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

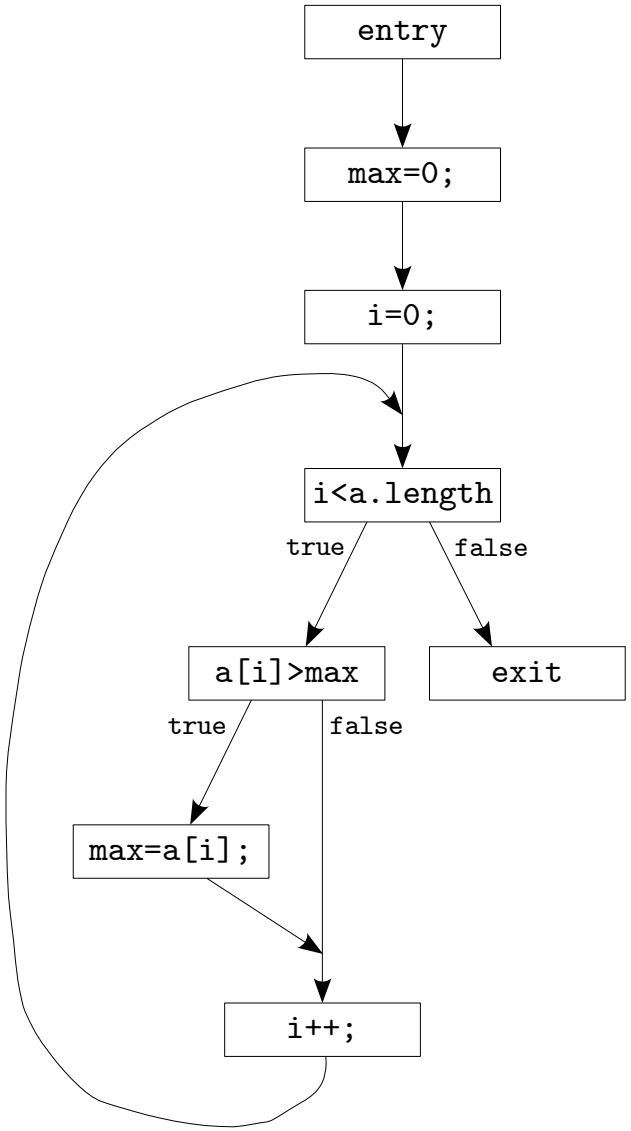
Do so within KeY itself.

Example Program

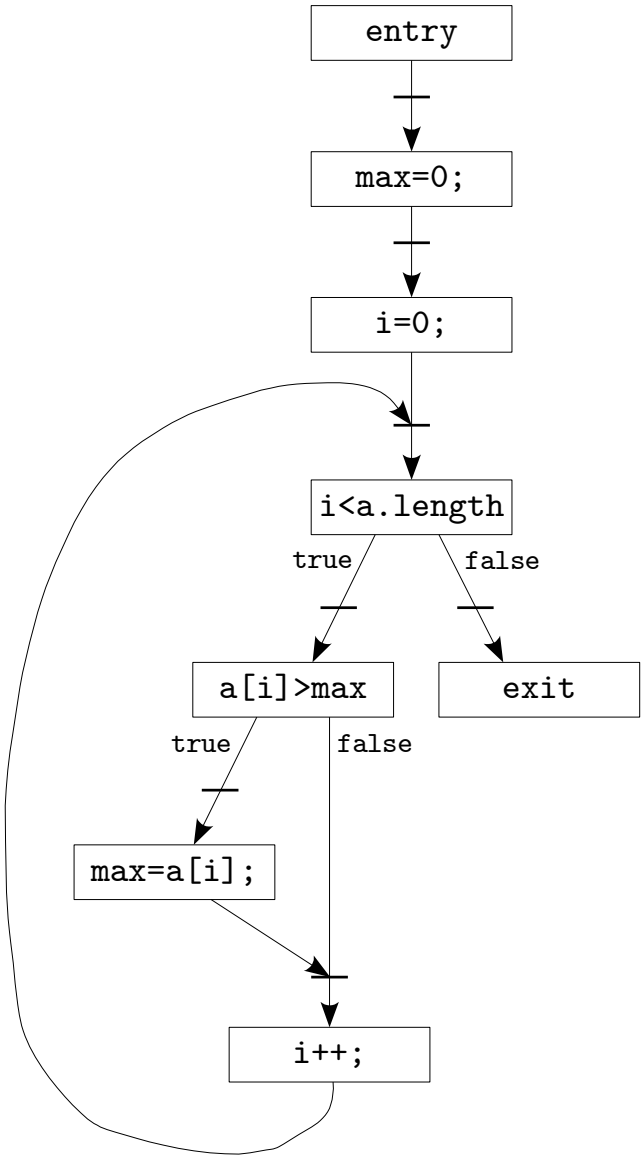
```
max = 0;
i = 0;
while(i < a.length) {
  if(a[i] > max)
    max = a[i];
  i++;
}
```

Example Program

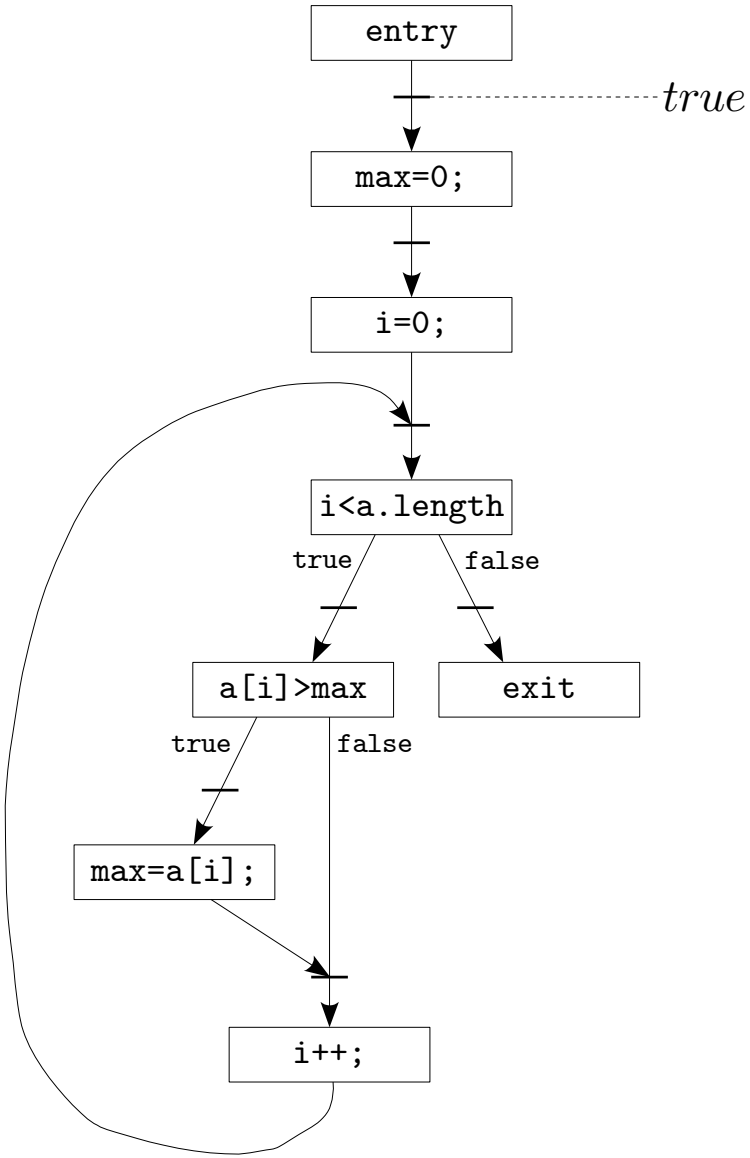
```
max = 0;
i = 0;
while(i < a.length) {
  if(a[i] > max)
    max = a[i];
  i++;
}
```



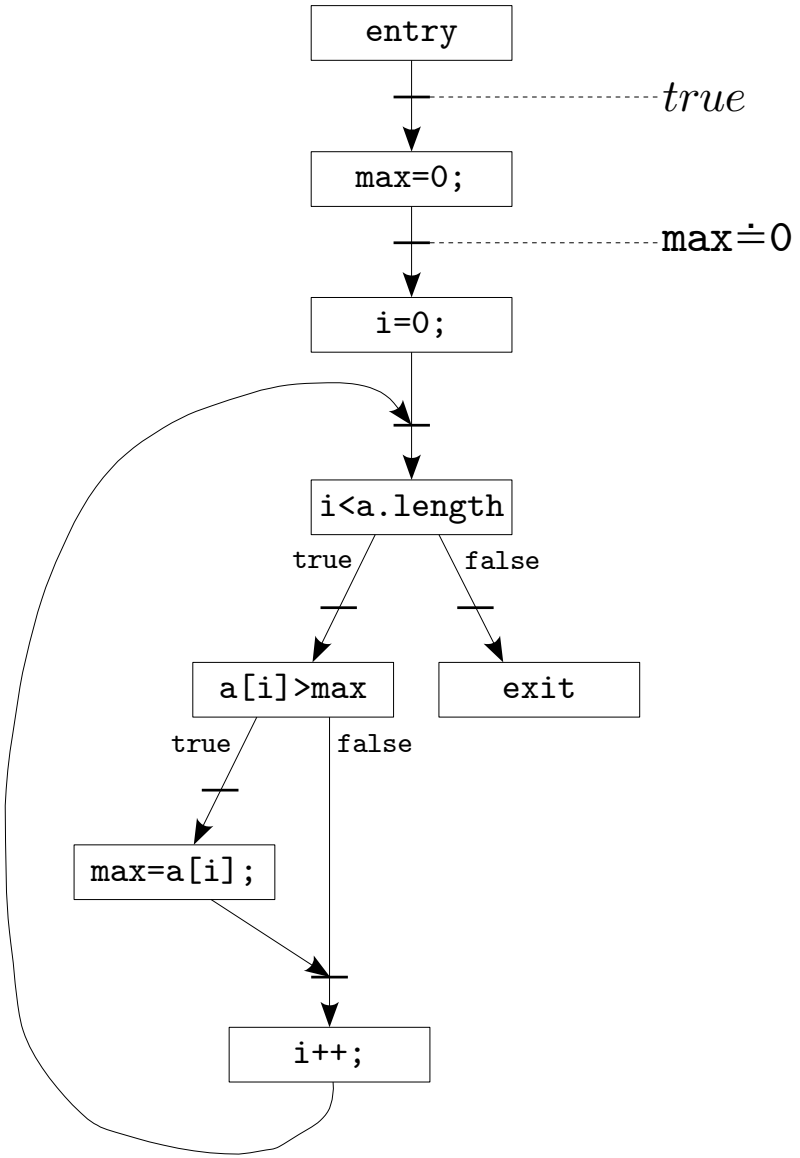
Example Program: Invariants



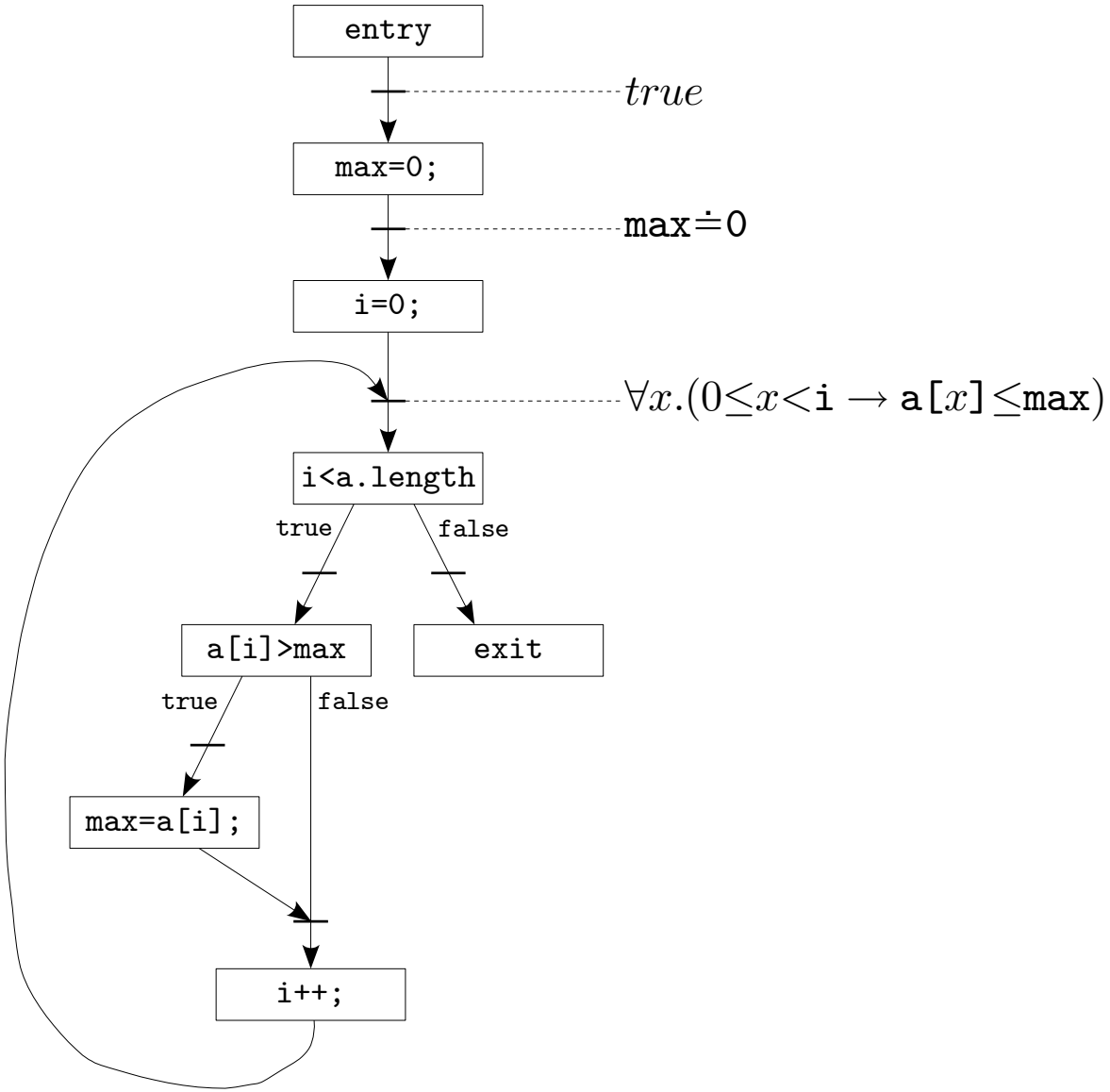
Example Program: Invariants



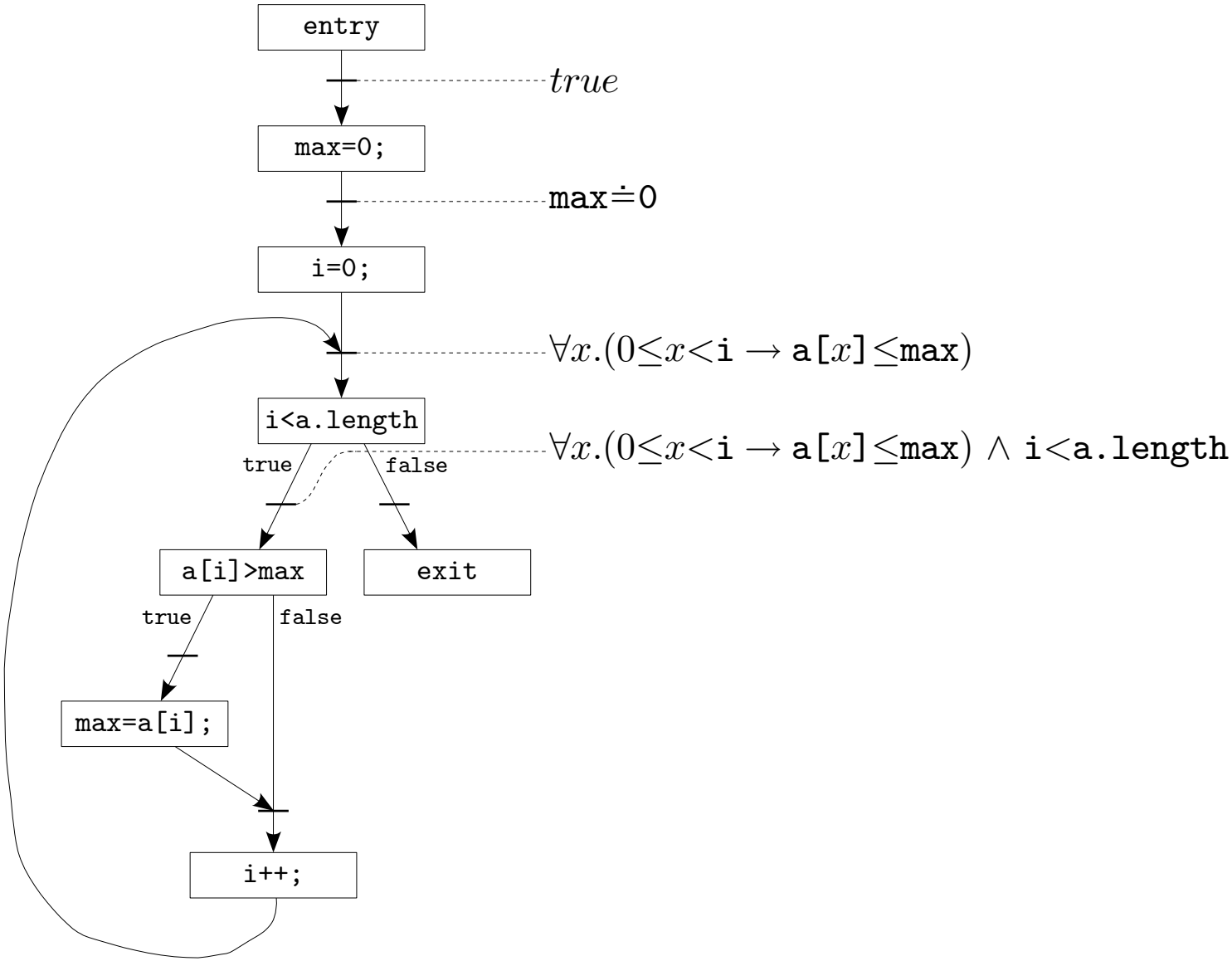
Example Program: Invariants



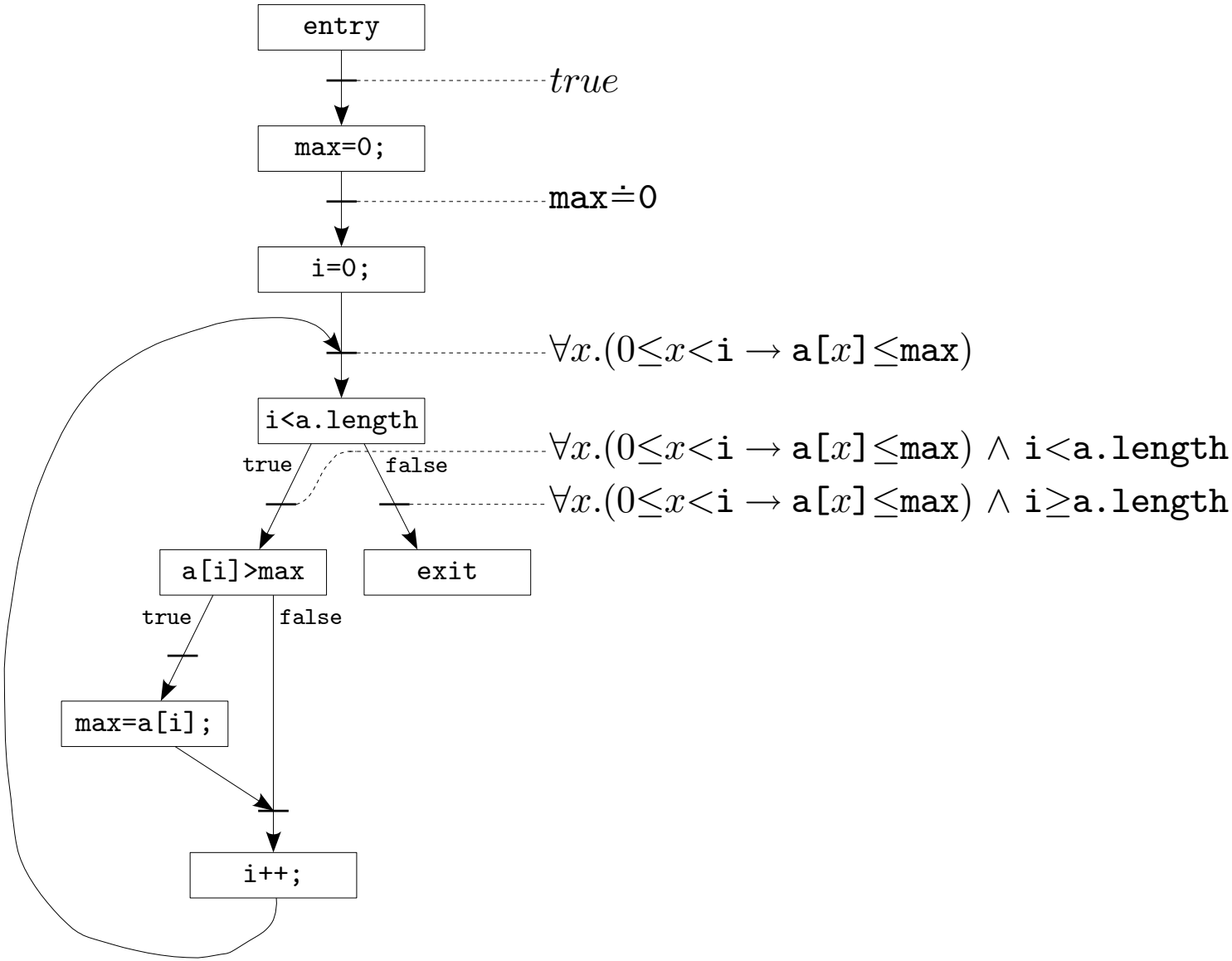
Example Program: Invariants



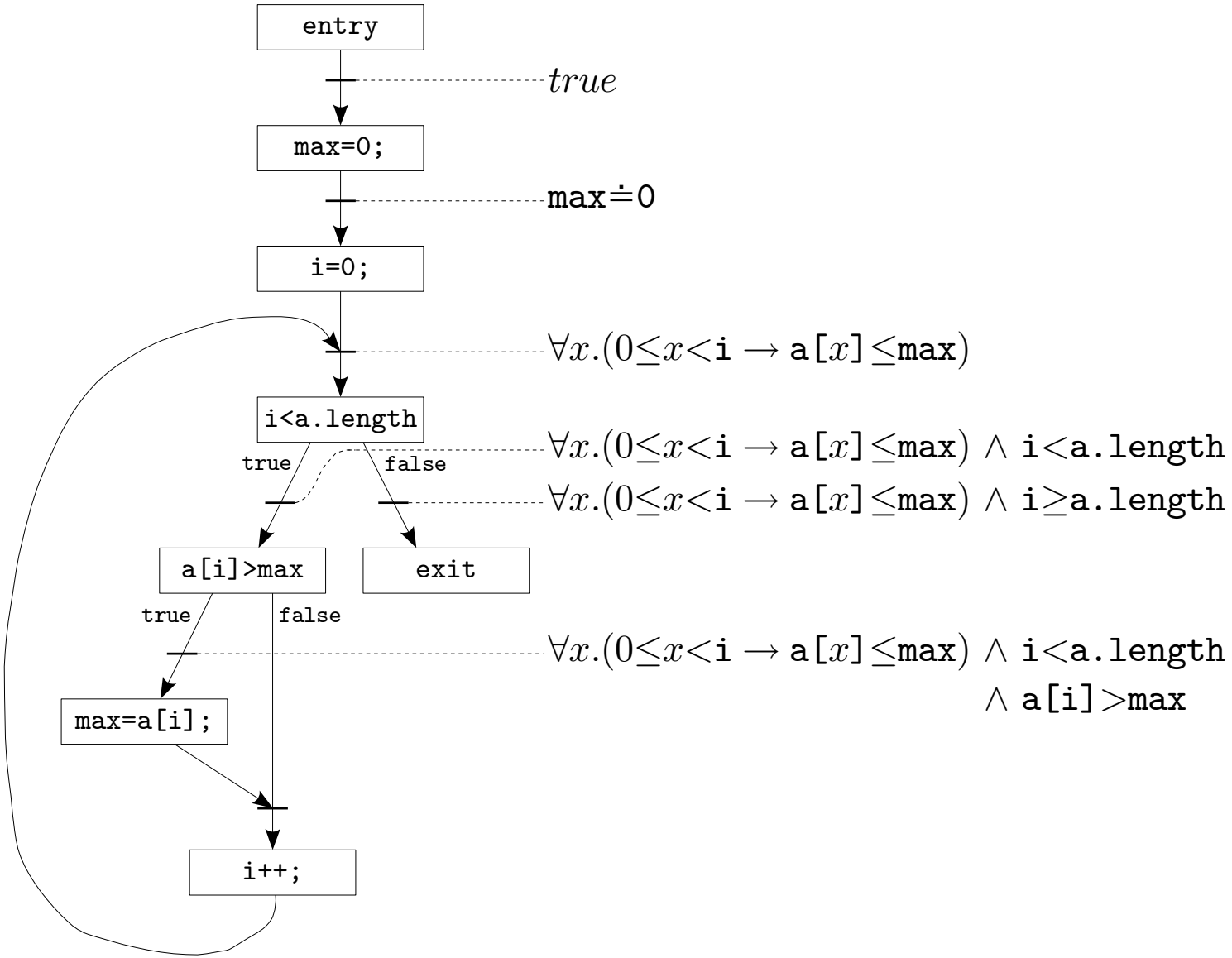
Example Program: Invariants



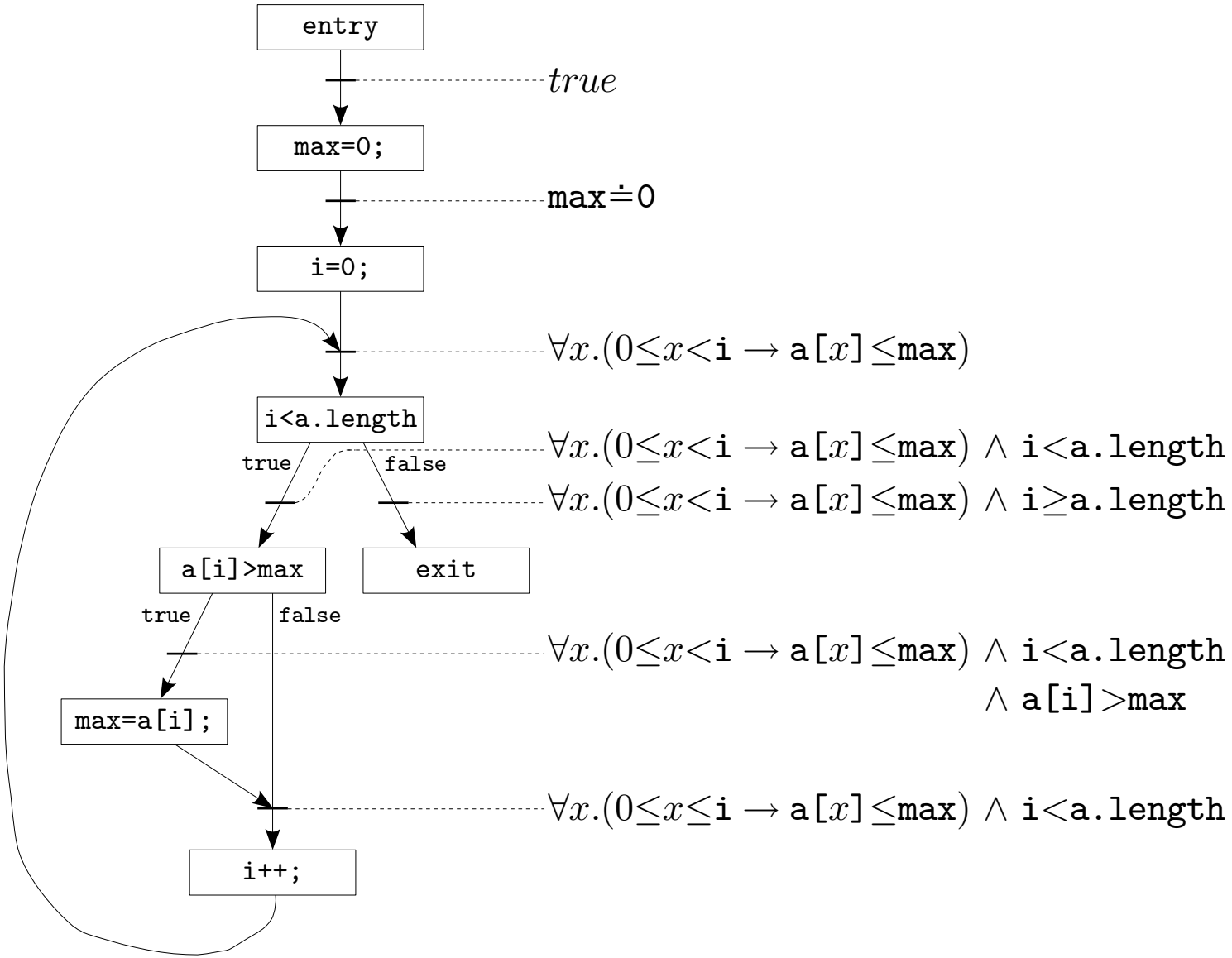
Example Program: Invariants



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:

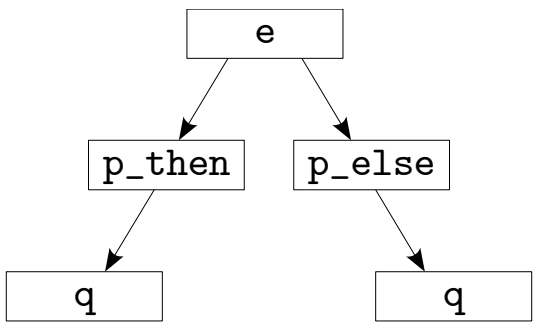
- Updates

KeY $\leftrightarrow$ Abstract Interpretation

Both can be seen as symbolic execution.

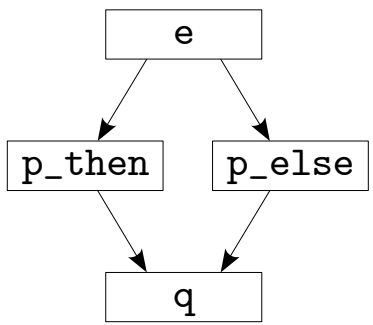
Differences:

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



KeY

vs.



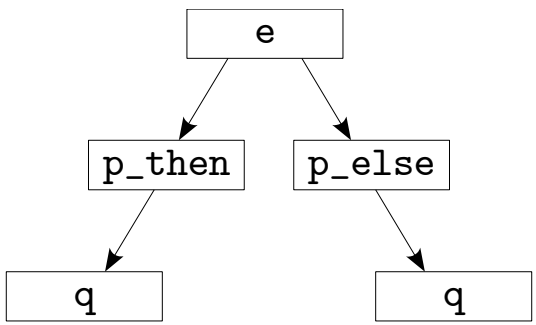
abstract interpretation

KeY $\leftrightarrow$ Abstract Interpretation

Both can be seen as symbolic execution.

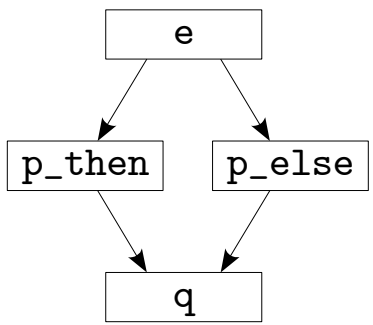
Differences:

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



KeY

vs.



abstract interpretation

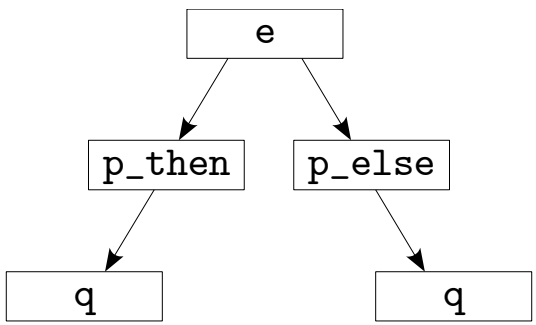
- Approximation

KeY $\leftrightarrow$ Abstract Interpretation

Both can be seen as symbolic execution.

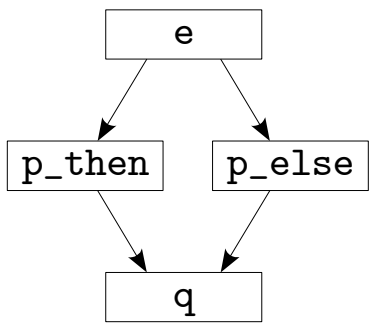
Differences:

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



KeY

vs.



abstract interpretation

- Approximation
- Fixed-point iteration

Invariant Inference in KeY

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

Invariant Inference in KeY

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

Invariant Inference in KeY

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

Invariant Inference in KeY

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

Invariant Inference in KeY

$$\texttt{max} \doteq 0 \land \texttt{i} \doteq 0 \rightarrow [\ \ \texttt{while(i < a.length) \{...\}}\ ]\psi$$

Invariant Inference in KeY

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

Invariant Inference in KeY

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

Invariant Inference in KeY

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

$$\wedge$$

$$\varphi_1 \wedge \texttt{i} \geq \texttt{a.length} \rightarrow []\psi$$

Invariant Inference in KeY

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

$$\wedge$$

$$\varphi_1 \wedge \texttt{i} \geq \texttt{a.length} \rightarrow []\psi$$

Invariant Inference in KeY

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

Invariant Inference in KeY

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

Invariant Inference in KeY

$$\varphi_4 \rightarrow [\ \texttt{i++;}\ \texttt{while(i < a.length) \{...\}}\ ]\psi$$

$\wedge$

$$\varphi_3 \rightarrow [\ \texttt{i++;}\ \texttt{while(i < a.length) \{...\}}\ ]\psi$$

$\wedge$

$$\varphi_1 \wedge \texttt{i} \geq \texttt{a.length} \rightarrow []\psi$$

Invariant Inference in KeY

$$\varphi_4 \rightarrow [\ \texttt{i++; while(i < a.length) \{...\}}\ ]\psi$$

$$\wedge$$

$$\varphi_3 \rightarrow [\ \texttt{i++; while(i < a.length) \{...\}}\ ]\psi$$

$$\wedge$$

$$\varphi_1 \wedge \texttt{i} \geq \texttt{a.length} \rightarrow []\psi$$

Invariant Inference in KeY

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

$$\wedge$$

$$\varphi_1 \wedge \texttt{i} \geq \texttt{a.length} \rightarrow []\psi$$

Invariant Inference in KeY

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

$$\wedge$$

$$\varphi_1 \wedge \texttt{i} \geq \texttt{a.length} \rightarrow []\psi$$

Invariant Inference in KeY

$$\varphi_5 \rightarrow [\ \texttt{while(i < a.length) \{...\}}\ ]\psi$$

$$\wedge$$

$$\varphi_1 \wedge \texttt{i} \geq \texttt{a.length} \rightarrow []\psi$$

Invariant Inference in KeY

$$\varphi_5 \rightarrow [\ \texttt{while(i < a.length) \{...\}}\ ]\psi$$

$$\wedge$$

$$\varphi_1 \wedge \texttt{i} \geq \texttt{a.length} \rightarrow [\,]\psi$$

Invariant Inference in KeY

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

Invariant Inference in KeY

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

Predicate abstraction

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

Invariant Inference in KeY

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

Predicate abstraction

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

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

Invariant Inference in KeY

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

Predicate abstraction

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

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

Invariant Inference in KeY

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

Predicate abstraction

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

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

Invariant Inference in KeY

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

Predicate abstraction

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

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

Invariant Inference in KeY

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

Predicate abstraction

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

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

Invariant Inference in KeY

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

Predicate abstraction

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

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

Invariant Inference in KeY

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

Predicate abstraction

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

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

Invariant Inference in KeY

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

Predicate abstraction

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

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

Invariant Inference in KeY

$$p_2 \wedge p_4 \wedge \mathtt{i} \geq \mathtt{a.length} \rightarrow [\,]\psi$$

Predicate abstraction

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

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

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 x Simplify
  - 10 s
- Selection sort:
  - 17000 predicates
  - 800 x Simplify
  - 10 min

Experiments

- Array maximum:
  - 260 predicates
  - 60 x Simplify
  - 10 s
- Selection sort:
  - 17000 predicates
  - 800 x 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 (?)