

# Inferring Invariants by Static Analysis in KeY

Benjamin Weiß

June 14, 2007

# Goal

Automatically find **invariants**:

- (First-order) formulas which always hold at specific points in the program code
- In particular: Loop invariants
- Can be used for verification with KeY

# Goal

Automatically find **invariants**:

- (First-order) formulas which always hold at specific points in the program code
- In particular: Loop invariants
- Can be used for verification with KeY

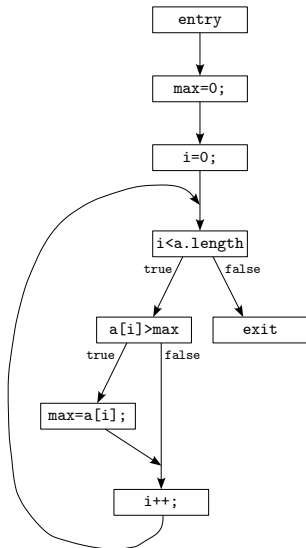
Do so **within KeY** itself.

## Example Program

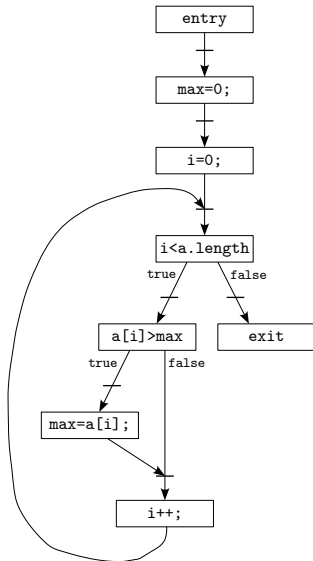
```
max = 0;
i = 0;
while(i < a.length) {
    if(a[i] > max)
        max = a[i];
    i++;
}
```

## Example Program

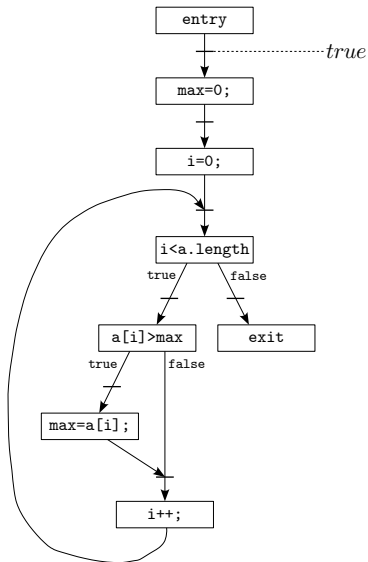
```
max = 0;  
i = 0;  
while(i < a.length) {  
    if(a[i] > max)  
        max = a[i];  
    i++;  
}
```



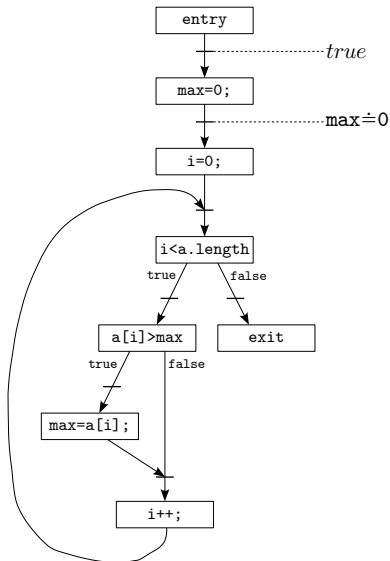
## Example Program: Invariants



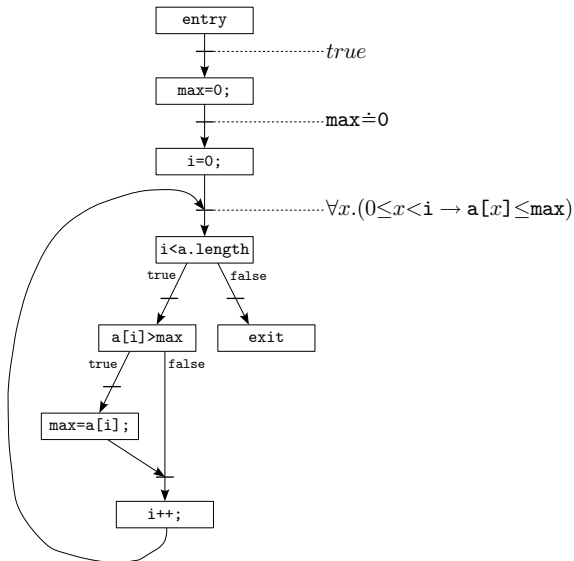
# Example Program: Invariants



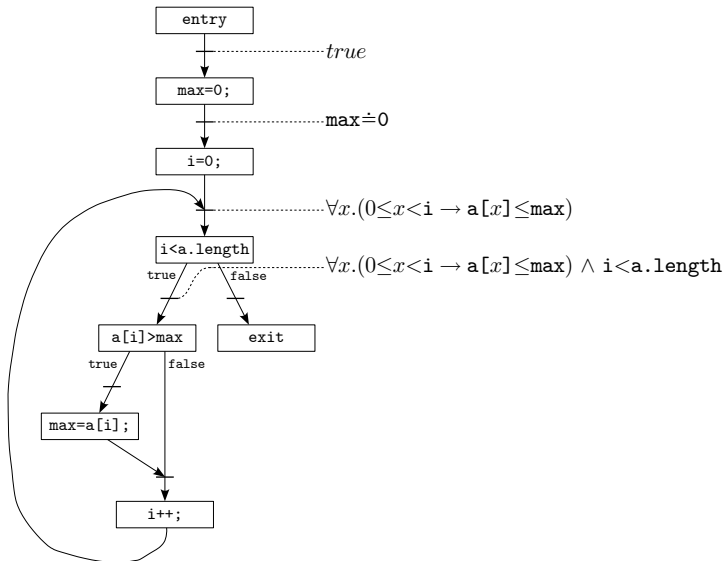
## Example Program: Invariants



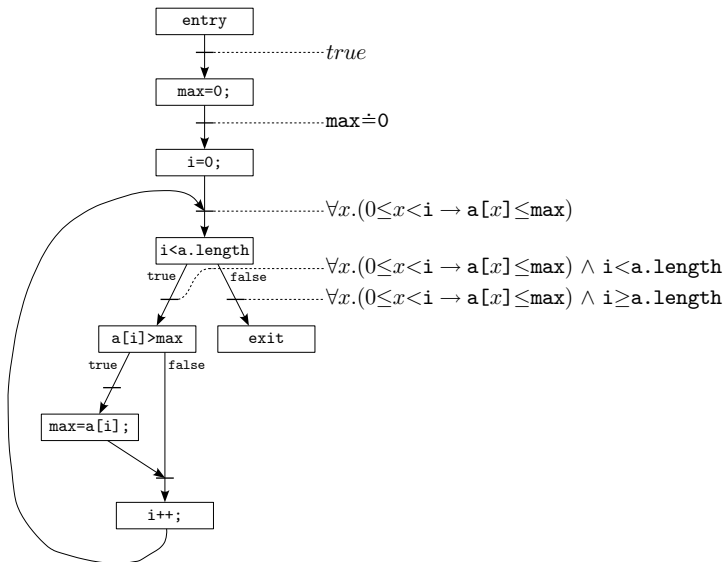
## Example Program: Invariants



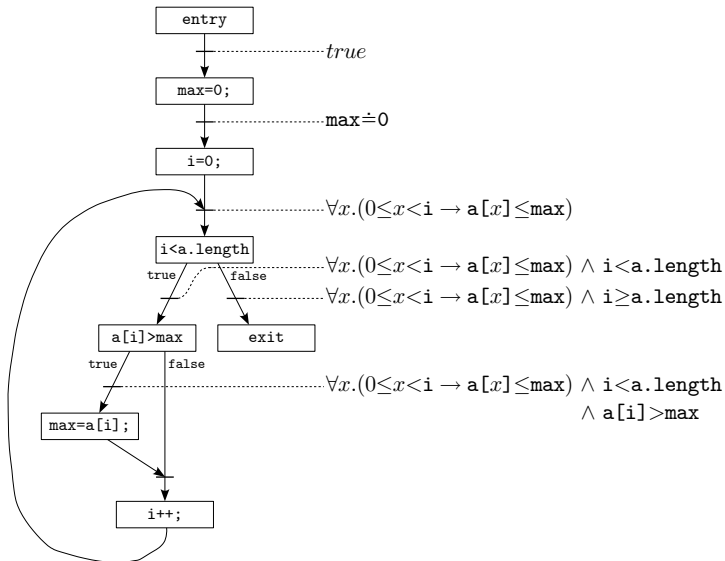
# Example Program: Invariants



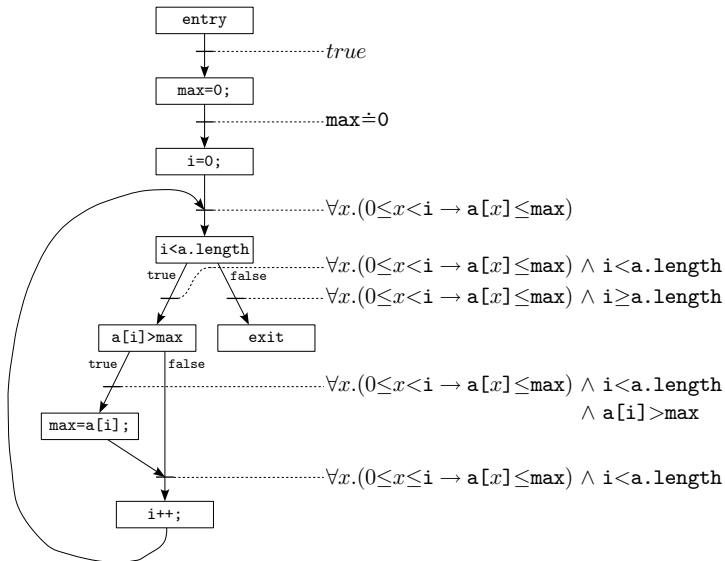
# Example Program: Invariants



# Example Program: Invariants



# Example Program: Invariants



# Dynamic and Static Program Analysis

## Dynamic Analysis:

An analysis which includes executing the program on concrete input values. (e.g. testing, Daikon)

# Dynamic and Static Program Analysis

## Dynamic Analysis:

An analysis which includes executing the program on concrete input values. (e.g. testing, Daikon)

## Static Analysis:

An analysis which does *not* actually execute the program. (e.g. KeY, abstract interpretation)

# KeY $\leftrightarrow$ Abstract Interpretation

Both can be seen as symbolic execution.

# KeY $\leftrightarrow$ Abstract Interpretation

Both can be seen as symbolic execution.

Differences:

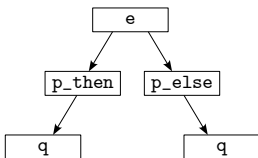
- Updates

# KeY $\leftrightarrow$ Abstract Interpretation

Both can be seen as symbolic execution.

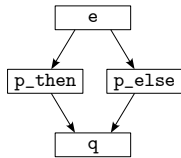
Differences:

- Updates
- `if(e) p_then else p_else; q`



KeY

vs.



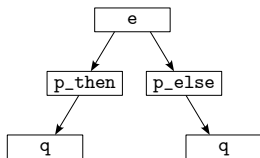
abstract interpretation

# KeY $\leftrightarrow$ Abstract Interpretation

Both can be seen as symbolic execution.

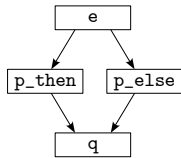
Differences:

- Updates
- `if(e) p_then else p_else; q`



KeY

vs.



abstract interpretation

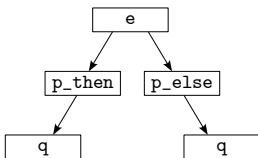
- Approximation

# KeY $\leftrightarrow$ Abstract Interpretation

Both can be seen as symbolic execution.

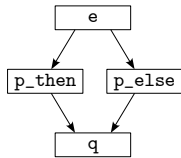
Differences:

- Updates
- `if(e) p_then else p_else; q`



KeY

vs.



abstract interpretation

- Approximation
- Fixed-point iteration

## Invariant Inference in KeY

```
true → [ max = 0;  
          i = 0;  
          while(i < a.length) {...}  
        ] $\psi$ 
```

## Invariant Inference in KeY

```
true → [  max = 0;  
          i = 0;  
          while(i < a.length) {...}  
          ] $\psi$ 
```

## Invariant Inference in KeY

```
max  $\doteq$  0  $\rightarrow$  [ i = 0;  
                  while(i < a.length) {...}  
                  ] $\psi$ 
```

## Invariant Inference in KeY

```
max  $\doteq$  0  $\rightarrow$  [ i = 0;  
                  while(i < a.length) {...}  
                  ] $\psi$ 
```

## Invariant Inference in KeY

$$\text{max} \doteq 0 \wedge i \doteq 0 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$

## Invariant Inference in KeY

$$\varphi_1 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$

# Invariant Inference in KeY

$$\varphi_1 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$

## Invariant Inference in KeY

$$\varphi_1 \wedge i < a.length \rightarrow [ \text{if}(a[i] > \text{max}) \text{max} = a[i];$$
$$i++;$$
$$\text{while}(i < a.length) \{...\}$$
$$]\psi$$
$$\wedge$$
$$\varphi_1 \wedge i \geq a.length \rightarrow []\psi$$

## Invariant Inference in KeY

$$\varphi_1 \wedge i < a.length \rightarrow [ \text{if}(a[i] > \text{max}) \text{max} = a[i];$$
$$i++;$$
$$\text{while}(i < a.length) \{ \dots \}$$
$$] \psi$$
$$\wedge$$
$$\varphi_1 \wedge i \geq a.length \rightarrow [] \psi$$

## Invariant Inference in KeY

$$\begin{array}{l} \varphi_2 \rightarrow [ \text{max} = \text{a}[i]; \\ \quad \text{i}++; \\ \quad \text{while}(\text{i} < \text{a.length}) \{ \dots \} \\ \quad ]\psi \\ \wedge \\ \varphi_3 \rightarrow [ \text{i}++; \\ \quad \text{while}(\text{i} < \text{a.length}) \{ \dots \} \\ \quad ]\psi \\ \wedge \\ \varphi_1 \wedge \text{i} \geq \text{a.length} \rightarrow []\psi \end{array}$$

## Invariant Inference in KeY

$$\begin{array}{l} \varphi_2 \rightarrow [ \text{max} = \text{a}[i]; \\ \quad \text{i}++; \\ \quad \text{while}(\text{i} < \text{a.length}) \{ \dots \} \\ \quad ]\psi \\ \wedge \\ \varphi_3 \rightarrow [ \text{i}++; \\ \quad \text{while}(\text{i} < \text{a.length}) \{ \dots \} \\ \quad ]\psi \\ \wedge \\ \varphi_1 \wedge \text{i} \geq \text{a.length} \rightarrow []\psi \end{array}$$

## Invariant Inference in KeY

$$\begin{array}{l} \varphi_4 \rightarrow [ \quad i++; \\ \quad \quad \text{while}(i < \text{a.length}) \{ \dots \} \\ \quad \quad ]\psi \\ \wedge \\ \varphi_3 \rightarrow [ \quad i++; \\ \quad \quad \text{while}(i < \text{a.length}) \{ \dots \} \\ \quad \quad ]\psi \\ \wedge \\ \varphi_1 \wedge i \geq \text{a.length} \rightarrow []\psi \end{array}$$

# Invariant Inference in KeY

$$\varphi_4 \rightarrow [ \quad i++;$$
$$\quad \text{while}(i < \text{a.length}) \{ \dots \}$$
$$\quad ]\psi$$
$$\wedge$$
$$\varphi_3 \rightarrow [ \quad i++;$$
$$\quad \text{while}(i < \text{a.length}) \{ \dots \}$$
$$\quad ]\psi$$
$$\wedge$$
$$\varphi_1 \wedge i \geq \text{a.length} \rightarrow []\psi$$

## Invariant Inference in KeY

$$\varphi_4 \vee \varphi_3 \rightarrow [ \quad i++;$$
$$\quad \text{while}(i < a.\text{length}) \{ \dots \}$$
$$\quad ]\psi$$
$$\wedge$$
$$\varphi_1 \wedge i \geq a.\text{length} \rightarrow []\psi$$

## Invariant Inference in KeY

$$\varphi_4 \vee \varphi_3 \rightarrow [ \text{ i++;} \\ \text{ while(i < a.length) \{...\}} \\ ]\psi$$
$$\wedge$$
$$\varphi_1 \wedge i \geq \text{a.length} \rightarrow []\psi$$

## Invariant Inference in KeY

$$\varphi_5 \rightarrow [ \text{while}(i < \text{a.length}) \{ \dots \} ] \psi$$
$$\wedge$$
$$\varphi_1 \wedge i \geq \text{a.length} \rightarrow [] \psi$$

# Invariant Inference in KeY

$$\varphi_5 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$
$$\wedge$$
$$\varphi_1 \wedge i \geq a.\text{length} \rightarrow [] \psi$$

## Invariant Inference in KeY

$$\varphi_1 \vee \varphi_5 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$

# Invariant Inference in KeY

$$\varphi_1 \vee \varphi_5 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$

## Predicate abstraction

$$\text{abstract}(\varphi) = \bigwedge \{ p \in P \mid \varphi \rightarrow p \text{ is valid} \}$$

# Invariant Inference in KeY

$$\varphi_1 \vee \varphi_5 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$

## Predicate abstraction

$$\text{abstract}(\varphi) = \bigwedge \{ p \in P \mid \varphi \rightarrow p \text{ is valid} \}$$

$$P = \{ \underbrace{i \doteq 0}_{p_1}, \underbrace{0 \leq i}_{p_2}, \underbrace{i \leq 1}_{p_3}, \underbrace{\forall x. (0 \leq x < i \rightarrow a[x] \leq \text{max})}_{p_4} \}$$

# Invariant Inference in KeY

$$\varphi_1 \vee \varphi_5 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$

## Predicate abstraction

$$\text{abstract}(\varphi) = \bigwedge \{ p \in P \mid \varphi \rightarrow p \text{ is valid} \}$$

$$P = \{ \underbrace{i \doteq 0}_{p_1}, \underbrace{0 \leq i}_{p_2}, \underbrace{i \leq 1}_{p_3}, \underbrace{\forall x. (0 \leq x < i \rightarrow a[x] \leq \text{max})}_{p_4} \}$$

# Invariant Inference in KeY

$$p_2 \wedge p_3 \wedge p_4 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$

## Predicate abstraction

$$\text{abstract}(\varphi) = \bigwedge \{ p \in P \mid \varphi \rightarrow p \text{ is valid} \}$$

$$P = \{ \underbrace{i \doteq 0}_{p_1}, \underbrace{0 \leq i}_{p_2}, \underbrace{i \leq 1}_{p_3}, \underbrace{\forall x. (0 \leq x < i \rightarrow a[x] \leq \text{max})}_{p_4} \}$$

# Invariant Inference in KeY

$$p_2 \wedge p_3 \wedge p_4 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$

## Predicate abstraction

$$\text{abstract}(\varphi) = \bigwedge \{ p \in P \mid \varphi \rightarrow p \text{ is valid} \}$$

$$P = \{ \underbrace{i \doteq 0}_{p_1}, \underbrace{0 \leq i}_{p_2}, \underbrace{i \leq 1}_{p_3}, \underbrace{\forall x. (0 \leq x < i \rightarrow a[x] \leq \text{max})}_{p_4} \}$$

# Invariant Inference in KeY

$$p_2 \wedge p_4 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$

## Predicate abstraction

$$\text{abstract}(\varphi) = \bigwedge \{ p \in P \mid \varphi \rightarrow p \text{ is valid} \}$$

$$P = \{ \underbrace{i \doteq 0}_{p_1}, \underbrace{0 \leq i}_{p_2}, \underbrace{i \leq 1}_{p_3}, \underbrace{\forall x. (0 \leq x < i \rightarrow a[x] \leq \text{max})}_{p_4} \}$$

# Invariant Inference in KeY

$$p_2 \wedge p_4 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$

## Predicate abstraction

$$\text{abstract}(\varphi) = \bigwedge \{ p \in P \mid \varphi \rightarrow p \text{ is valid} \}$$

$$P = \{ \underbrace{i \doteq 0}_{p_1}, \underbrace{0 \leq i}_{p_2}, \underbrace{i \leq 1}_{p_3}, \underbrace{\forall x. (0 \leq x < i \rightarrow a[x] \leq \text{max})}_{p_4} \}$$

# Invariant Inference in KeY

$$p_2 \wedge p_4 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$

## Predicate abstraction

$$\text{abstract}(\varphi) = \bigwedge \{ p \in P \mid \varphi \rightarrow p \text{ is valid} \}$$

$$P = \{ \underbrace{i \doteq 0}_{p_1}, \underbrace{0 \leq i}_{p_2}, \underbrace{i \leq 1}_{p_3}, \underbrace{\forall x. (0 \leq x < i \rightarrow a[x] \leq \text{max})}_{p_4} \}$$

# Invariant Inference in KeY

$$p_2 \wedge p_4 \rightarrow [ \text{while}(i < a.\text{length}) \{ \dots \} ] \psi$$

## Predicate abstraction

$$\text{abstract}(\varphi) = \bigwedge \{ p \in P \mid \varphi \rightarrow p \text{ is valid} \}$$

$$P = \{ \underbrace{i \doteq 0}_{p_1}, \underbrace{0 \leq i}_{p_2}, \underbrace{i \leq 1}_{p_3}, \underbrace{\forall x. (0 \leq x < i \rightarrow a[x] \leq \text{max})}_{p_4} \}$$

# Invariant Inference in KeY

$$p_2 \wedge p_4 \wedge i \geq a.length \rightarrow \llbracket \psi \rrbracket$$

## Predicate abstraction

$$abstract(\varphi) = \bigwedge \{p \in P \mid \varphi \rightarrow p \text{ is valid}\}$$

$$P = \left\{ \underbrace{i \doteq 0}_{p_1}, \underbrace{0 \leq i}_{p_2}, \underbrace{i \leq 1}_{p_3}, \underbrace{\forall x. (0 \leq x < i \rightarrow a[x] \leq \max)}_{p_4} \right\}$$

# Implementation

- Rules
  - Handling updates
  - Merging of control flow paths
  - Predicate abstraction (via Simplify / SMT)
  - Ending loop iteration when fixed-point reached

# Implementation

- Rules
  - Handling updates
  - Merging of control flow paths
  - Predicate abstraction (via Simplify / SMT)
  - Ending loop iteration when fixed-point reached
- Heuristic predicate generator

# Implementation

- Rules
  - Handling updates
  - Merging of control flow paths
  - Predicate abstraction (via Simplify / SMT)
  - Ending loop iteration when fixed-point reached
- Heuristic predicate generator
- Proof search strategy

# Experiments

- Array maximum:
  - 260 predicates
  - 60 x Simplify
  - 10 s

# Experiments

- Array maximum:
  - 260 predicates
  - 60 × Simplify
  - 10 s
- Selection sort:
  - 17000 predicates
  - 800 × Simplify
  - 10 min

# Experiments

- Array maximum:
  - 260 predicates
  - 60 × Simplify
  - 10 s
- Selection sort:
  - 17000 predicates
  - 800 × Simplify
  - 10 min
- In both cases:
  - A few predicates were added by hand ( $\rightsquigarrow$  quantified invariants)
  - Inferred invariants strong enough for verification

# Summary & Future Work

## Summary

- Invariant inference using KeY's symbolic execution framework
- Limitations: Available predicates, capabilities of Simplify & Co
- First experiments promising

# Summary & Future Work

## Summary

- Invariant inference using KeY's symbolic execution framework
- Limitations: Available predicates, capabilities of Simplify & Co
- First experiments promising

## Future work

- More experiments
- Improve implementation
- Establish soundness of all of the new rules
- Incorporate “counterexample-guided abstraction refinement” techniques (?)