



Programming Languages II

Unless otherwise stated, exercises should be submitted in electronic form, via the collaborative learning environment `moodle.softlab.ntua.gr`. Deadlines will be strict. You are allowed at most one late exercise.

Exercise 2 Garbage collection

Deadline: 15/12/2019

Implement a garbage collector for the extension of the virtual machine of exercise 1, defined below. The language to be implemented will be called **Befunge-93+** and will be the extension of **Befunge-93** with three new instructions, `c`, `h` και `t`.

We recommend that you use C/C++ as the implementation language.

Values. The values stored in the machine's stack are now of two types: signed integer numbers of at least 62 bits, and pairs (cons cells) consisting of two values, which are stored on the heap (boxed) and are represented by their location.

Instructions. Three more instructions are added to the virtual machine of exercise 1:

- `c` (cons): removes from the top of the stack in turn two elements, `b` and `a`, creates in the heap a new pair (cons cell) containing these two elements (`a` in the first position and `b` in the second) and pushes on top of the stack the pair's location.
- `h` (head): removes an element from the top of the stack, which must be the location of a pair (cons cell) in the heap; if pushes on top of the stack the value that is contained in the first position of this pair (head).
- `t` (tail: same as `h` but for the second position of the pair (tail).

Example of execution. The commands `base64` and `zcat` come from some operating system that respects itself (e.g., Linux).

```
$ base64 -d << __EOF__ | zcat > pp.b
H4sIAOgryl0AAy2PQw7DIBBF95wCY7KZMU2wgcAIod6imwhXiqomqRaVeVVDt/BNYvPE7z5gp
hwlQ0Mf1Ba6Sz3tS5u1LmsgpG3/7MtB/WiNmDZGr4v9CzWhDMU6hq11KBxX+d6Es0s+nWvLbVc
rnSzSJHbemFXNm2+WXsmZhrv07AHjvcyQpsyj7p0E6rZJybuGctqzClumMgFuQNRQCHKIEZC
s4uYpnZ9DyVxBi0KILi5saHhjpiNx+Gj2o7vPx/f0mCVXD3F5qjHzcr1/3D/EHPJ9Xox4BAAA=
__EOF__

$ ./befunge93+ pp.b
.....$
(snip: 17 such lines in total)
```

Caution: Set your virtual machine to use 2^{20} words (each one with 64 bits) for the stack and a total of 2^{24} words for the heap.

The program of the previous example generates pairs (cons cells) with a total size of 8,33GB. Unless you violate the above limits and you have too much memory on your computer, this program cannot run unless you implement a garbage collector.