

Compiler Construction 2018/19

— Exercise Sheet 11 —

Hand in until January 14th, 2019 before the exercise class.

General Remarks

- This is the last exercise sheet. If you have gained at least **114 points in total** (50% of 228), you are admitted to the exam.

Exercise 1

(5 Points)

Consider the following EPL program:

```

in/out x;
var y;
proc P;
    x := 3 * x
proc Q;
    x := x - y
if x <= 100 or y == x then
    P()
else
    y := 5;
    Q()
  
```

- Translate the EPL program to abstract machine (AM) code.
- Translate the EPL program to AM code with jumping code.
Hint: you only need to give the parts of the code which need to be changed.

Exercise 2

(5 Points)

Let X, Y and Z be global variables in the program given below.

Listing 1: A simple pointer program.

```

1  class Node {
2      Node* left = NULL;
3      int value = 0;
4      Node* right = NULL;
5
6      public Node(int v) { this.value = v; }
7  }
8
9  Node* X := NULL;
10 Node* Y := NULL;
11 Node* Z := NULL;
12
13 func main() {
14     X := new Node(1);
15     X.left = X;
  
```

```

16 X.right = new Node(2);
17 Y := X.right;
18 Z := getTree(Y); // see heap diagram for result
19
20 // ← The heap presented belongs to the program in this state
21
22 X := new Node(3);
23 X.left := new Node(4);
24 X.value = 42;
25 X.right := new Node(5);
26 Z.left := NULL;
27 }

```

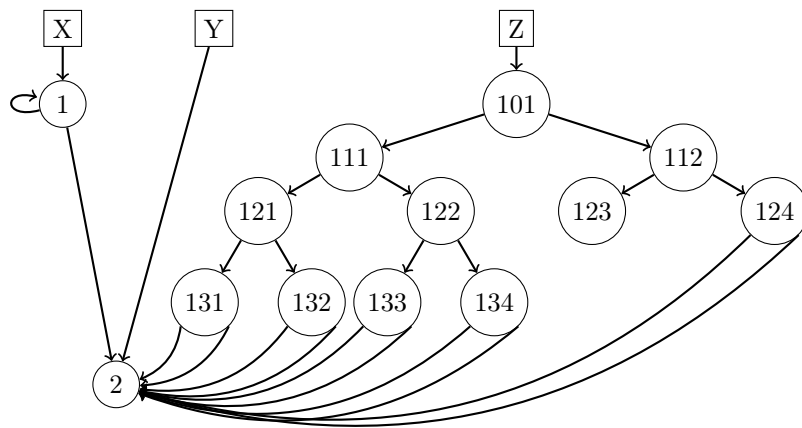


Figure 11.1: The state of the heap at the marked program location.

- Calculate the reference count for each object `Node` at the marked location in the program.
- Show the heap after executing the remaining statements of `main()`, including the reference counts, but before any garbage collection has occurred.
- Perform a garbage collection run using mark-and-sweep and indicate which nodes get “collected” by drawing the resulting heap.
- Perform a garbage collection run using reference counting and indicate which nodes get “collected” by drawing the resulting heap.