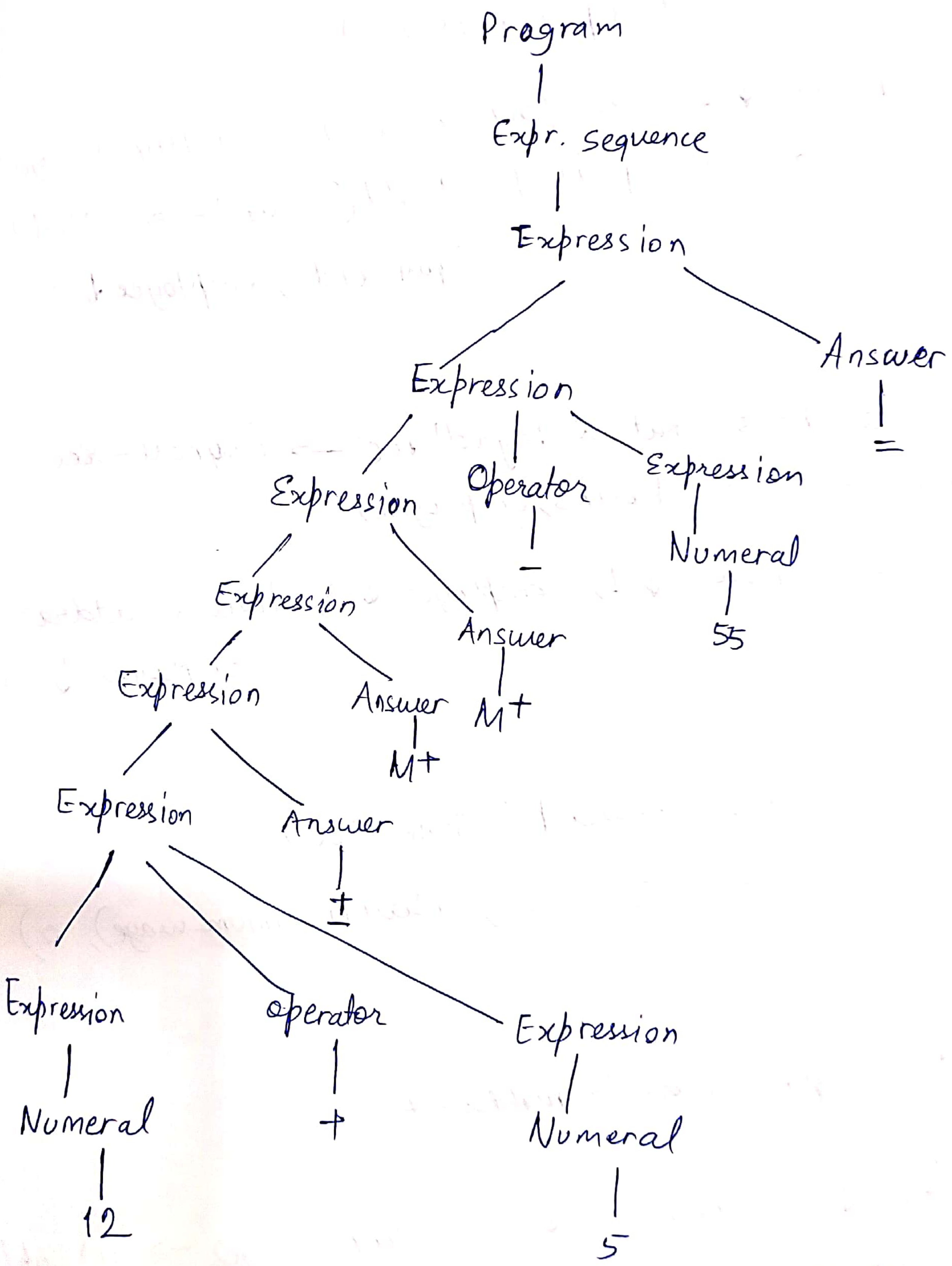
Numeral ::= Digit | Numerical Digit

Digit ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Itish. Agarwal
18CS30021

PTO

Abstract Syntax tree for $12 + 5 \pm M^{+}M^{+} - 55$



Itish Agarwal
18CS30021

Itish Agarwal
18CS30021

Q4.

(a) (i) update-payrate : Rat X Payroll-rec → Payroll-rec

update-payrate (pay, employee)

= (employee ↓ 1, (cases (employee ↓ 2) of isDay (dwage)
→ inDay (pay) [] isNight (nwage) → inNight
(pay) end), employee ↓ 3)

(ii) update - hours : Rat X Payroll-rec → Payroll-rec

update-hours (hours, employee)

= (employee ↓ 1, employee ↓ 2, hours addrat
employee ↓ 3)

(b) (i) jdoe = newemp ("Jane Doe")

= ("Jane Doe", inDay (minimum_wage), 0)

(ii) jdoe_night = move_to_nightshift (jdoe)

= (jdoe ↓ 1, (cases (jdoe ↓ 2) of isDay (dwage) → inNight
(dwage) [] isNight (nwage) → inNight (nwage) end)
job ↓ 3)

= ("Jane Doe", (cases (inDay (minimium_wage) of isDay (dwage)
→ inNight (dwage) [] isNight (nwage) → inNight (nwage)

end), 0)

= ("Jane Doe", inNight (minimum_wage), 0)

(iii) jdoe_hours = update_hours (makerat (38, 1), jdoe_night)

= (jdoe_night ↓ 1, jdoe_night ↓ 2, makerat(38,1) addrat
jdoe_night ↓ 3)

= ("Jane Doe", inNight (minimum_wage), makerat (38,1)
addrat 0)

= ("Jane Doe", inNight (minimum_wage), makerat (38,1))

(iv) jdoe_pay = update_payrate (makerat (9, 1), jdoe_hours)

= (jdoe_hours ↓ 1, (cases (jdoe_hours ↓ 2) of isDay
(dwage) → inDay (makerat (9,1)) [] isNight (nwage)
→ inNight (makerat(9,1)) end), jdoe_hours ↓ 3)

Itish Agarwal
18CS30021

PTO

= ("Jane Doe", (cases(inNight (minimium_wage)) of isDay(dwage)

→ inDay (makerat (9,1)) [] isNight(nwage) → inNight

(makerat (9,1) end), makerat (38,1))

= ("Jane Doe", inNight (makerat (9,1)), makerat (38,1))

compute_pay (jdoe_pay) =

(cases jdoe_pay ↓ 2 of isDay (dwage) → dwage multrat

(jdoe_pay ↓ 3) [] isNight (nwage) → (nwage multrat

1.5) multrat (jdoe_pay ↓ 3) end)

= (cases inNight (makerat (9,1)) of isDay (dwage) → dwage

multrat makerat (38,1) [] isNight (nwage) → (nwage

multrat 1.5) mutlrat makerat (38,1) end)

= (inNight (makerat (9,1) multrat 1.5) multrat

makerat(38,1)

Itish Agarwal
18CS30021