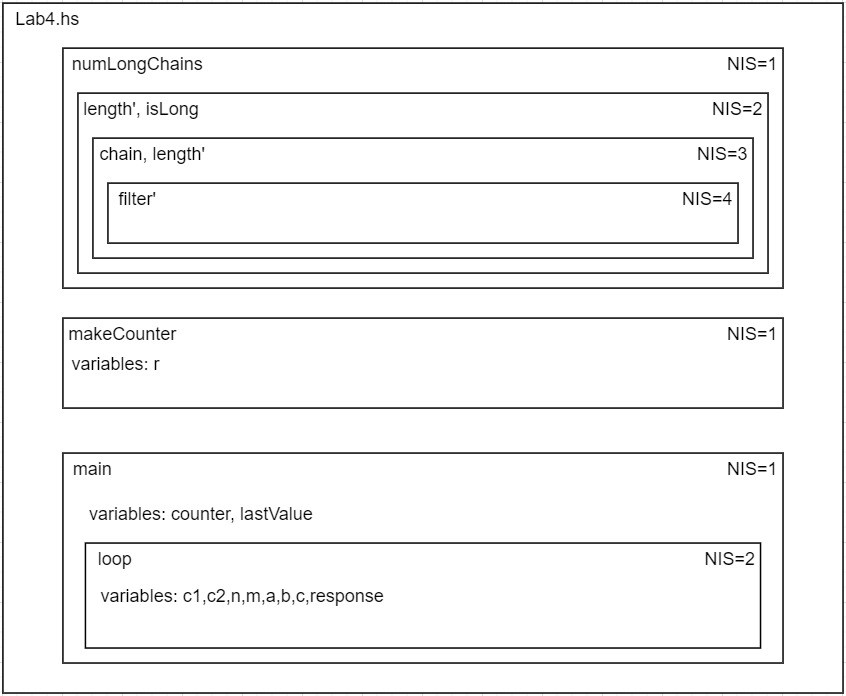
Problem and solution description

The problem I chose to solve is the following: given an integer upper bound and minimum length, find out the number of numbers smaller than the upper bound whose Collatz sequence chain is longer than the minimum length. Each implementation adopts the following structure: a function or combination of functions which generates the chain for a given number or multiple given numbers; a function or combinations of functions which generate the lengths of all the chains of all the numbers smaller than the specified bound; a function or combination of functions which filters the chains whose length is smaller than the required minimum and a function which counts the remaining valid chains. In the case of Haskell, the simulation of a global mutable variable was also implemented for didactic purposes.

Haskell program analysis

Diagram



There are no dangling references problems since this feature is not available in Haskell (as far as I studied).

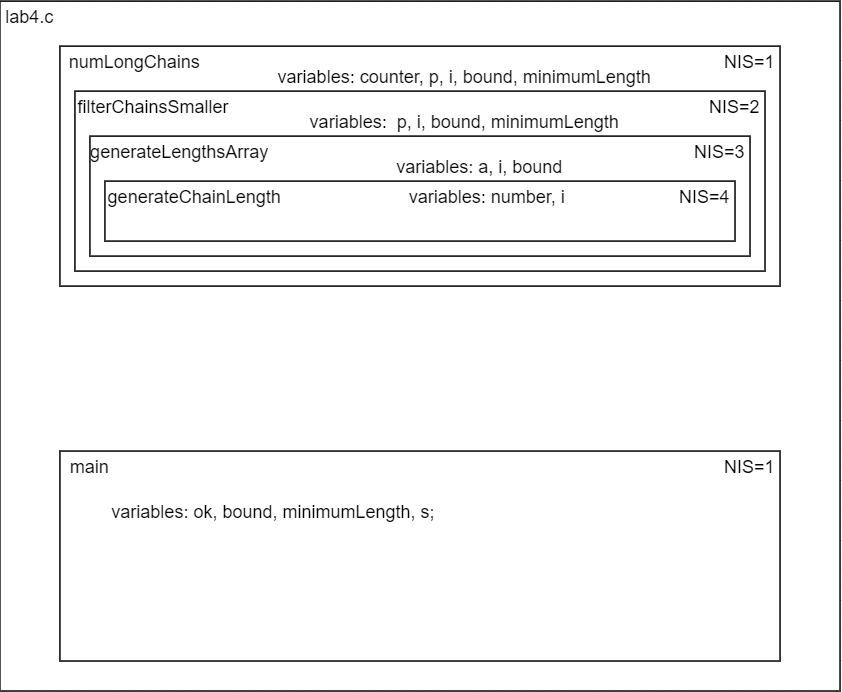
Variables Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable | Moment of allocation | Place of allocation | Visibility Domain (VD) | Life  Duration (LD) | Relationship between the last 2 |
| r | Run time | Heap | makeCounter function and any subsequent local function | from the moment it is declared in makeCounter until it finishes | VD determines LD |
| counter | Run time | Heap | main function and any subsequent local function | from the moment it is declared in main until it finishes | VD determines LD |
| lastValue | Run time | Heap | main function and any subsequent local function | from the moment it is declared in main until it finishes | VD determines LD |
| c1 | Run time | Heap | loop function and any subsequent local function | from the moment it is declared in loop until it finishes | VD determines LD |
| c2 | Run time | Heap | loop function and any subsequent local function | from the moment it is declared in loop until it finishes | VD determines LD |
| n | Run time | Heap | loop function and any subsequent local function | from the moment it is declared in loop until it finishes | VD determines LD |
| m | Run time | Heap | loop function and any subsequent local function | from the moment it is declared in loop until it finishes | VD determines LD |
| a | Run time | Heap | loop function and any subsequent local function | from the moment it is declared in loop and until it finishes | VD determines LD |
| b | Run time | Heap | loop function and any subsequent local function | from the moment it is declared in loop until it finishes | VD determines LD |
| c | Run time | Heap | loop function and any subsequent local function | from the moment it is declared in loop until it finishes | VD determines LD |
| response | Run time | Heap | loop function and any subsequent local function | from the moment it is declared in loop until it finishes | VD determines LD |

The dynamic evolution is strictly determined by the static structure in this case since there is no place which is directly affected by the lazy nature of Haskell.

C++ program analysis

Diagram



A dangling reference could appear if the variable a was not made static or, after being made static, no pointer to it was kept or returned after exiting the function generateLengthArray.

Variables Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable | Moment of allocation | Place of allocation | Visibility Domain (VD) | Life  Duration (LD) | Relationship between the last 2 |
| ok | Compile time | Stack | main function and any subsequent local function | from the moment it is declared in main until it finishes | VD determines LD |
| minimumLength\_main | Compile time | Stack | main function and any subsequent local function | from the moment it is declared in main until it finishes | VD determines LD |
| bound\_main | Compile time | Stack | main function and any subsequent local function | from the moment it is declared in main until it finishes | VD determines LD |
| s | Compile time | Stack | main function and any subsequent local function | from the moment it is declared in main until it finishes | VD determines LD |
| counter | Compile time | Stack | numLongChains function and any subsequent local function | from the moment it is declared in numLongChains until it finishes | VD determines LD |
| p\_numLongChains | Compile time | Stack | numLongChains function and any subsequent local function | from the moment it is declared in numLongChains until it finishes | VD determines LD |
| i\_numLongChains | Compile time | Stack | numLongChains function and any subsequent local function | from the moment it is declared in numLongChains until it finishes | VD determines LD |
| minimumLength\_main\_numLongChains | Compile time | Stack | numLongChains function and any subsequent local function | from the moment it is declared in numLongChains until it finishes | VD determines LD |
| bound\_filterChainsSmaller | Compile time | Stack | filterChainsSmaller function and any subsequent local function | from the moment it is declared in filterChainsSmaller until it finishes | VD determines LD |
| p\_filterChainsSmaller | Compile time | Stack | filterChainsSmaller function and any subsequent local function | from the moment it is declared in filterChainsSmaller until it finishes | VD determines LD |
| i\_filterChainsSmaller | Compile time | Stack | filterChainsSmaller function and any subsequent local function | from the moment it is declared in filterChainsSmaller until it finishes | VD determines LD |
| minimumLength\_main\_filterChainsSmaller | Compile time | Stack | filterChainsSmaller function and any subsequent local function | from the moment it is declared in filterChainsSmaller until it finishes | VD determines LD |
| bound\_filterChainsSmaller | Compile time | Stack | filterChainsSmaller function and any subsequent local function | from the moment it is declared in filterChainsSmaller until it finishes | VD determines LD |
| a | Compile time | GDS | generateLengthsArray function and any subsequent local function | from the moment it is declared in generateLengthsArray until the program finishes | Link is broken by the static keyword |
| i\_generateLengthsArray | Compile time | Stack | generateLengthsArray function and any subsequent local function | from the moment it is declared in generateLengthsArray until it finishes | VD determines LD |
| bound\_generateLengthsArray | Compile time | Stack | generateLengthsArray function and any subsequent local function | from the moment it is declared in generateLengthsArray until it finishes | VD determines LD |
| number\_generateChainLength | Compile time | Stack | generateChainLength function and any subsequent local function | from the moment it is declared in generateChainLength until it finishes | VD determines LD |
| i\_generateChainLength | Compile time | Stack | generateChainLength function and any subsequent local function | from the moment it is declared in generateChainLength  until it finishes | VD determines LD |

The dynamic evolution is strictly determined by the static structure in this case since there are no dynamic variables which can change the structure.