Decomposing Software Verification using Distributed Summary Synthesis

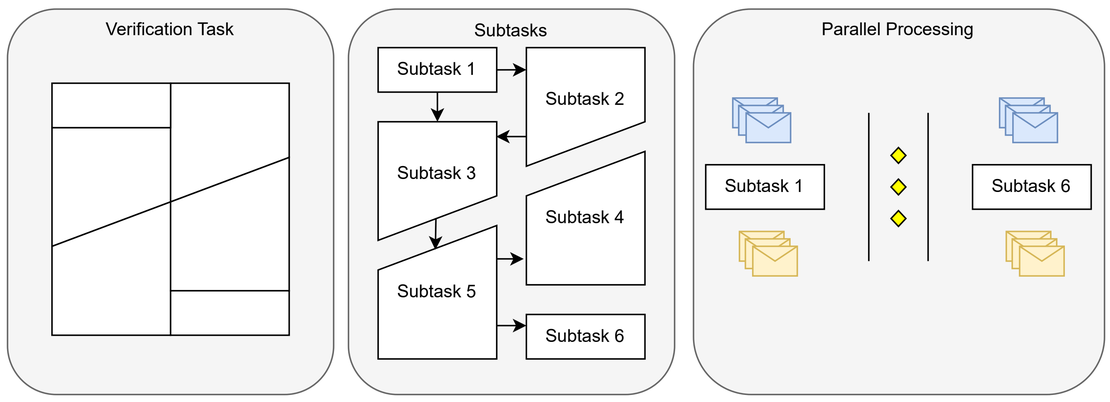
Dirk Beyer, Matthias Kettl, and Thomas Lemberger

Motivation

- Formal verification is a time-consuming task.
- This hinders the integration of formal verification to CIs.
- Therefore, we propose distributed summary synthesis (DSS) [3], a domain-independent framework for distributing verification algorithms (cf. also [2, 4, 1]).

https://www.sosy-lab.org/research/distributed-summary-synthesis/

Approach



A task is divided into multiple subtasks, each with their own pre- and violation conditions. They can now be verified in parallel.

Program P

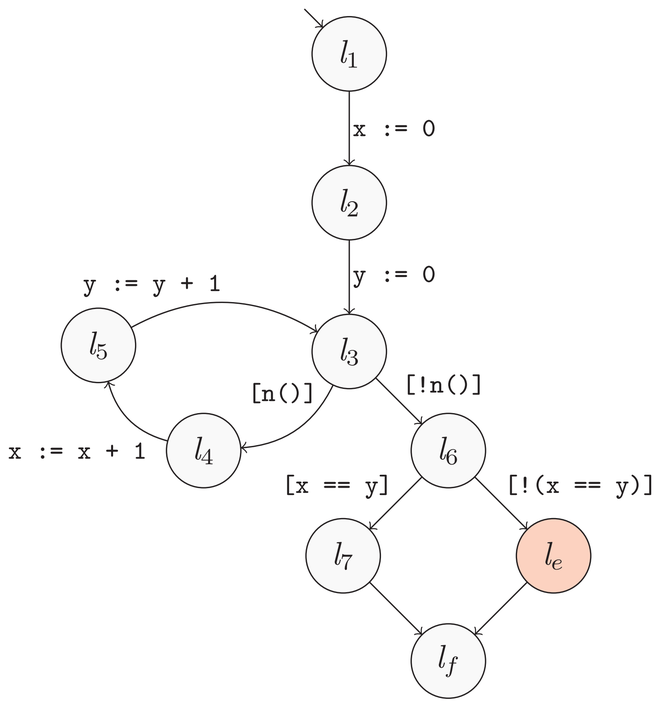
Safe program:

- Values x and y are initialized to 0.
- The program increments x and y nondeterministically often by one.
- The assert in line 8 always holds.

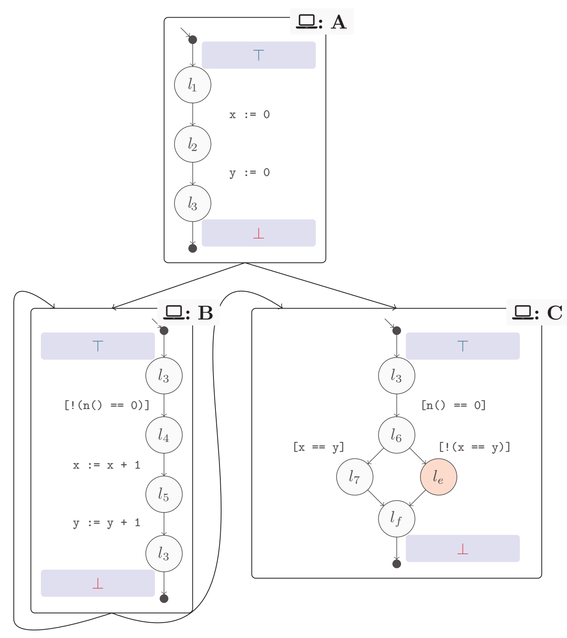
```
1 int main() {
2   int x = 0;
3   int y = 0;
4   while (n()) {
5     x++;
6     y++;
7   }
8   assert(x == y);
9 }
```

We represent programs as control-flow automata (CFA).

CFA P

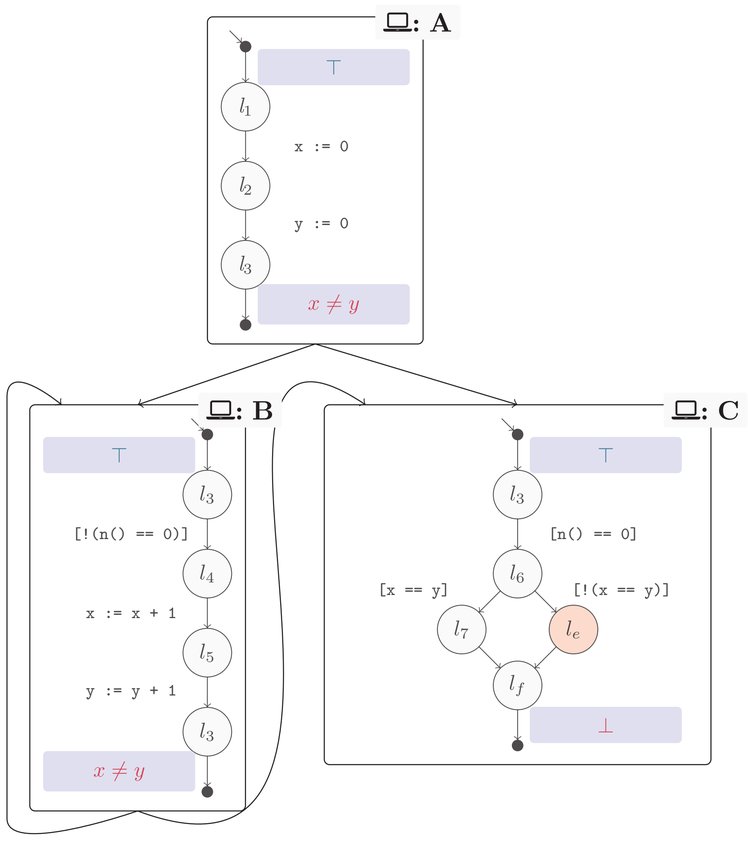


Decomposition



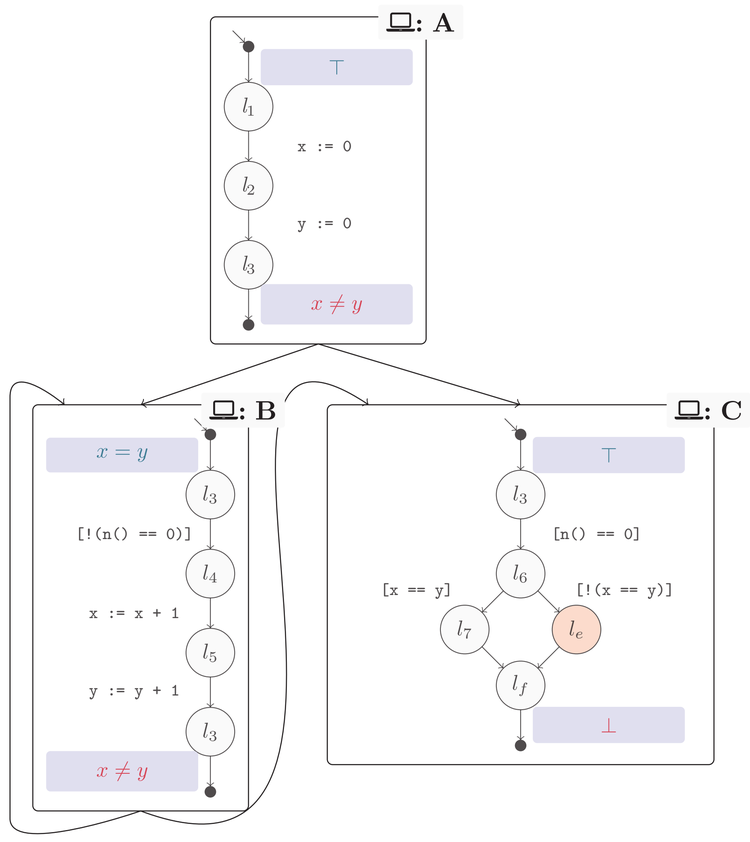
Iteration 1

Block	Result
A	$\downarrow \boxtimes_{B,C} : \top$
B	$\downarrow \boxtimes_{B,C} : \top$
C	$\uparrow \boxtimes_{A,B} : x \neq y$



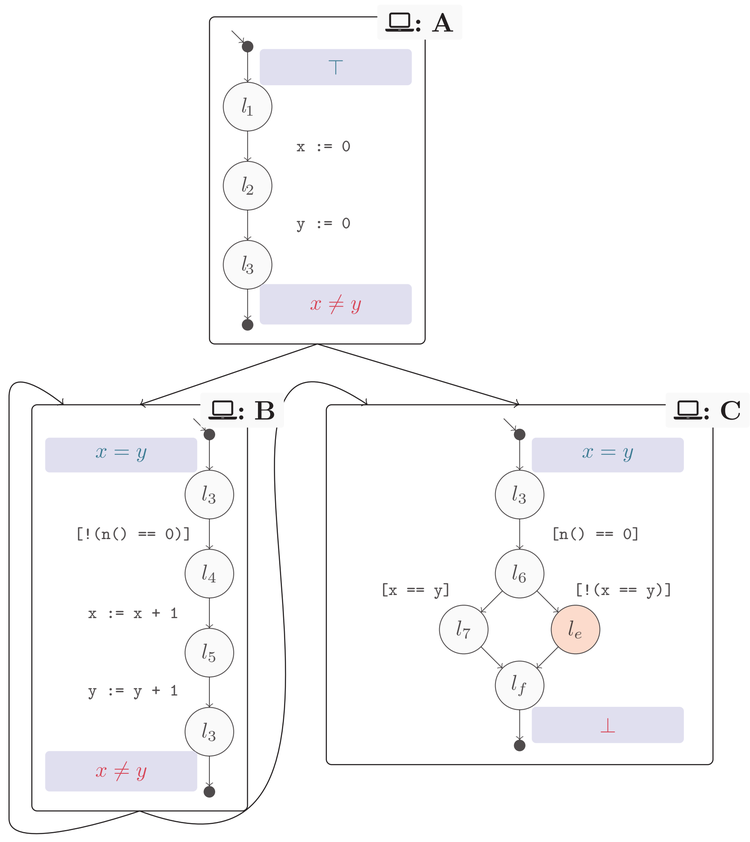
Iteration 2

Block	Result
A	$\downarrow \boxtimes_{B,C} : x = y$
B	$\uparrow \boxtimes_{A,B} : x \neq y$
C	*idle*

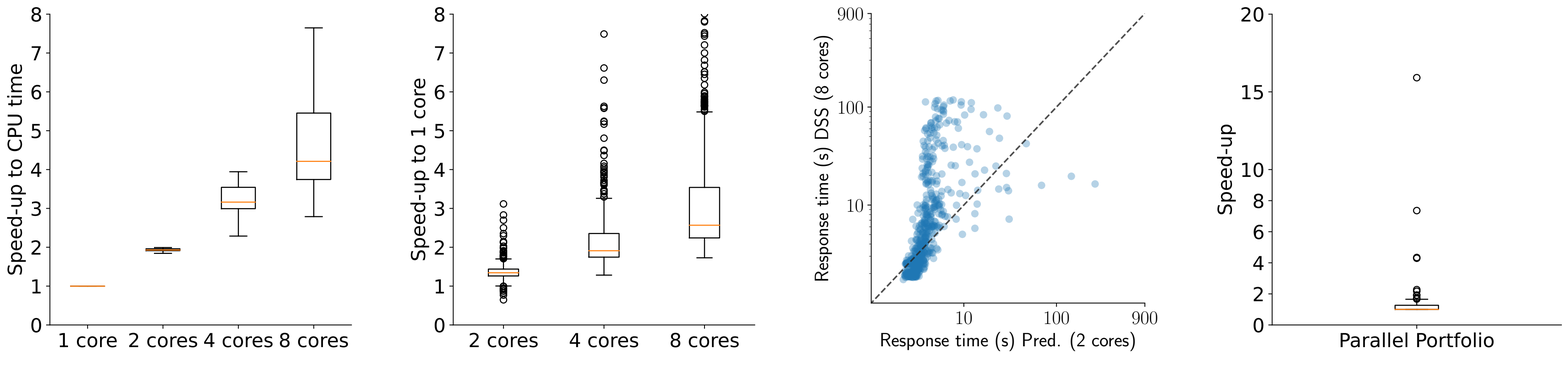


Iteration 3

Block	Result
A	$\downarrow \boxtimes_{B,C} : x = y$
B	$\downarrow \boxtimes_{B,C} : x = y$
C	*idle*



Evaluation



DSS effectively distributes the workload and can decrease the response time.

References

[1] Alt, L., Asadi, S., Chockler, H., Even-Mendoza, K., Fedyukovich, G., Hyvärinen, A.E.J., Sharygina, N.: HiFrog: SMT-based function summarization for software verification. In: Proc. TACAS. pp. 207–213. LNCS 10206 (2017). https://doi.org/10.1007/978-3-662-54580-5_12

[2] Beyer, D., Friedberger, K.: Domain-independent interprocedural program analysis using block-abstraction memoization. In: Proc. ESEC/FSE. pp. 50–62. ACM (2020). https://doi.org/10.1145/3368089.3409718

[3] Beyer, D., Kettl, M., Lemberger, T.: Decomposing software verification using distributed summary synthesis. pp. Article 59 (July 2024), 23 pages. FSE 2024 (2024). https://doi.org/10.1145/3660766

[4] Calcagno, C., Distefano, D., Dubreil, J., Gabi, D., Hooimeijer, P., Luca, M., O'Hearn, P.W., Papakonstantinou, I., Purbrick, J., Rodriguez, D.: Moving fast with software verification. In: Proc. NFM. pp. 3–11. LNCS 9058, Springer (2015). https://doi.org/10.1007/978-3-319-17524-9_1