Better termination proving through cooperation

Marc Brockschmidt [1] Byron Cook [2,3] Carsten Fuhs [3]

[1]RWTH Aachen University

[2]Microsoft Research Cambridge

[3]University College London

CAV 2013

Termination Analysis: Invariants and Rank Functions

Example

```
y := 1;
while x > 0 do
    x := x − y;
    y := y + 1;
done
```

- Invariant $y > 0$ and rank function x prove termination
- How do we know that we need $y > 0$? $\curvearrowright$ x requires it

Termination Analysis: Invariants and Rank Functions

Example

```
y := 1;
while x > 0 do
    x := x − y;
    y := y + 1;
done
```

- Invariant $y > 0$ and rank function x prove termination
- How do we know that we need $y > 0$? ↷ x requires it
- How do we know that x is a RF? ↷ $y > 0$ proves it

Termination by iterative strengthening: Idea

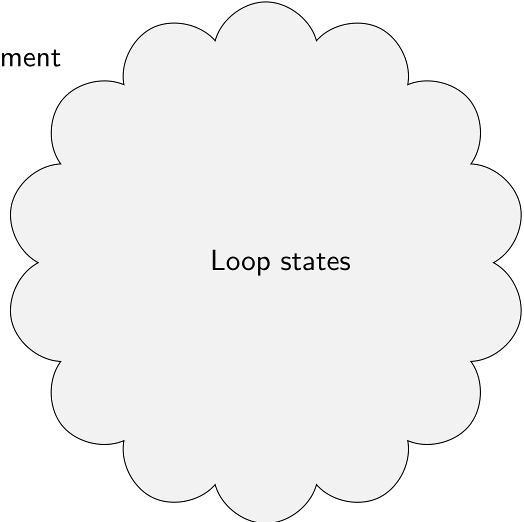
1. Safety: Provide samples (Counterexamples)
2. Rank tool: Find specific termination argument
3. Safety: Prove generality, or 1

Termination by iterative strengthening: Idea

1. Safety: Provide samples (Counterexamples)
2. Rank tool: Find specific termination argument
3. Safety: Prove generality, or 1

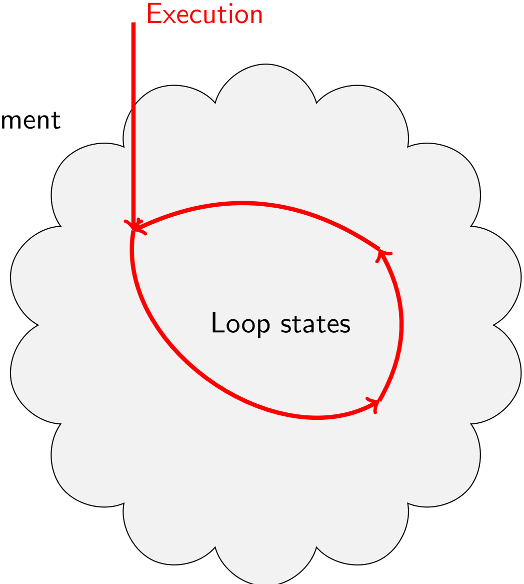
Termination by iterative strengthening

Find counterexample
then strengthen argument



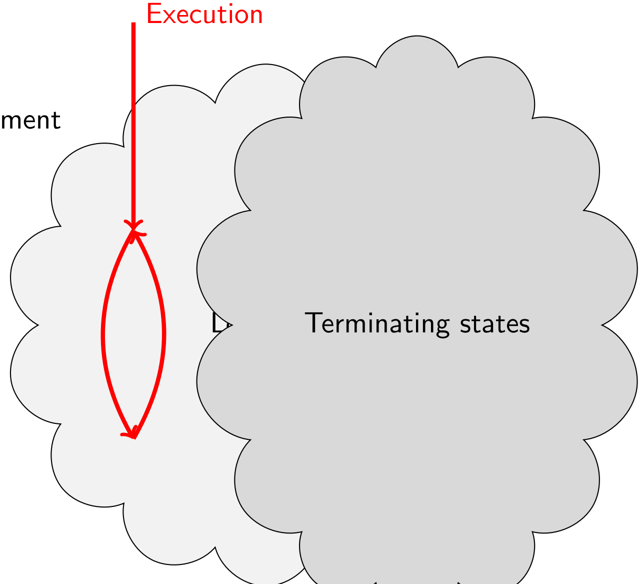
Termination by iterative strengthening

Find counterexample
then strengthen argument



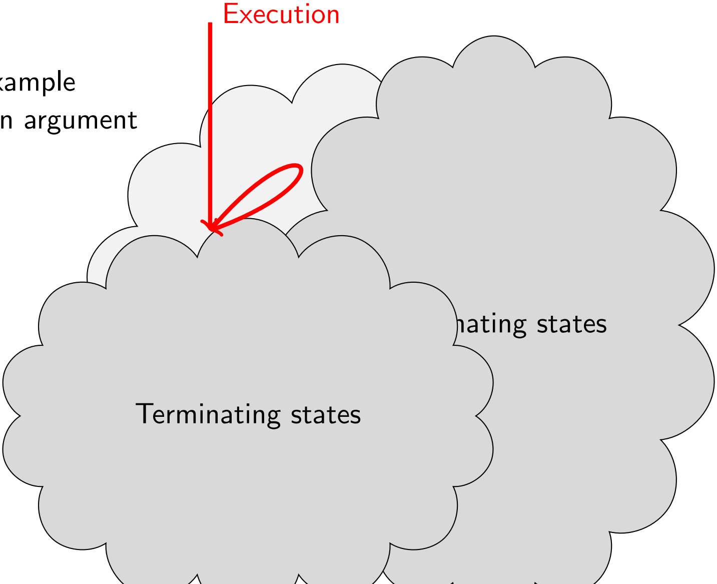
Termination by iterative strengthening

Find counterexample
then strengthen argument



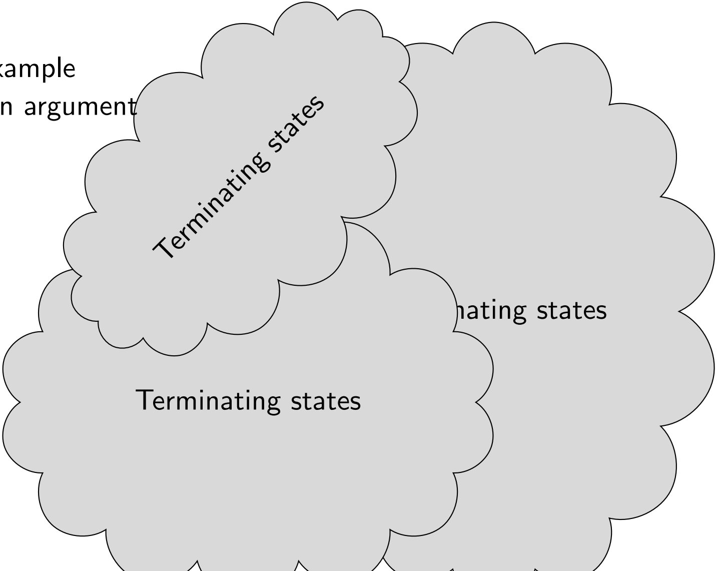
Termination by iterative strengthening

Find counterexample
then strengthen argument



Termination by iterative strengthening

Find counterexample
then strengthen argument



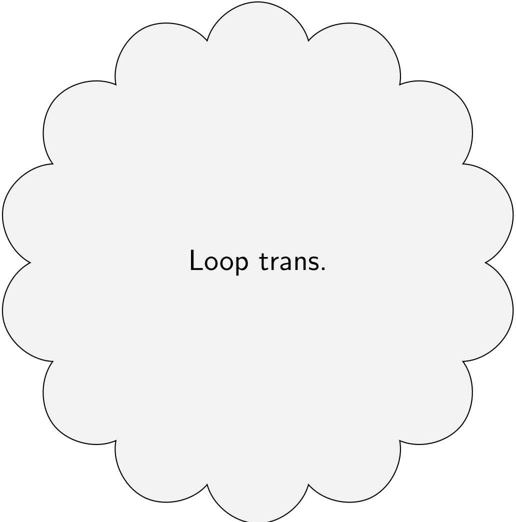
Termination by iterative strengthening: Worst case

1. Safety: Look at everything, then return old sample
2. Rank tool: Find **too** specific termination argument
3. Safety: Can't prove generality, repeat 1

Termination by iterative strengthening: Worst case

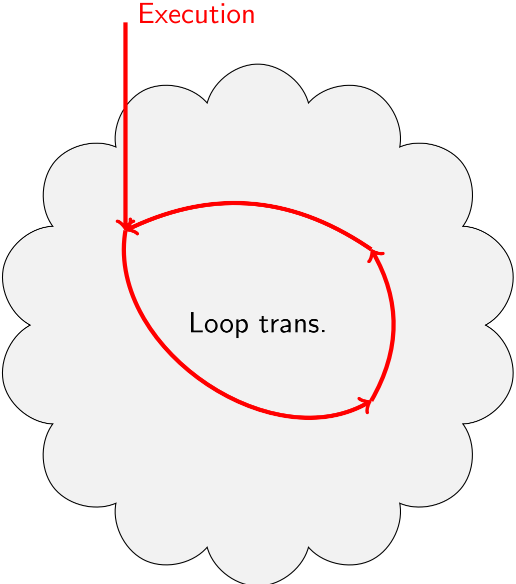
1. Safety: Look at everything, then return old sample
2. Rank tool: Find **too** specific termination argument
3. Safety: Can't prove generality, repeat 1

Termination by iterative simplification



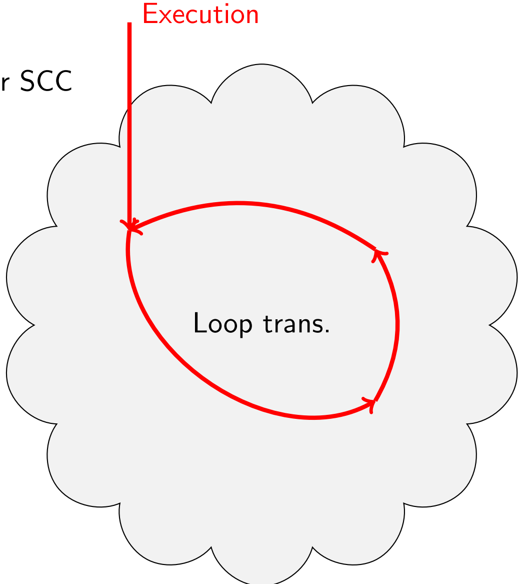
Loop trans.

Termination by iterative simplification



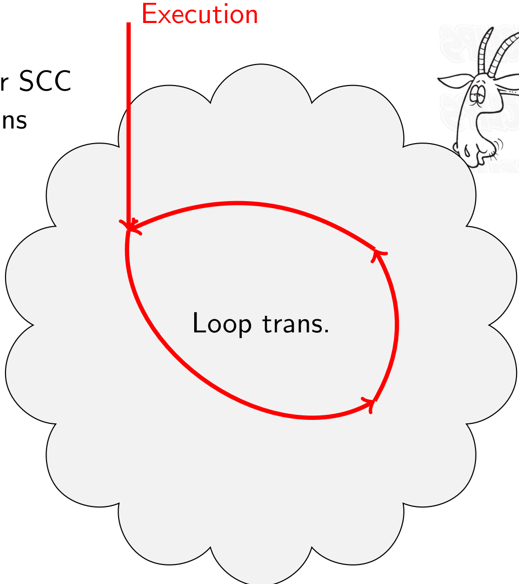
Termination by iterative simplification

Find rank function for SCC



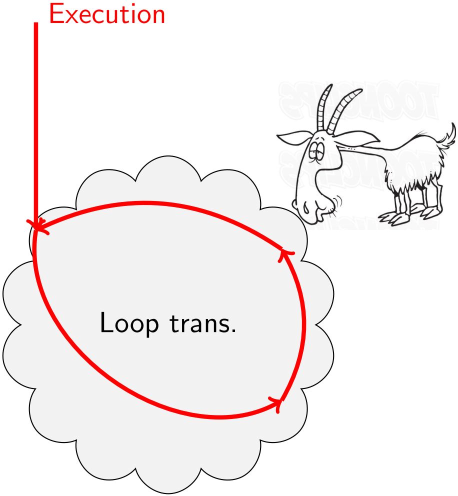
Termination by iterative simplification

Find rank function for SCC
then remove transitions



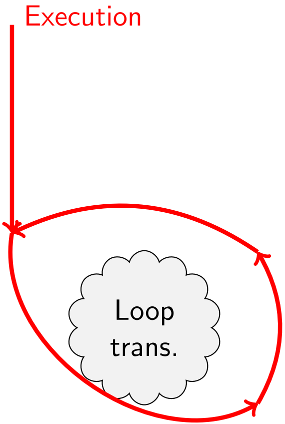
Termination by iterative simplification

Find rank function for SCC
then remove transitions



Termination by iterative simplification

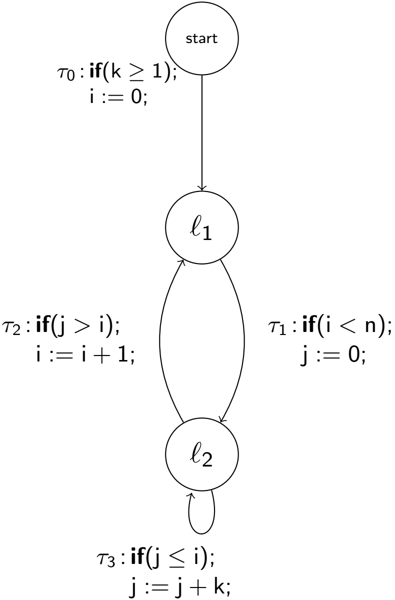
Find rank function for SCC
then remove transitions



Termination by cooperation

1. Safety: Provide samples (Counterexamples)
2. Rank tool: Find termination argument **in context**
3. Rank tool: Mark definitely terminating parts
4. Safety: Prove generality for rest, or 1

A nested example

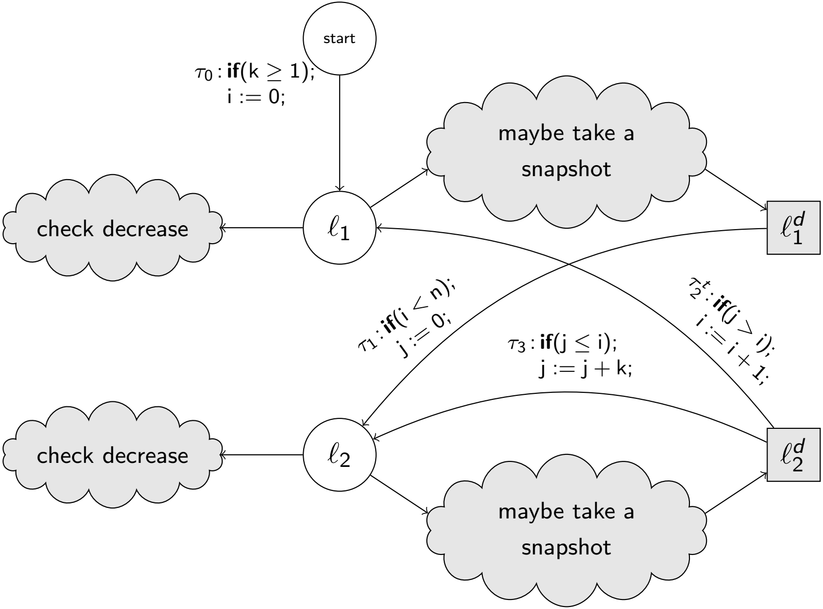


start
τ_0 : **if**(k $\geq$ 1); i := 0;
ℓ_1
τ_2 : **if**(j > i); i := i + 1;
τ_1 : **if**(i < n); j := 0;
ℓ_2
τ_3 : **if**(j $\leq$ i); j := j + k;

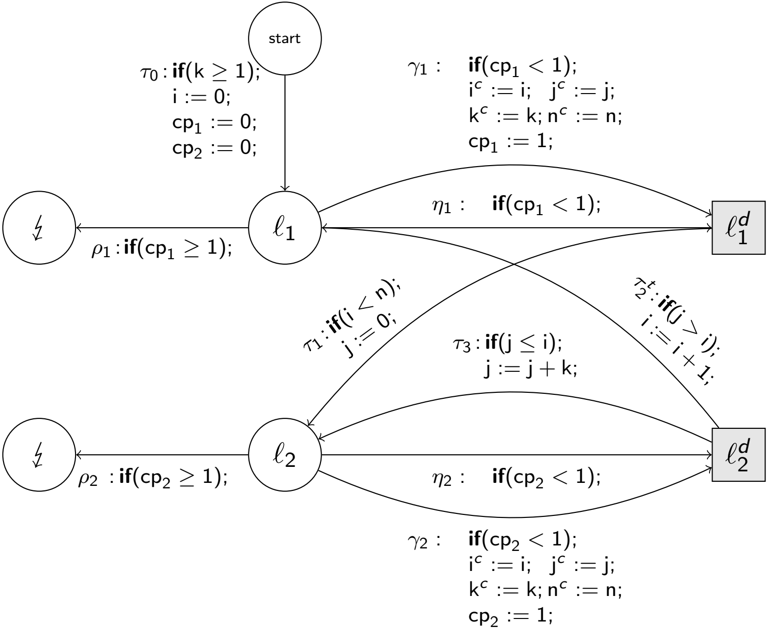
Example (Source)

```
if k ≥ 1 then
  i := 0;
ℓ1  while i < n do
      j := 0;

ℓ2    while j ≤ i do
        j := j + k;
      done

      i := i + 1;
    done
fi
```

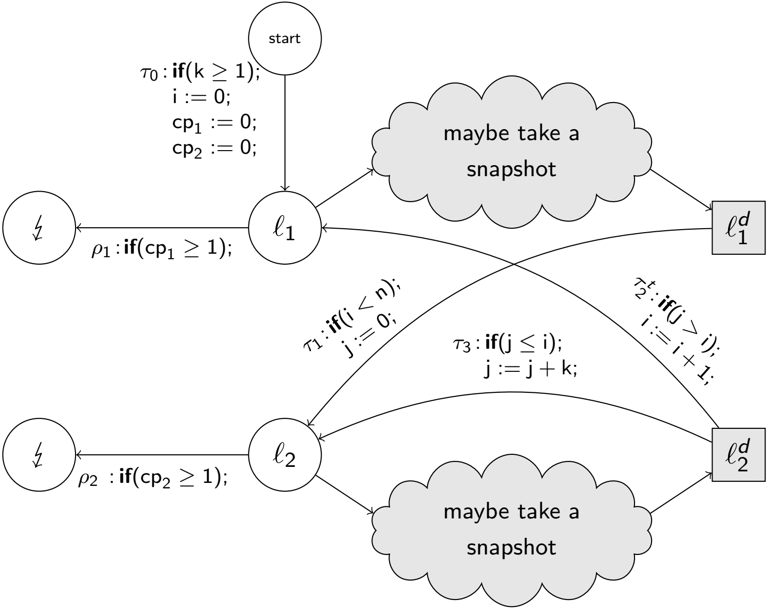
A nested example: Instrumentation



A nested example: Instrumentation


start
τ_0 : **if**(k $\geq$ 1);
i := 0;
cp_1 := 0;
cp_2 := 0;
γ_1 : **if**(cp_1 < 1);
i^c := i; j^c := j;
k^c := k; n^c := n;
cp_1 := 1;
η_1 : **if**(cp_1 < 1);
ρ_1 : **if**($cp_1 \geq 1$);
ℓ_1
ℓ_1^d
τ_1 : **if**(i < n);
j := 0;
τ_2^t : **if**(j > i);
i := i + 1;
τ_3 : **if**(j $\leq$ i);
j := j + k;
ℓ_2
ℓ_2^d
ρ_2 : **if**($cp_2 \geq 1$);
η_2 : **if**(cp_2 < 1);
γ_2 : **if**(cp_2 < 1);
i^c := i; j^c := j;
k^c := k; n^c := n;
cp_2 := 1;

A nested example: Instrumentation


start
τ_0 : **if**(k $\geq$ 1);
i := 0;
cp$_1$:= 0;
cp$_2$:= 0;
maybe take a snapshot
ℓ_1
ℓ_1^d
ρ_1 : **if**(cp$_1$ $\geq$ 1);
τ_1 : **if**(i < n);
j := 0;
τ_2^t : **if**(j > i);
i := i + 1;
τ_3 : **if**(j $\leq$ i);
j := j + k;
ℓ_2
ℓ_2^d
ρ_2 : **if**(cp$_2$ $\geq$ 1);
maybe take a snapshot

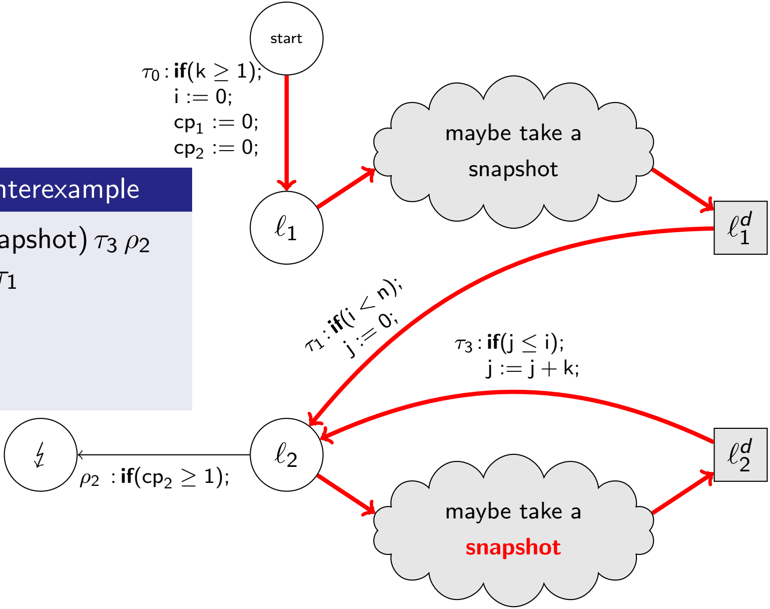
A nested example: Strengthening

First Counterexample

$\tau_0\ \tau_1$ (snapshot) $\tau_3\ \rho_2$

Stem: $\tau_0\ \tau_1$

Loop: τ_3


start
τ_0 : **if**(k $\geq$ 1);
i := 0;
cp$_1$:= 0;
cp$_2$:= 0;
ℓ_1
maybe take a snapshot
ℓ_1^d
τ_1 : **if**(i < n);
j := 0;
τ_3 : **if**(j $\leq$ i);
j := j + k;
ℓ_2
ℓ_2^d
maybe take a **snapshot**
ρ_2 : **if**(cp$_2$ $\geq$ 1);
↯

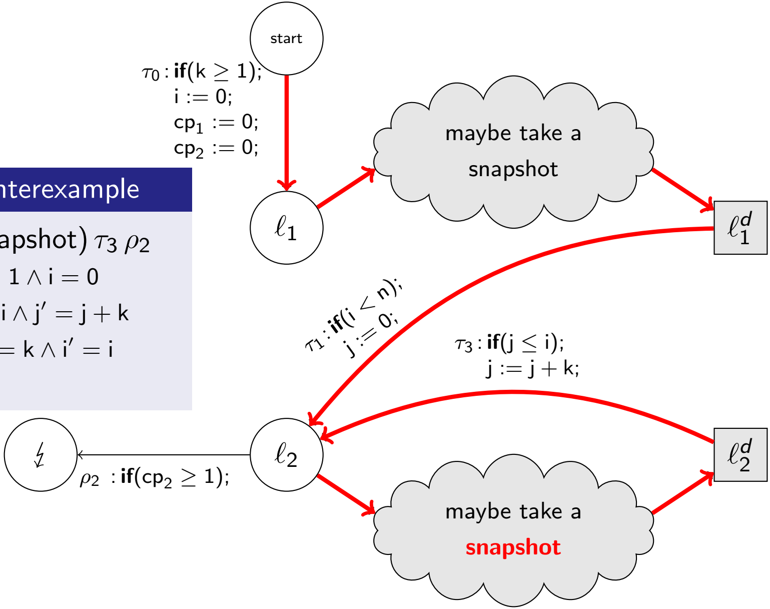
A nested example: Strengthening

First Counterexample

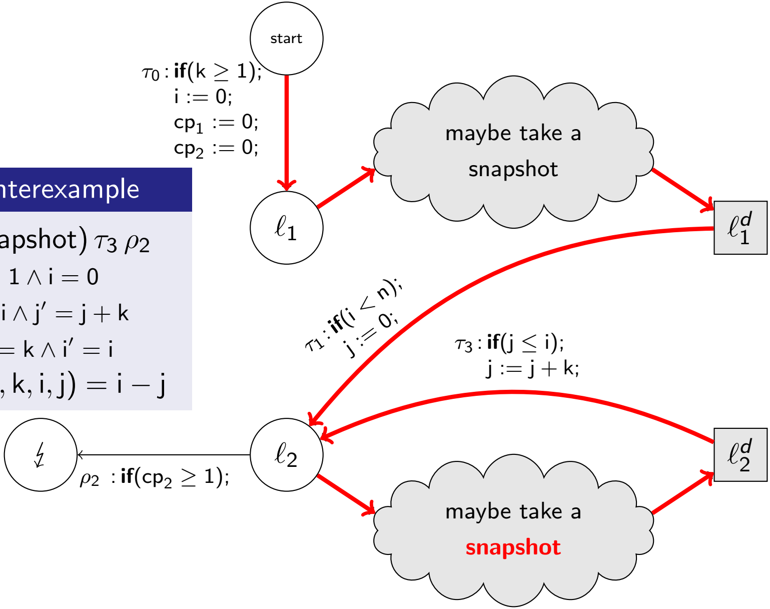
$\tau_0\ \tau_1$ (snapshot) $\tau_3\ \rho_2$

Stem: $\mathsf{k} \geq 1 \land \mathsf{i} = 0$

Loop: $\mathsf{j} \leq \mathsf{i} \land \mathsf{j}' = \mathsf{j} + \mathsf{k}$
$\land\ \mathsf{k}' = \mathsf{k} \land \mathsf{i}' = \mathsf{i}$



A nested example: Strengthening


start
$\tau_0 : \textbf{if}(\mathsf{k} \geq 1);$
$\mathsf{i} := 0;$
$\mathsf{cp}_1 := 0;$
$\mathsf{cp}_2 := 0;$
maybe take a snapshot
ℓ_1
ℓ_1^d
$\tau_1 : \textbf{if}(\mathsf{i} < \mathsf{n});$
$\mathsf{j} := 0;$
$\tau_3 : \textbf{if}(\mathsf{j} \leq \mathsf{i});$
$\mathsf{j} := \mathsf{j} + \mathsf{k};$
ℓ_2
ℓ_2^d
$\lightning$
$\rho_2 : \textbf{if}(\mathsf{cp}_2 \geq 1);$
maybe take a snapshot

First Counterexample

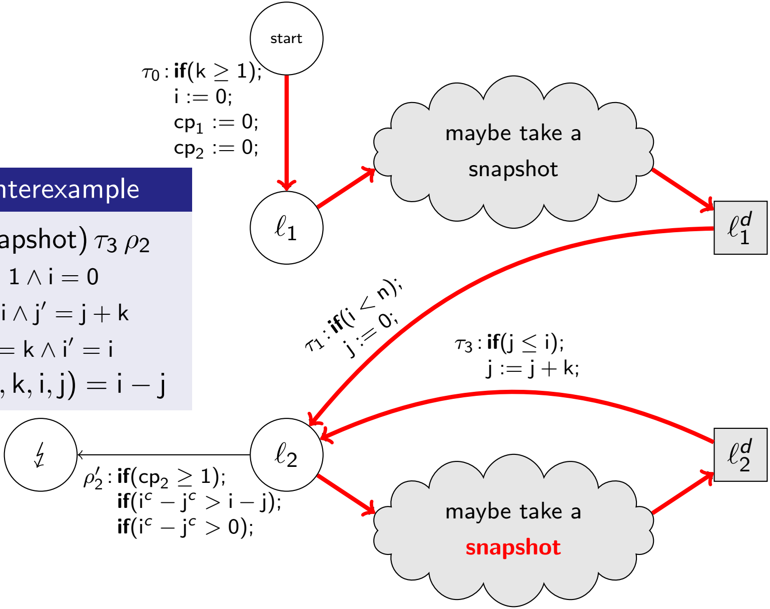
$\tau_0\ \tau_1$ (snapshot) $\tau_3\ \rho_2$

Stem: $\mathsf{k} \geq 1 \wedge \mathsf{i} = 0$

Loop: $\mathsf{j} \leq \mathsf{i} \wedge \mathsf{j}' = \mathsf{j} + \mathsf{k}$
$\wedge\ \mathsf{k}' = \mathsf{k} \wedge \mathsf{i}' = \mathsf{i}$

RF: $f_{\ell_2}(\mathsf{n}, \mathsf{k}, \mathsf{i}, \mathsf{j}) = \mathsf{i} - \mathsf{j}$

A nested example: Strengthening



start
$\tau_0:$ **if**$(\mathsf{k} \geq 1)$;
$\mathsf{i} := 0$;
$\mathsf{cp}_1 := 0$;
$\mathsf{cp}_2 := 0$;
ℓ_1
maybe take a snapshot
ℓ_1^d
$\tau_1:$ **if**$(\mathsf{i} < \mathsf{n})$;
$\mathsf{j} := 0$;
$\tau_3:$ **if**$(\mathsf{j} \leq \mathsf{i})$;
$\mathsf{j} := \mathsf{j} + \mathsf{k}$;
ℓ_2
ℓ_2^d
↯
$\rho_2':$ **if**$(\mathsf{cp}_2 \geq 1)$;
if$(\mathsf{i}^c - \mathsf{j}^c > \mathsf{i} - \mathsf{j})$;
if$(\mathsf{i}^c - \mathsf{j}^c > 0)$;
maybe take a **snapshot**

First Counterexample

$\tau_0\ \tau_1\ (\text{snapshot})\ \tau_3\ \rho_2$

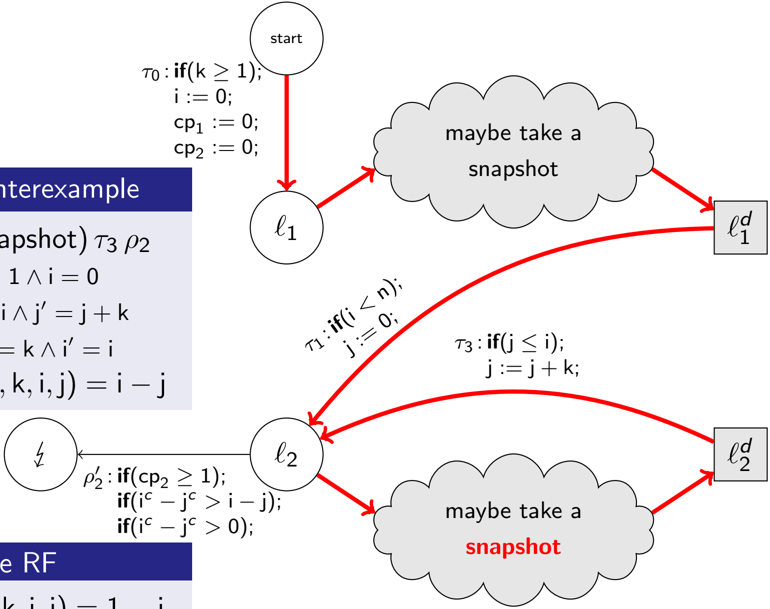
Stem: $\mathsf{k} \geq 1 \land \mathsf{i} = 0$

Loop: $\mathsf{j} \leq \mathsf{i} \land \mathsf{j}' = \mathsf{j} + \mathsf{k}$

$\land\ \mathsf{k}' = \mathsf{k} \land \mathsf{i}' = \mathsf{i}$

RF: $f_{\ell_2}(\mathsf{n}, \mathsf{k}, \mathsf{i}, \mathsf{j}) = \mathsf{i} - \mathsf{j}$

A nested example: Strengthening


start
τ_0 : **if**(k $\geq$ 1);
i := 0;
cp$_1$:= 0;
cp$_2$:= 0;
maybe take a
snapshot
ℓ_1
ℓ_1^d
τ_1 : **if**(i < n);
j := 0;
τ_3 : **if**(j $\leq$ i);
j := j + k;
ℓ_2
ℓ_2^d
$\lightning$
ρ_2' : **if**(cp$_2$ $\geq$ 1);
if(i^c − j^c > i − j);
if(i^c − j^c > 0);
maybe take a
snapshot

First Counterexample

$\tau_0\ \tau_1$ (snapshot) $\tau_3\ \rho_2$

Stem: $\mathsf{k} \geq 1 \land \mathsf{i} = 0$

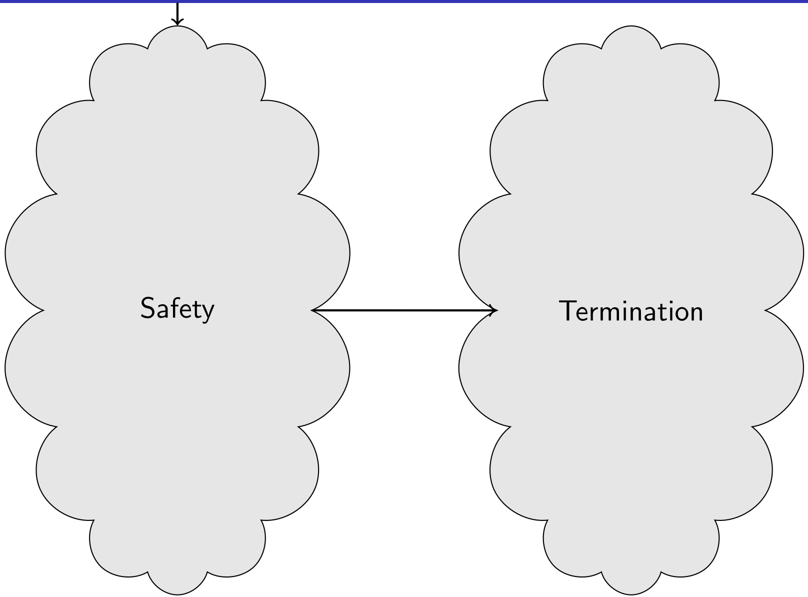
Loop: $\mathsf{j} \leq \mathsf{i} \land \mathsf{j}' = \mathsf{j} + \mathsf{k}$
$\land\ \mathsf{k}' = \mathsf{k} \land \mathsf{i}' = \mathsf{i}$

RF: $f_{\ell_2}(\mathsf{n}, \mathsf{k}, \mathsf{i}, \mathsf{j}) = \mathsf{i} - \mathsf{j}$

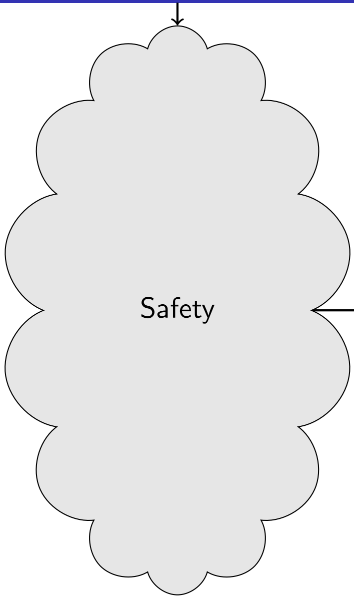
Alternative RF

RF: $f'_{\ell_2}(\mathsf{n}, \mathsf{k}, \mathsf{i}, \mathsf{j}) = 1 - \mathsf{j}$

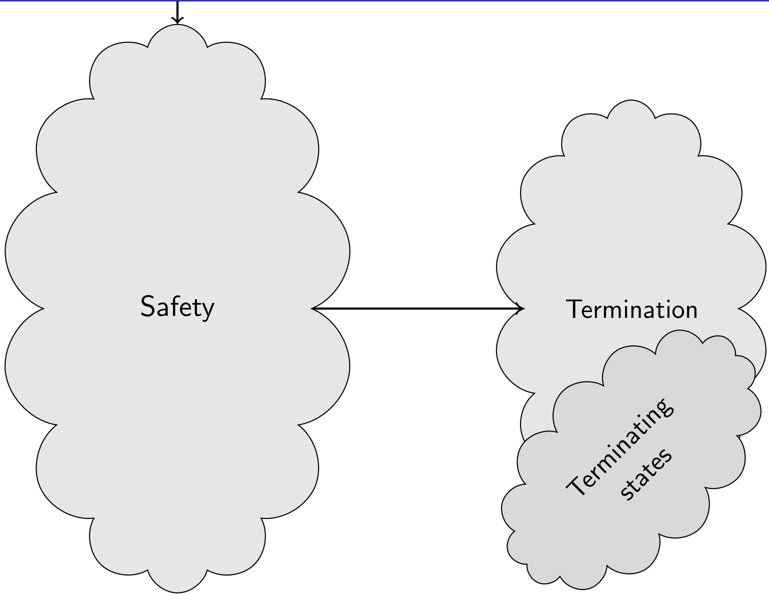
Cooperation: High-level view



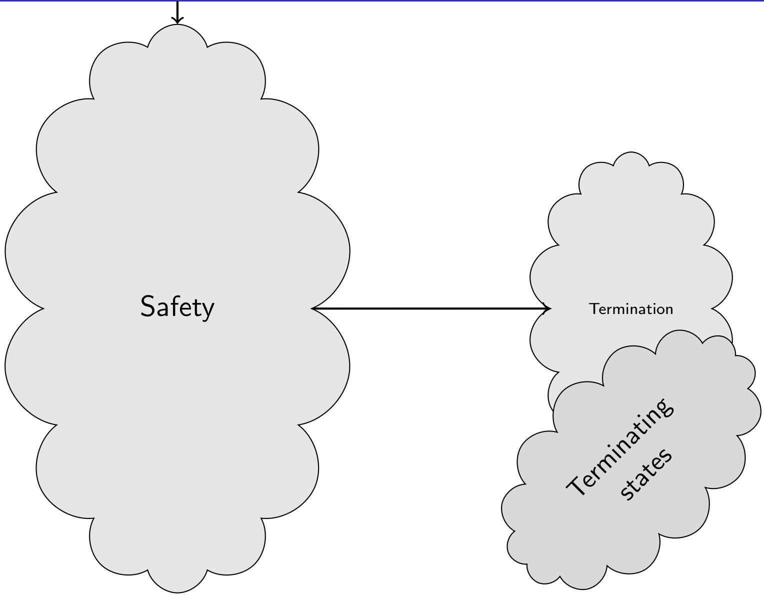
Cooperation: High-level view



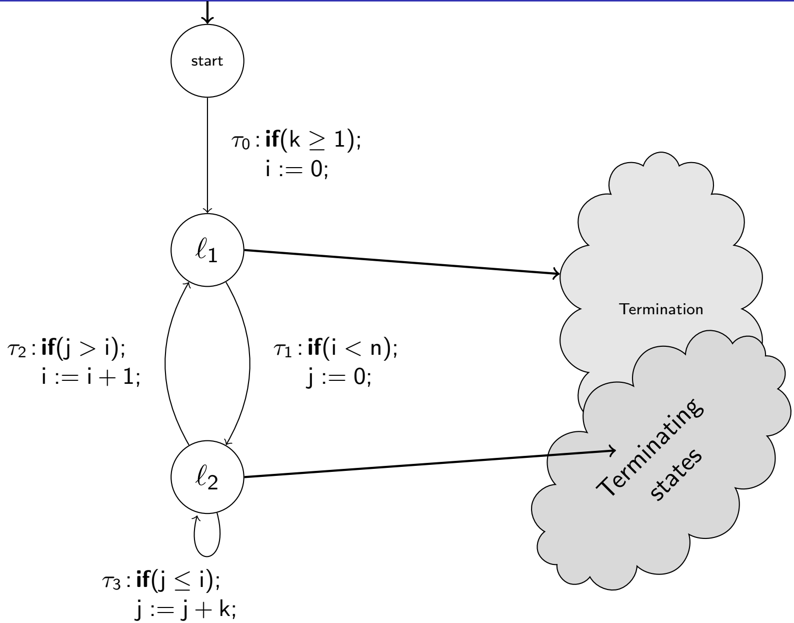
Cooperation: High-level view



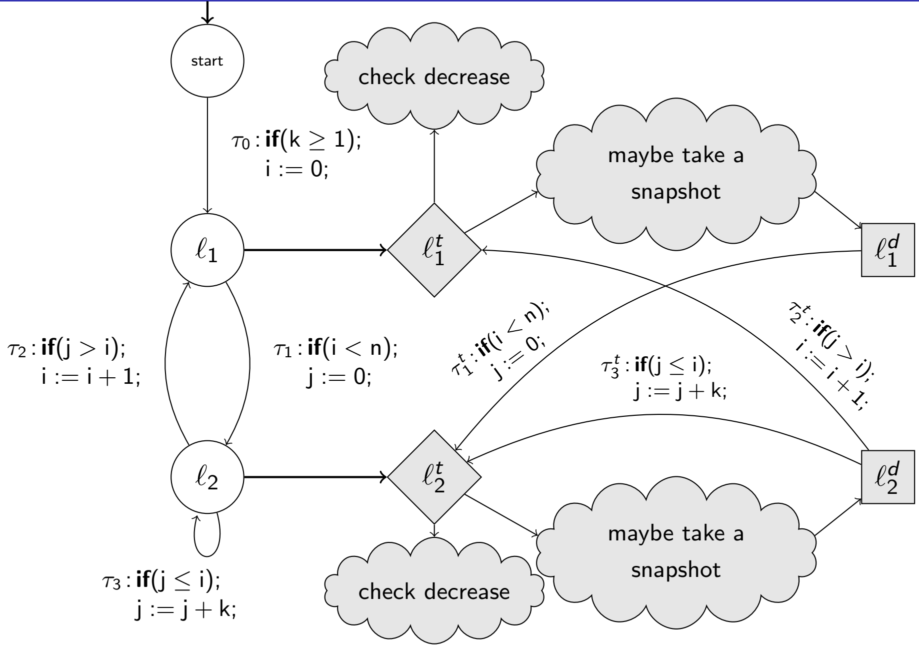
Cooperation: High-level view



Cooperation: High-level view


start
τ_0 : **if**(k $\geq$ 1);
i := 0;
ℓ_1
Termination
τ_2 : **if**(j > i);
i := i + 1;
τ_1 : **if**(i < n);
j := 0;
Terminating
states
ℓ_2
τ_3 : **if**(j $\leq$ i);
j := j + k;

Cooperation: High-level view


start
check decrease
maybe take a snapshot
τ_0 : **if**(k $\geq$ 1); i := 0;
ℓ_1
ℓ_1^t
ℓ_1^d
τ_2 : **if**(j > i); i := i + 1;
τ_1 : **if**(i < n); j := 0;
τ_1^t : **if**(i < n); j := 0;
τ_2^t : **if**(j > i); i := i + 1;
τ_3^t : **if**(j $\leq$ i); j := j + k;
ℓ_2
ℓ_2^t
ℓ_2^d
maybe take a snapshot
τ_3 : **if**(j $\leq$ i); j := j + k;
check decrease

Cooperation

Intuition:

- **Safety subgraph**: original program
- **Termination subgraph**: instrumented copy

Cooperation

Intuition:

- **Safety subgraph**: original program
- **Termination subgraph**: instrumented copy

- **Ranking**: Simplify problem, “point out hard bits”

Cooperation

Intuition:

- **Safety subgraph**: original program
- **Termination subgraph**: instrumented copy

- **Ranking**: Simplify problem, “point out hard bits”
- **Safety**: Analyze whole program, “point out invariants”

Cooperation

Intuition:

- **Safety subgraph**: original program
- **Termination subgraph**: instrumented copy

- **Ranking**: Simplify problem, “point out hard bits”
- **Safety**: Analyze whole program, “point out invariants”

Approach:

- Analyze whole SCC, not counterexample slice

Cooperation

Intuition:

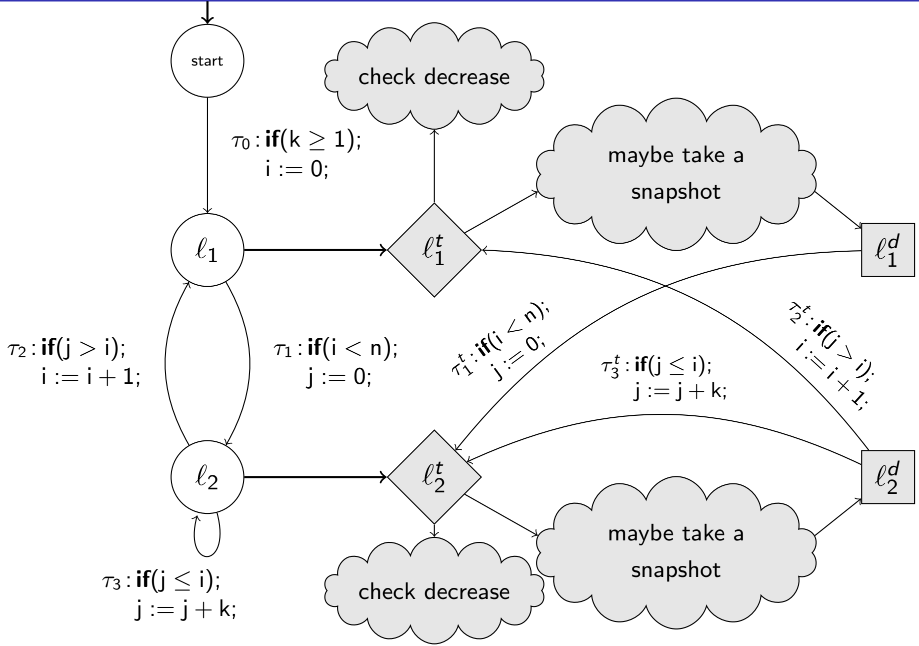
- **Safety subgraph**: original program
- **Termination subgraph**: instrumented copy

- **Ranking**: Simplify problem, “point out hard bits”
- **Safety**: Analyze whole program, “point out invariants”

Approach:

- Analyze whole SCC, not counterexample slice
- Remove transitions after proof

Cooperation: Simplification



start

τ_0 : **if**(k $\geq$ 1);
i := 0;

check decrease

maybe take a snapshot

ℓ_1

ℓ_1^t

ℓ_1^d

τ_2 : **if**(j > i);
i := i + 1;

τ_1 : **if**(i < n);
j := 0;

τ_1^t : **if**(i < n);
j := 0;

τ_3^t : **if**(j $\leq$ i);
j := j + k;

τ_2^t : **if**(j > i);
i := i + 1;

ℓ_2

ℓ_2^t

ℓ_2^d

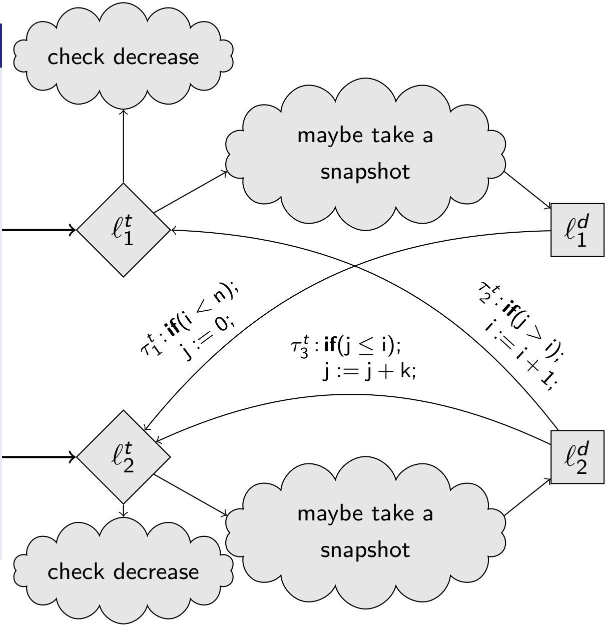
maybe take a snapshot

check decrease

τ_3 : **if**(j $\leq$ i);
j := j + k;

Cooperation: Simplification

Simplification

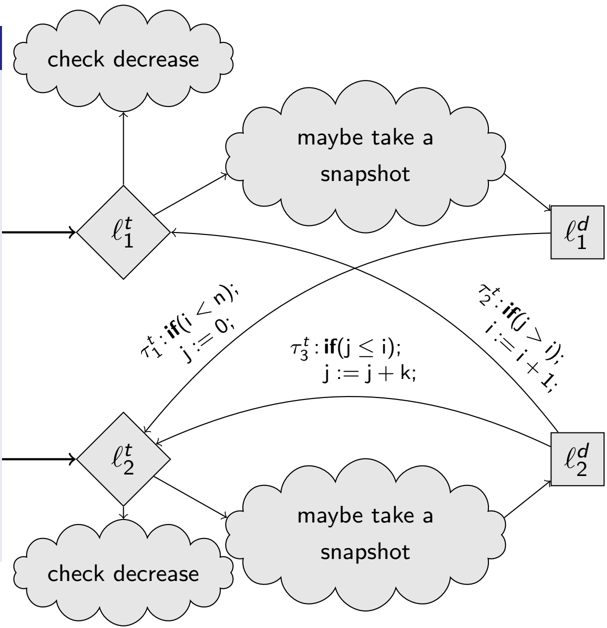

check decrease
maybe take a snapshot
ℓ_1^t
ℓ_1^d
$\tau_1^t : \textbf{if}(\mathsf{i} < \mathsf{n}); \; \mathsf{j} := 0;$
$\tau_2^t : \textbf{if}(\mathsf{j} > \mathsf{i}); \; \mathsf{i} := \mathsf{i} + 1;$
$\tau_3^t : \textbf{if}(\mathsf{j} \leq \mathsf{i}); \; \mathsf{j} := \mathsf{j} + \mathsf{k};$
ℓ_2^t
ℓ_2^d
check decrease
maybe take a snapshot

Cooperation: Simplification

Simplification

1. Find SCC $\mathcal{S}$ in termination graph:

   $\ell_1^t, \ell_1^d, \ell_2^t, \ell_2^d$


check decrease
maybe take a snapshot
ℓ_1^t
ℓ_1^d
τ_1^t : **if**(i < n); j := 0;
τ_2^t : **if**(j > i); i := i + 1;
τ_3^t : **if**(j ≤ i); j := j + k;
ℓ_2^t
ℓ_2^d
maybe take a snapshot
check decrease

Cooperation: Simplification

Simplification

1. Find SCC $\mathcal{S}$ in termination graph:

$$\ell_1^t, \ell_1^d, \ell_2^t, \ell_2^d$$

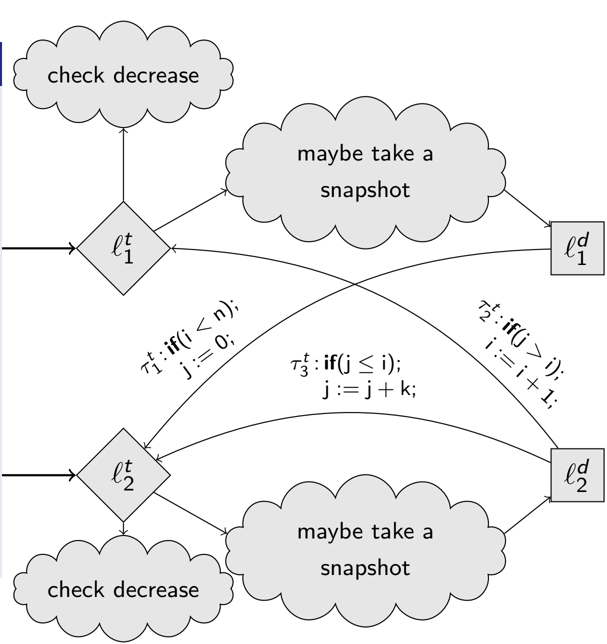
2. Find $\mathcal{S}$-orienting RF:

$$f^1_{\ell_1^t}(\mathrm{i,j,k,n}) = \mathrm{n} - \mathrm{i} + 1$$

$$f^1_{\ell_1^d}(\mathrm{i,j,k,n}) = \mathrm{n} - \mathrm{i} + 1$$

$$f^1_{\ell_2^t}(\mathrm{i,j,k,n}) = \mathrm{n} - \mathrm{i}$$

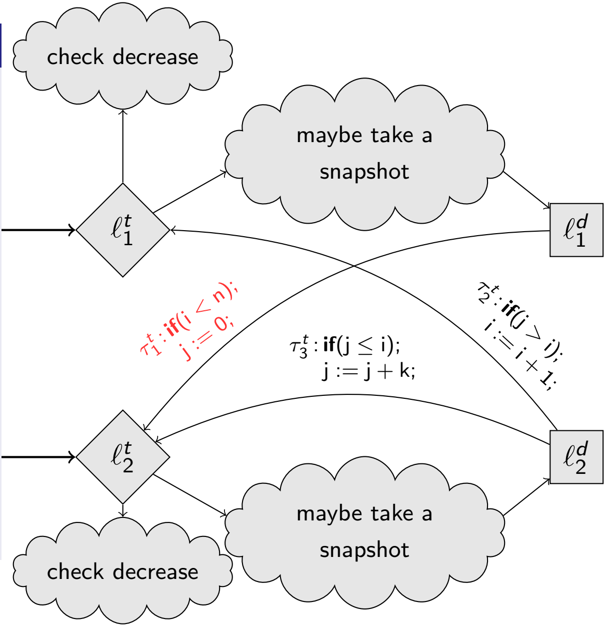
$$f^1_{\ell_2^d}(\mathrm{i,j,k,n}) = \mathrm{n} - \mathrm{i}$$


check decrease
maybe take a snapshot
ℓ_1^t
ℓ_1^d
τ_1^t : **if**(i < n); j := 0;
τ_3^t : **if**(j ≤ i); j := j + k;
τ_2^t : **if**(j > i); i := i + 1;
ℓ_2^t
ℓ_2^d
maybe take a snapshot
check decrease

Cooperation: Simplification

Simplification

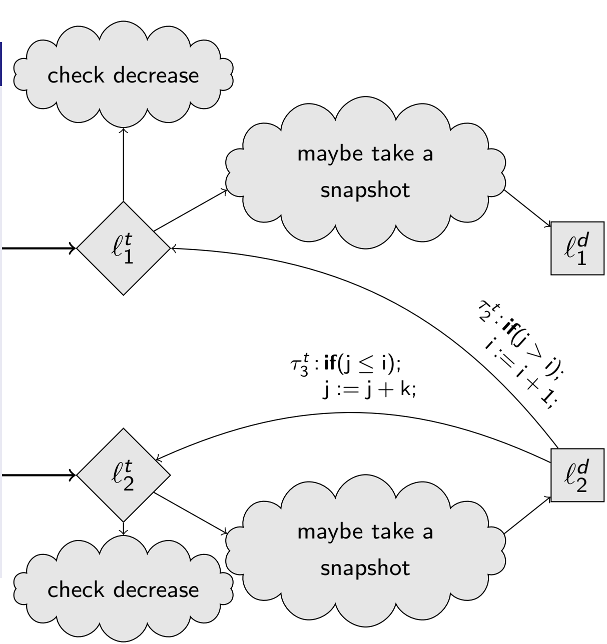
1. Find SCC $\mathcal{S}$ in termination graph:
   $$\ell_1^t, \ell_1^d, \ell_2^t, \ell_2^d$$
2. Find $\mathcal{S}$-orienting RF:
   $$f^1_{\ell_1^t}(\mathsf{i,j,k,n}) = \mathsf{n} - \mathsf{i} + 1$$
   $$f^1_{\ell_1^d}(\mathsf{i,j,k,n}) = \mathsf{n} - \mathsf{i} + 1$$
   $$f^1_{\ell_2^t}(\mathsf{i,j,k,n}) = \mathsf{n} - \mathsf{i}$$
   $$f^1_{\ell_2^d}(\mathsf{i,j,k,n}) = \mathsf{n} - \mathsf{i}$$
3. Delete decr./bounded


check decrease
maybe take a snapshot
ℓ_1^t
ℓ_1^d
τ_1^t : **if**(i < n); j := 0;
τ_2^t : **if**(j > i); i := i + 1;
τ_3^t : **if**(j ≤ i); j := j + k;
ℓ_2^t
ℓ_2^d
maybe take a snapshot
check decrease

Cooperation: Simplification

Simplification

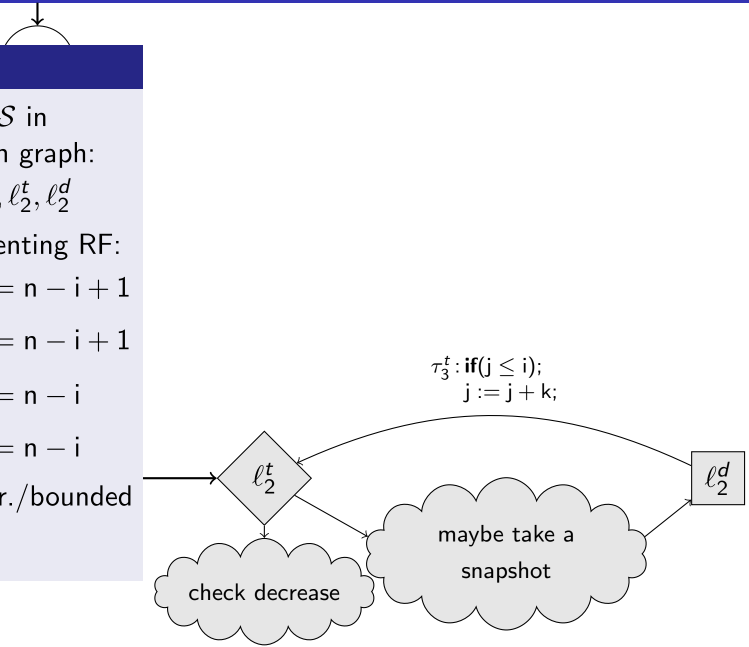
1. Find SCC $\mathcal{S}$ in termination graph:
   $$\ell_1^t, \ell_1^d, \ell_2^t, \ell_2^d$$
2. Find $\mathcal{S}$-orienting RF:
   $$f^1_{\ell_1^t}(\text{i,j,k,n}) = \text{n} - \text{i} + 1$$
   $$f^1_{\ell_1^d}(\text{i,j,k,n}) = \text{n} - \text{i} + 1$$
   $$f^1_{\ell_2^t}(\text{i,j,k,n}) = \text{n} - \text{i}$$
   $$f^1_{\ell_2^d}(\text{i,j,k,n}) = \text{n} - \text{i}$$
3. Delete decr./bounded



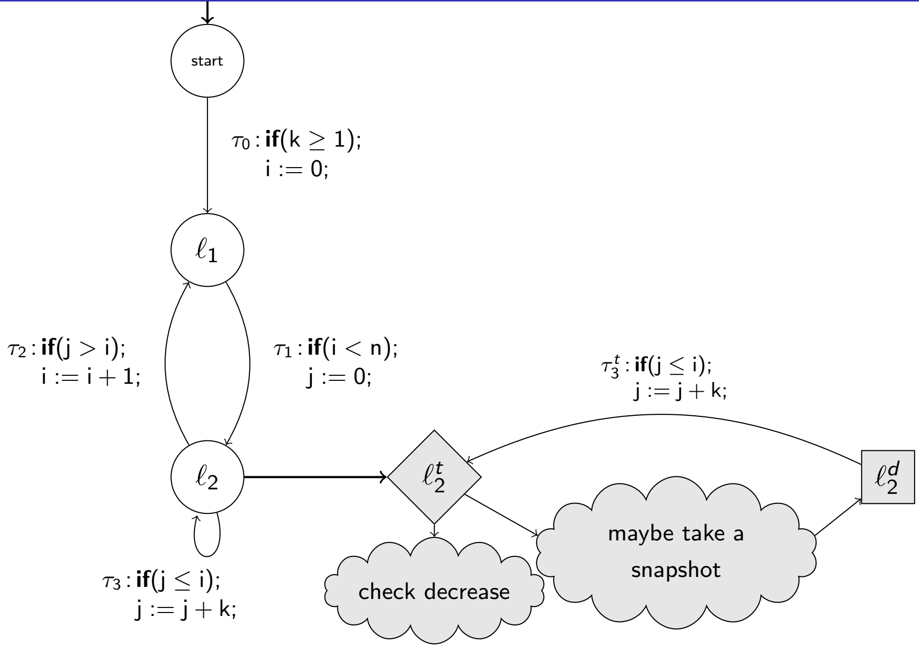
Cooperation: Simplification

Simplification

1. Find SCC $\mathcal{S}$ in termination graph:
   $$\ell_1^t, \ell_1^d, \ell_2^t, \ell_2^d$$
2. Find $\mathcal{S}$-orienting RF:
   $$f^1_{\ell_1^t}(\mathsf{i},\mathsf{j},\mathsf{k},\mathsf{n}) = \mathsf{n} - \mathsf{i} + 1$$
   $$f^1_{\ell_1^d}(\mathsf{i},\mathsf{j},\mathsf{k},\mathsf{n}) = \mathsf{n} - \mathsf{i} + 1$$
   $$f^1_{\ell_2^t}(\mathsf{i},\mathsf{j},\mathsf{k},\mathsf{n}) = \mathsf{n} - \mathsf{i}$$
   $$f^1_{\ell_2^d}(\mathsf{i},\mathsf{j},\mathsf{k},\mathsf{n}) = \mathsf{n} - \mathsf{i}$$
3. Delete decr./bounded
4. Clean up



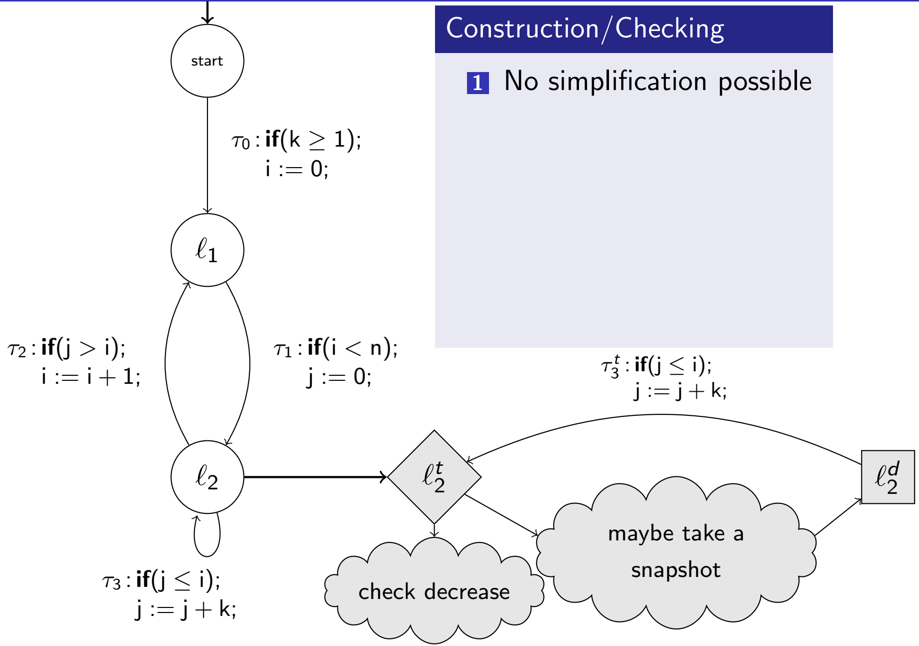
Cooperation


start
$\tau_0 : \textbf{if}(k \geq 1);$
$i := 0;$
ℓ_1
$\tau_2 : \textbf{if}(j > i);$
$i := i + 1;$
$\tau_1 : \textbf{if}(i < n);$
$j := 0;$
$\tau_3^t : \textbf{if}(j \leq i);$
$j := j + k;$
ℓ_2
ℓ_2^t
ℓ_2^d
maybe take a snapshot
check decrease
$\tau_3 : \textbf{if}(j \leq i);$
$j := j + k;$

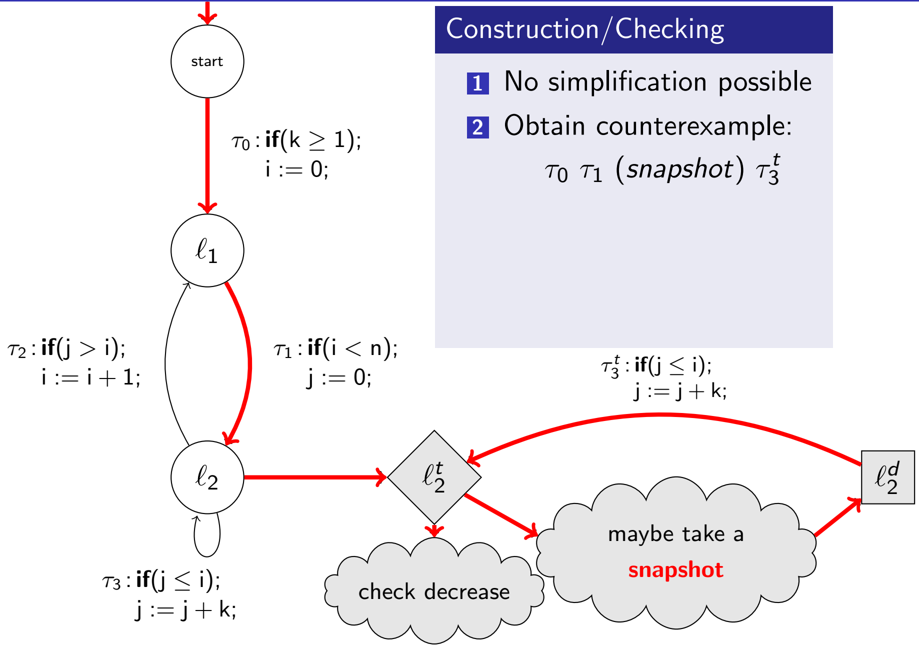
Cooperation: Invariants

Construction/Checking

1. No simplification possible


start
τ_0 : **if**(k $\geq$ 1);
i := 0;
ℓ_1
τ_2 : **if**(j > i);
i := i + 1;
τ_1 : **if**(i < n);
j := 0;
ℓ_2
τ_3 : **if**(j $\leq$ i);
j := j + k;
ℓ_2^t
τ_3^t : **if**(j $\leq$ i);
j := j + k;
ℓ_2^d
check decrease
maybe take a snapshot

Cooperation: Invariants

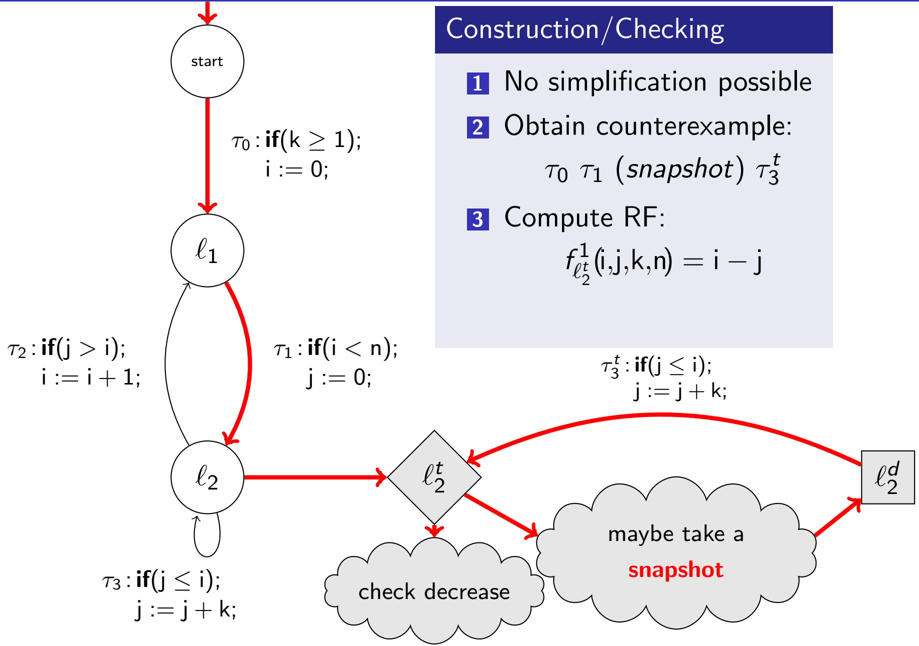

start
τ_0 : **if**(k $\geq$ 1);
i := 0;
ℓ_1
τ_2 : **if**(j > i);
i := i + 1;
τ_1 : **if**(i < n);
j := 0;
ℓ_2
τ_3 : **if**(j $\leq$ i);
j := j + k;
ℓ_2^t
τ_3^t : **if**(j $\leq$ i);
j := j + k;
ℓ_2^d
check decrease
maybe take a
snapshot

Construction/Checking

1. No simplification possible
2. Obtain counterexample:

$$\tau_0\ \tau_1\ (\mathit{snapshot})\ \tau_3^t$$

Cooperation: Invariants


start
τ_0 : **if**(k $\geq$ 1);
i := 0;
ℓ_1
τ_2 : **if**(j > i);
i := i + 1;
τ_1 : **if**(i < n);
j := 0;
ℓ_2
τ_3 : **if**(j $\leq$ i);
j := j + k;
ℓ_2^t
τ_3^t : **if**(j $\leq$ i);
j := j + k;
ℓ_2^d
check decrease
maybe take a
snapshot

Construction/Checking

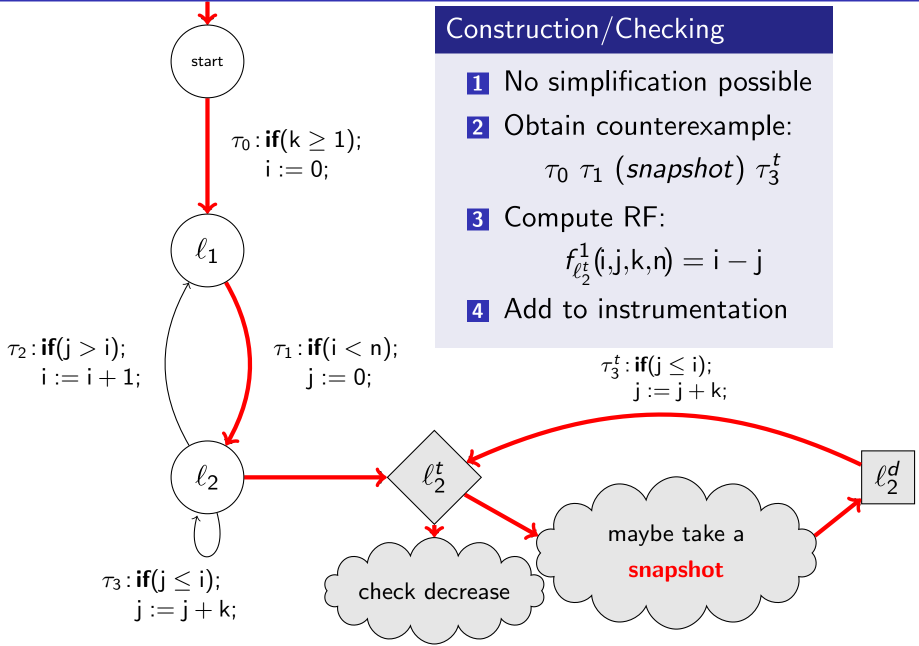
1. No simplification possible
2. Obtain counterexample:

   $\tau_0\ \tau_1\ (\textit{snapshot})\ \tau_3^t$

3. Compute RF:

   $f^1_{\ell_2^t}(\text{i,j,k,n}) = \text{i} - \text{j}$

Cooperation: Invariants


start
$\tau_0 : \textbf{if}(k \geq 1);$
$i := 0;$
ℓ_1
$\tau_2 : \textbf{if}(j > i);$
$i := i + 1;$
$\tau_1 : \textbf{if}(i < n);$
$j := 0;$
ℓ_2
$\tau_3 : \textbf{if}(j \leq i);$
$j := j + k;$
ℓ_2^t
$\tau_3^t : \textbf{if}(j \leq i);$
$j := j + k;$
ℓ_2^d
check decrease
maybe take a
snapshot

Construction/Checking

1. No simplification possible
2. Obtain counterexample:

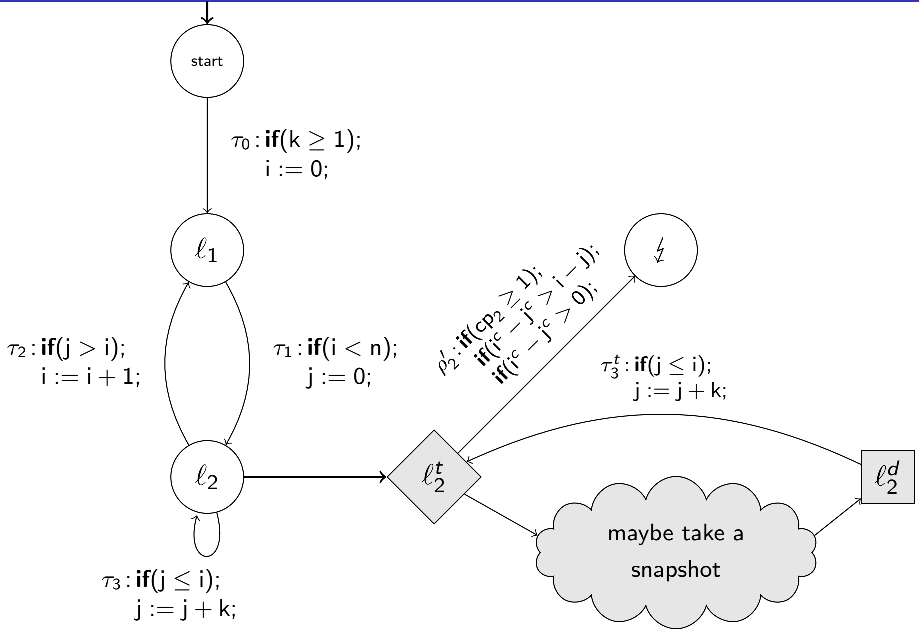
   $$\tau_0 \ \tau_1 \ (\textit{snapshot}) \ \tau_3^t$$

3. Compute RF:

   $$f^1_{\ell_2^t}(i,j,k,n) = i - j$$

4. Add to instrumentation

Cooperation: Invariants


start
τ_0 : **if**(k $\geq$ 1);
i := 0;
ℓ_1
τ_2 : **if**(j > i);
i := i + 1;
τ_1 : **if**(i < n);
j := 0;
ℓ_2
τ_3 : **if**(j $\leq$ i);
j := j + k;
ρ_2' : **if**($\mathrm{cp}_2 \geq 1$);
if($i^c - j^c > i - j$);
if($i^c - j^c > 0$);
↯
τ_3^t : **if**(j $\leq$ i);
j := j + k;
ℓ_2^t
maybe take a snapshot
ℓ_2^d

Cooperation: Evaluation

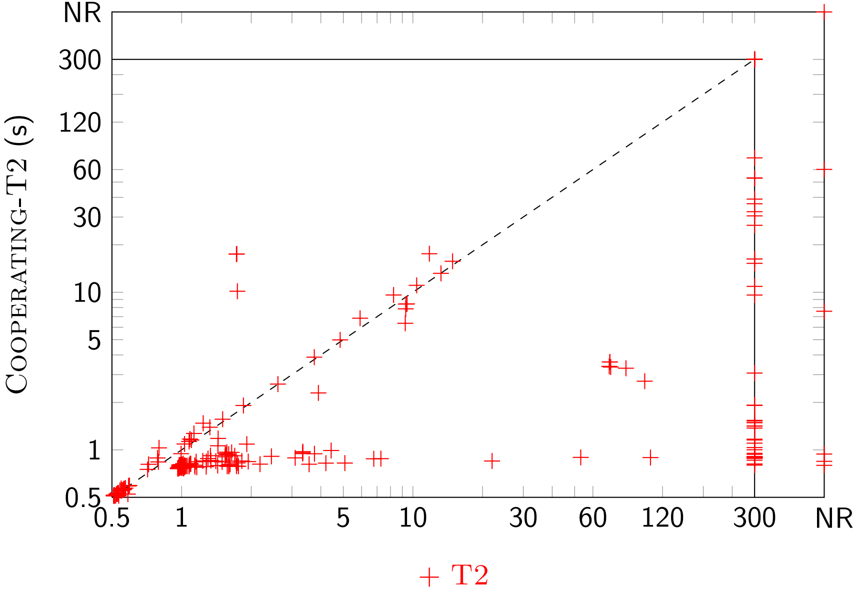
Evaluated on 449 termination proving benchmarks
260 known terminating, 181 known non-terminating, 8 unknown
Sources: Windows drivers, APACHE, POSTGRESQL, ...

Cooperation: Evaluation

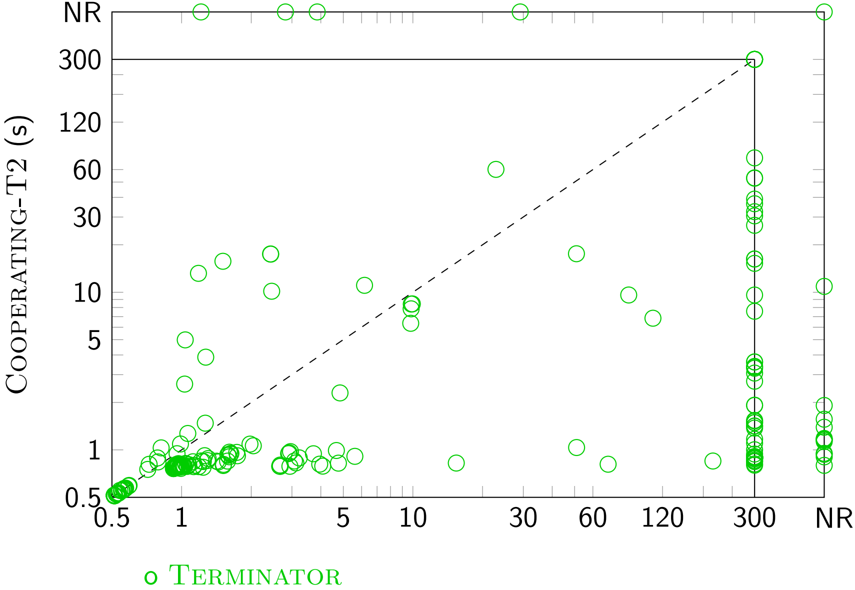
Evaluated on 449 termination proving benchmarks
260 known terminating, 181 known non-terminating, 8 unknown
Sources: Windows drivers, APACHE, POSTGRESQL, ...

	Term (#)	Term (avg. s)
COOPERATING-T2	245	3.42
APROVE	197	2.21
KITTEL	196	4.65
T2	189	5.15
APROVE+INTERPROC	185	1.53
TERMINATOR	177	4.99
SIZE-CHANGE/MCNP	156	17.50
ARMC	138	16.16

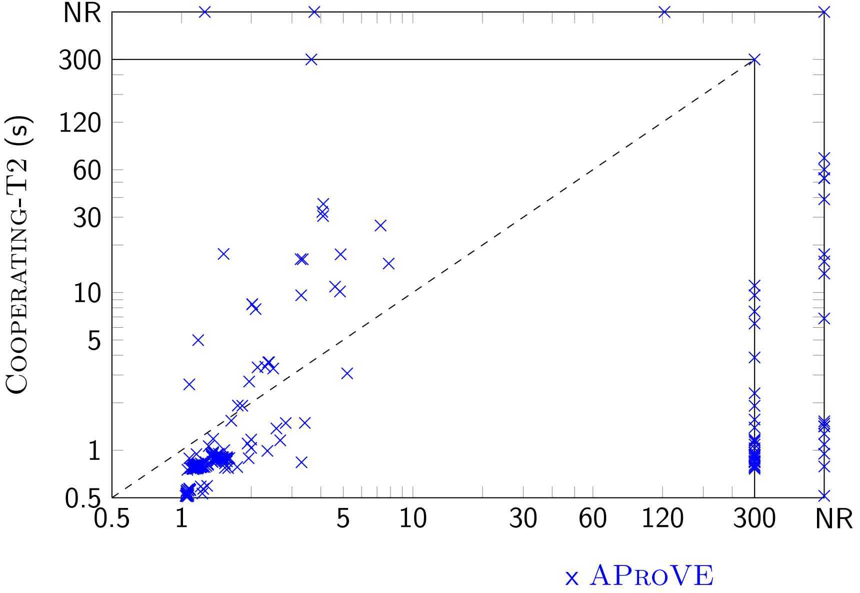
Cooperation: Evaluation



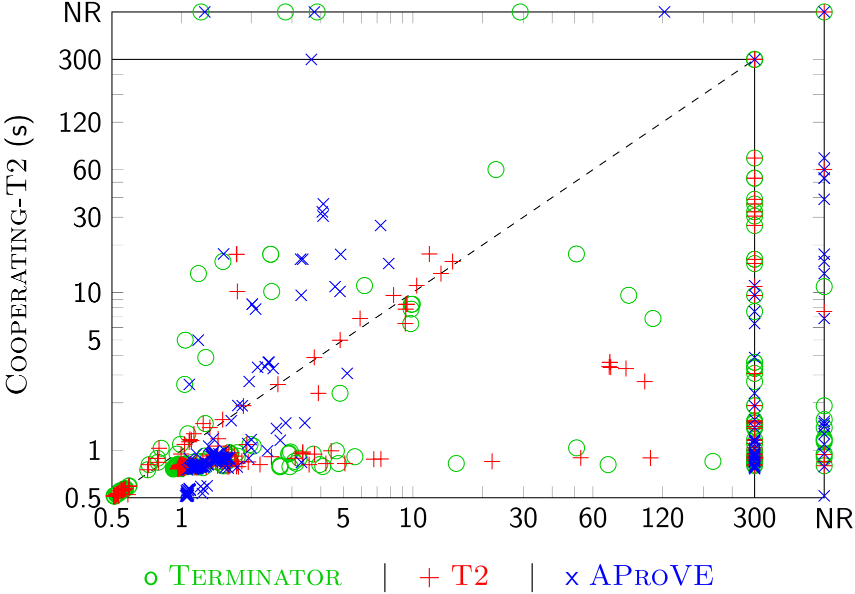
Cooperation: Evaluation



Cooperation: Evaluation



Cooperation: Evaluation



Cooperation: Evaluation

Evaluated on 449 termination proving benchmarks
260 known terminating, 181 known non-terminating, 8 unknown
Sources: Windows drivers, Apache, PostgreSQL, ...

	Term (#)	Term (avg. s)
Cooperating-T2	245	3.42
AProVE	197	2.21
KITTeL	196	4.65
T2	189	5.15
AProVE+Interproc	185	1.53
Terminator	177	4.99
Size-Change/MCNP	156	17.50
ARMC	138	16.16

Sources available: `http://research.microsoft.com/en-us/projects/t2/`