Iekkë

A SAT-Based Bounded-Round Verifier for Multi-Threaded Programs

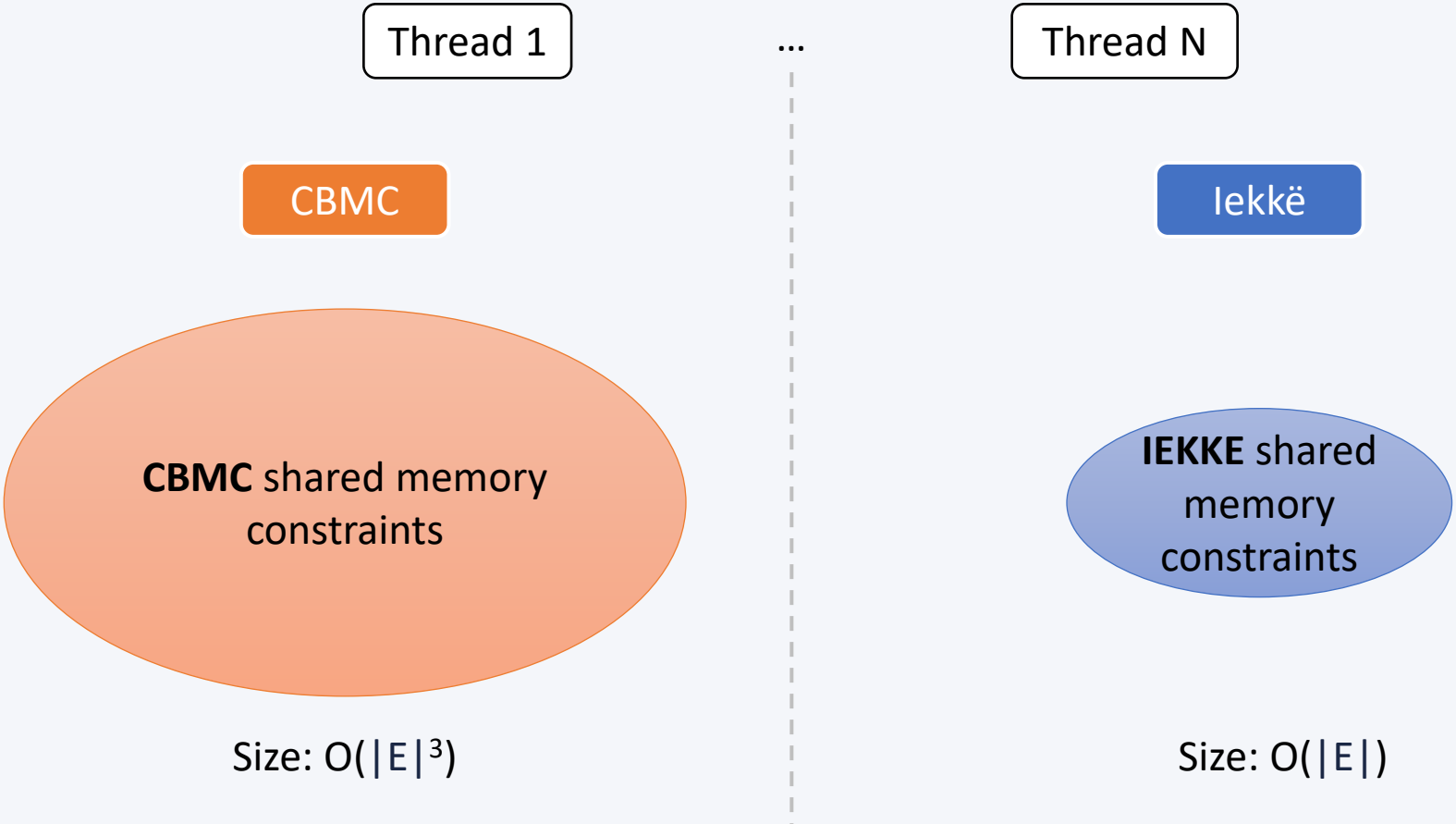
Competition Contribution

Paolo Di Biase[1,5] · Bernd Fischer[2] · Salvatore La Torre[3] · Peter Schrammel[4,6] · Gennaro Parlato[5]

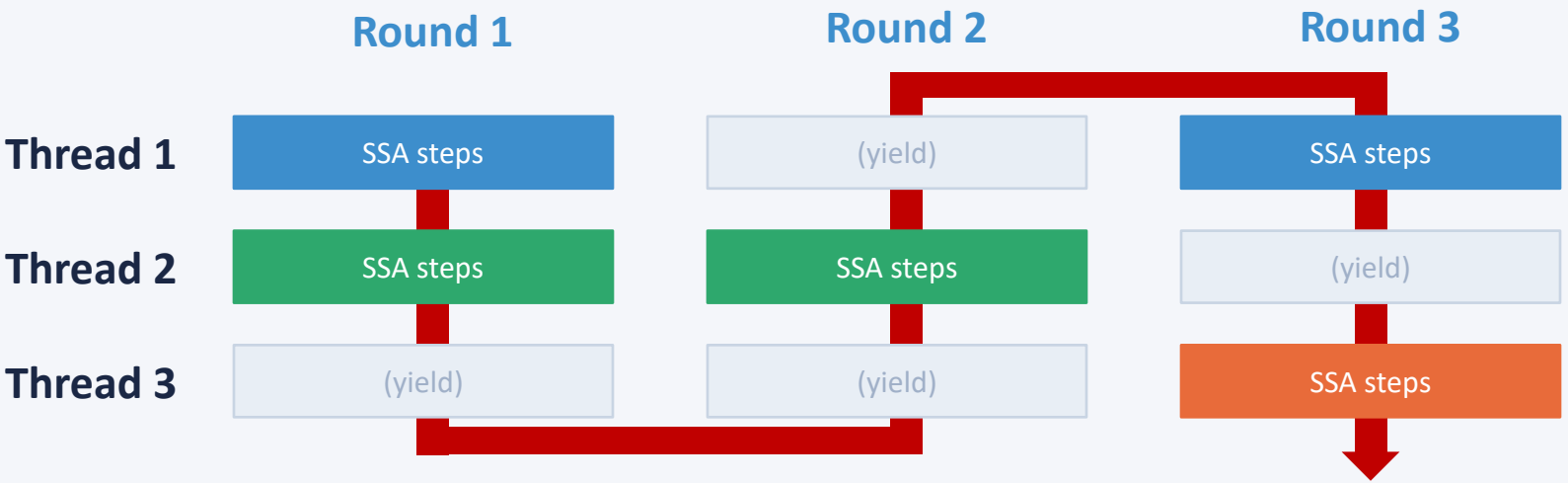
[1] GSSI [2]Stellenbosch Univ. [3]Univ. of Salerno [4]Univ. of Sussex [5]Univ. of Molise [6]Diffblue Ltd

TACAS 2026 · SV-COMP

Motivation: Verifying Shared-Memory Concurrency

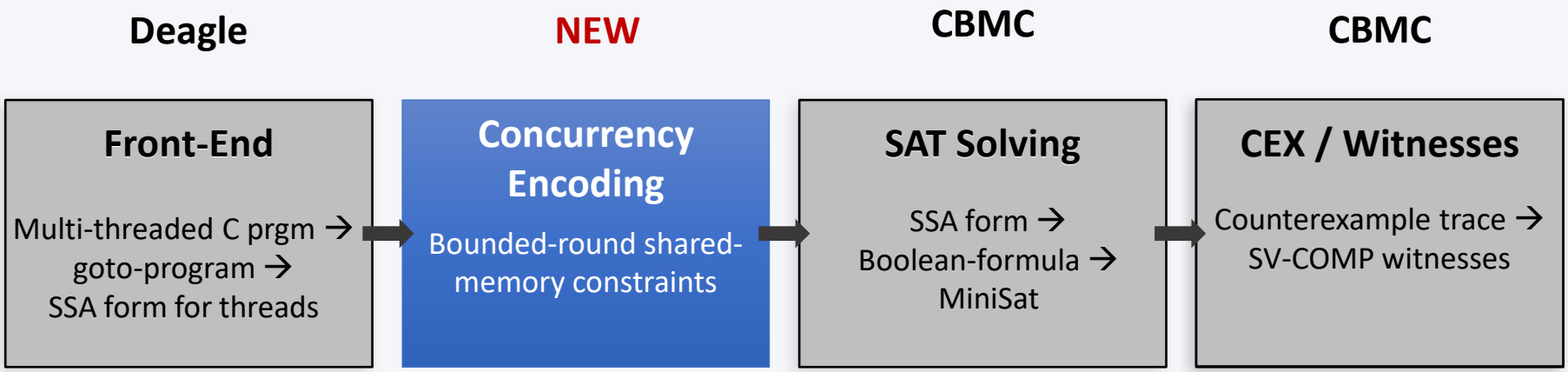


Key Idea: Bounded-Round Semantics



- Iekkë under-approximates the interleaving space by restricting executions to at most k rounds *[Musuvathi, Qadeer: PLDI 2007]*
- Inspired by **Lazy-CSeq**, but encoded directly at the formula level (no code replication)
- Constraints size: O(|E| · k) — linear in events AND in round bound k

Software Architecture



SV-COMP 2026 Results – Concurrency Category

3428

points

308 Confirmed Bugs

216 Unconfirmed Bugs

44 False alarms
(implementation bugs)

22 Missed bugs
(round bound too low)

Notable weakness: no-data-race sub-category (646 / 1030 benchmarks solved, 47 incorrect results)

Strengths, Weaknesses & Future Directions

✓ Strengths

- Compact propositional encoding
- Linear constraint size
- Effective for bug-finding

✗ Weaknesses

- Round bound must be set manually (no automatic increase)

Future Work

Iterative deepening

Automatically increase k until property is proved or budget exhausted

Weak memory models

Extend to TSO / PSO consistency models

Fix data-race handling

Address remaining implementation gaps from Deagle heritage

Thank you!