CSCI-GA.2110.001 Programming Language October 24, 2021

Solutions of Homework 1

Name: Xiao Ma (xm2074) *Due: 11.55 pm Sunday, October 24*

1. Regular Expression

(a) $([a-z] \mid [A-Z][a-z]^*[0-9][a-z]^*[A-Z] \mid [0-9][a-z]^*[A-Z][a-z]^*[0-9])^*$

(b) $(-|\epsilon)(0 \mid [1-9][0-9]^*)\ .\ [0-9][0-9]^*E(-|\epsilon)(0 \mid [1-9][0-9]^*)$

(c) $[a-zA-Z][a-zA-Z0-9]^*\ (\epsilon|\_)\ ([a-zA-Z0-9]\_)^*[a-zA-Z0-9]^*$

2. Context-free Grammar

(a) id: representing the syntax of numbers like 0, 1 and names like function name: foo,fac, variable name x,y.
PROG ⟶ FUNS
FUNS ⟶ FUN | FUN FUNS
FUN ⟶ fun DEFS
DEFS ⟶ DEF | DEF ”|” DEFS
DEF ⟶ id ARGUDEFS = STATES
ARGUDEFS ⟶ id | id ARGUDEFS
STATES ⟶ CALCULATION | EXP
EXP ⟶ id = CALCULATION | LET
CALCULATION ⟶ CALCULATION OP CALCULATION
CALCULATION ⟶ id
CALCULATION ⟶ id | FUN_CALL
OP ⟶ + | - | * | /
FUN_CALL ⟶ id(ARGU)
ARGU ⟶ CALCULATION
LET ⟶ let VALFUNS in IDS (EXP |ϵ) end
VALFUNS ⟶ VALDEFS FUNS
IDS ⟶ id | id IDS
VALDEFS ⟶ val EXP

(b) The pictures are as below in next two pages.

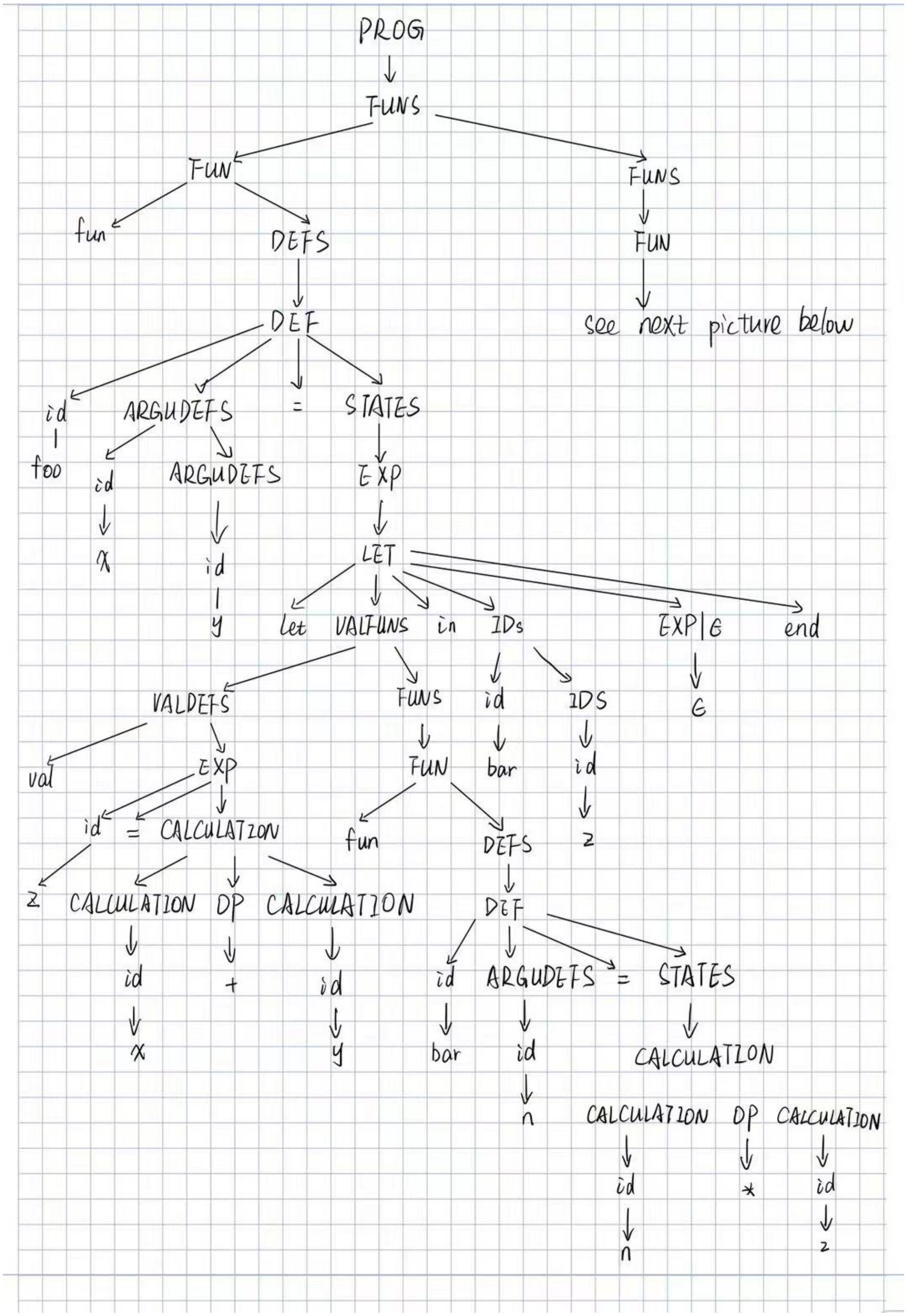


Figure 1: subtree1

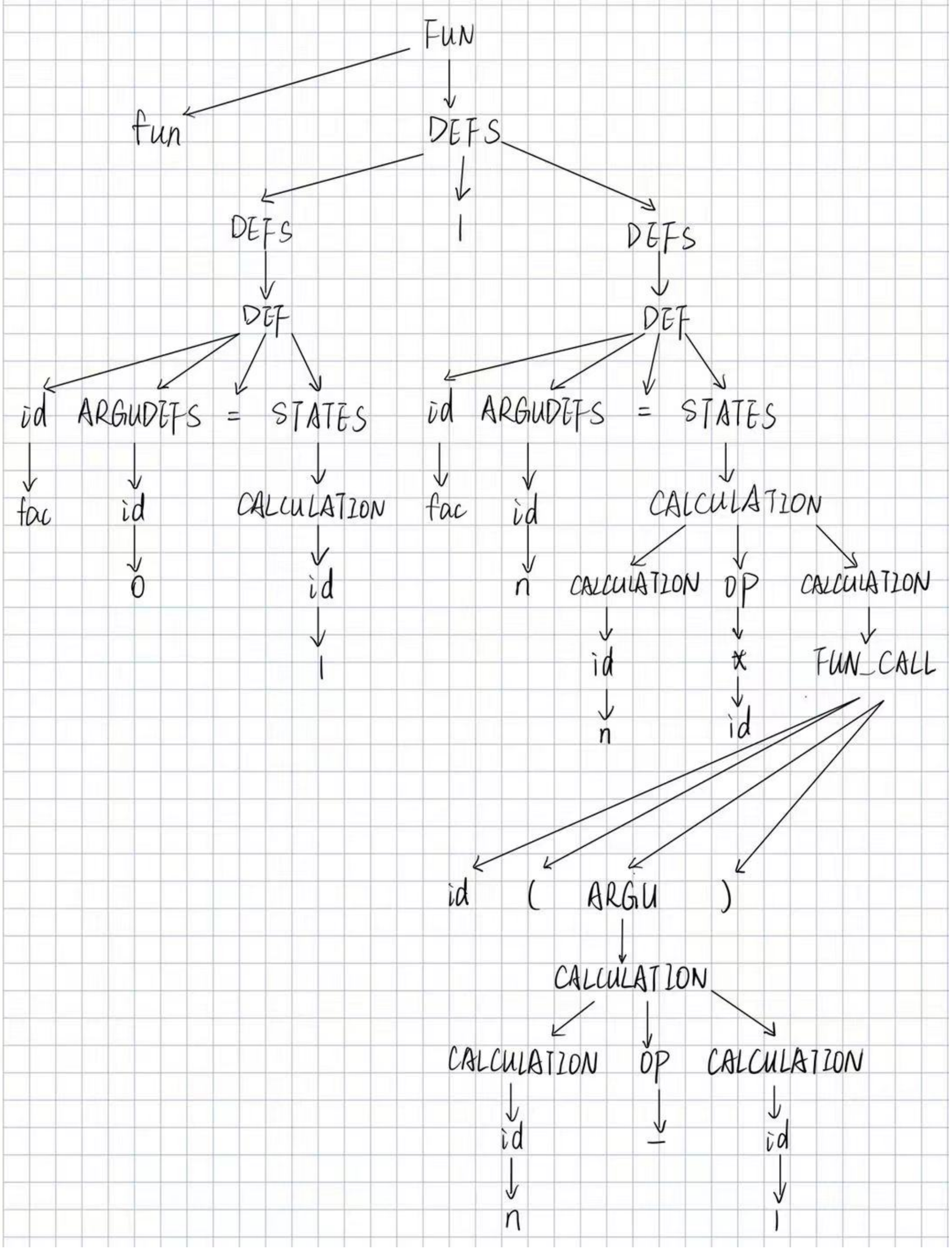


Figure 2: subtree2

3. Scoping

(a) **static scoping:** the body of a function is evaluated in the environment of its **definition**.
dynamic scoping: the body of a function is evaluated in the environment of its **call**.

(b) **Ada code:**

```
procedure test is
        int x = 2;
        procedure print() is
        begin
                print(x);
        end;
        procedure call() is
                int x = 4;
        begin
                print(x);
        end;
begin
        call();
end;
```

For **static scoping**, the result is 2, because the outer scope that x is defined is test(). So the output is 2.
For **dynamic scoping**, the result is 4, because the function which calls print() is call(), where x is defined. The value of x in call() is 4. So the output is 4.

(c) 1. Easy to figure out the value of variables.
2. Implementing more efficiently, with less expense.
3. Easy to keep the code modular.

4. Stack

(a) The picture is as following in next page.
Dynamic link: blue line;
static link: red line;
EP: orange line;

(b) It would print 20,17.

(c) It would print 20,20.

5. Parameter Passing

(a) **pass by value:** 1 2 3 4 5 6 7 8 9 10

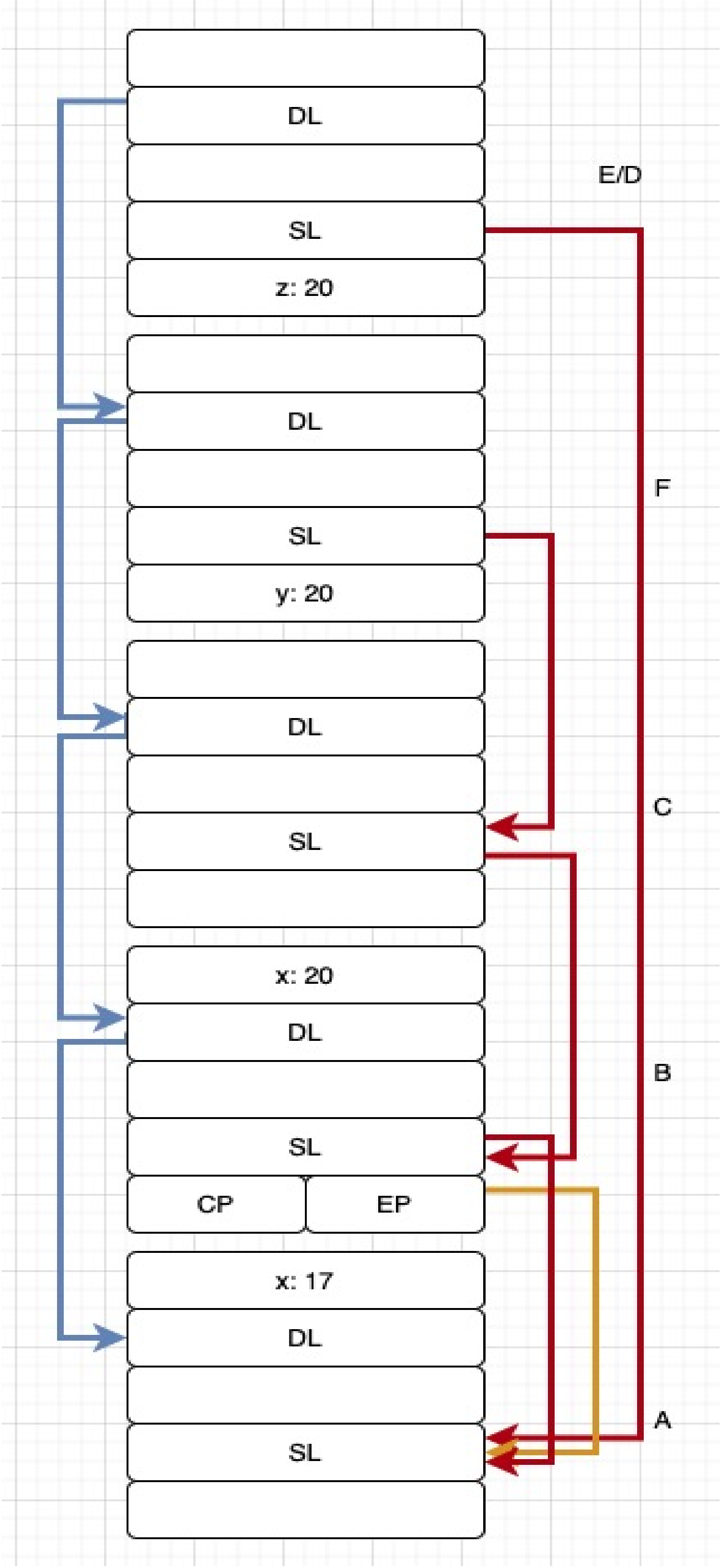


Figure 3: stack for the program

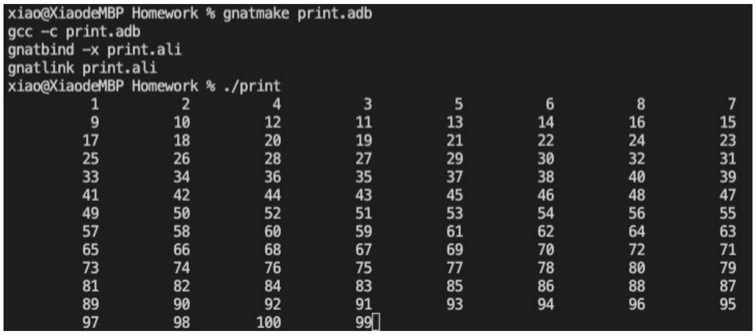
(b) **pass by reference:** 1 45 3 4 5 6 7 8 9 10

(c) **pass by value-result:** 1 15 3 4 5 6 7 8 9 10

(d) **pass by name:** 1 2 3 4 5 6 7 8 45 10

6. Ada Ada code:

```
with text_io;
use text_io;

procedure print is
    package Int_Io is new Integer_Io(Integer);
    use Int_Io;

    i: integer:= 1;
    j: integer:= 1;
    N: integer:= 100;
    task printOdd is
        entry waitOdd;
    end printOdd;

    task printEven is
        entry waitEven;
    end printEven;

    task body printOdd is
    begin
        for i in 1..N loop
            if (i mod 2) = 1 then
                put(i);
            else
                if j-i=1 then
                    printEven.waitEven;
                else
                    accept waitOdd;
                end if;
            end if;
        end loop;
    end printOdd;

    task body printEven is
    begin
        for j in 2..N loop
            if (j mod 2)= 0 then
                put(j);
            else
```



```
xiao@XiaodeMBP Homework % gnatmake print.adb
gcc -c print.adb
gnatbind -x print.ali
gnatlink print.ali
xiao@XiaodeMBP Homework % ./print
          1          2          4          3          5          6          8          7
          9         10         12         11         13         14         16         15
         17         18         20         19         21         22         24         23
         25         26         28         27         29         30         32         31
         33         34         36         35         37         38         40         39
         41         42         44         43         45         46         48         47
         49         50         52         51         53         54         56         55
         57         58         60         59         61         62         64         63
         65         66         68         67         69         70         72         71
         73         74         76         75         77         78         80         79
         81         82         84         83         85         86         88         87
         89         90         92         91         93         94         96         95
         97         98        100         99
```

Figure 4: output for Q6-Ada

```
                if j-i = 1 then
                    accept waitEven;
                else
                    printOdd.waitOdd;
                end if;
            end if;
        end loop;
    end printEven;

begin
null;
end print;
```

output:
The picture above.