Template-based Synthesis of Heap Abstractions

Viktor Malík

imalik@fit.vutbr.cz

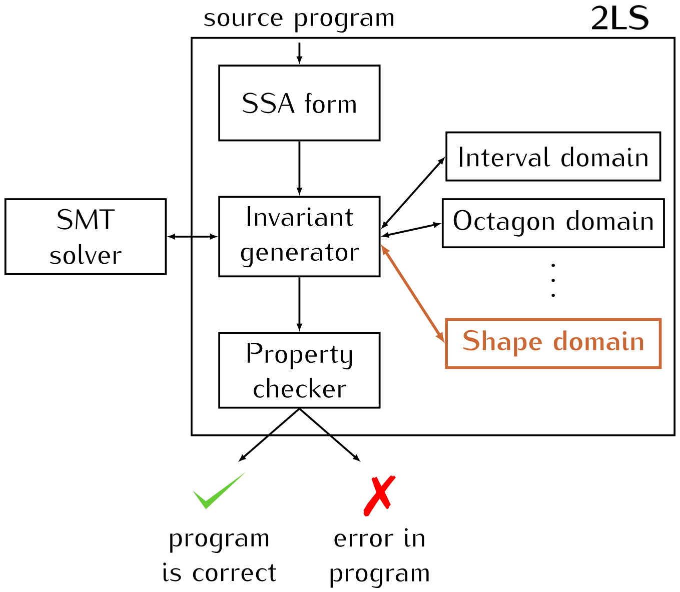
Faculty of Information Technology, Brno University of Technology

Supervisor: prof. Tomáš Vojnar

Motivation

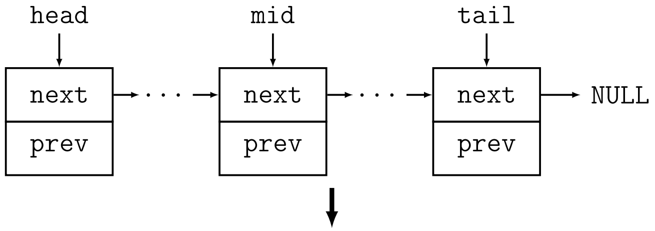
2LS is a program analysis framework for C programs. Currently, it is well-usable for analysis of numerical variables in programs, but it lacks the ability to analyse programs manipulating dynamic data structures.

In this work, we give a solution to the integration of shape analysis into 2LS, which is aimed to analyse the shape of dynamic data structures.



We propose a new abstract domain to describe the shape of the heap, which is used by 2LS to analyse programs manipulating dynamic data structures.

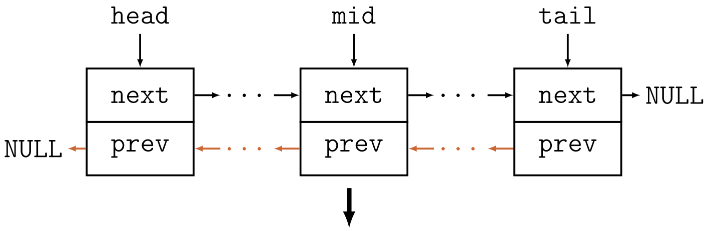
Example



Invariant

$head = \&do_0 \ \wedge \ mid = \&do_1 \ \wedge \ tail = \&do_2$
$path(do_0, \texttt{next}, \texttt{NULL})[do_0, do_1, do_2]$
$path(do_1, \texttt{next}, \texttt{NULL})[do_1, do_2]$
$path(do_2, \texttt{next}, \texttt{NULL})[do_2]$

... transformation into doubly linked list ...



New invariant

$path(do_0, \texttt{prev}, \texttt{NULL})[do_0]$
$path(do_1, \texttt{prev}, \texttt{NULL})[do_0, do_1]$
$path(do_2, \texttt{prev}, \texttt{NULL})[do_0, do_1, do_2]$

$\Longrightarrow$ Ordering of nodes did not change.

Methodology

2LS requires its abstract domains to describe program properties using logical formulae. We use an approach based on *points-to* relation and on *access paths*.

$p = \&do_0$

$path(do_0, \texttt{next}, \texttt{null})[do_1]$

Experiments

2LS without and with our extension on 173 tasks from SV-COMP'17 Heap Reachability category.

Shape analysis	Correct	Incorrect	**Score**
Without	76	18	-240
With	82	4	32