Programming Language
Homework #1
Yuqian Zhang
N19945556

1.
a) ([A-Za-z0-9]*) [A-Z] ([A-Za-z0-9]*)[0-9] ([A-Za-z0-9]*)
| ([A-Za-z0-9]*) [0-9]([A-Za-z0-9]*)[A-Z] ([A-Za-z0-9]*)

b) [0-9][0-9]*.[0-9]*[0-9]E[0-9][0-9]*

c) [A-Za-z]([A-Za-z0-9]|_|ε)*]([A-Za-z0-9]|_|ε)*]([A-Za-z0-9]|_|ε)*]([A-Za-z0-9]|_|ε)*]([A-Za-z0-9]|_|ε)*]([A-Za-z0-9]|_|ε)*

2.
a)
P:= program name ; VD FD PD BE|PP
VD:= var name|var name,|var name;| ε|VD VD
FD:= function name (VD) VD BE|ε|FD FD
PD:= procedure name (VD) VD BE|ε|PD PD
BE:= begin EXPR end name;
EXPR:= EXPR:=EXPR;|return EXPR;|name|number|EXPR OP EXPR
|name(VD)||name(EXPR)|print(EXPR);|EXPR EXPR
OP:= +|-|*

b) Please the attached document for the parse tree

3.

a)

Static scoping: the body of a function is evaluated in the environment in which the function is defined.

Dynamic scoping: the body of a function is evaluated in the environment in which the function is called.

b) The difference between static scoping and dynamic scoping.

Consider the following code:

```
x: integer :=7 //line(1)
procedure a()
begin
        b();
end
procedure b ()
        x: integer = 20 //line(2)
        procedure f ()
        begin
                x:= x+1;
        end
begin
        f();
end
```

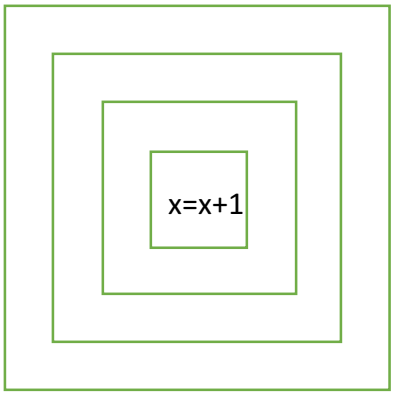
If a() is called, it will start b(), and then call f(). In procedure f, it reassigns the value of variable x.

In the static scoping, x refers to where the function defines, which is in procedure b(line (2)), thus it gets reassigned with the value 21.

In dynamic scoping, x refers to where the function is called, which is in procedure a(line (1)), thus it gets reassigned with the value 8.

c) Rule for resolving variable references in a block structured, statically scoped language.

A variable reference is resolved to the corresponding declaration in the innermost surrounding block.



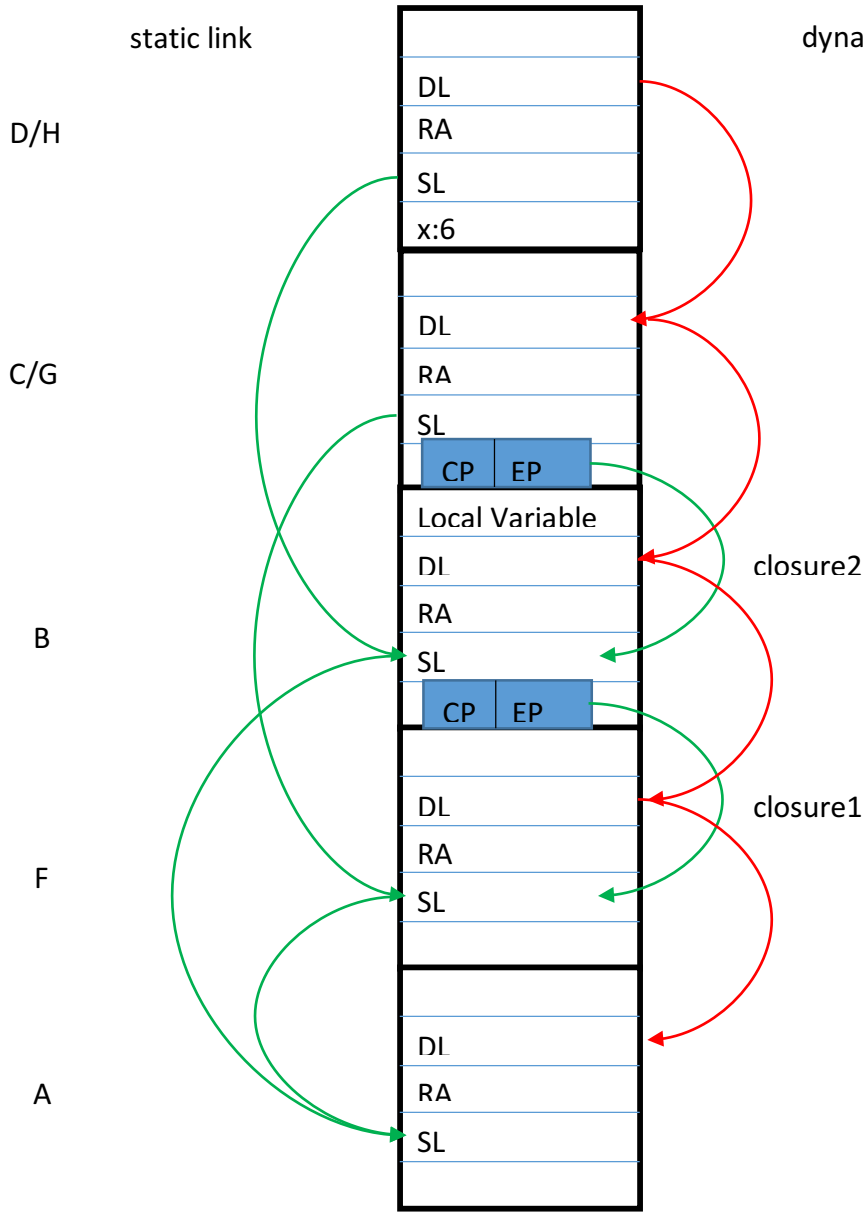
Finding the reference inside out.

d) Rule for resolving variable references in a block structured, dynamically scoped language.

Tracing back to the head of call chain. (looking backwards)

a → b → c → d → e → f

4. a)



b) There are two parts of a closure: code pointer (CP)+ environment pointer(EP)
code pointer (CP)is pointing to the code for the procedure being passed.
environment pointer (EP)is the static link to be used when the passed procedure is called.

c) Dynamic link points in to the stack frame of the calling procedure.
Return address points to the place in the code of the calling procedure to return to.

5.
a) pass by value
10 20 30 40 50
b) pass by reference
31 20 30 40 50
c) pass by value-result
21 20 30 40 50
d) pass by name
10 20 31 40 50

6.
a)

```
with text_io;

procedure main is
    use text_io;

    Task one is
    end one;
    Task two is
         entry Start;
    end two;

    Task body one is
        begin
            loop
            Put("one");
            New_Line;
            two.Start;
            end loop;
    end one;

    Task body two is
        begin
            loop
            accept Start do
            Put("two");
            New_Line;
            end Start;
            end loop;
    end two;

begin
    null;
end main;
```

b) The printing of ‘One’ and the printing of ‘Two’ are occurring concurrently.
It is Rendezvous concurrency.
In my code, two printing tasks communicate by exchanging messages, in other words, message passing.
In the loop, Task one print ‘one’ and then blocks until Task two finishes its printing of ‘two’.
This synchronous message passing method promises that ‘one ‘ and ‘two’ are printed in right order.