Static Program Analysis

Lecture 16: Abstract Interpretation VI
(Counterexample-Guided Abstraction Refinement)

Winter Semester 2016/17

Thomas Noll
Software Modeling and Verification Group
RWTH Aachen University

https://moves.rwth-aachen.de/teaching/ws-1617/spa/
Nacht der Professoren präsentiert von:

StudierenOhneGrenzen

am 13.01.2017
ab 22:30 Uhr – im Apollo

Line-Up

- DJ Prof. Katoen
- Informatik
- DJ Prof. Fischer
- Biomaterialforschung
- DJ Prof. Schröder
- Strukturmechanik & Leichtbau
- DJ Prof. Böhm
- Wirtschaftswissenschaften
- DJ Prof. Richter
- Politikwissenschaft
- DJ AIXTRA-Beats
- Medizin

EST 2005
Recap: Predicate Abstraction

Outline of Lecture 16

Recap: Predicate Abstraction

Computation of Postconditions

Counterexample-Guided Abstraction Refinement
Recap: Predicate Abstraction

Counterexample-Guided Abstraction Refinement (CEGAR)

1. Start with (coarse) initial abstraction $A$
2. Property $\varphi$ satisfied in $A$?
   - yes: Verification successful
   - no: Find run violating $\varphi$
      - spurious
      - real: Analyse counterexample
         - Error found
Recap: Predicate Abstraction

Predicate Abstraction I

Definition (Predicate abstraction)

Let $\text{Var}$ be a set of variables.

- A predicate is a Boolean expression $p \in BExp$ over $\text{Var}$.
- A state $\sigma \in \Sigma$ satisfies $p \in BExp$ ($\sigma \models p$) if $\text{val}_\sigma(p) = \text{true}$.
- $p$ implies $q$ ($p \models q$) if $\sigma \models q$ whenever $\sigma \models p$ (or: $p$ is stronger than $q$, $q$ is weaker than $p$).
- $p$ and $q$ are equivalent ($p \equiv q$) if $p \models q$ and $q \models p$.
- Let $P = \{p_1, \ldots, p_n\} \subseteq BExp$ be a finite set of predicates, and let $\neg P := \{\neg p_1, \ldots, \neg p_n\}$. An element of $P \cup \neg P$ is called a literal. The predicate abstraction lattice is defined by:

$$\text{Abs}(P) := \left( \left\{ \bigwedge Q \mid Q \subseteq P \cup \neg P \right\}, \models \right).$$

Abbreviations: true := $\bigwedge \emptyset$, false := $\bigwedge \{p_i, \neg p_i, \ldots\}$
Recap: Predicate Abstraction

Predicate Abstraction II

Lemma

**Abs**(*P*) is a complete lattice with

- \( \bot = \text{false} \), \( \top = \text{true} \)
- \( Q_1 \sqcap Q_2 = Q_1 \land Q_2 \) where \( \overline{b} := \bigwedge \{ q \in P \cup \neg P \mid b \models q \} \) (i.e., strongest formula in **Abs**(*P*) that implies \( Q_1 \land Q_2 \))
- \( Q_1 \sqcup Q_2 = Q_1 \lor Q_2 \) (i.e., strongest formula in **Abs**(*P*) that is implied by \( Q_1 \lor Q_2 \))

Example

Let \( P := \{ p_1, p_2, p_3 \} \) with \( p_1 := (x = 1), p_2 := (y = 2), p_3 := (z = 3) \).

1. For \( Q_1 := p_1 \land \neg p_2 \) and \( Q_2 := \neg p_2 \land p_3 \), we obtain
   \[
   \begin{align*}
   Q_1 \cap Q_2 &= Q_1 \land Q_2 = p_1 \land \neg p_2 \land p_3 \\
   Q_1 \cup Q_2 &= Q_1 \lor Q_2 = \neg p_2 \land (p_1 \lor p_3) = \neg p_2
   \end{align*}
   
   2. For \( Q_1 := p_1 \land p_2 \) and \( Q_2 := p_1 \land \neg p_2 \), we obtain
   \[
   \begin{align*}
   Q_1 \cap Q_2 &= Q_1 \land Q_2 = \text{false} \\
   Q_1 \cup Q_2 &= Q_1 \lor Q_2 = p_1 \land (p_2 \lor \neg p_2) = p_1
   \end{align*}
   
   6 of 18 Static Program Analysis
   Winter Semester 2016/17
   Lecture 16: Abstract Interpretation VI
   (Counterexample-Guided Abstraction Refinement)
Recap: Predicate Abstraction

Predicate Abstraction III

Definition (Galois connection for predicate abstraction)

The Galois connection for predicate abstraction is determined by

\[ \alpha : 2^\Sigma \rightarrow \text{Abs}(P) \quad \text{and} \quad \gamma : \text{Abs}(P) \rightarrow 2^\Sigma \]

with

\[ \alpha(S) := \bigsqcup \{ Q_\sigma \mid \sigma \in S \} \quad \text{and} \quad \gamma(Q) := \{ \sigma \in \Sigma \mid \sigma \models Q \} \]

where

\[ Q_\sigma := \bigwedge(\{ p_i \mid 1 \leq i \leq n, \sigma \models p_i \} \cup \{ \neg p_i \mid 1 \leq i \leq n, \sigma \not\models p_i \}) \]

Example

- Let \( \text{Var} := \{ x, y \} \) and \( P := \{ p_1, p_2, p_3 \} \) where \( p_1 := (x \leq y) \), \( p_2 := (x = y) \), \( p_3 := (x > y) \)
- If \( S = \{ \sigma_1, \sigma_2 \} \subseteq \Sigma \) with \( \sigma_1 = [x \mapsto 1, y \mapsto 2] \), \( \sigma_2 = [x \mapsto 2, y \mapsto 2] \),
  then \( \alpha(S) = Q_{\sigma_1} \sqcup Q_{\sigma_2} \)
  \[ = (p_1 \land \neg p_2 \land \neg p_3) \sqcup (p_1 \land p_2 \land \neg p_3) \]
  \[ = (p_1 \land \neg p_2 \land \neg p_3) \lor (p_1 \land p_2 \land \neg p_3) \]
  \[ \equiv p_1 \land \neg p_3 \]
- If \( Q = p_1 \land \neg p_2 \in \text{Abs}(P) \), then \( \gamma(Q) = \{ \sigma \in \Sigma \mid \sigma(x) < \sigma(y) \} \)
Recap: Predicate Abstraction

Abstract Semantics for Predicate Abstraction

Definition (Execution relation for predicate abstraction)

If \( c \in \text{Cmd} \) and \( Q \in \text{Abs}(P) \), then \( \langle c, Q \rangle \) is called an abstract configuration. The execution relation for predicate abstraction is defined by the following rules:

\[
\begin{align*}
\langle \text{skip}, Q \rangle & \Rightarrow \langle \downarrow, Q \rangle \\
\langle c_1, Q \rangle & \Rightarrow \langle c'_1, Q' \rangle \quad c'_1 \neq \downarrow \\
\langle c_1 ; c_2, Q \rangle & \Rightarrow \langle c'_1 ; c_2, Q' \rangle \\
\langle \text{if } b \text{ then } c_1 \text{ else } c_2 \text{ end}, Q \rangle & \Rightarrow \langle c_1, Q \land b \rangle \\
\langle \text{if } b \text{ then } c_1 \text{ else } c_2 \text{ end}, Q \rangle & \Rightarrow \langle c_2, Q \land \neg b \rangle \\
\langle \text{while } b \text{ do } c \text{ end}, Q \rangle & \Rightarrow \langle c; \text{while } b \text{ do } c \text{ end}, Q \land b \rangle \\
\langle \text{while } b \text{ do } c \text{ end}, Q \rangle & \Rightarrow \langle \downarrow, Q \land \neg b \rangle
\end{align*}
\]
Computation of Postconditions

Outline of Lecture 16

Recap: Predicate Abstraction

Computation of Postconditions

Counterexample-Guided Abstraction Refinement
Computation of Postconditions

Problem: $\overline{b} = \bigwedge \{ q \in P \cup \neg P \mid b \models q \}$ (i.e., the strongest formula in $\text{Abs}(P)$ that is implied by $b$) is generally not computable (due to undecidability of implication in certain logics)
Computation of Postconditions

Problem: \( \overline{b} = \bigwedge \{ q \in P \cup \neg P \mid b \models q \} \) (i.e., the strongest formula in \( \text{Abs}(P) \) that is implied by \( b \)) is generally not computable (due to undecidability of implication in certain logics)

Solutions:
- **Over-approximation**: fall back to non-strongest postconditions
  - in practice, (automatic) theorem proving
  - for every \( i \in \{1, \ldots, n\} \), try to prove \( b \models p_i \) and \( b \models \neg p_i \)
  - approximate \( \overline{b} \) by conjunction of all provable literals
Computation of Postconditions

Problem: \( \overline{b} = \bigwedge \{ q \in P \cup \neg P \mid b \models q \} \) (i.e., the strongest formula in \( \text{Abs}(P) \) that is implied by \( b \)) is generally not computable (due to undecidability of implication in certain logics)

Solutions:

- **Over-approximation:** fall back to non-strongest postconditions
  - in practice, (automatic) theorem proving
  - for every \( i \in \{1, \ldots, n\} \), try to prove \( b \models p_i \) and \( b \models \neg p_i \)
  - approximate \( \overline{b} \) by conjunction of all provable literals

- **Restriction of programs:**
  - \( \models \) decidable for certain logics
  - example: Presburger arithmetic (first-order theory of \( \mathbb{N} \) with +)
  - thus \( \overline{b} \) computable for WHILE programs without multiplication
Computation of Postconditions

Problem: \( \bar{b} = \bigwedge \{ q \in P \cup \neg P \mid b \models q \} \) (i.e., the strongest formula in \( \text{Abs}(P) \) that is implied by \( b \)) is generally not computable (due to undecidability of implication in certain logics)

Solutions:
- **Over-approximation:** fall back to non-strongest postconditions
  - in practice, (automatic) theorem proving
  - for every \( i \in \{1, \ldots, n\} \), try to prove \( b \models p_i \) and \( b \models \neg p_i \)
  - approximate \( \bar{b} \) by conjunction of all provable literals
- **Restriction of programs:**
  - \( \models \) decidable for certain logics
  - example: Presburger arithmetic (first-order theory of \( \mathbb{N} \) with \(+\))
  - thus \( \bar{b} \) computable for WHILE programs without multiplication
- **Restriction to finite domains:**
  - for example, binary numbers of fixed size
  - thus everything (domain, Galois connection, ...) exactly computable
  - problem: exponential blowup in \( n \) \( \Longrightarrow \) solution: Binary Decision Diagrams (BDDs)
Outline of Lecture 16

Recap: Predicate Abstraction

Computation of Postconditions

Counterexample-Guided Abstraction Refinement
Counterexample-Guided Abstraction Refinement

Reminder: CEGAR

Start with (coarse) initial abstraction $A$

Property $\varphi$ satisfied in $A$?

Verification successful

yes

no

Find run violating $\varphi$

spurious

Analyze counterexample

real

Error found

Remove counterexample by refining $A$
Counterexample-Guided Abstraction Refinement

Reminder: CEGAR

Start with (coarse) initial abstraction $A$

Property $\varphi$ satisfied in $A$?

- Verification successful
  - yes
  - no

Find run violating $\varphi$

Remove counterexample by refining $A$

Problems:
- How to decide realness of counterexample?

Analyze counterexample

Error found
Counterexample-Guided Abstraction Refinement

Reminder: CEGAR

- Start with (coarse) initial abstraction $A$
- Property $\varphi$ satisfied in $A$?
  - yes
    - Verification successful
  - no
    - Find run violating $\varphi$
      - spurious
    - Analyze counterexample
      - real
        - Error found
      - spurious

Problems:
- How to decide realness of counterexample?
- How to extract new predicates from spurious counterexample?
Counterexample-Guided Abstraction Refinement

Properties of Interest

- A certain program location is not reachable (dead code)
- Division by zero is excluded
- The value of $x$ never becomes negative
- After program termination, the value of $y$ is even
Counterexample-Guided Abstraction Refinement

Properties of Interest

- A certain program location is not reachable (dead code)
- Division by zero is excluded
- The value of $x$ never becomes negative
- After program termination, the value of $y$ is even

⇒ All representable as (non-)reachability of “bad locations”
⇒ Counterexample = path to bad locations
Counterexample-Guided Abstraction Refinement

Properties of Interest

- A certain program location is not reachable (dead code)
- Division by zero is excluded
- The value of $x$ never becomes negative
- After program termination, the value of $y$ is even

⇒ All representable as (non-)reachability of “bad locations”
⇒ Counterexample = path to bad locations

Definition 16.1 (Counterexample)

- A counterexample is a sequence of $k \geq 1$ abstract transitions of the form
  \[ \langle c_0, \text{true} \rangle \Rightarrow \langle c_1, Q_1 \rangle \Rightarrow \ldots \Rightarrow \langle c_k, Q_k \rangle \]
  where
  - $c_0, \ldots, c_k \in Cmd$ (or $c_k = \downarrow$)
  - $Q_1, \ldots, Q_k \in \text{Abs}(P)$ with $Q_k \neq \text{false}$
Counterexample-Guided Abstraction Refinement

Properties of Interest

- A certain program location is not reachable (dead code)
- Division by zero is excluded
- The value of $x$ never becomes negative
- After program termination, the value of $y$ is even

⇒ All representable as (non-)reachability of “bad locations”
⇒ Counterexample = path to bad locations

Definition 16.1 (Counterexample)

- A counterexample is a sequence of $k \geq 1$ abstract transitions of the form
  $\langle c_0, \text{true} \rangle \Rightarrow \langle c_1, Q_1 \rangle \Rightarrow \ldots \Rightarrow \langle c_k, Q_k \rangle$
  where
  - $c_0, \ldots, c_k \in \text{Cmd}$ (or $c_k = \bot$)
  - $Q_1, \ldots, Q_k \in \text{Abs}(P)$ with $Q_k \not\equiv \text{false}$
- It is called real if there exist concrete states $\sigma_0, \ldots, \sigma_k \in \Sigma$ such that
  $\forall i \in \{1, \ldots, k\} : \sigma_i \models Q_i$ and $\langle c_{i-1}, \sigma_{i-1} \rangle \rightarrow \langle c_i, \sigma_i \rangle$
- Otherwise it is called spurious.
Lemma 16.2

If $\langle c_0, \text{true} \rangle \Rightarrow \langle c_1, Q_1 \rangle \Rightarrow \ldots \Rightarrow \langle c_k, Q_k \rangle$ is a spurious counterexample, there exist Boolean expressions $b_0, \ldots, b_k$ with $b_0 \equiv \text{true}$, $b_k \equiv \text{false}$, and

$$\forall i \in \{1, \ldots, k\}, \sigma, \sigma' \in \Sigma : \sigma \models b_{i-1} \land \langle c_{i-1}, \sigma \rangle \rightarrow \langle c_i, \sigma' \rangle \implies \sigma' \models b_i$$
Lemma 16.2

If \( \langle c_0, \text{true} \rangle \Rightarrow \langle c_1, Q_1 \rangle \Rightarrow \ldots \Rightarrow \langle c_k, Q_k \rangle \) is a spurious counterexample, there exist Boolean expressions \( b_0, \ldots, b_k \) with \( b_0 \equiv \text{true} \), \( b_k \equiv \text{false} \), and

\[
\forall i \in \{1, \ldots, k\}, \sigma, \sigma' \in \Sigma : \sigma \models b_{i-1} \land \langle c_{i-1}, \sigma \rangle \rightarrow \langle c_i, \sigma' \rangle \implies \sigma' \models b_i
\]

Proof (idea).

Inductive definition of \( b_i \) as strongest postconditions:

1. \( b_0 := \text{true} \)
2. for \( i = 1, \ldots, k \): definition of \( b_i \) depending on \( b_{i-1} \) and on (axiom) transition rule applied in \( \langle c_{i-1}, . \rangle \Rightarrow \langle c_i, . \rangle \):
   
   - (skip) \( b_i := b_{i-1} \)
   - (asgn) \( b_i := \exists x'. (b_{i-1}[x \mapsto x'] \land x = a[x \mapsto x']) \)
     (for \( x := a \); \( x' = \) previous value of \( x \))

(yields \( b_k \equiv \text{false} \); by induction on \( k \))
Example 16.3

- Let \( c_0 := [x := z^0];[z := z + 1]^1;[y := z]^2; \)
  \hspace{1cm} \text{if } [x = y]^3 \text{ then } [\text{skip}]^4 \text{ else } [\text{skip}]^5 \text{ end}
Example 16.3

- Let $c_0 := [x := z]^0; [z := z + 1]^1; [y := z]^2;
  \quad \text{if } [x = y]^3 \text{ then } [\text{skip}]^4 \text{ else } [\text{skip}]^5 \text{ end}
- \textbf{Interesting property:} after termination, } x \neq y, \text{ i.e., label 4 unreachable}
Example 16.3

- Let $c_0 := [x := z]^0;[z := z + 1]^1;[y := z]^2;$
  
  \[\text{if } x = y\text{ then } \text{[skip]}^4 \text{ else } \text{[skip]}^5 \text{ end}\]

- **Interesting property:** after termination, $x \neq y$, i.e., label 4 unreachable

- **Initial abstraction:** $P = \emptyset$ ($\implies$ $\text{Abs}(P) = \{\text{true}, \text{false}\}$)
Counterexample-Guided Abstraction Refinement

Elimination of Spurious Counterexamples II

Example 16.3

- Let \( c_0 := [x := z; z := z + 1; y := z]; \) if \( x = y \) then [skip] else [skip] end
- **Interesting property:** after termination, \( x \neq y \), i.e., label 4 unreachable
- **Initial abstraction:** \( P = \emptyset \) (\( \Rightarrow \) \( \text{Abs}(P) = \{\text{true}, \text{false}\} \))
- **(Spurious) counterexample:** \( \langle 0, \text{true} \rangle \Rightarrow \langle 1, \text{true} \rangle \Rightarrow \langle 2, \text{true} \rangle \Rightarrow \langle 3, \text{true} \rangle \Rightarrow \langle 4, \text{true} \rangle \)
Counterexample-Guided Abstraction Refinement

Elimination of Spurious Counterexamples II

Example 16.3

- Let \( c_0 := [x := z]^0; [z := z + 1]^1; [y := z]^2; \)
  \[\text{if } [x = y]^3 \text{ then } [\text{skip}]^4 \text{ else } [\text{skip}]^5 \text{ end}\]
- **Interesting property:** after termination, \( x \neq y \), i.e., label 4 unreachable
- **Initial abstraction:** \( P = \emptyset \) (\( \Rightarrow \) \( \text{Abs}(P) = \{\text{true}, \text{false}\} \))
- (Spurious) **counterexample:** \( \langle 0, \text{true} \rangle \Rightarrow \langle 1, \text{true} \rangle \Rightarrow \langle 2, \text{true} \rangle \Rightarrow \langle 3, \text{true} \rangle \Rightarrow \langle 4, \text{true} \rangle \)
- **Forward construction of strongest postconditions:**
  - \( b_0 := \text{true} \)
Example 16.3

- Let \( c_0 := [x := z]^0; [z := z + 1]^1; [y := z]^2; \)
  \[ \text{if } [x = y]^3 \text{ then } [\text{skip}]^4 \text{ else } [\text{skip}]^5 \text{ end} \]
- Interesting property: after termination, \( x \neq y \), i.e., label 4 unreachable
- Initial abstraction: \( P = \emptyset \) (\( \implies \text{Abs}(P) = \{\text{true, false}\} \))
- (Spurious) counterexample: \( \langle 0, \text{true} \rangle \Rightarrow \langle 1, \text{true} \rangle \Rightarrow \langle 2, \text{true} \rangle \Rightarrow \langle 3, \text{true} \rangle \Rightarrow \langle 4, \text{true} \rangle \)
- Forward construction of strongest postconditions:
  - \( b_0 := \text{true} \)
  - (asgn) \( b_i := \exists x'. (b_{i-1}[x \mapsto x'] \land x = a[x \mapsto x']) \)
    \[ \implies b_1 := \exists x'. (b_0[x \mapsto x'] \land x = z[x \mapsto x']) \equiv (x = z) \]
Example 16.3

- Let \( c_0 := [x := z]^0;[z := z + 1]^1;y := z]^2; \)
  \[ \text{if } [x = y]^3 \text{ then } [\text{skip}]^4 \text{ else } [\text{skip}]^5 \text{ end} \]
- Interesting property: after termination, \( x \neq y \), i.e., label 4 unreachable
- Initial abstraction: \( P = \emptyset \) \( \implies \) \( \text{Abs}(P) = \{\text{true, false}\} \)
- (Spurious) counterexample: \( \langle 0, \text{true} \rangle \implies \langle 1, \text{true} \rangle \implies \langle 2, \text{true} \rangle \implies \langle 3, \text{true} \rangle \implies \langle 4, \text{true} \rangle \)
- Forward construction of strongest postconditions:
  - \( b_0 := \text{true} \)
  - (asgn) \( b_1 := \exists x'.(b_{i-1}[x \mapsto x'] \land x = a[x \mapsto x']) \)
    \[ \implies b_1 := \exists x'.(b_0[x \mapsto x'] \land x = z[x \mapsto x']) \equiv (x = z) \]
  - (asgn) \( b_2 := \exists x'.(b_{i-1}[x \mapsto x'] \land x = a[x \mapsto x']) \)
    \[ \implies b_2 := \exists z'.(b_1[z \mapsto z'] \land z = z + 1[z \mapsto z']) \]
    \[ = \exists z'.(x = z' \land z = z' + 1) \equiv (x + 1 = z) \]
Counterexample-Guided Abstraction Refinement

Elimination of Spurious Counterexamples II

Example 16.3

- Let $c_0 := [x := z]^0;[z := z + 1]^1;[y := z]^2$;
  if $[x = y]^3$ then $[\text{skip}]^4$ else $[\text{skip}]^5$ end
- Interesting property: after termination, $x \neq y$, i.e., label 4 unreachable
- Initial abstraction: $P = \emptyset$ (implies $\text{Abs}(P) = \{\text{true, false}\}$)
- (Spurious) counterexample: $\langle 0, \text{true} \rangle \Rightarrow \langle 1, \text{true} \rangle \Rightarrow \langle 2, \text{true} \rangle \Rightarrow \langle 3, \text{true} \rangle \Rightarrow \langle 4, \text{true} \rangle$
- Forward construction of strongest postconditions:
  - $b_0 := \text{true}$
  - (asgn) $b_1 := \exists x'.(b_{i-1}[x \mapsto x'] \land x = a[x \mapsto x'])$
    $\implies b_1 := \exists x'.(b_0[x \mapsto x'] \land x = z[x \mapsto x']) \equiv (x = z)$
  - (asgn) $b_2 := \exists z'.(b_{i-1}[x \mapsto x'] \land x = a[x \mapsto x'])$
    $\implies b_2 := \exists z'.(b_1[z \mapsto z'] \land z = z + 1[z \mapsto z'])$
    $= \exists z'.(x = z' \land z = z' + 1) \equiv (x + 1 = z)$
  - (asgn) $b_3 := \exists y'.(b_{i-1}[y \mapsto y'] \land y = z[y \mapsto y']) \equiv (x + 1 = z \land y = z)$
Example 16.3

Let \( c_0 := [x := z^0];[z := z + 1]^1;[y := z]^2; \)
if [x = y]^3 then [skip]^4 else [skip]^5 end

- **Interesting property:** after termination, \( x \neq y \), i.e., label 4 unreachable
- **Initial abstraction:** \( P = \emptyset \) (\( \implies \) \( \text{Abs}(P) = \{\text{true}, \text{false}\} \))
- **(Spurious) counterexample:** \( \langle 0, \text{true} \rangle \Rightarrow \langle 1, \text{true} \rangle \Rightarrow \langle 2, \text{true} \rangle \Rightarrow \langle 3, \text{true} \rangle \Rightarrow \langle 4, \text{true} \rangle \)
- **Forward construction of strongest postconditions:**
  - \( b_0 := \text{true} \)
  - (asgn) \( b_i := \exists x'. (b_{i-1}[x \mapsto x'] \land x = a[x \mapsto x']) \)
    \( \implies b_1 := \exists x'. (b_0[x \mapsto x'] \land x = z[x \mapsto x']) \equiv (x = z) \)
  - (asgn) \( b_i := \exists x'. (b_{i-1}[x \mapsto x'] \land x = a[x \mapsto x']) \)
    \( \implies b_2 := \exists z'. (b_1[z \mapsto z'] \land z = z + 1[z \mapsto z']) \)
    \( = \exists z'. (x = z' \land z = z' + 1) \equiv (x + 1 = z) \)
  - (asgn) \( b_i := \exists x'. (b_{i-1}[x \mapsto x'] \land x = a[x \mapsto x']) \)
    \( \implies b_3 := \exists y'. (b_2[y \mapsto y'] \land y = z[y \mapsto y']) \equiv (x + 1 = z \land y = z) \)
  - (if1) \( b_i := b_{i-1} \land b \)
    \( \implies b_4 := (b_3 \land x = y) \equiv (x + 1 = z \land y = z \land x = y) \equiv \text{false} \)
Counterexample-Guided Abstraction Refinement

Abstraction Refinement

- Using $b_1, \ldots, b_{k-1}$ as computed before, let $P' := P \cup \{p_1, \ldots, p_n\}$ where $p_1, \ldots, p_n$ are the atomic conjuncts occurring in $b_1, \ldots, b_{k-1}$
- Refine $\text{Abs}(P)$ to $\text{Abs}(P')$

Lemma 16.4

After refinement, the spurious counterexample $\langle c_0, \text{true} \rangle \Rightarrow \langle c_1, Q_1 \rangle \Rightarrow \ldots \Rightarrow \langle c_k, Q_k \rangle$ with $Q_k \not\equiv \text{false}$ does not exist anymore.

Proof. omitted
Counterexample-Guided Abstraction Refinement

Abstraction Refinement

- Using $b_1, \ldots, b_{k-1}$ as computed before, let $P' := P \cup \{p_1, \ldots, p_n\}$ where $p_1, \ldots, p_n$ are the atomic conjuncts occurring in $b_1, \ldots, b_{k-1}$
- Refine $Abs(P)$ to $Abs(P')$

Lemma 16.4

After refinement, the spurious counterexample

$$\langle c_0, \text{true} \rangle \Rightarrow \langle c_1, Q_1 \rangle \Rightarrow \ldots \Rightarrow \langle c_k, Q_k \rangle$$

with $Q_k \not\equiv \text{false}$ does not exist anymore.

Proof.

omitted
Example 16.5 (cf. Example 16.3)

- Let $c_0 := [x := z]^0;[z := z + 1]^1;[y := z]^2$;
  \[\text{if } [x = y]^3 \text{ then } [\text{skip}]^4 \text{ else } [\text{skip}]^5 \text{ end}\]
- $P = \emptyset$, $P' = \{\underbrace{x = z}_{p_1}, \underbrace{x + 1 = z}_{p_2}, \underbrace{y = z}_{p_3}\}$
Counterexample-Guided Abstraction Refinement

A Simple Example

Example 16.5 (cf. Example 16.3)

- Let $c_0 := [x := z]^0; [z := z + 1]^1; [y := z]^2$
  if $[x = y]^3$ then $[\text{skip}]^4$ else $[\text{skip}]^5$ end

- $P = \emptyset$, $P' = \{x = z, x + 1 = z, y = z\}$

- Refined abstract transitions:
  $\langle 0, \text{true} \rangle$
Counterexample-Guided Abstraction Refinement

A Simple Example

Example 16.5 (cf. Example 16.3)

- Let $c_0 := [x := z]^0;[z := z + 1]^1;[y := z]^2$;
  \[\text{if } [x = y]^3 \text{ then } [\text{skip}]^4 \text{ else } [\text{skip}]^5 \text{ end}\]
- $P = \emptyset$, $P' = \{x = z, x + 1 = z, y = z\}$
- Refined abstract transitions:
  \[\langle 0, \text{true} \rangle \Rightarrow \langle 1, p_1 \land \neg p_2 \rangle\]
Counterexample-Guided Abstraction Refinement

A Simple Example

Example 16.5 (cf. Example 16.3)

- Let $c_0 := [x := z]^0; [z := z + 1]^1; [y := z]^2$;
  if $[x = y]^3$ then $[\text{skip}]^4$ else $[\text{skip}]^5$ end
- $P = \emptyset$, $P' = \{x = z, x + 1 = z, y = z\}$
- Refined abstract transitions:
  \[
  \langle 0, \text{true} \rangle \Rightarrow \langle 1, p_1 \land \neg p_2 \rangle \\
  \Rightarrow \langle 2, \neg p_1 \land p_2 \rangle
  \]
Counterexample-Guided Abstraction Refinement

A Simple Example

Example 16.5 (cf. Example 16.3)

• Let $c_0 := [x := z]^0;[z := z + 1]^1;[y := z]^2$;
  if $[x = y]^3$ then [skip]$^4$ else [skip]$^5$ end

• $P = \emptyset$, $P' = \{x = z, x + 1 = z, y = z\}$

• Refined abstract transitions:
  \[
  \langle 0, \text{true} \rangle \Rightarrow \langle 1, p_1 \land \lnot p_2 \rangle
  \Rightarrow \langle 2, \lnot p_1 \land p_2 \rangle
  \Rightarrow \langle 3, \lnot p_1 \land p_2 \land p_3 \rangle
  \]
Counterexample-Guided Abstraction Refinement

A Simple Example

Example 16.5 (cf. Example 16.3)

- Let $c_0 := [x := z]^0; [z := z + 1]^1; [y := z]^2;$
  if $[x = y]^3$ then $[\text{skip}]^4$ else $[\text{skip}]^5$ end
- $P = \emptyset$, $P' = \{ x = z, x + 1 = z, y = z \}$
- Refined abstract transitions:

  $\langle 0, \text{true} \rangle \Rightarrow \langle 1, p_1 \land \neg p_2 \rangle$
  $\Rightarrow \langle 2, \neg p_1 \land p_2 \rangle$
  $\Rightarrow \langle 3, \neg p_1 \land p_2 \land p_3 \rangle$
  $\Rightarrow \langle 4, \neg p_1 \land p_2 \land p_3 \land x = y \rangle$
  $\equiv \text{false}$
Counterexample-Guided Abstraction Refinement

Another Example: Multiplication

Example 16.6

- Let $c_0 := [z := 0]^0$; while $[x > 0]^1$ do $[z := z + y]^2$; $[x := x - 1]^3$ end;
  if $[z \mod y = 0]^4$ then $[\text{skip}]^5$ else $[\text{skip}]^6$ end;

- Global assumption: $y > 0$

- Interesting property: label 6 unreachable (since $z$ multiple of $y$)
Counterexample-Guided Abstraction Refinement

Another Example: Multiplication

Example 16.6

- Let $c_0 := [z := 0]^0$;
  
  while $[x > 0]^1$ do
  
  $[z := z + y]^2$;
  
  $[x := x - 1]^3$
  
  end;
  
  if $[z \mod y = 0]^4$ then
  
  [skip]$^5$
  
  else
  
  [skip]$^6$
  
  end;

- Global assumption: $y > 0$

- Interesting property: label 6 unreachable (since $z$ multiple of $y$)

- Initial abstraction: $P = \emptyset$ ( $\implies$ Abs($P$) = \{true, false\})

- Abstraction refinement: on the board