The Skink Static Analysis Tool

Franck Cassez, Anthony Sloane,
Matthew Roberts, Pongsak Suvanpong

SAPLING'17

MACQUARIE
University
SYDNEY·AUSTRALIA

http://science.mq.edu.au/~fcassez/home.html

Program Analysis

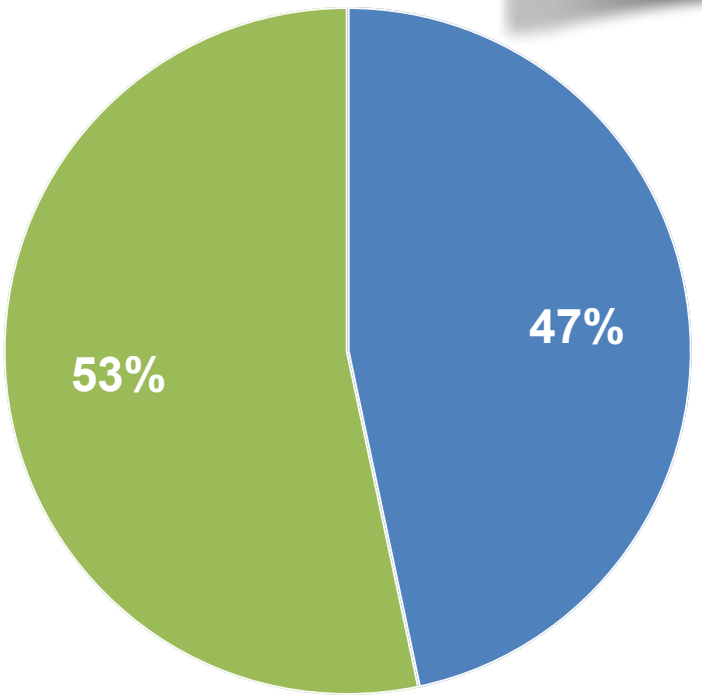
```
1   var i:int;
2   assume i >= 0;
3   while (i >= 0) do
4     i = i - 1;
5     assert (i + 1 >= 0);
6   done;
```

Analysis Result \ Grand Truth	*Incorrect*	*Correct*
Incorrect (warning)	***True Positive***	***False Positive***
Correct (No warning)	***False Negative***	***True Negative***

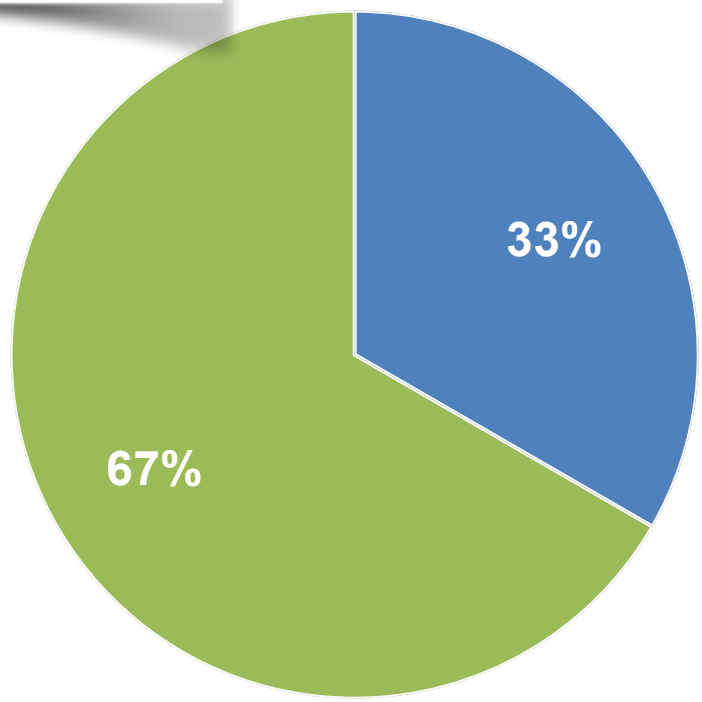
Commercial Static Analysers

30 selected bug reports

Asterisk



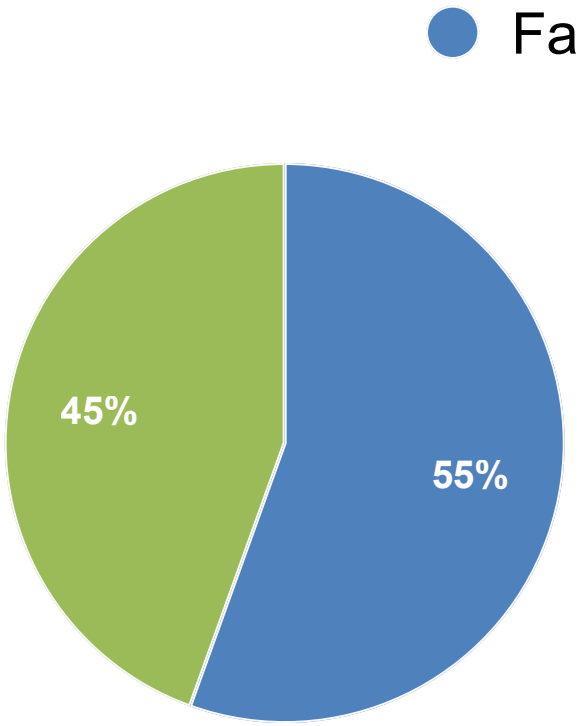
Wireshark



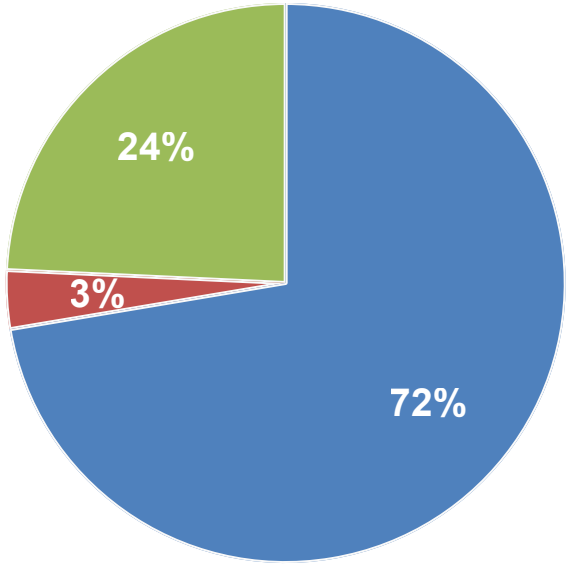
False Positive True Positive

NIST SATE V Workshop, 2014

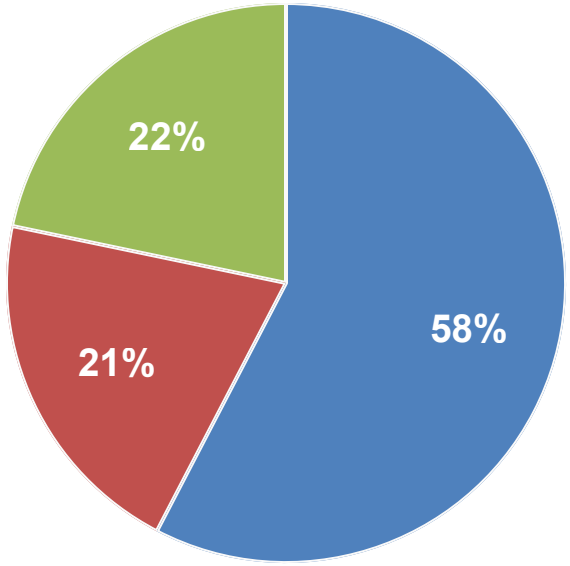
Results Juliet Test Suite (NIST)



NULL-pointer deref
CWE 476



Array-out-of-bounds
CWE 124, 126



Divide-by-zero
CWE 369

Juliet test suite: known programs status

Static Analysis: Challenges

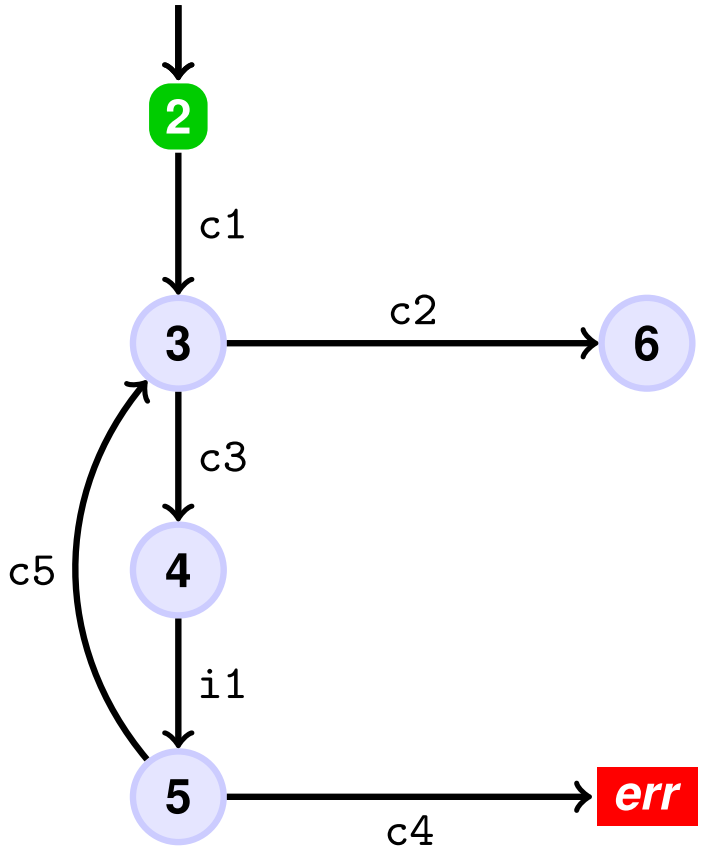
Skink

Refinement of trace abstraction

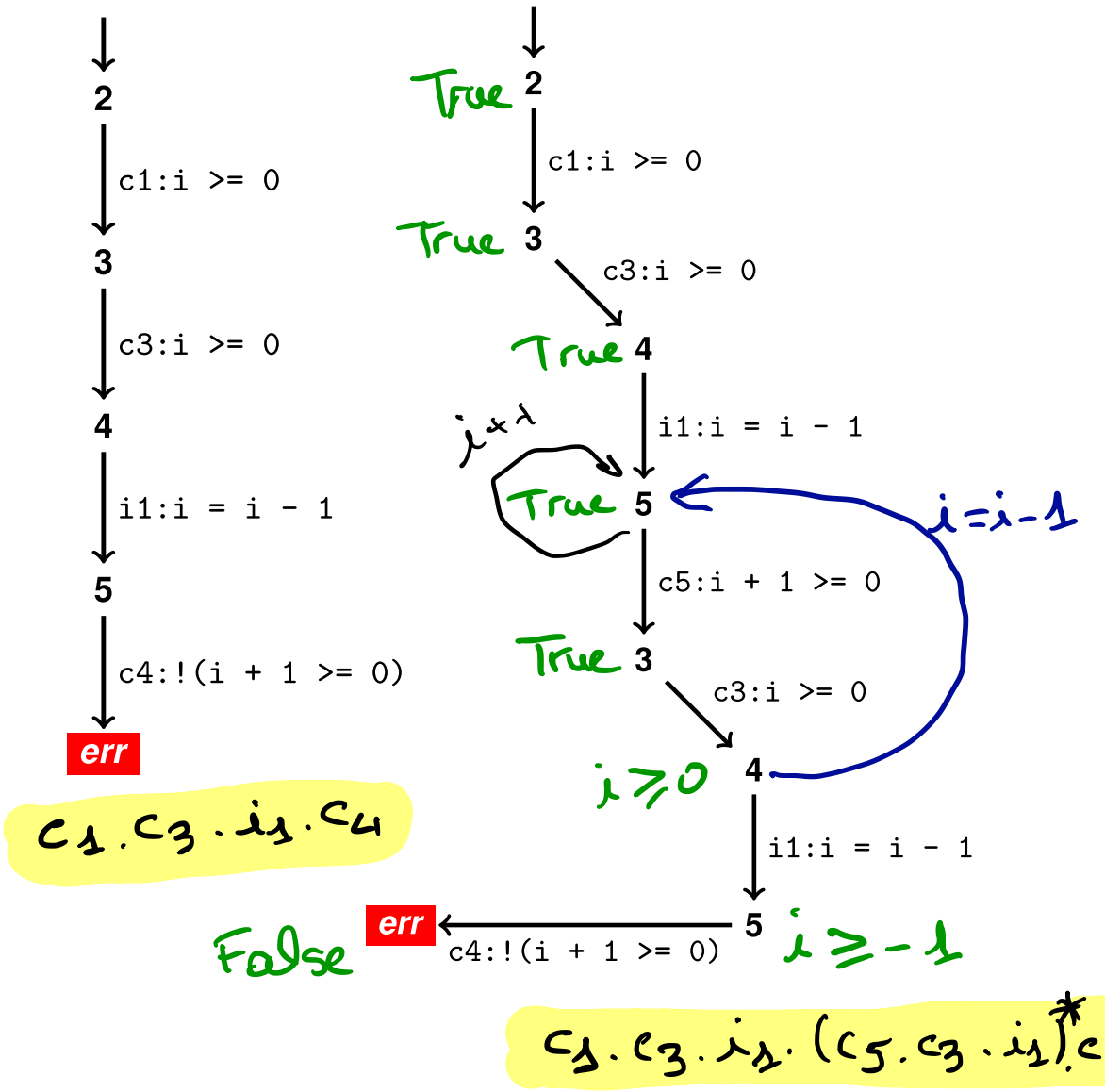
Intra-procedural Analysis

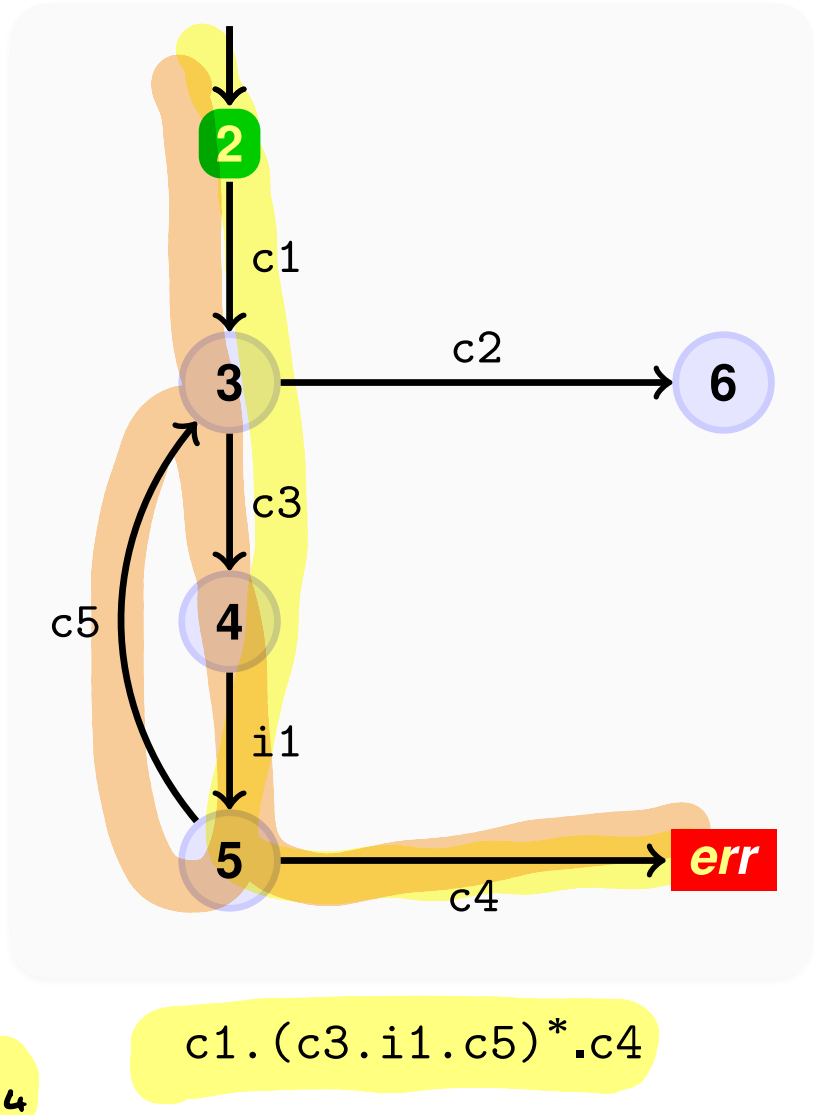
```
1   var i:int;
2   assume i >= 0;
3   while (i >= 0) do
4     i = i - 1;
5     assert (i + 1 >= 0);
6   done;
```

Name	Semantics
c1	i >= 0
c2	!(i >= 0)
c3	i >= 0
c4	!(i + 1 >= 0)
c5	i + 1 >= 0
i1	i = i - 1

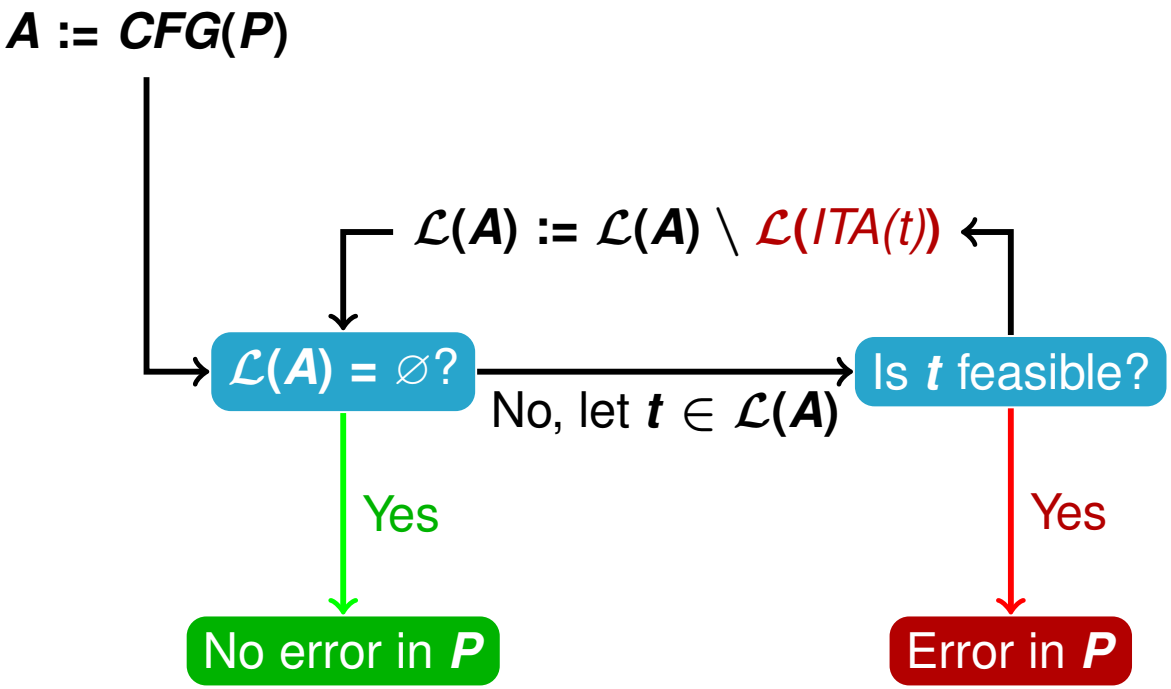


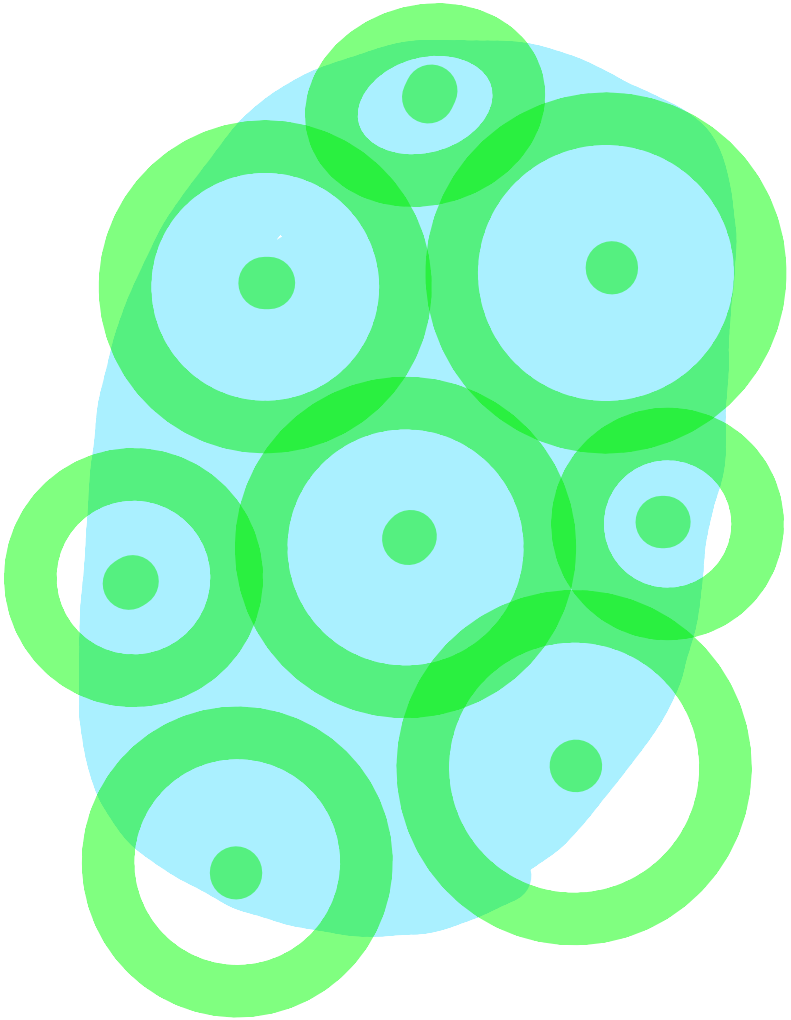
Refinement of Trace Abstraction





Refinement of Trace Abstraction Refinement





Heizmann, M., Hoenicke, J., Podelski, A.,
Refinement of trace abstraction Static Analysis Symposium, 2009.

Inter-procedural

```
1   proc main() : (n) {
2     assume(m >= 1);
3     n = inc(1, m);
4     assert(n >= 0);
5   }
```

```
7   proc inc(p,q) : (r) {
8     assert(p >= 0);
9     if (p >= 1)
10      r = q + 1;
11    else
12      r = q;
13    endif;
14  }
```

Build summaries – no inlining

Summary-Based Inter-Procedural Analysis via Modular Trace Refinement. Cassez, F.; Müller, C.; and Burnett, K. In 34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, December 15-17, 2014, New Delhi, India, pages 545--556, 2014.

Analysis of program code. Cassez, F.; and Müller, C. September~12 2017. US Patent 9,760,469

Multiple threads

```
2   // thread T1
3   thread T1
4   x = 0;
5   lock(m);
6   if (x == y) {
7     unlock(m);
8     d = 3;
9   } else {
10    unlock(m);
11  }
12  /* end */
```

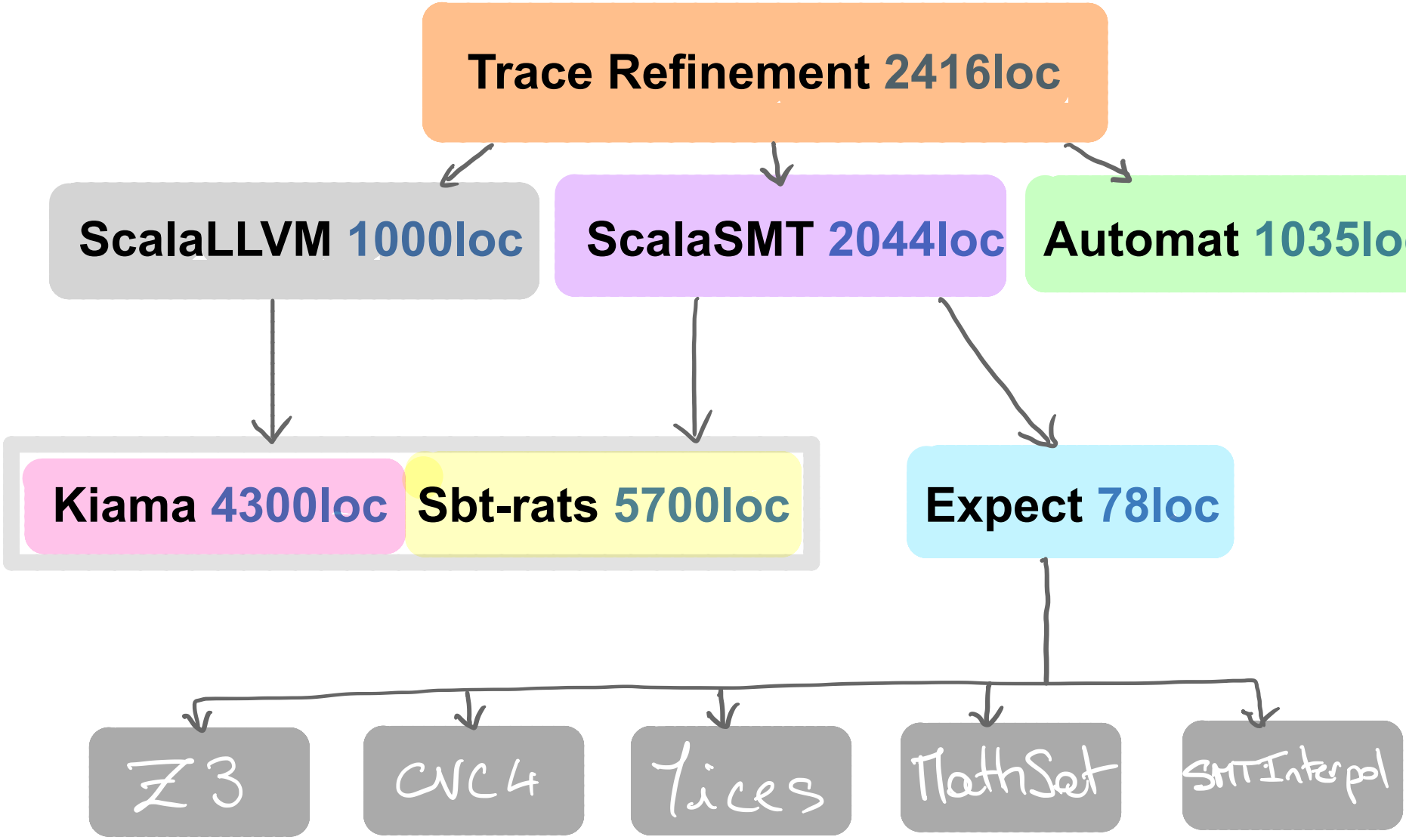
```
14  // Thread T2
15  thread T2
16  y = 1;
17  lock(m);
18  if (x <= y) {
19    unlock(m);
20    d = 2;
21  } else {
22    unlock(m);
23  }
```

Trace refinement + partial order reduction

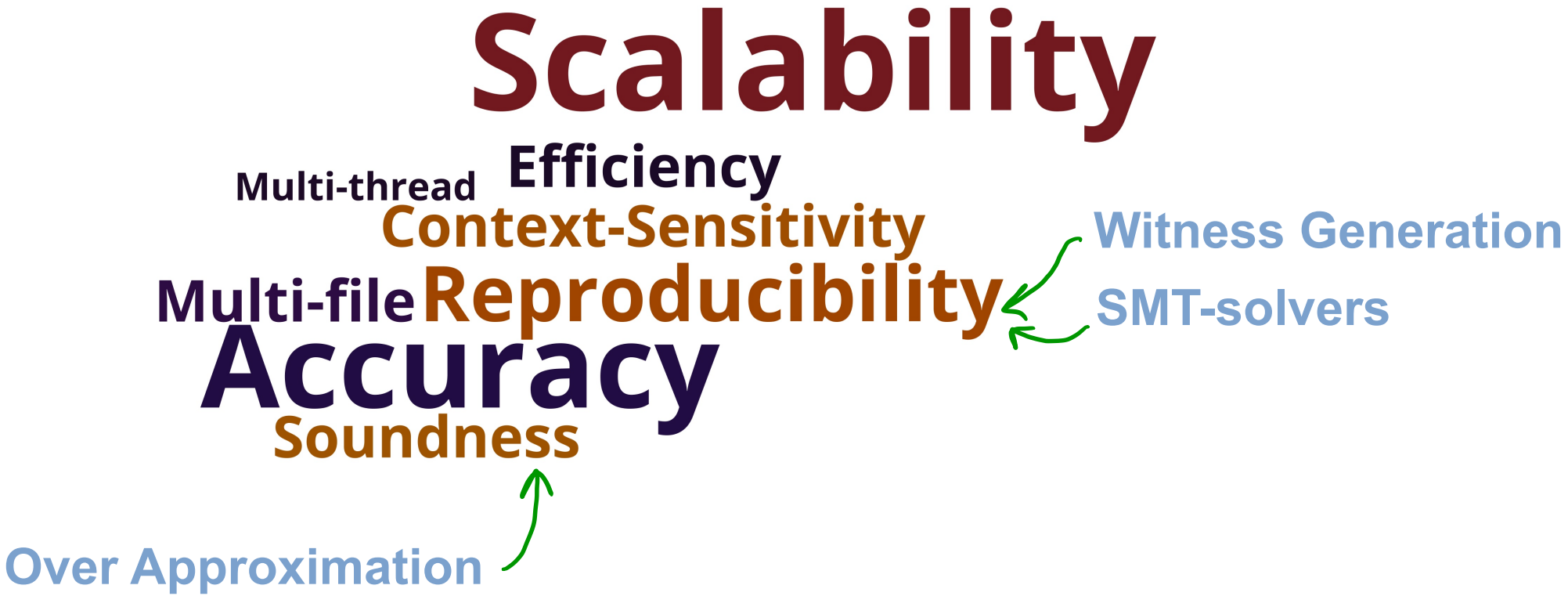
Verification of Concurrent Programs Using Trace Abstraction Refinement. Cassez, F.; and Ziegler, F. In Logic for Programming, Artificial Intelligence, and Reasoning - 20th International Conference, LPAR-20 2015, Suva, Fiji, November 24-28, 2015, Proceedings, volume 9450, of Lecture Notes in Computer Science, pages 233--248, 2015. Springer

Static Analysis: Challenges

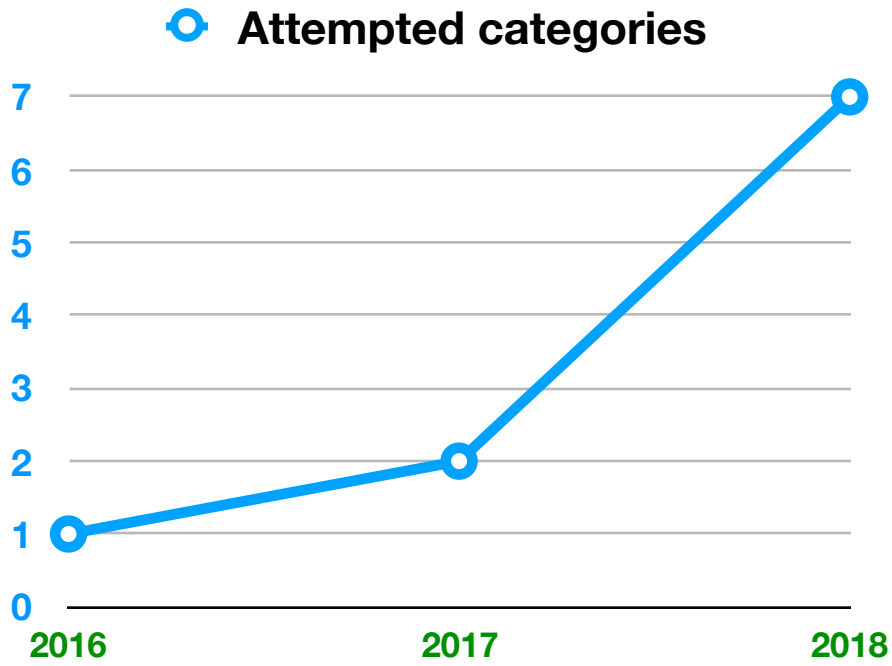
Skink: Tool Architecture

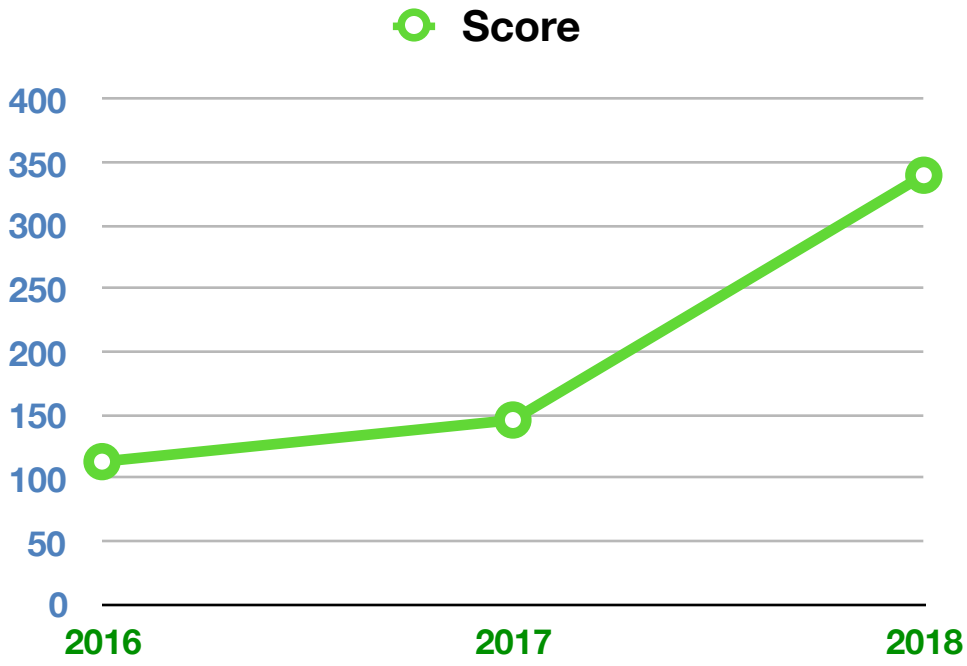


Static Analysis: Challenges



Skink@SV-COMP 16, 17, 18





Skink: Static Analysis of Programs in LLVM Intermediate Representation (Competition contribution). Cassez, F.; Sloane, A.; Roberts, M.; Pigram, M.; Suvanpong, P.; and de Aledo Marugán, P. G. In Tools and Algorithms for the Construction and Analysis of Systems - 23rd International Conference, TACAS 2017.

Current/Ongoing Work




Parallel Analysis

Invariants Synthesis

Machine Learning

Security Analysis

Multi-thread Analysis

Termination Analysis

Test harness

Proof certificates

Code coverage

References

Skink: Static Analysis of Programs in LLVM Intermediate Representation (Competition contribution). Cassez, F.; Sloane, A.; Roberts, M.; Pigram, M.; Suvanpong, P.; and de Aledo Marugán, P. G. In Tools and Algorithms for the Construction and Analysis of Systems - 23rd International Conference, TACAS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017. Proceedings, of LNCS, pages 380--384, 2017. Springer

ScalaSMT: Satisfiability Modulo Theory in Scala. Cassez, F.; and Sloane, A. In SCALA'17, October 23--27, 2017, Vancouver, BC, Canada. Proceedings., 2017.

Analysis of program code. Cassez, F.; and Müller, C. September~12 2017. US Patent 9,760,469

The Sbt-rats Parser Generator Plugin for Scala. Sloane, A.; Cassez, F.; and Buckley, S. In Proceedings of the 2016 7th ACM SIGPLAN Symposium on Scala, of SCALA 2016, pages 110--113, New York, NY, USA, 2016. ACM

Verification of Concurrent Programs Using Trace Abstraction Refinement. Cassez, F.; and Ziegler, F. In Davis, M.; Fehnker, A.; McIver, A.; and Voronkov, A., editor(s), Logic for Programming, Artificial Intelligence, and Reasoning - 20th International Conference, LPAR-20 2015, Suva, Fiji, November 24-28, 2015, Proceedings, volume 9450, of Lecture Notes in Computer Science, pages 233--248, 2015. Springer LPAR Best paper award

Summary-Based Inter-Procedural Analysis via Modular Trace Refinement. Cassez, F.; Müller, C.; and Burnett, K. In 34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, December 15-17, 2014, New Delhi, India, pages 545--556, 2014.

Sloane, A. M. Lightweight language processing in Kiama . In Generative and Transformational Techniques in Software Engineering III. Volume 6491 of Lecture Notes in Computer Science, Springer, 2011.