Exercises 14 (Run-time organization)

14A *(Data representation)*

Draw diagrams showing how the types of Exercise 2B would be represented in a computer.

14B *(Global and local storage allocation)*

Draw diagrams showing the allocation of storage to the global and local variables of the C programs in Exercise 9A(a) and (b).

14C *(Global, local, and heap storage allocation)*

Draw diagrams showing the allocation of storage to the global, local, and heap variables of the Java program in Exercise 9B.

14D *(Garbage collection)*

A copying or generational garbage collector copies live heap variables from one space to another. All pointers to copied heap variables must be redirected. Suggest how this could be done efficiently.

Exercises 15 (Native code generation)

15A *(Register allocation)*

Consider the following C function:

```
void f (int a) {
  int b = a+1;
  int c = b*b-1;
  int d = a-c;
  b = d+7;
  return b*2;
}
```

Translate the function’s body into a single basic-block. Each instruction in the basic-block should contain at most one operator. Complex expressions should be broken down using temporary variables for intermediate results.

Beside the basic-block, show where each local/temporary variable is live.

Allocate registers to all local and temporary variables. Use as few registers as possible.

15B *(Code selection)*

Consider a C program in which:

- Local variables `d` and `i` are of type `int`, and are located at offsets 8 and 12 (respectively) relative to the base of the activation frame.
- Global array `a` is of type `int*`, and a pointer to the base of the array (i.e., a pointer to `a[0]`) is contained at global address 16.

(a) Show how the following C statement:

```
d = i - 1;
```

would be represented as an IR tree, using the IR summarized on slide 15-15 of the course notes.

Find at least one way to “cover” the IR tree, using the Jouette instruction patterns of slides 15-19 and 15-20. For that covering, write down the corresponding Jouette object code, assuming a reasonable register allocation.

(b) Show how the following C statement:

```
d = a[i] – a[1];
```

would be represented as an IR tree.

Find at least one way to “cover” the IR tree. For that covering, write down the corresponding Jouette object code, assuming a reasonable register allocation.