

Example

main

procedure P

begin

procedure S begin ... end S

if random(1) < 1 then P()

else { S(); Q() }

end P;

procedure Q begin ... end Q;

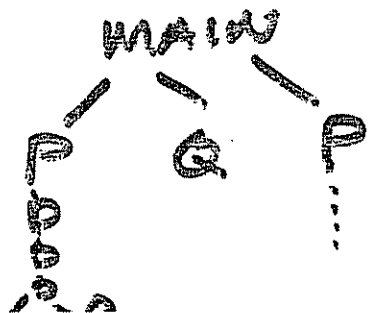
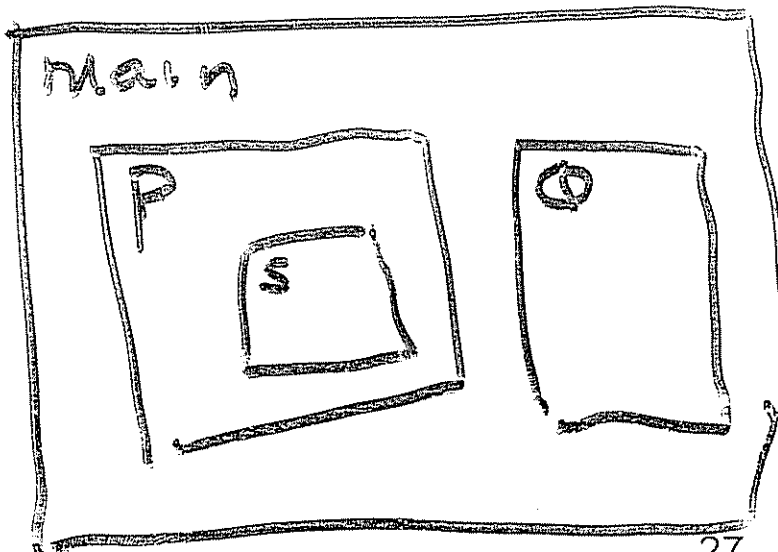
P; —

Q; —

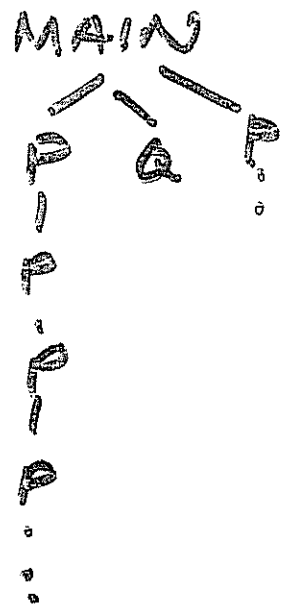
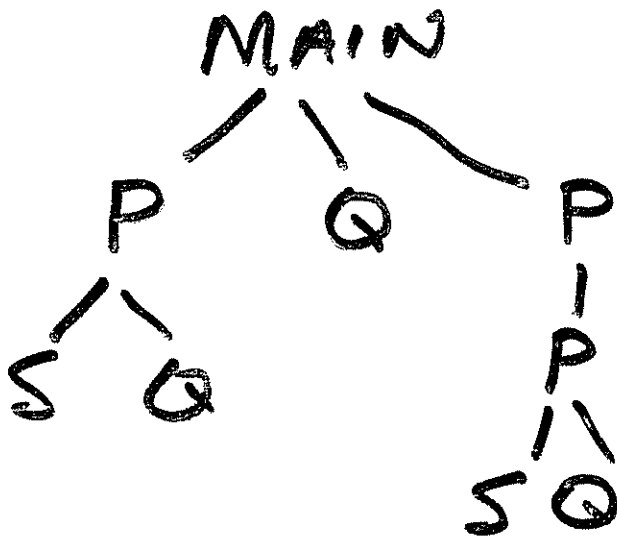
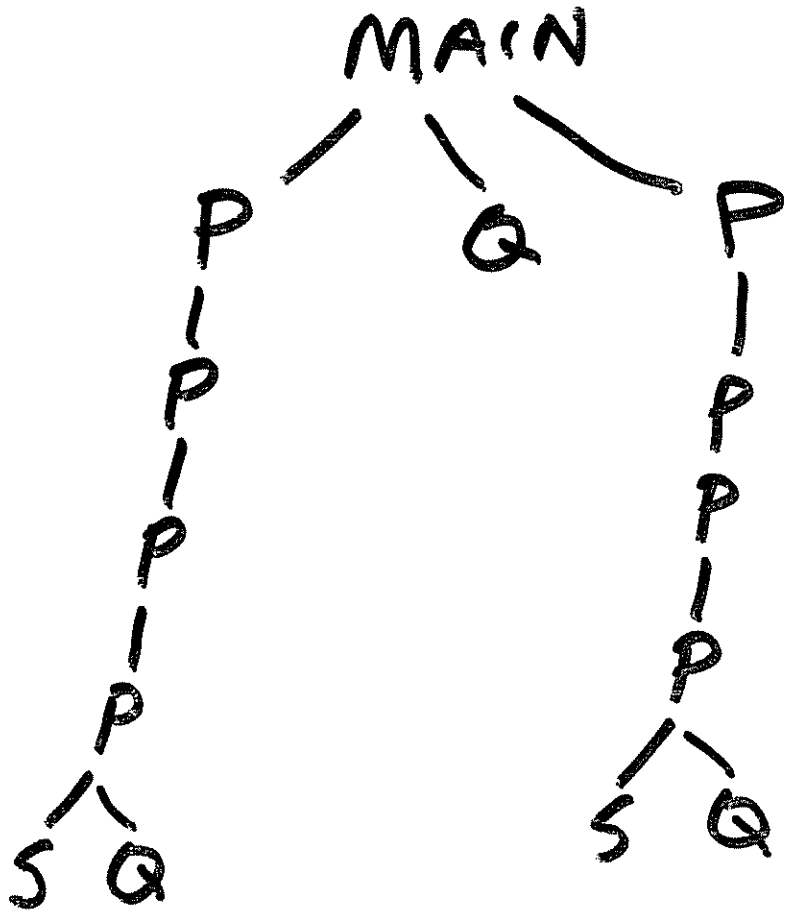
P; —

end

Note Recursive



Sample Activation Trees



ACTIVATION TREE

MAIN

MAIN

P

MAIN

P

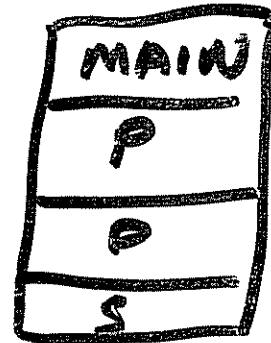
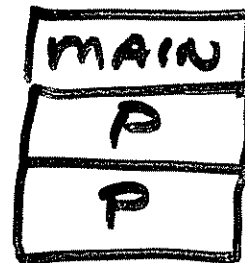
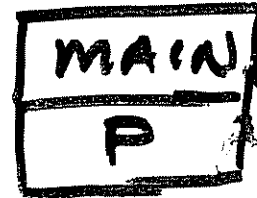
MAIN

S

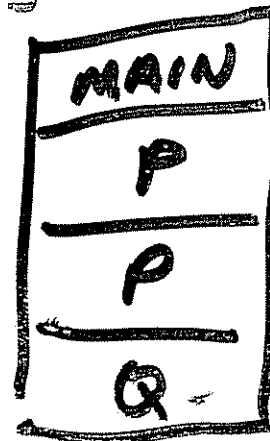
MAIN

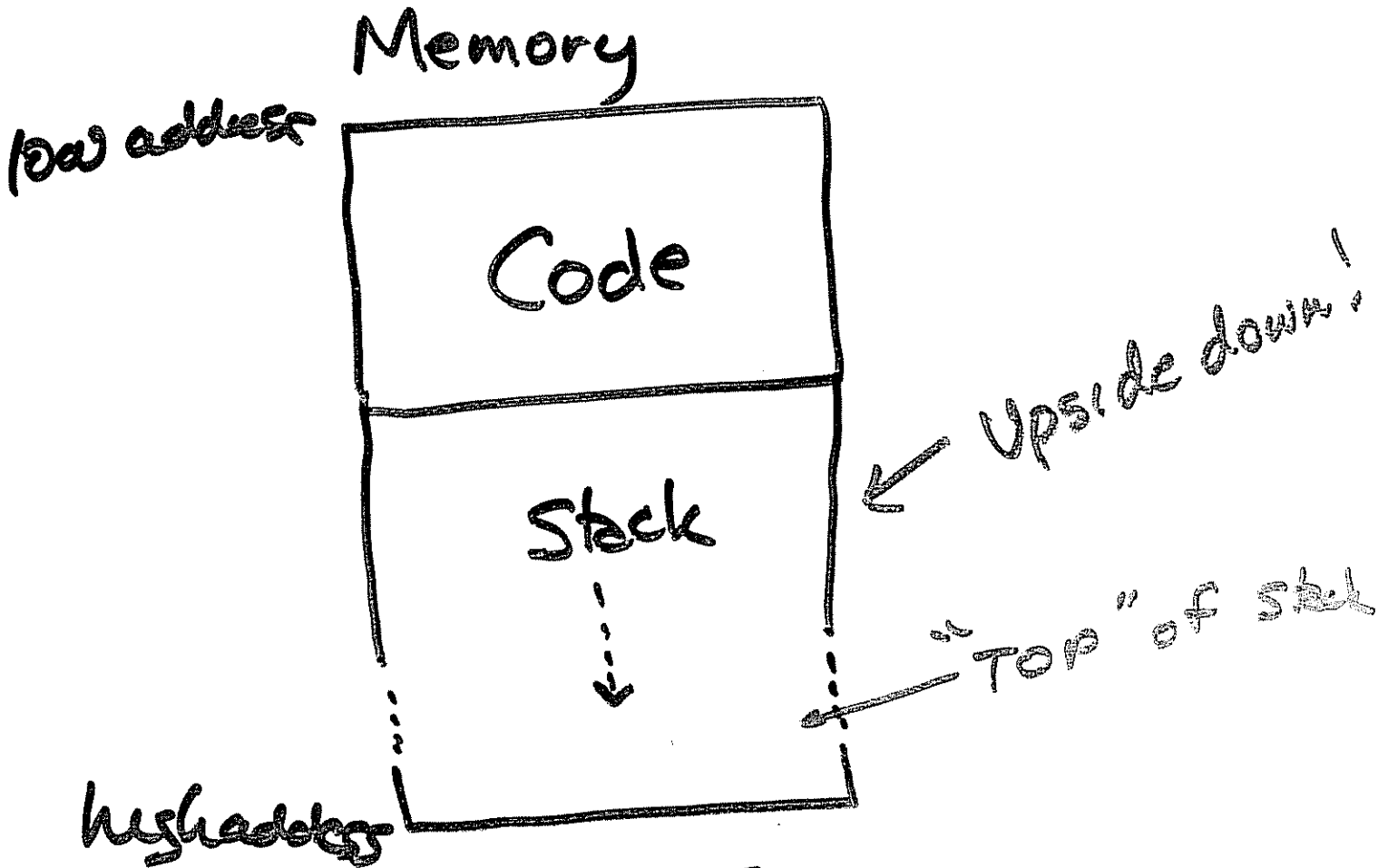
P

RUNTIME STACK

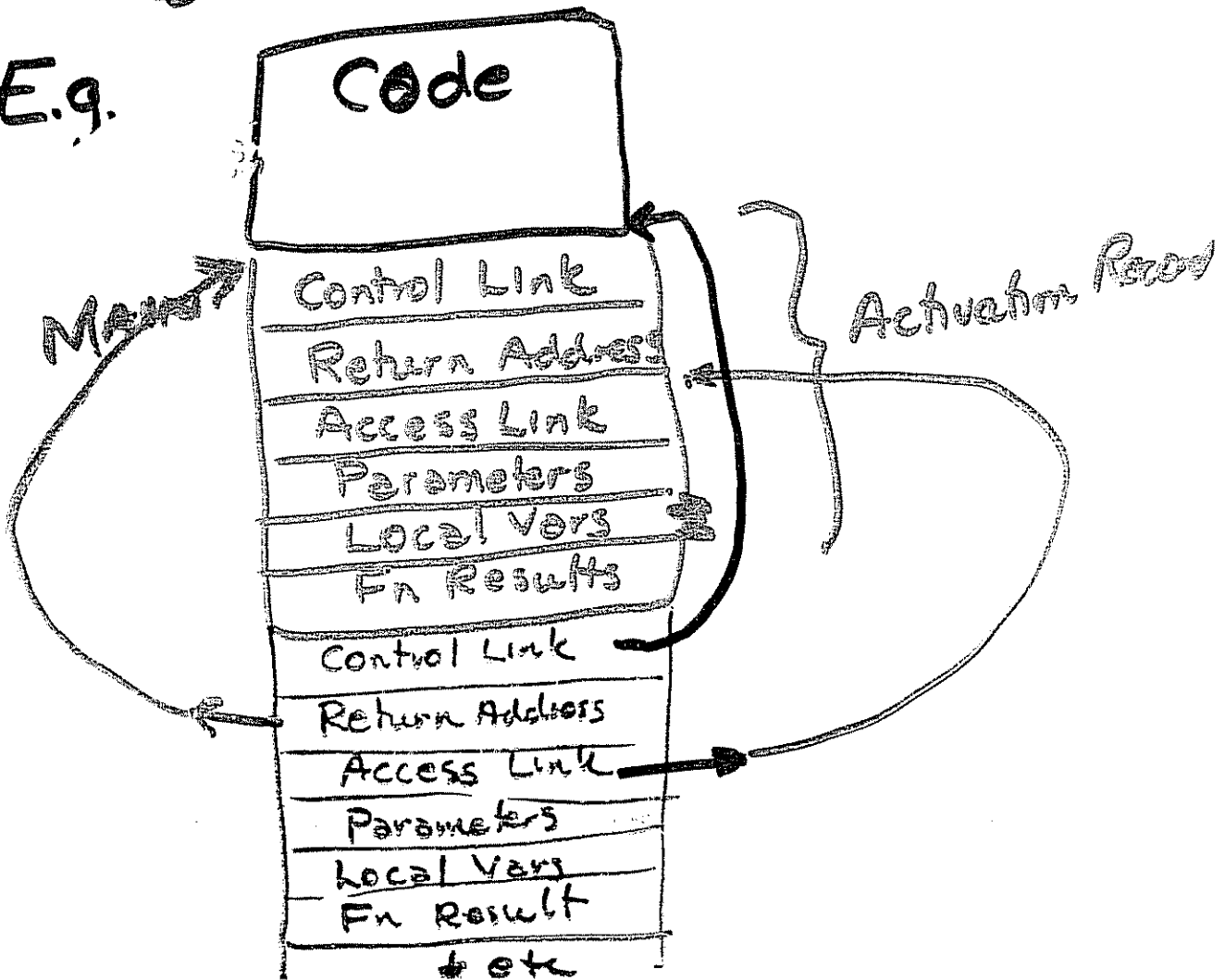


POP S



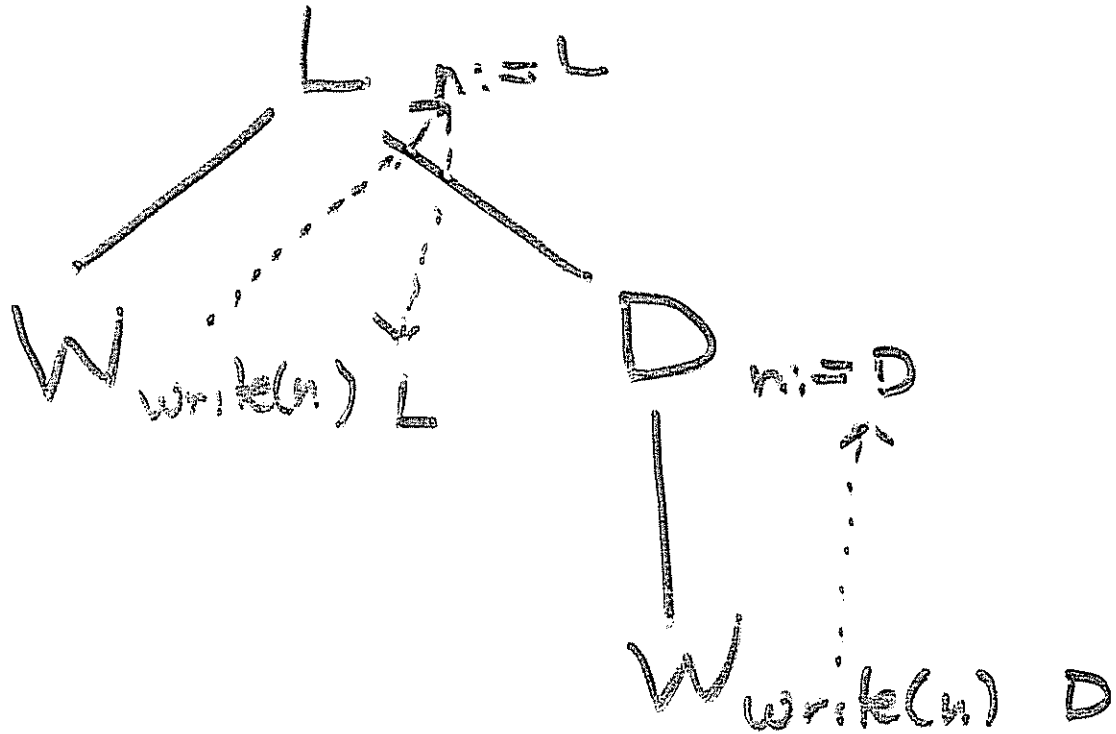


E.g.



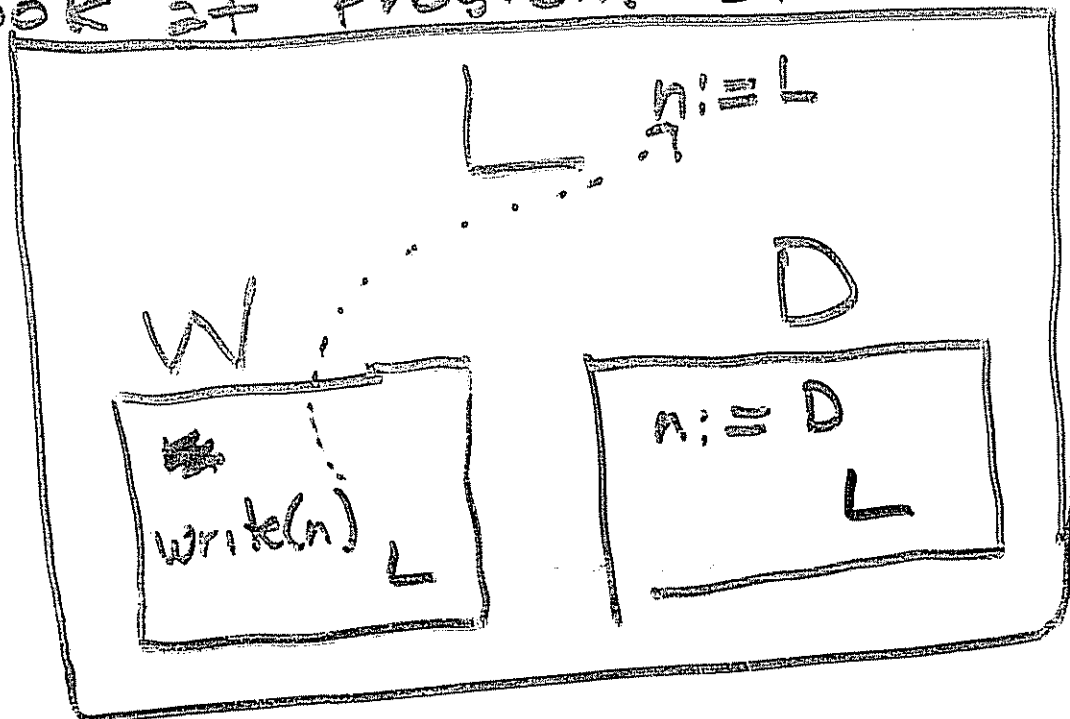
Dynamic Scope

⇒ Look @ Activation Tree

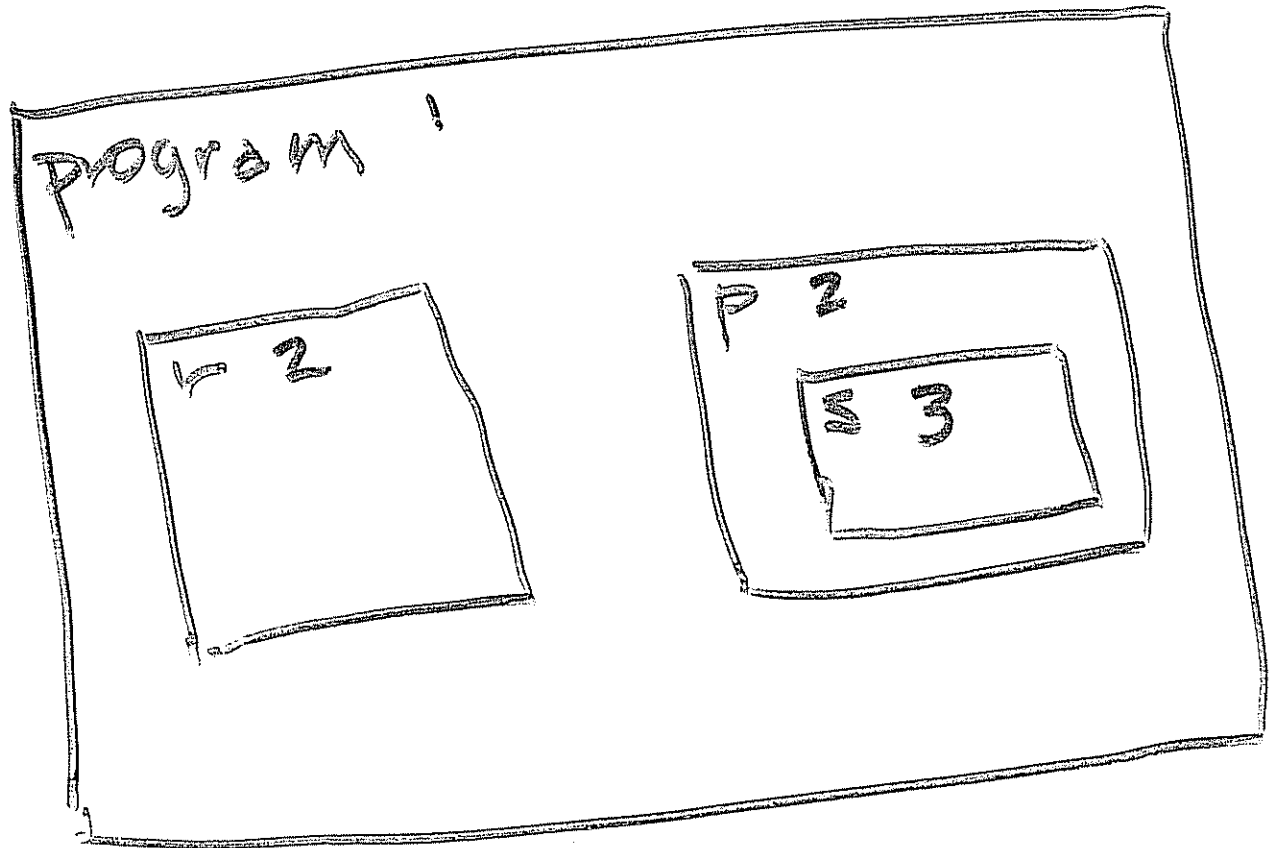


Static/Lexical Scope

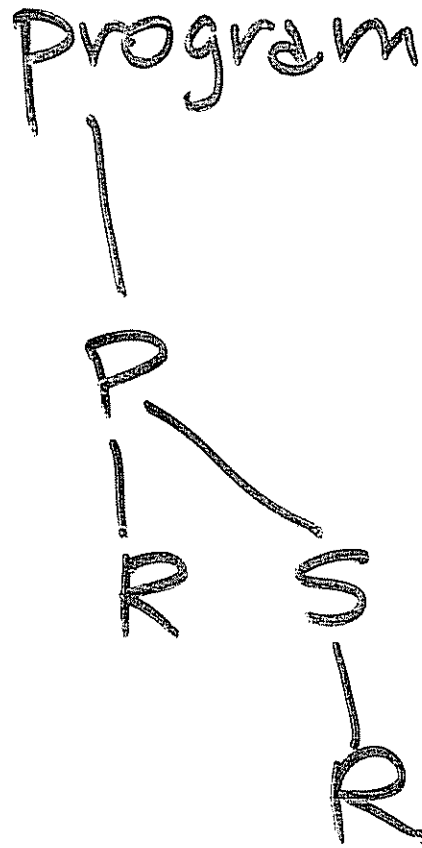
⇒ Look at Program Structure



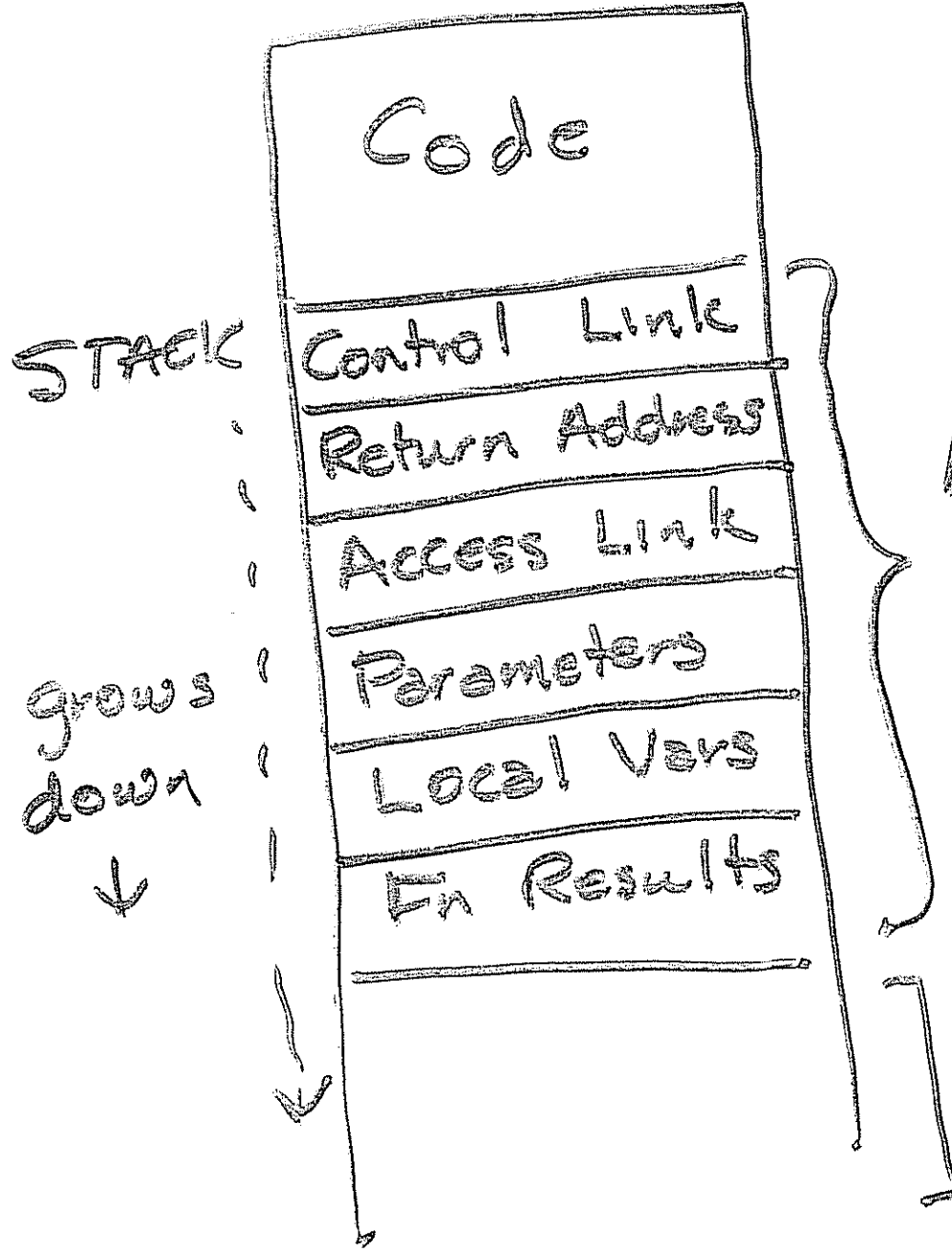
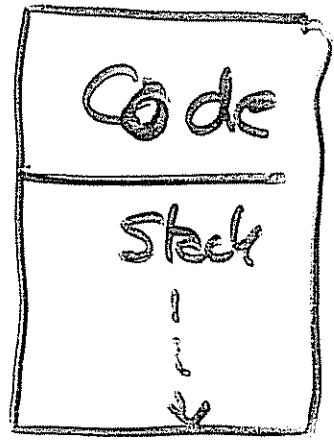
Draw the Program Structure



What's the Activation Tree?

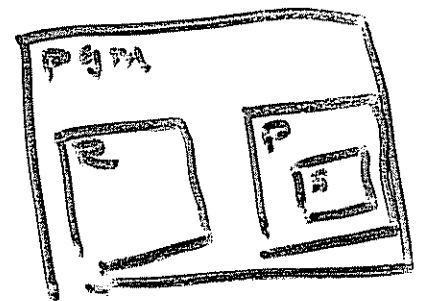
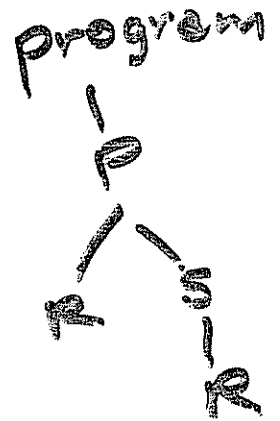
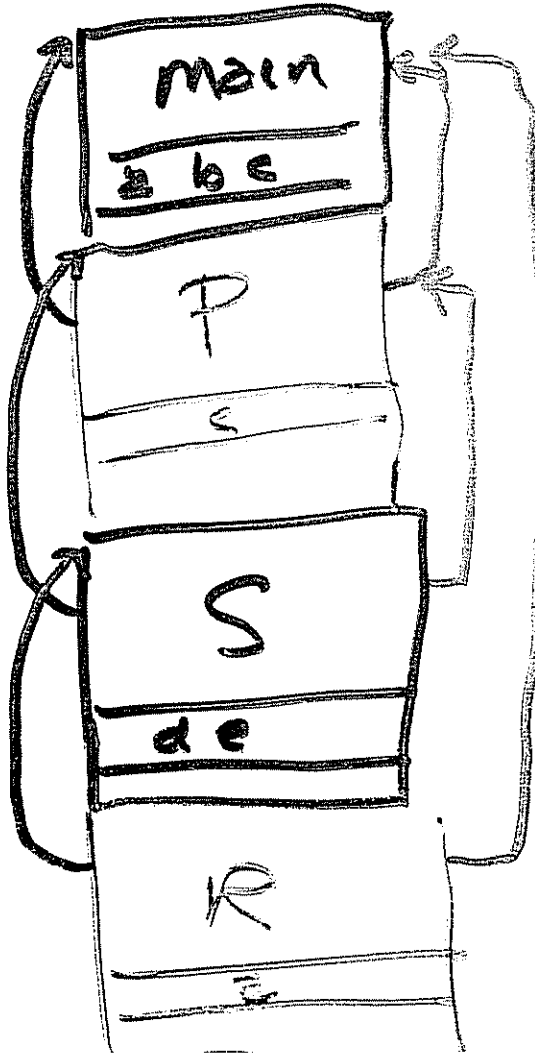
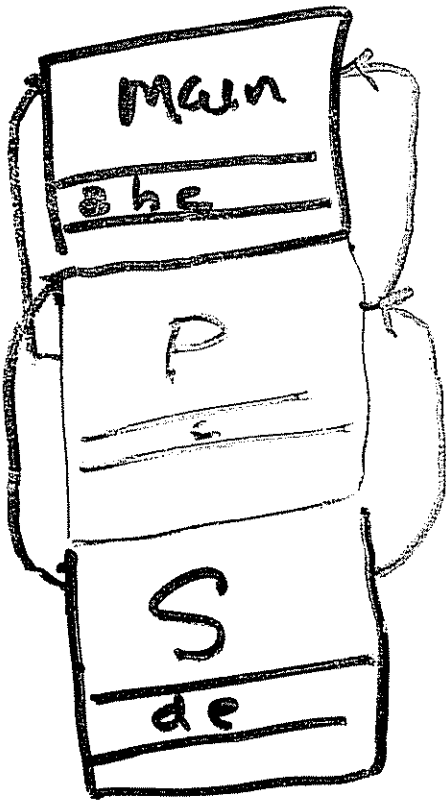
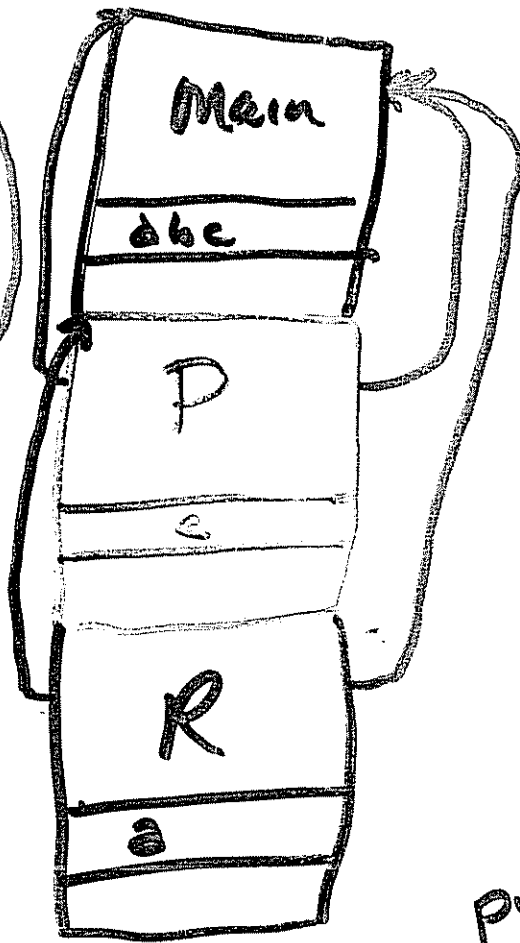
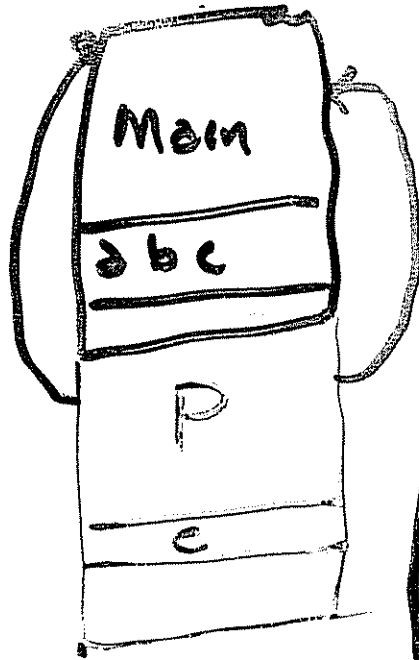
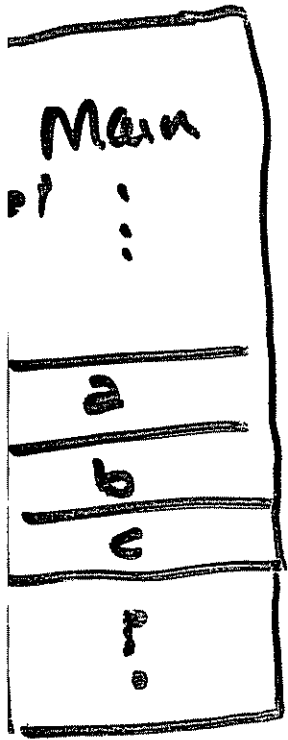


Recall What's In
Our Stack!



A typical
Activation
Record

Another record
pushed onto
stack



To determine the access link
for name a, b, c in S
follow $n-m$ access links from
procedure S (where a, b, c are used)

① Nesting Depth, n , of S

for a $n = 3$

for b $n = 3$

for c $n = 3$

② Nesting Depth of declaration, m , of a, b, c

for a $m = 1$

for b $m = 1$

for c

$m = 2$

← because
re declared
in procedure P

∴ For a, b follow $n-m = 2$ links
For c follow $n-m = 1$ link

Dynamic Scope Example

program

a : integer;

procedure z

a : integer; ...

a := 1;

y;

output a; ←←

end z;

procedure w

a : integer; ...

a := 2;

y;

output a; ←←

end w;

procedure y ...

a := 0;

end y;

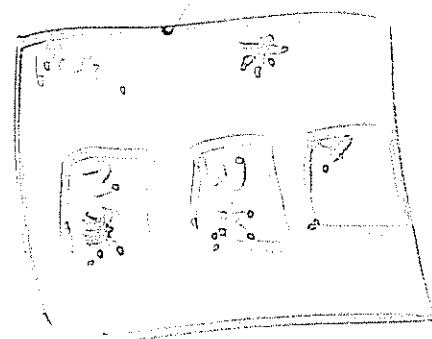
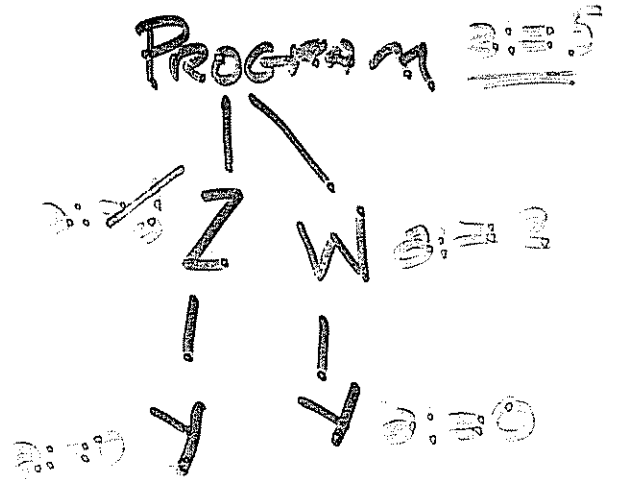
a := 5;

z;

w;

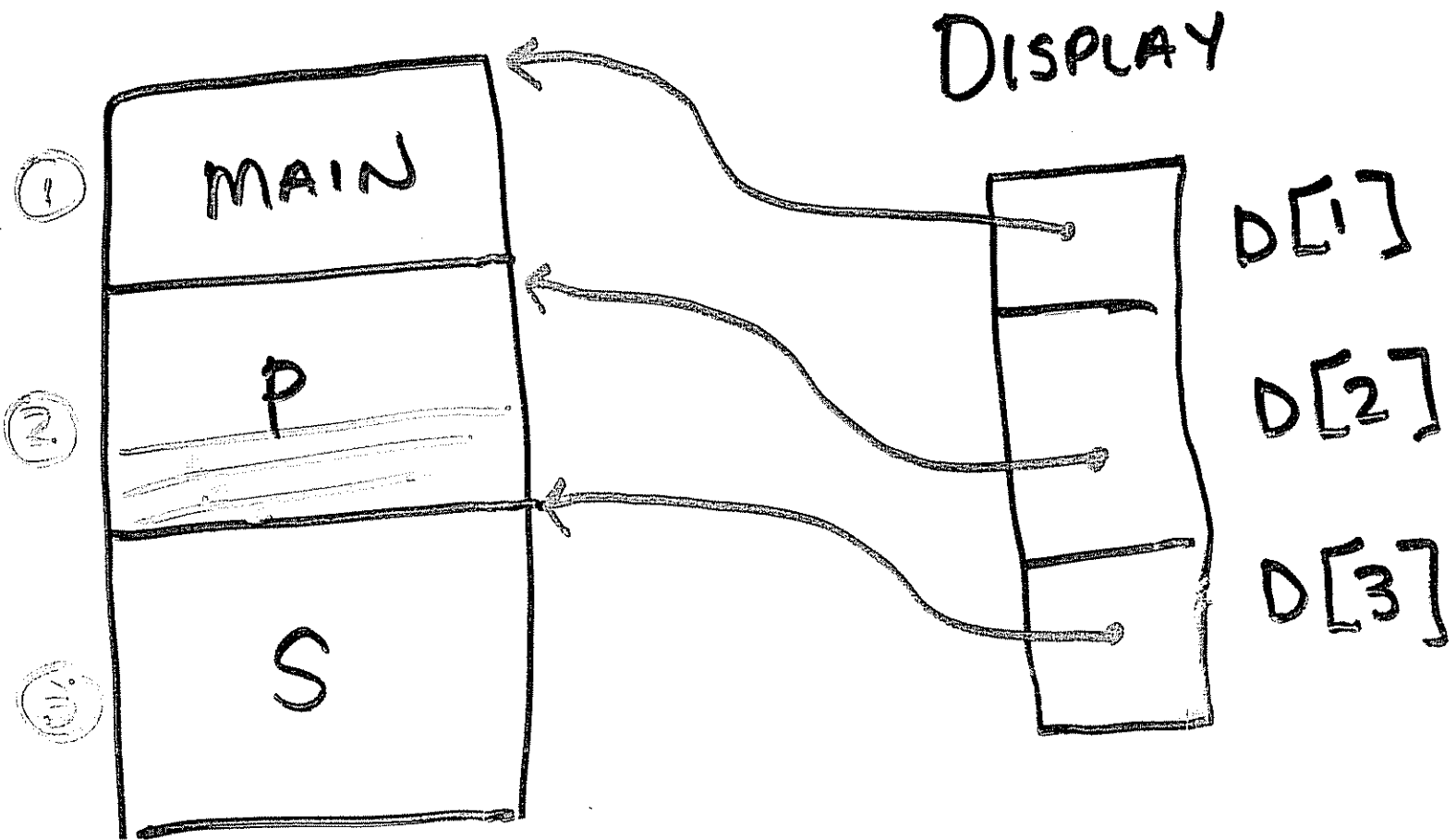
output a; ←←

end



no declaration

005



ISA Mother(x)

①

female(x), parent(x, -)

x = Janet

⑩

parent(Janet, -)

⑩

Success

x = Janet

Janet

x = Diane

parent(Diane, -)

|

Success

x = Wil

x = Cathy

parent(Cathy, -)

|

Success

x = Kit

isa(x) ^{top(x)}

16 female(x), parent(x, -)!
x=jane

parent(jane, -)!
/ 10

SUCCESS

x=jane

top2(x)

female(x) isa2(x)

x=jane

x=diane

x=cothy

isa2(jane)

parent(jane, -)!

/ 10

SUCCESS x=jane

- ① holiday (fri, apr 14)
- ② weather (fri, fair)
- ③ weather (sat, fair)
- ④ weather (sun, fair)
- ⑤ weekend (sat)
- ⑥ weekend (sun)
- ⑦ picnic (Day): - weather (Day, fair), weekend (Day)
- ⑧ picnic (Day): - holiday (Day, apr 14)

