Ultimate GemCutter

Dominik Klumpp[1] Daniel Dietsch[1] Matthias Heizmann[1] Frank Schüssele[1]
Marcel Ebbinghaus[1] Azadeh Farzan[2] Andreas Podelski[1]

[1]University of Freiburg, Freiburg im Breisgau, Germany

[2]University of Toronto, Toronto, Canada

SV-COMP 2023

Example Program

$$\{\ x = y = i = j = 0\ \}$$

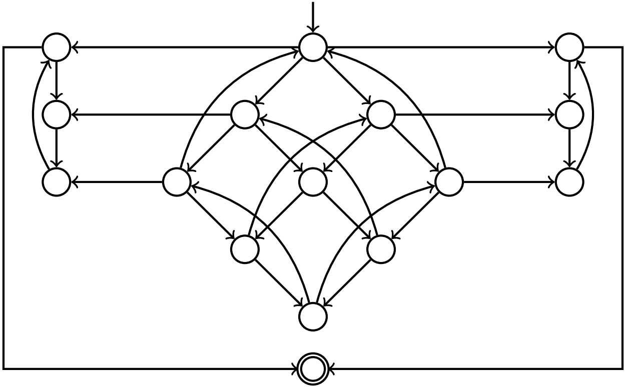
```
while (i < n) {
      x += A[i];
      i++;
}
```

$$\|$$

```
while (j < n) {
      y += A[j];
      j++;
}
```

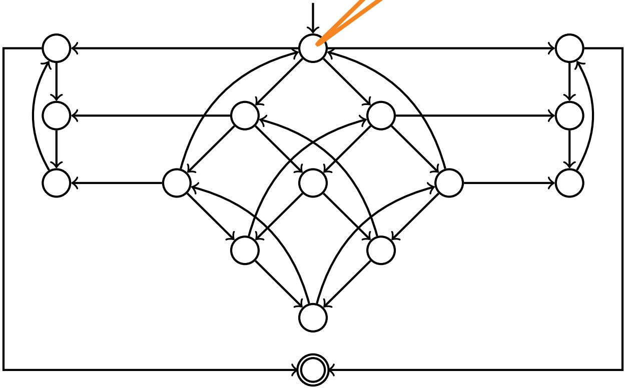
$$\{\ x = y\ \}$$

Key Idea

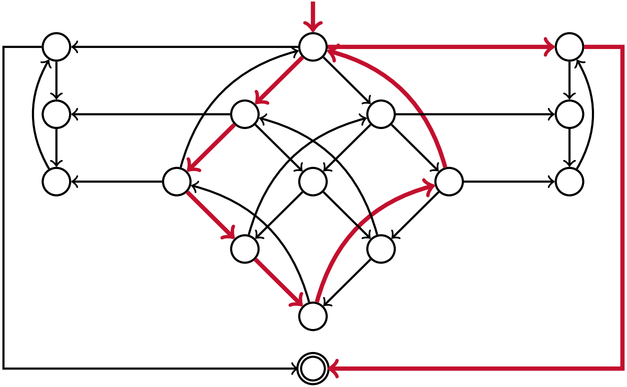


Key Idea

Invariant: $x = \sum_{k=0}^{i} A[k] \land y = \sum_{k=0}^{j} A[k] \land i \leq n \land j \leq n$



Key Idea



Counterexample:

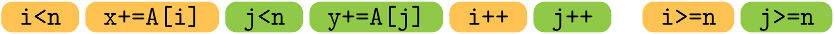
$\tau =$ `i<n` `x+=A[i]` `j<n` `y+=A[j]` `i++` `j++` `i>=n` `j>=n`

Generalization

`i<n` `x+=A[i]` `j<n` `y+=A[j]` `i++` `j++`   `i>=n` `j>=n`

Generalization

$$\{ \ x = y = i = j = 0 \ \}$$



```
i<n  x+=A[i]  j<n  y+=A[j]  i++  j++    i>=n  j>=n
```

$$\{ \ x = y \ \}$$

Generalization

generalization
across loop iterations
using interpolation

$$\{\ x = y = i = j = 0\ \}$$

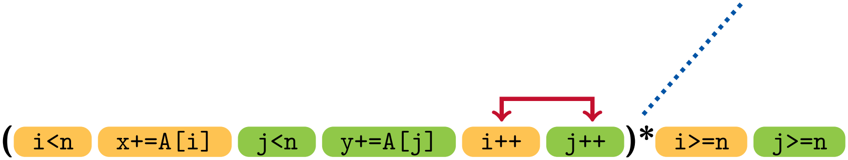
(`i<n` `x+=A[i]` `j<n` `y+=A[j]` `i++` `j++`)* `i>=n` `j>=n`

$$\{\ x = y\ \}$$

Generalization

generalization
across loop iterations
using interpolation

$$\{\ x = y = i = j = 0\ \}$$



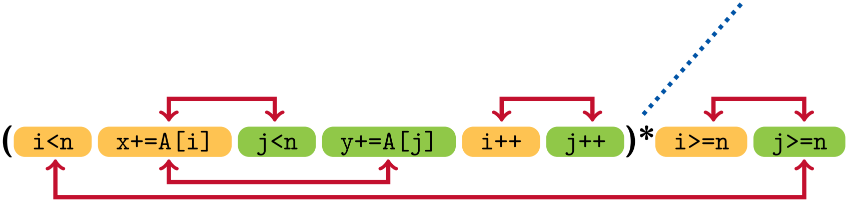
(i<n x+=A[i] j<n y+=A[j] i++ j++)* i>=n j>=n

$$\{\ x = y\ \}$$

Generalization

$$\{\ x = y = i = j = 0\ \}$$

generalization
across loop iterations
using interpolation



`( i<n x+=A[i] j<n y+=A[j] i++ j++ )* i>=n j>=n`

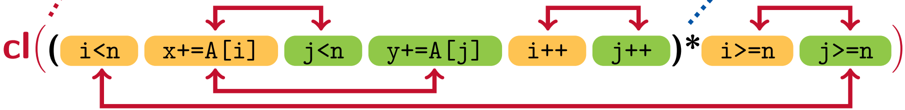
$$\{\ x = y\ \}$$

Generalization

generalization
across interleavings
using commutativity

generalization
across loop iterations
using interpolation

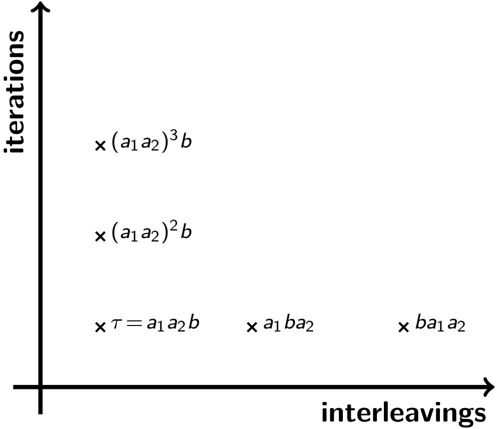
$$\{\ x = y = i = j = 0\ \}$$



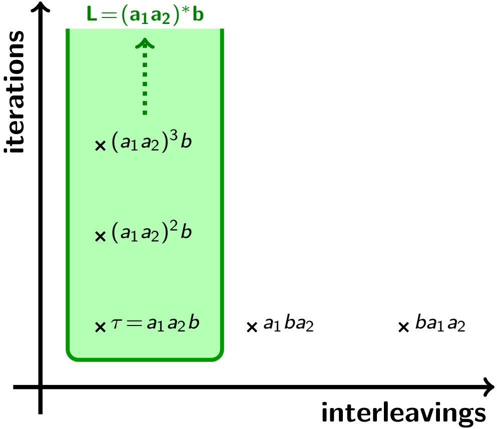
cl((`i<n` `x+=A[i]` `j<n` `y+=A[j]` `i++` `j++`)* `i>=n` `j>=n`)

$$\{\ x = y\ \}$$

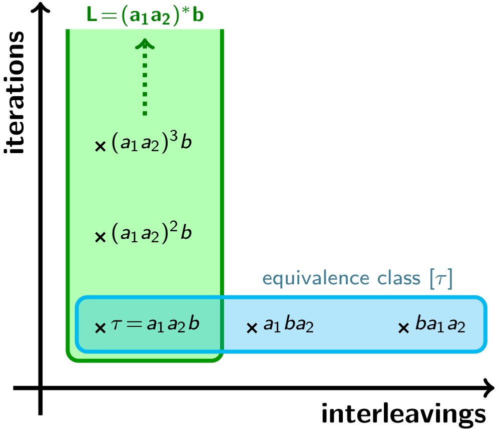
Generalization



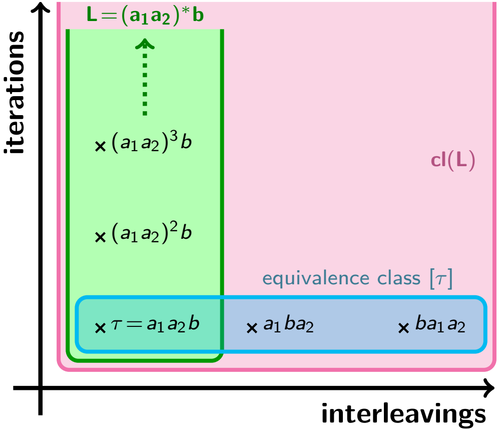
Generalization



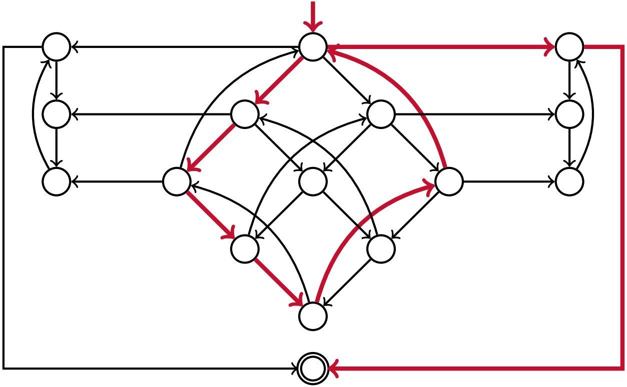
Generalization



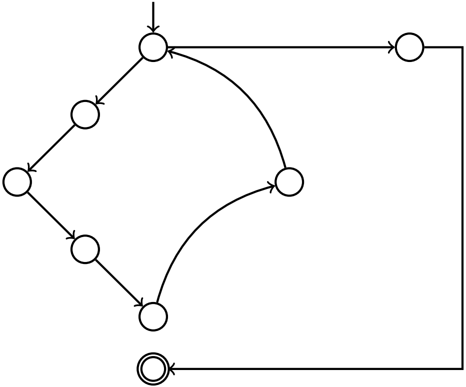
Generalization



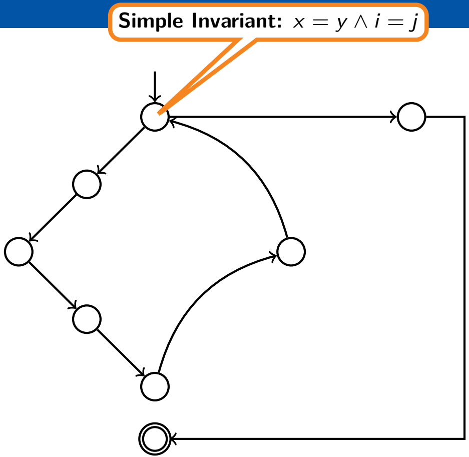
Key Idea



Key Idea

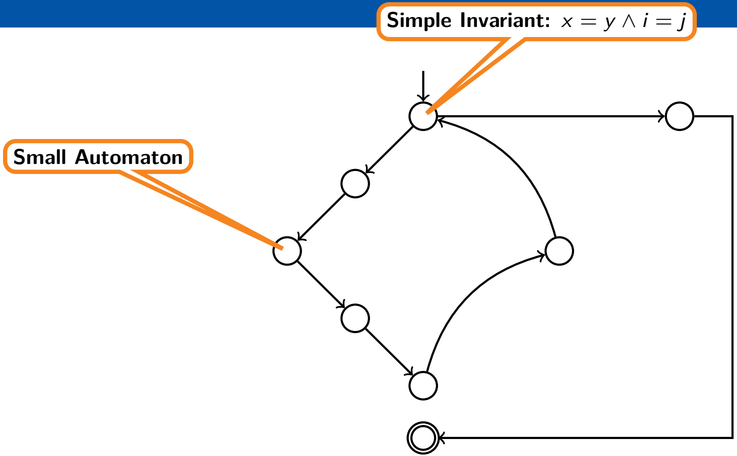


Benefits



Simple Invariant: $x = y \land i = j$

Benefits



Simple Invariant: $x = y \wedge i = j$

Small Automaton

Summary

- **CEGAR**-based verification of concurrent programs
- **Generalization** across interleavings via **commutativity**
- **Sound sequentialization** using **Partial Order Reduction** methods

$\Rightarrow$ **simple** proofs, **efficient** proof check

Summary

- **CEGAR**-based verification of concurrent programs
- **Generalization** across interleavings via **commutativity**
- **Sound sequentialization** using **Partial Order Reduction** methods

$\Rightarrow$ **simple** proofs, **efficient** proof check

Find out more: `ultimate-pa.org`

SV-COMP'22	Dominik Klumpp et al. *Ultimate GemCutter and the Axes of Generalization (Competition Contribution)*.
PLDI'22	Azadeh Farzan, Dominik Klumpp and Andreas Podelski. *Sound Sequentialization for Concurrent Program Verification*.
POPL'23	Azadeh Farzan, Dominik Klumpp and Andreas Podelski. *Stratified Commutativity in Verification Algorithms for Concurrent Programs*.