

## TDT 4205

### Problem Set 4

#### 1. Memory layout

1.1. Does a VSL program require any heap memory at run time?. Explain.

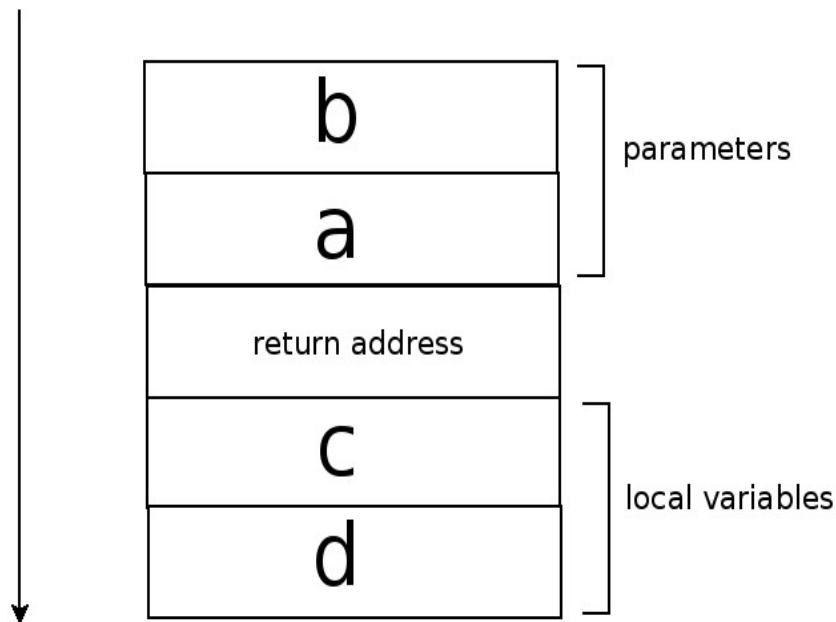
No, because VSL is not going to allow dynamic memory allocation, so it is not necessary.

1.1 Describe the layout for a stack frame of the following function

The push-sequence into the stack would be:

- Push each actual parameter (in reverse order): push b, push a.
- Push return address
- Push local variables: push c, push d.

So at this moment, the view of the stack would be:



# Scoping

2.1 Which values would you expect the above code to print under these assumptions?.

It would print:

- $2 * 3 * (5^{(3-1)}) = 150$
- $3 * 4 * (5^{(4-1)}) = 1500$

2.2 Show how you would structure the symbol table entries for `df_dx` and `dg_dx` in order for the language to handle closures properly.

We should create a new kind of type of variable which alert the compiler when is going to treat a function as a value. This would be the type stored in the symbol table.

The idea is:

- When we call the function `differentiate` we initialized the values of the activation record that we store in the position memory of the variable.
- We really execute the code when we have already an instance of the `variable-function` and we make a call with the parameter `x` making the substitution of the stack with its own activation record.