

Engineering Theory Tools: In Real Life, not Pretend

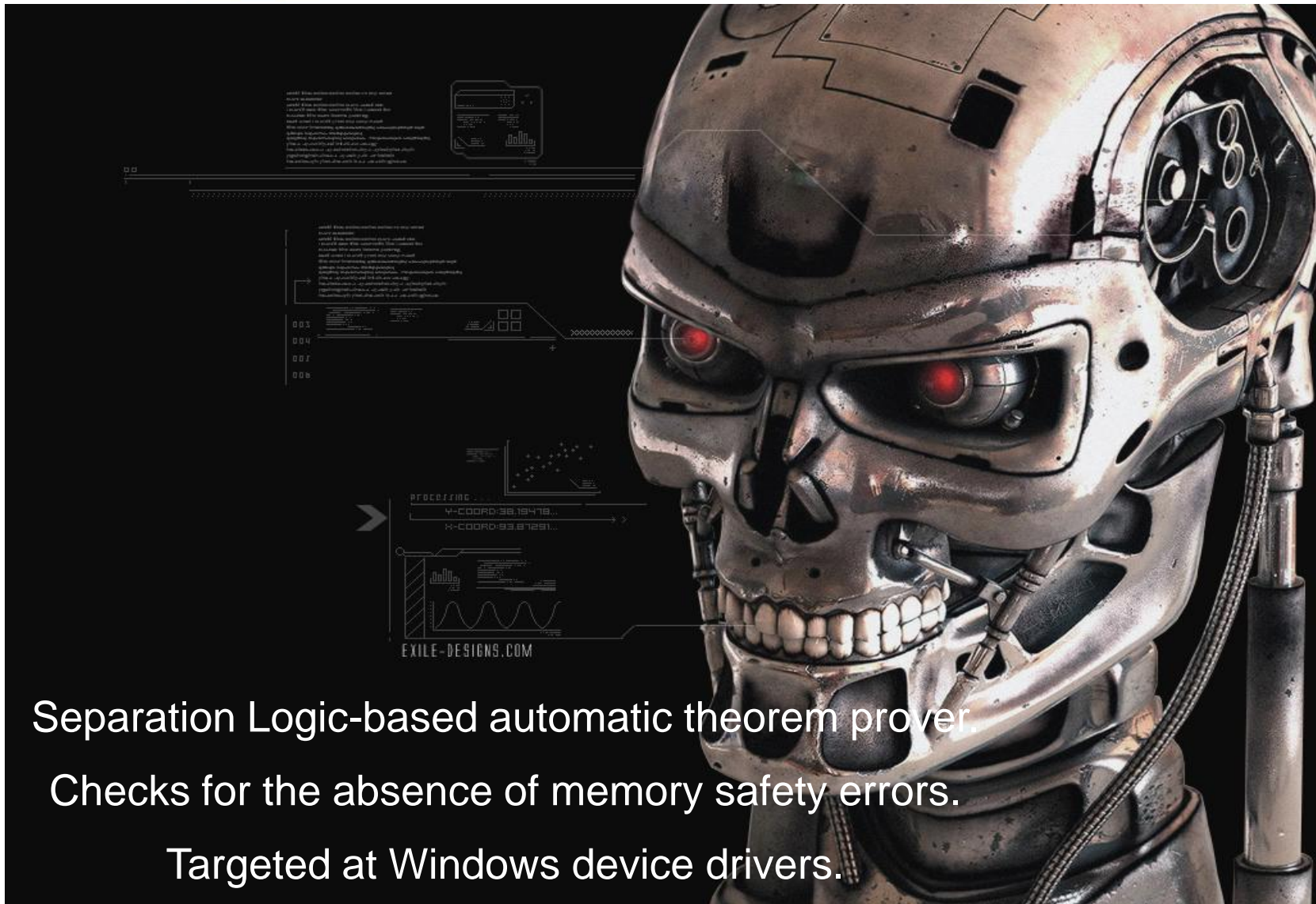
Samin Ishtiaq

(with: Josh Berdine, Byron Cook)

Microsoft Research Cambridge

Theory Engineering Workshop

9 Feb 2010, Cambridge



Separation Logic-based automatic theorem prover.

Checks for the absence of memory safety errors.

Targeted at Windows device drivers.

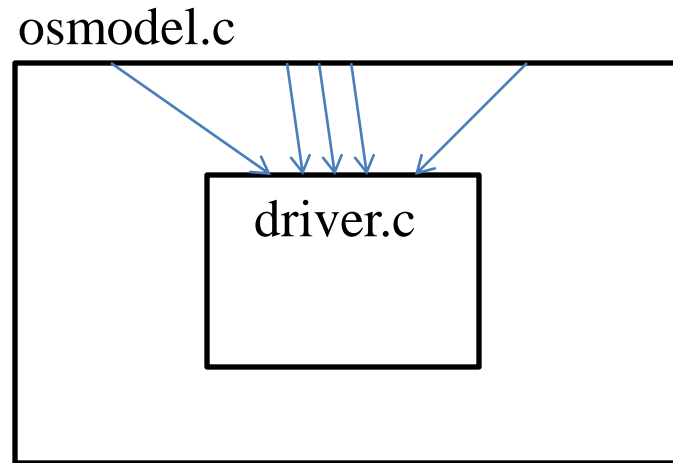
- Abstract interpreter using a domain with separation logic formulae together with linear arithmetic.
- Accepts C code as input.
- Failed proof partially leads to Cex.

Not a demo of a 20-line Pascal-like programs anymore

- Must plug into real parser at the frontend
- Must reason about C memory model (struct layout, p++)

```
//  
// Calculate the address of the base of the structure given its type, and an  
// address of a field within the structure.  
//  
#define CONTAINING_RECORD(address, type, field) ((type *) ( \  
                                                    (PCHAR) (address) - \  
                                                    (ULONG_PTR) (&((type *)0)->field)))
```

- Call-in solvers for sub-domains (Z3, Clousot)



A significant amount of effort is needed to model the environment (CRT, kmdf, Java libraries) in which the DUT code runs.

A tool will only be used if it finds new/significant bugs. Fast.

You only get cred if you can beat the targeted-test bug-hunters.

Without users, your tool is just a POPL adjunct.

samin.ishtiaq@microsoft.com

<http://research.microsoft.com/slayer>