

Abstraction-Based Interprocedural Verification

Márk Somorjai

Advisors: Mihály Dobos-Kovács, Levente Bajczi, András Vörös, PhD.



**Critical Systems
Research Group**

Abstraction-Based Software Verification

Program

locations
statements

CFA

(Control Flow Automaton)

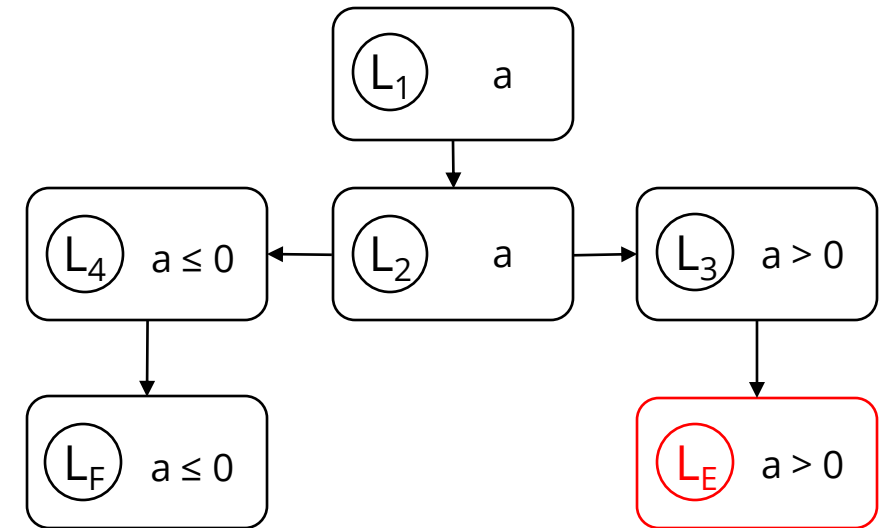
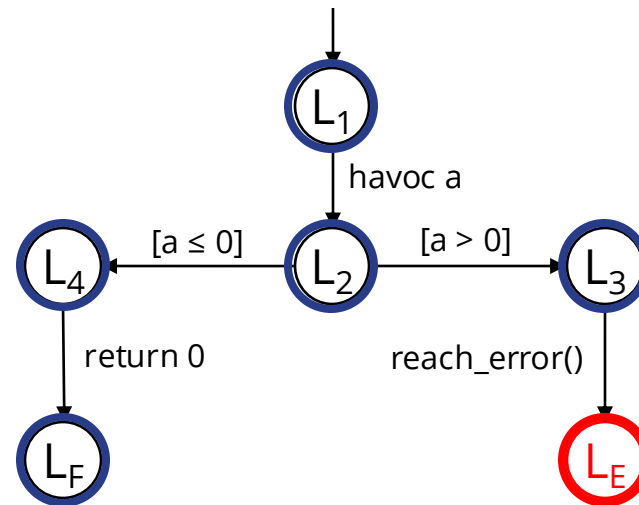
nodes
edges

ARG

(Abstract Reachability Graph)

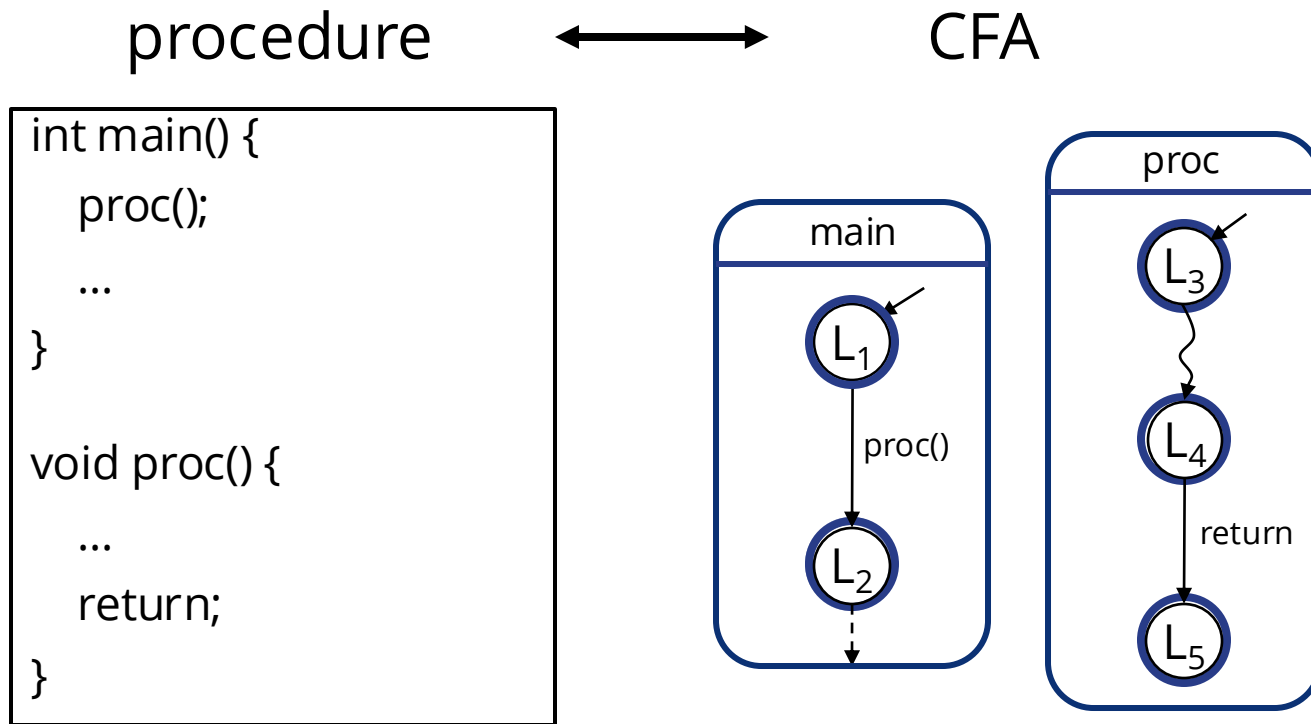
abstract states
transitions

```
int main() {  
  int a = input();  
  if (a > 0) {  
    reach_error();  
  }  
  return 0;  
}
```

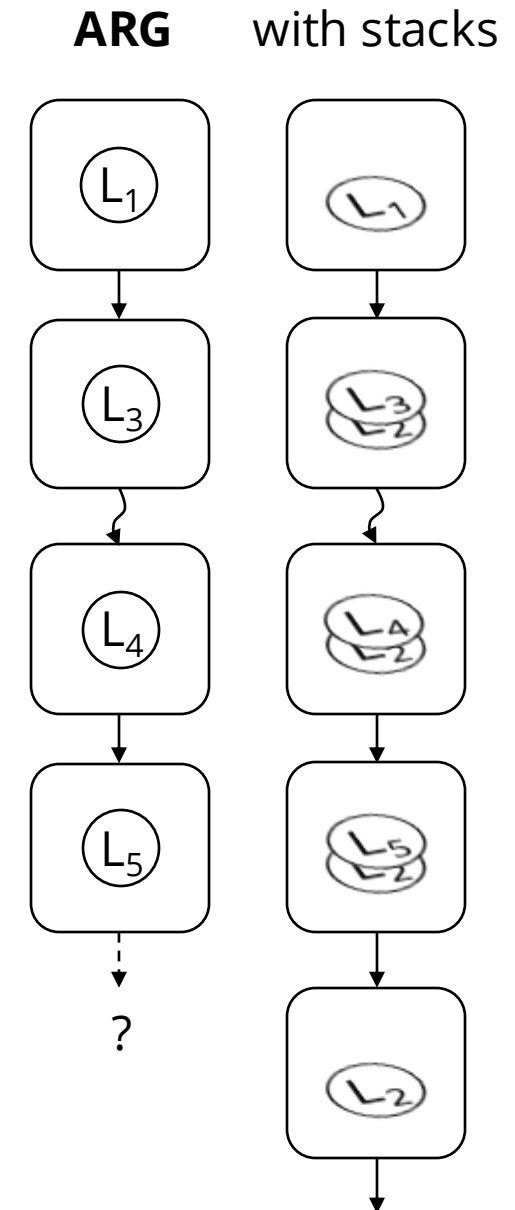


Interprocedural Verification

- Verification of programs with procedures

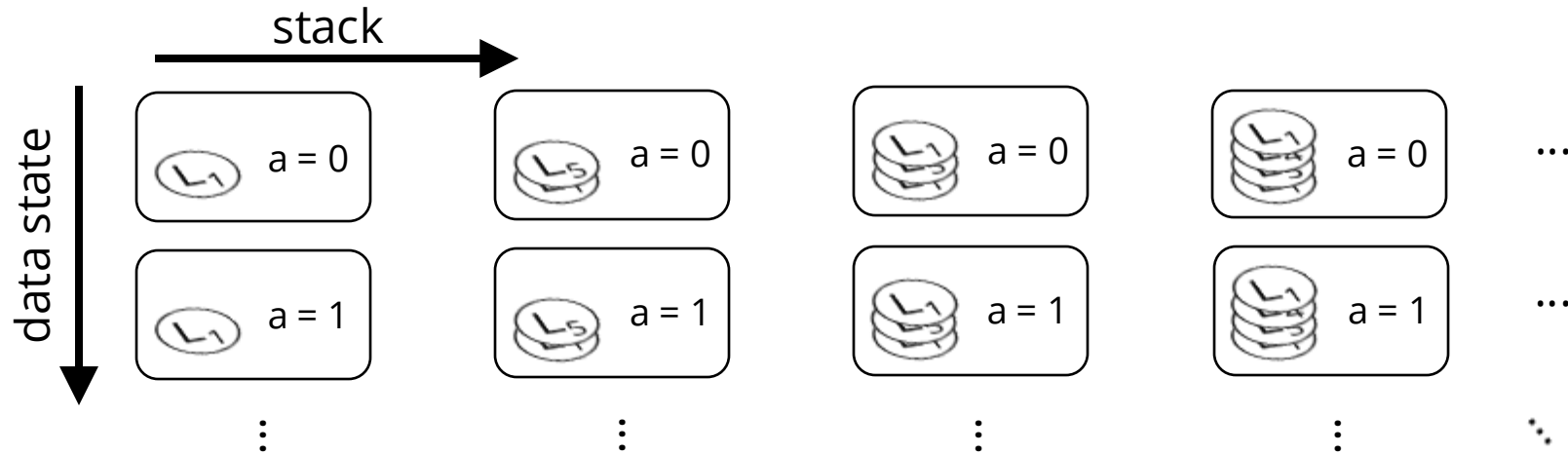


- Location stack is part of the program's state



Abstraction-Based Interprocedural Verification

- Location stacks leads to further state-space explosion



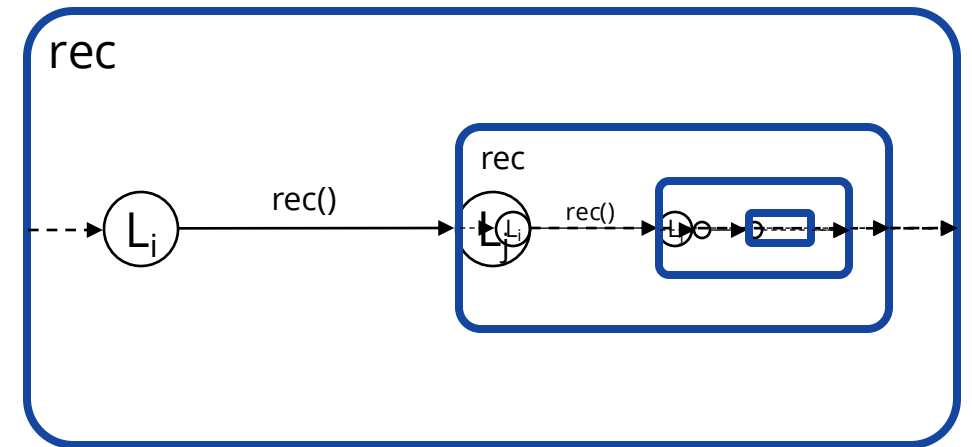
My work:

Reduce the abstract state-space of procedural programs by **applying abstraction to location stacks** in order to **improve the efficiency of verification**.

Procedure Inlining

- Procedure call is replaced with the CFA of the procedure
 - input parameter assignments before the call
 - output parameter assignments after the call

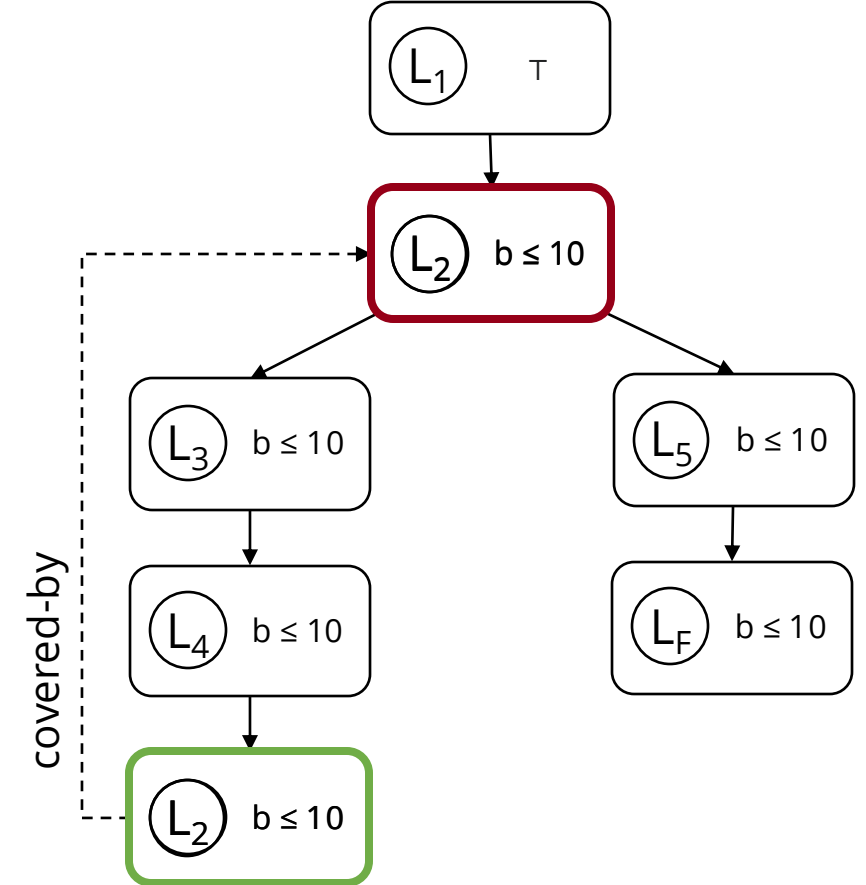
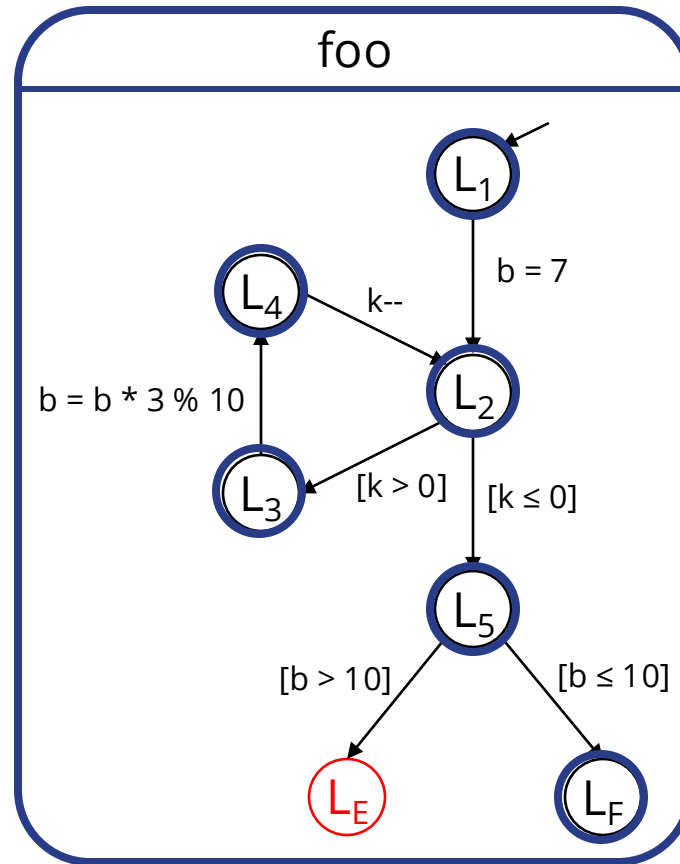
- Problems:
 - larger CFA
 - cannot handle recursive procedures



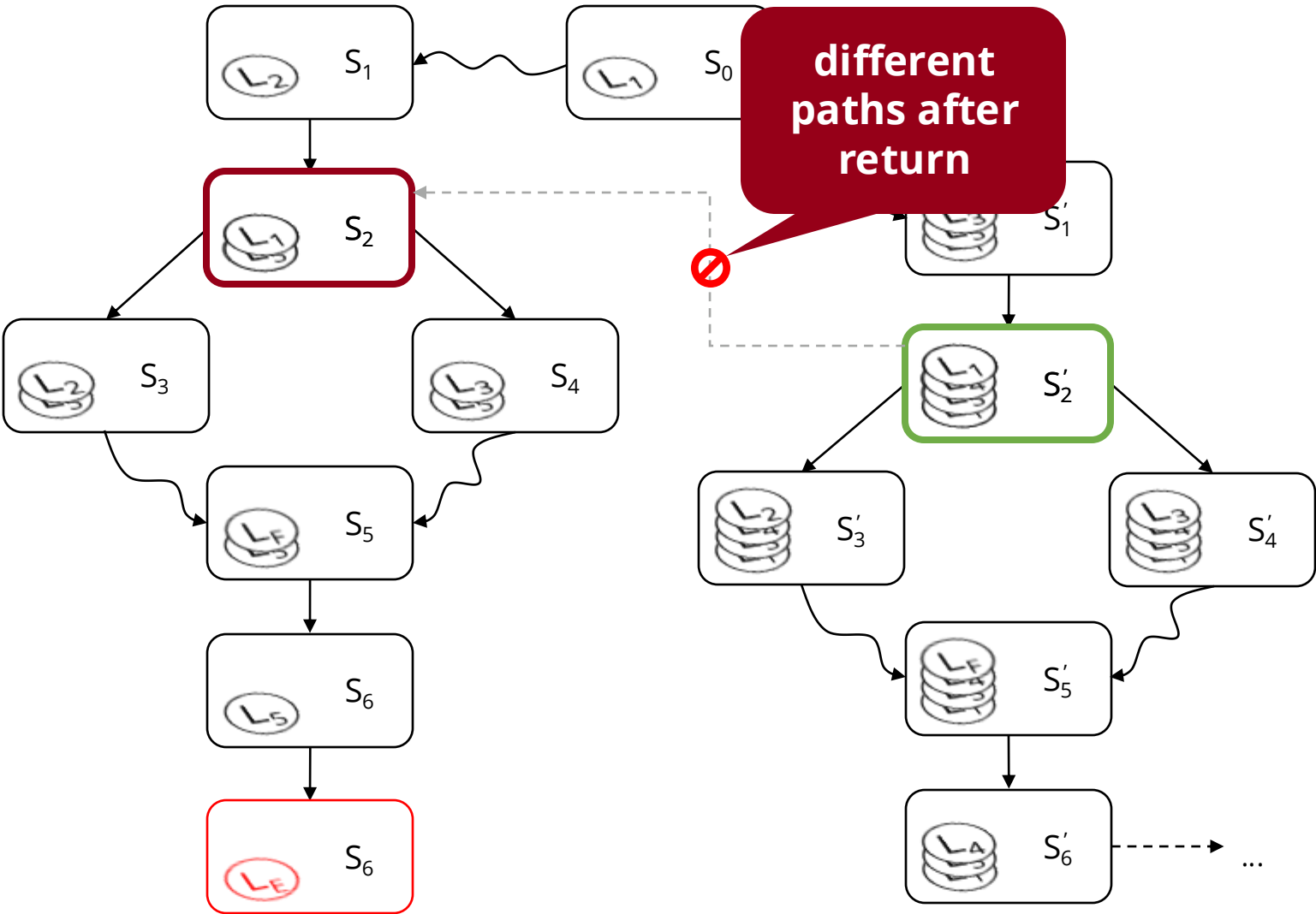
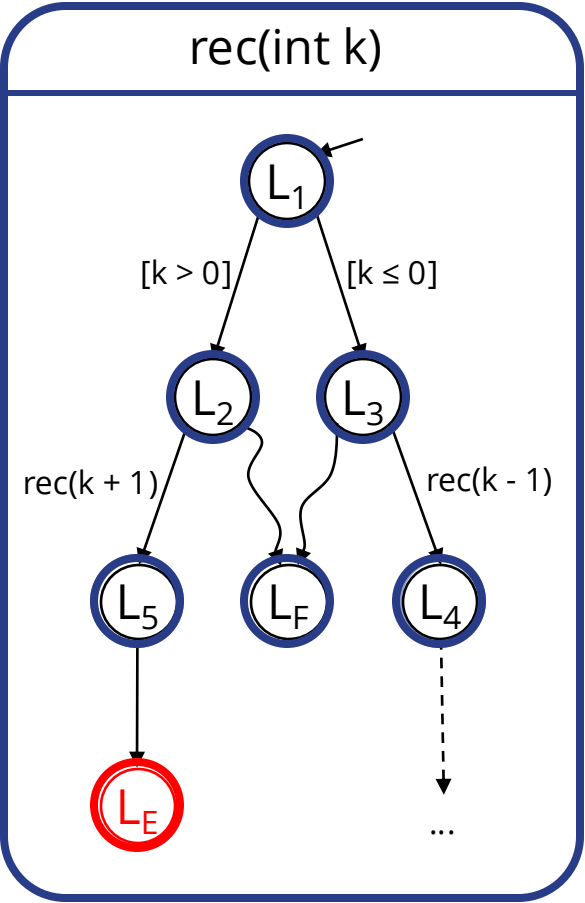
- Verifying recursive programs **requires** a more nuanced approach

Covering Abstract States

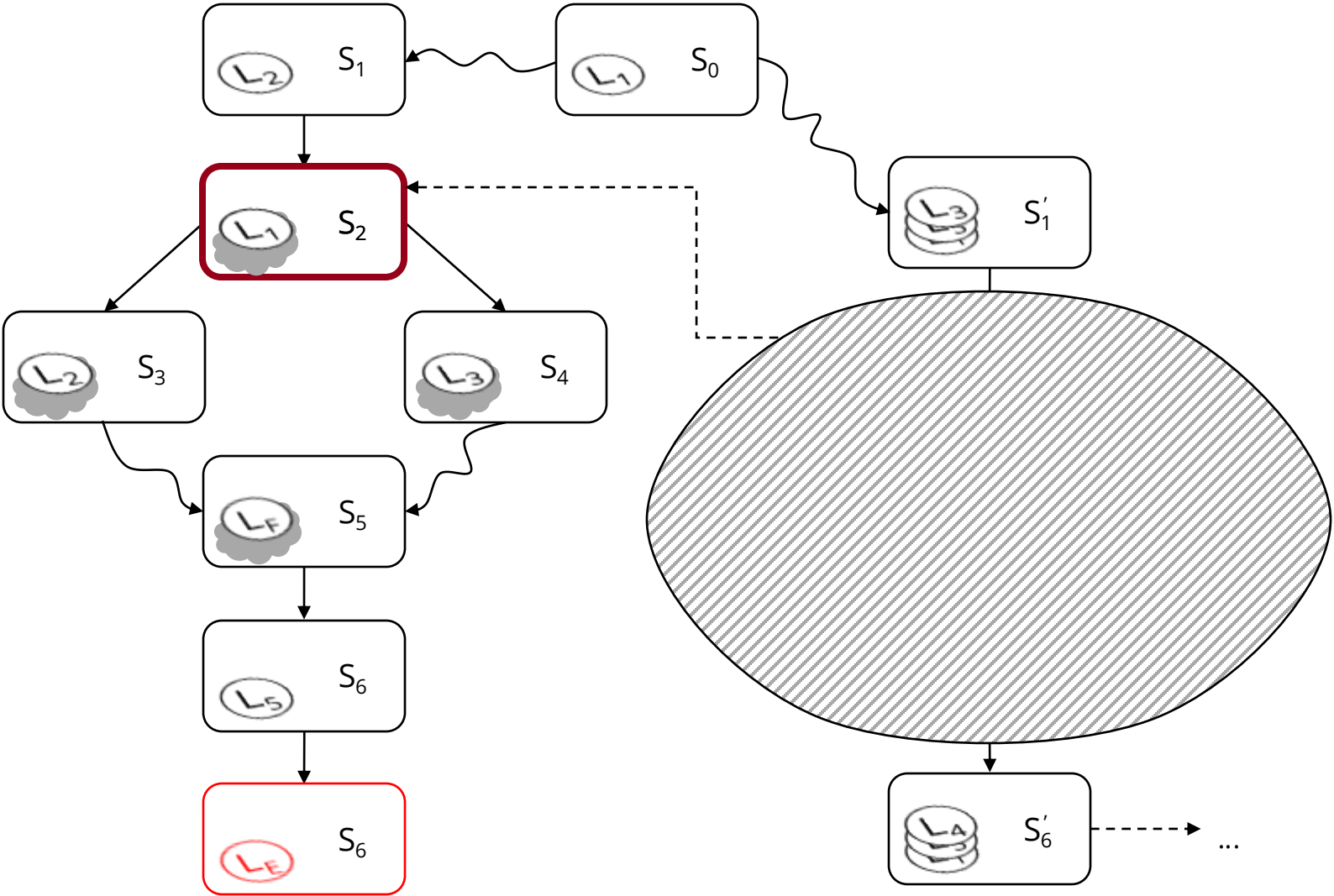
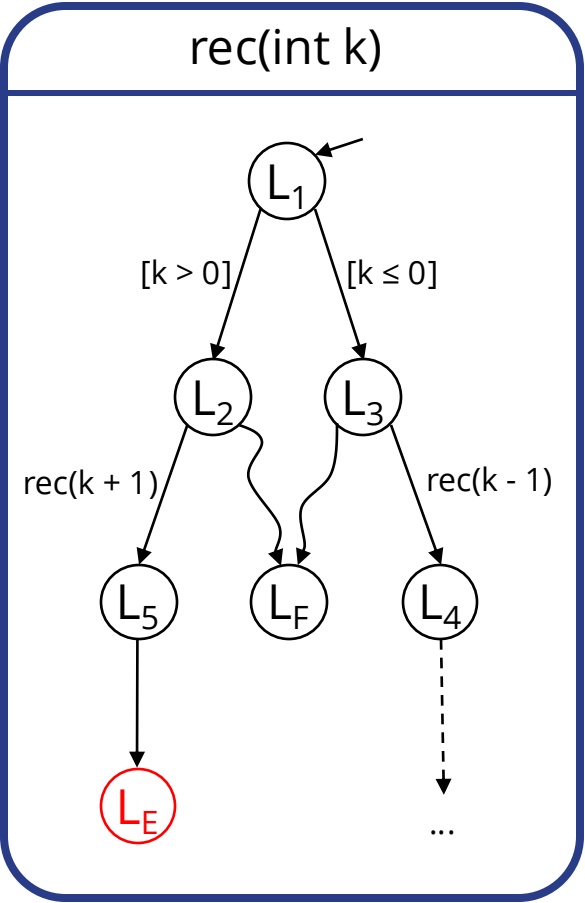
```
void foo(int k) {  
  int b = 7;  
  while (k > 0) {  
    b = b * 3 % 10;  
    k--;  
  }  
  assert(b ≤ 10);  
}
```



Abstract States with Stacks

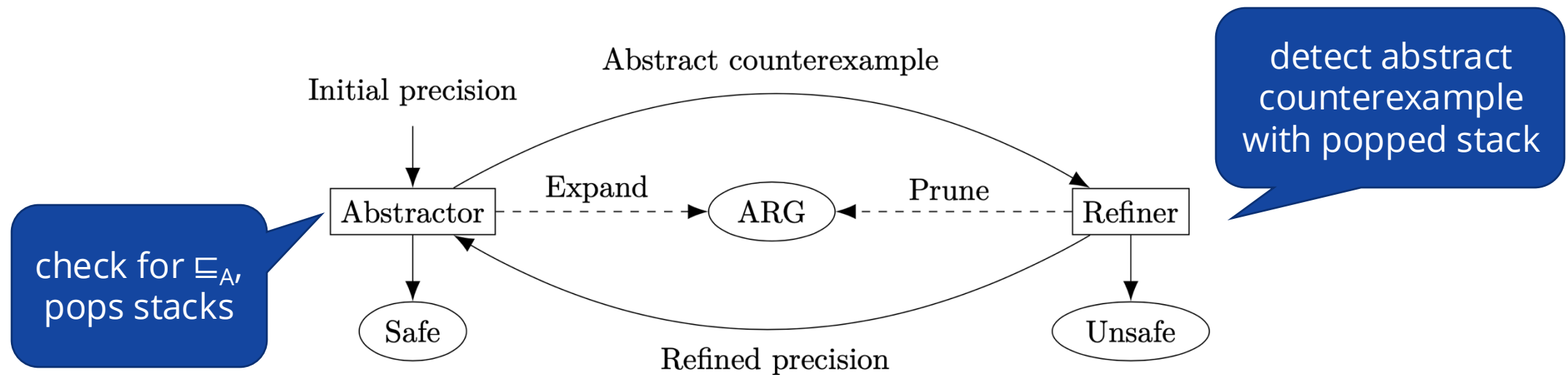


Applying Abstraction to Stacks



Effect on Verification Algorithm

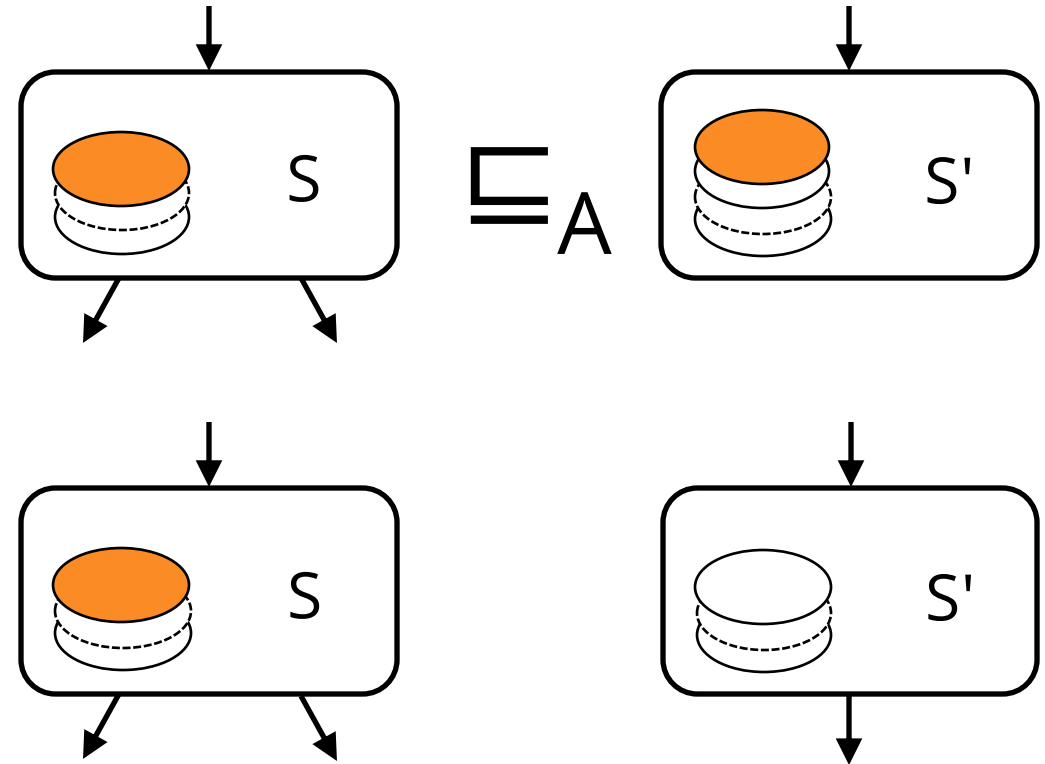
- Advantages:
 - reduced state space $\longrightarrow$ more efficient verification
 - can solve some infinitely recursive tasks
- Counterexample-Guided Abstraction Refinement (CEGAR)



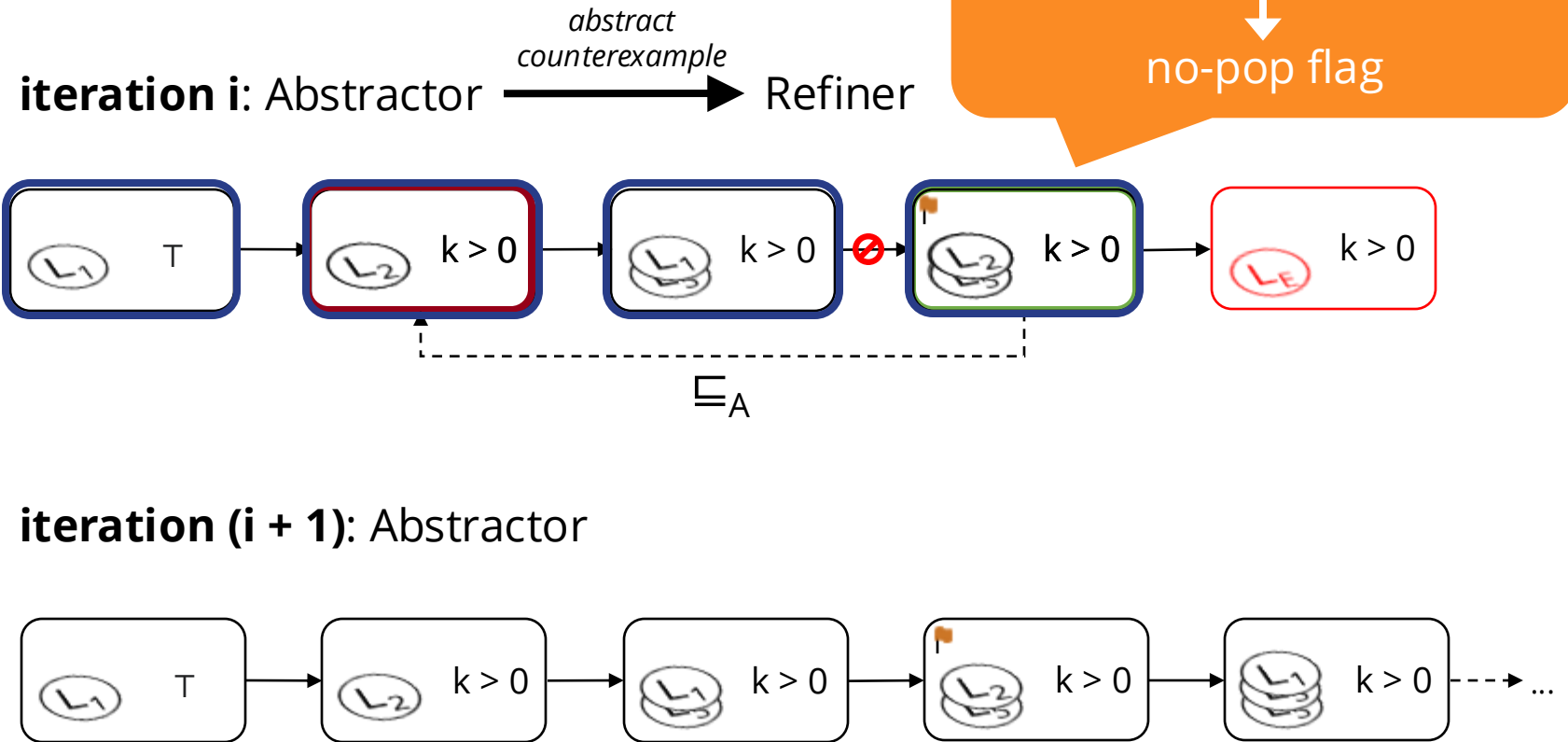
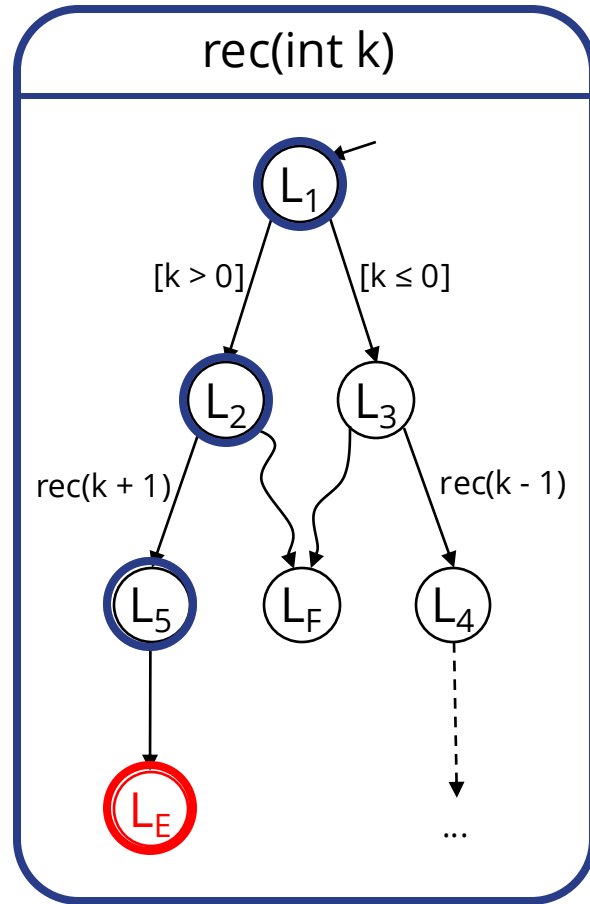
Stack Abstraction

$$(q, S) \sqsubseteq_A (q', S') \iff \text{top}(q) = \text{top}(q') \wedge S \sqsubseteq S'$$

1. Check $(q, S) \sqsubseteq_A (q', S')$:
 - top locations are the same
 - S overapproximates S'
2. If $(q, S) \sqsubseteq_A (q', S')$ found:
 - pop top location of q'
 - continue exploration



Spurious Counterexamples

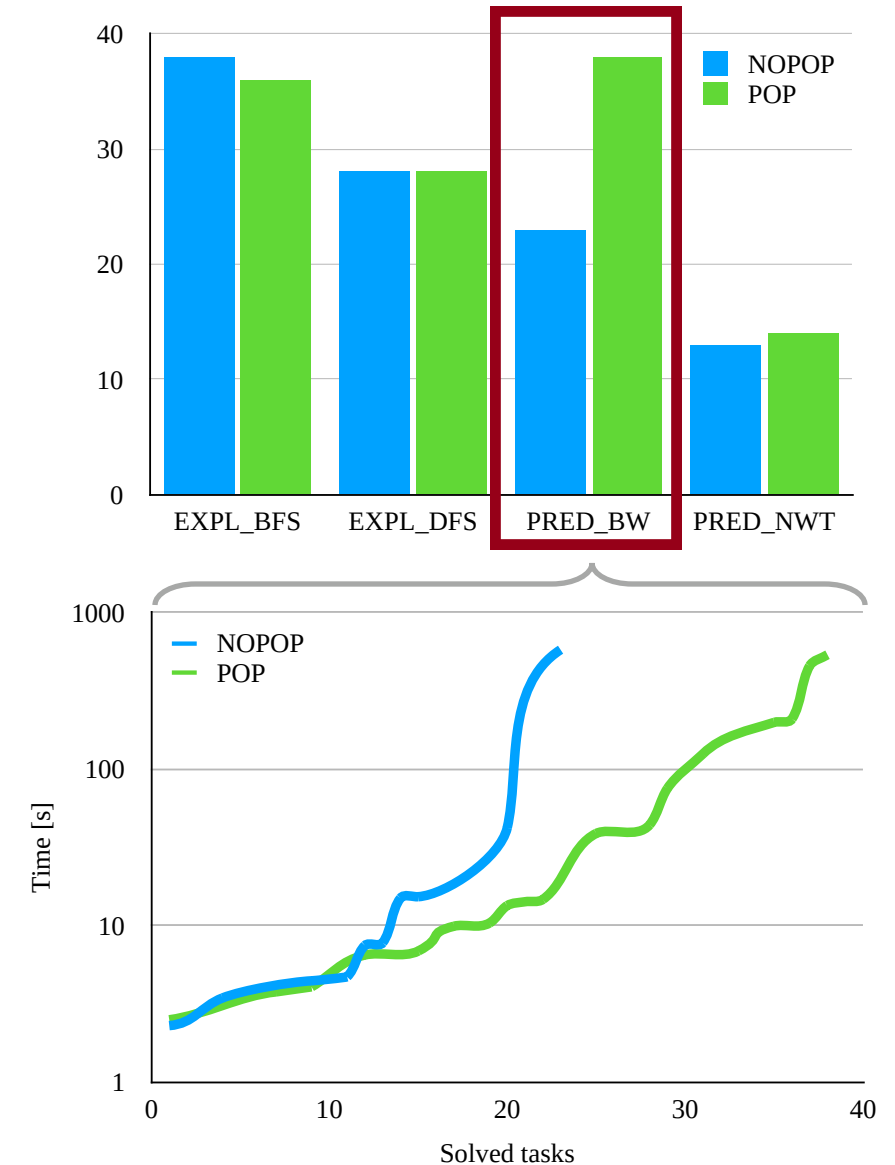


Evaluation

- Implementation in Theta¹, our model-checking framework
- 1219 C programs from the SV-COMP benchmark repository²
 - tasks without procedures to measure overhead
 - recursive tasks to measure performance gains
- Many configurations tested, using BenchExec by SoSyLab
 - abstraction domains
 - refinement strategies
- No wrong answers, # solved tasks (15-minute timeout) is shown

Benchmark Results

- Recursive: Improved performance with predicate abstraction.
 - 65% more programs verified
 - 20% of solved tasks could not be verified by any other configuration
 - solves difficult tasks quicker
- Non-procedural: slight performance overhead
 - but: recursion can be detected before verification



Summary

- Stack abstraction: novel interprocedural verification technique
 - reduced abstract state-space
 - integration into existing abstraction-based algorithm (CEGAR)
- Practical results
 - implementation in the open-source Theta
 - benchmark campaign: improved performance on recursive programs
- Future work:
 - publish work to international community
 - combination with other, orthogonal techniques (e.g. summaries)