TU WIEN
FAKULTÄT FÜR !NFORMATIK

for(syte)
Formal Methods in Systems Engineering

Upper and Lower Loop Bound Estimation by Symbolic Execution and Loop Acceleration

Pavel Čadek
TU Wien

Loop Bound Analysis

What is the maximal number of loop iterations?

Usage

- worst case execution time
- complexity analysis
- resource consumption
- schedulers
- ...

Reachability Bounds

What is the maximal number of executions of a specific program part?

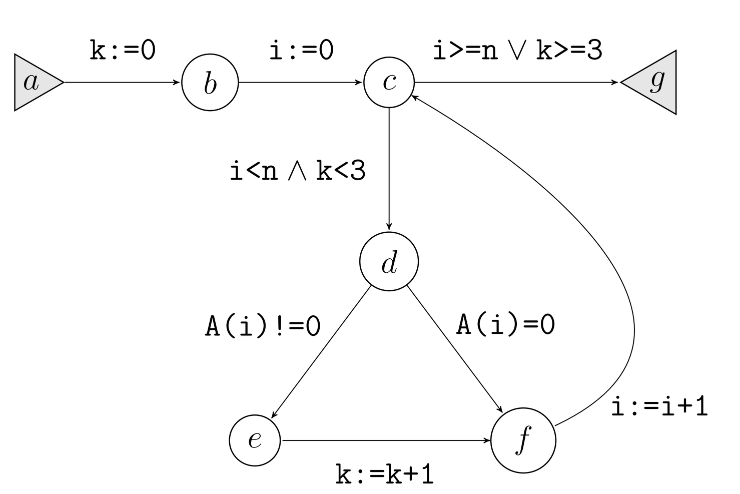
Lower Bounds

What is the minimal number of executions of a specific program part?

Usage

- improving precision of upper bounds
- best case computational complexity
- estimating tightness of upper bounds
- invariant generation

Example



Lower reachability bound:

$(b,c):\ 1$
$(c,d):\ \max\{\min\{n,3\},0\}$
$(d,e):\ 0$
$(d,f):\ \max\{n-3,0\}$

Upper reachability bound:

$(b,c):\ 1$
$(c,d):\ \max\{n,0\}$
$(d,e):\ \max\{\min\{n,3\},0\}$
$(d,f):\ \max\{n,0\}$

References

[1] J. Strejček and M. Trtík. “Abstracting Path Conditions”. In: ISSTA. 2012, pp. 155–165.
[2] Pavel Čadek, Jan Strejček, and Marek Trtík. “Tighter Loop Bound Analysis”. In: ATVA. 2016, pp. 512–527.

Loop Acceleration

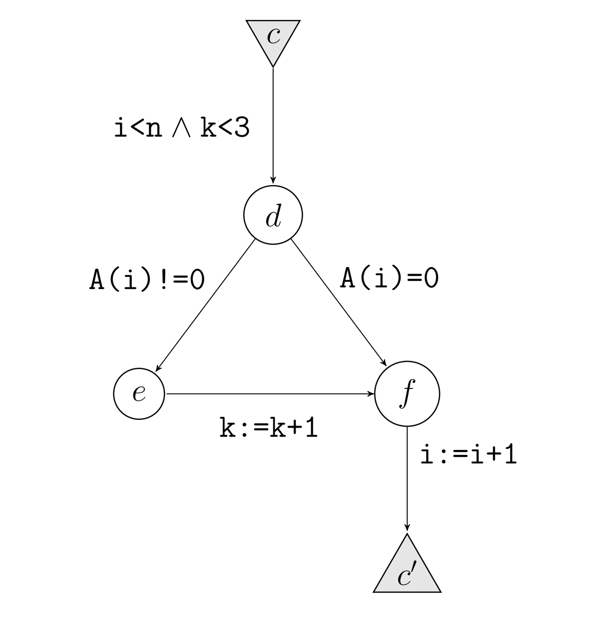
- representing several loop iterations by one transition
- solving path explosion for bounded model checking and symbolic execution

Usage

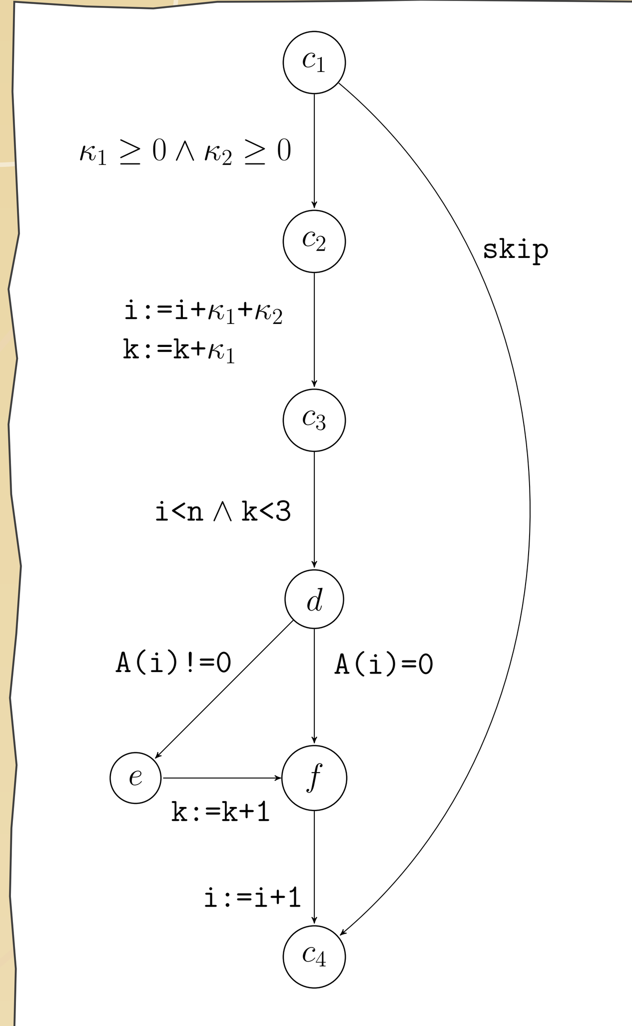
- Loop Underapproximation
  - an accelerated path added, others remain
  - bug finding (reducing depth of a bug)
- Loop Overapproximation
  - all paths replaced
  - resulting flowgraph acyclic
  - proving correctness

Our Acceleration Approach

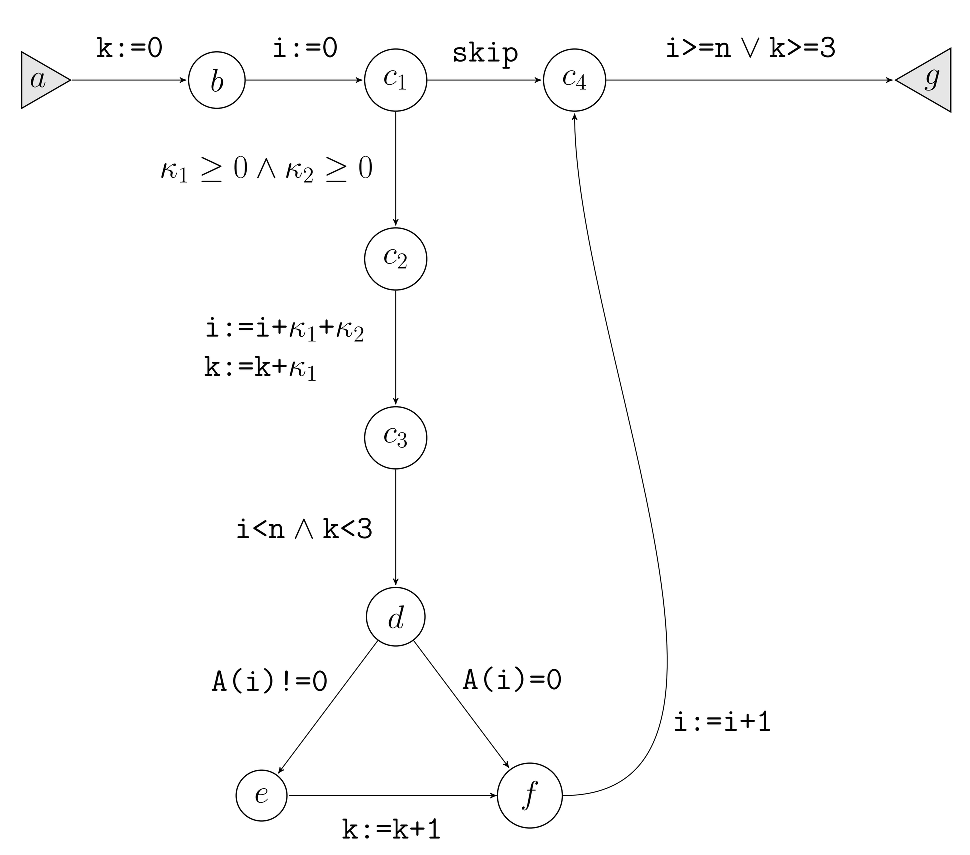
1. make a flowgraph induced by the loop, which represents one loop iteration
2. symbolically execute the induced flowgraph
3. introduce a counter κ_i for each loop path i
4. summarize the effect of $\kappa_1 + \kappa_2 + \ldots$ iterations
5. make 2 branches
   - 0 iterations
   - summary of $\kappa_1 + \kappa_2 + \ldots$ iterations and a copy of the last iteration



$\theta_1(\mathtt{i}) = \underline{i}+1$
$\theta_1(\mathtt{k}) = \underline{k}+1$
$\theta_2(\mathtt{i}) = \underline{i}+1$
$\theta_2(\mathtt{k}) = \underline{k}$
$\theta^{\vec{\kappa}}(\mathtt{i}) = \underline{i}+\kappa_1+\kappa_2$
$\theta^{\vec{\kappa}}(\mathtt{k}) = \underline{k}+\kappa_1$



Replacing the Loop with its Acyclic Overapproximation



Symbolic Execution

Instead of normal program inputs one supplies symbols representing arbitrary values. The execution splits into two at branching statements.

Symbolic Memory (θ)

- changes after each assignment
- values of variables as symbolic expressions

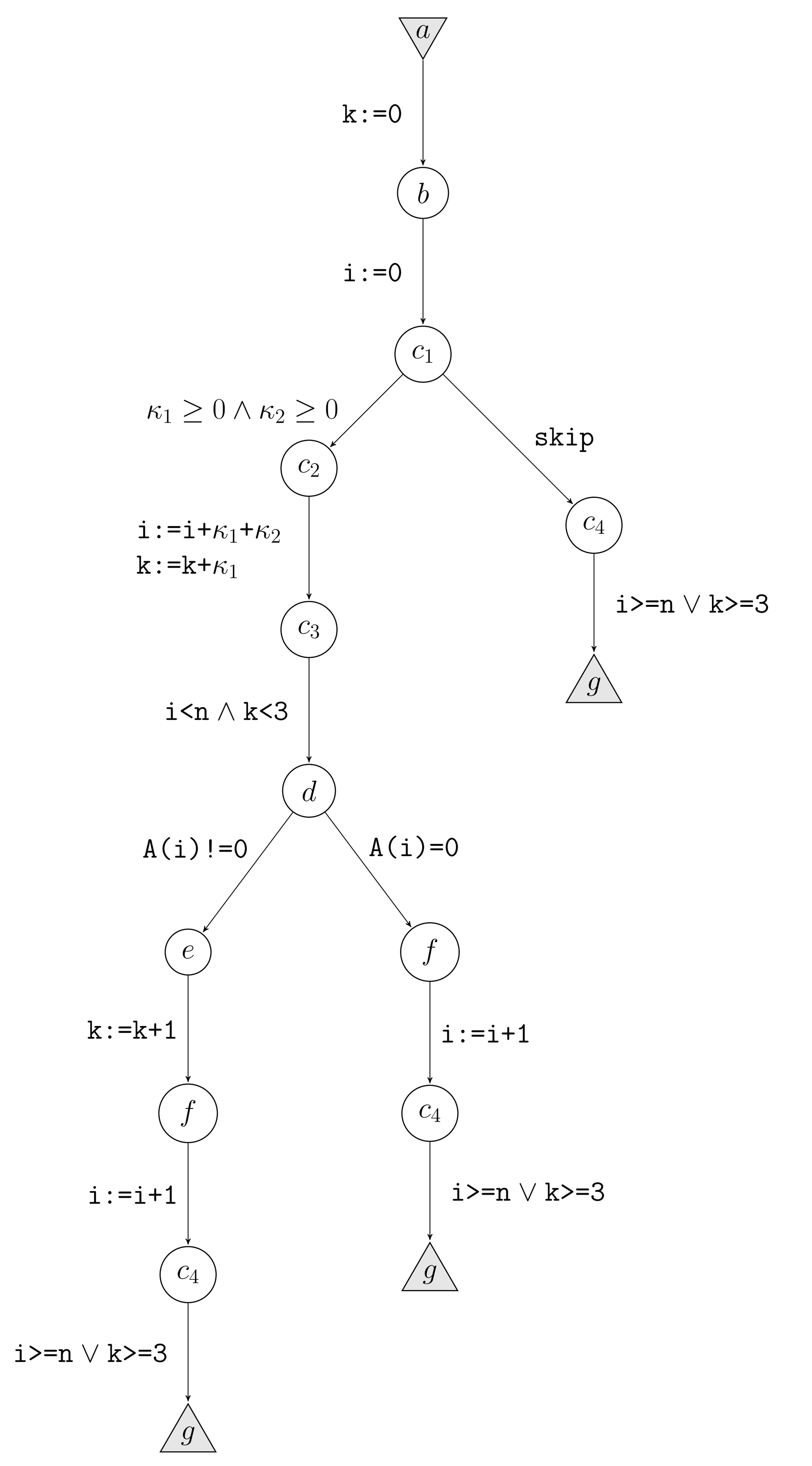
Path Condition (φ)

- changes after each branching condition
- a necessary and sufficient condition for an execution to follow the particular path

Usage

- summarizing effect of an acyclic piece of code
- bug finding

Symbolic Execution of the Acyclic Program



$\varphi_1 \equiv 0 \geq \underline{n} \ \vee\ 0 \geq 3$

$\varphi_2 \equiv \kappa_1 \geq 0 \ \wedge\ \kappa_2 \geq 0 \ \wedge\ \kappa_1+\kappa_2 < \underline{n} \ \wedge$
$\kappa_1 < 3 \ \wedge\ \underline{A}(\kappa_1+\kappa_2) \neq 0 \ \wedge$
$(\kappa_1+\kappa_2+1 \geq \underline{n} \ \vee\ \kappa_1+1 \geq 3)$

$\Rightarrow$
$\kappa_1+\kappa_2 < \underline{n}$
$\kappa_1+\kappa_2+1 \geq \max\{\underline{n},3\}$
$\kappa_1 < \min\{\underline{n},3\}$
$\kappa_1 \geq 0$
$\kappa_2 \geq \max\{\underline{n}-3,0\}$
$\kappa_2 < \underline{n}$

$\varphi_3 \equiv \kappa_1 \geq 0 \ \wedge\ \kappa_2 \geq 0 \ \wedge\ \kappa_1+\kappa_2 < \underline{n} \ \wedge$
$\kappa_1 < 3 \ \wedge\ \underline{A}(\kappa_1+\kappa_2) = 0 \ \wedge$
$(\kappa_1+\kappa_2+1 \geq \underline{n} \ \vee\ \kappa_1 \geq 3)$

$\Rightarrow \quad \ldots$

pcadek@forsyte.at