19. 11. 2013 in Brno

Predator
A Formal Verification Tool

David Kašpar (xkaspa34@stud.fit.vutbr.cz)

Abstract

Predator is an open-source plugin developed for GCC compiler and used for static analysis of C language programs. [1] It provides automated formal verification for low-level system code that manipulates dynamic data structures, such as Linux linked lists. [2] Because low-level code often uses some tricky programming techniques or hacks, it's not easy to manually verify correctness of the code. One example of this is proving absence of memory safety errors.

This is why Symbolic Memory Graphs (SMGs) were proposed as an abstract domain for shape analysis to cope with such low-level memory manipulation. [2] Some examples include pointer arithmetic, safe usage of invalid pointers, block operations with memory, reinterpretation of the memory contents, address alignment, etc. [1]

The plugin is based on Code Listener API and allows you to easily analyse source codes without need to manually preprocessing them first. [1] The tool has won 3 international competitions so far [1] and is mainly developed by Kamil Dudka, member of VeriFIT group.

References

[1] Predator [online webpage]. Available at:
http://www.fit.vutbr.cz/research/groups/verifit/tools/predator/

[2] K. Dudka, P. Peringer and T. Vojnar. Byte-precise Verification of Low-level List Manipulation. June 21, 2013. [slides] Available at:
http://www.fit.vutbr.cz/research/groups/verifit/tools/predator/predator-sas2013-slides.pdf