



Static Program Analysis

**Lecture 15: Abstract Interpretation VI
(Counterexample-Guided Abstraction Refinement)**

Summer Semester 2018

Thomas Noll

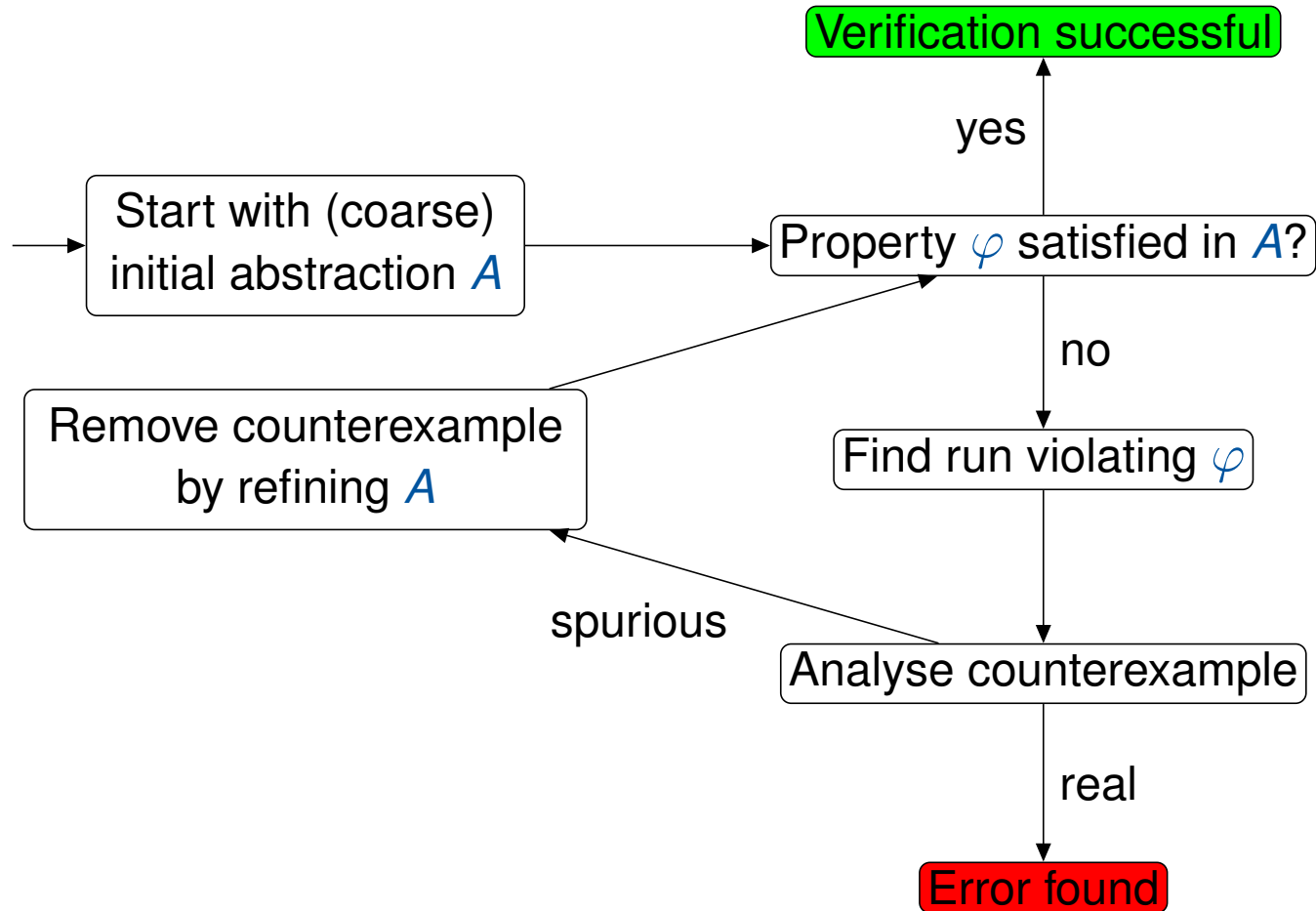
Software Modeling and Verification Group

RWTH Aachen University

<https://moves.rwth-aachen.de/teaching/ss-18/spa/>

Recap: Predicate Abstraction

Counterexample-Guided Abstraction Refinement (CEGAR)



Recap: Predicate Abstraction

Predicate Abstraction I

Definition (Predicate abstraction)

Let Var be a set of variables.

- A **predicate** is a Boolean expression $p \in BExp$ over Var .
- A state $\sigma \in \Sigma$ **satisfies** $p \in BExp$ ($\sigma \models p$) if $val_\sigma(p) = \text{true}$.
- 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 $q \models p$.
- p and q are **independent** if $p \not\models q$ and $q \not\models p$, otherwise **interdependent**.
- Let $P = \{p_1, \dots, p_n\} \subseteq BExp$ be a finite set of predicates, and let $\neg P := \{\neg p_1, \dots, \neg p_n\}$. An element of $P \cup \neg P$ is called a **literal**. The **predicate abstraction lattice** is defined by:

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

Abbreviation: $\text{true} := \bigwedge \emptyset$ (and $\text{false} \equiv \bigwedge \{p_i, \neg p_i, \dots\}$ if $P \neq \emptyset$)

Recap: Predicate Abstraction

Predicate Abstraction II

Lemma

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

- $\perp = \text{false}$, $\top = \text{true}$
- $Q_1 \sqcap Q_2 = \overline{Q_1 \wedge Q_2}$ where $\bar{b} := \bigwedge \{q \in P \cup \neg P \mid b \models q\}$
(i.e., \bar{b} is strongest formula in $Abs(P)$ that is implied by b)
- $Q_1 \sqcup Q_2 = \overline{Q_1 \vee Q_2}$

Example

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

1. For $Q_1 := p_1 \wedge \neg p_2$ and $Q_2 := \neg p_2 \wedge p_3$, we obtain

$$Q_1 \sqcap Q_2 = \overline{Q_1 \wedge Q_2} = \overline{p_1 \wedge \neg p_2 \wedge p_3} \quad (= \bigwedge (Q_1 \cup Q_2))$$

$$Q_1 \sqcup Q_2 = \overline{Q_1 \vee Q_2} \equiv \overline{\neg p_2 \wedge (p_1 \vee p_3)} = \neg p_2 \quad (= \bigwedge (Q_1 \cap Q_2))$$

2. For $Q_1 := p_1 \wedge p_2$ and $Q_2 := p_1 \wedge \neg p_2$, we obtain

$$Q_1 \sqcap Q_2 = \overline{Q_1 \wedge Q_2} = \overline{\text{false}} \quad (= \bigwedge (Q_1 \cup Q_2))$$

$$Q_1 \sqcup Q_2 = \overline{Q_1 \vee Q_2} \equiv \overline{p_1 \wedge (p_2 \vee \neg p_2)} = \overline{p_1} \quad (= \bigwedge (Q_1 \cap Q_2))$$

Recap: Predicate Abstraction

Predicate Abstraction III

Definition (Galois connection for predicate abstraction)

The **Galois connection for predicate abstraction** is determined by

$$\begin{aligned} & \alpha : 2^\Sigma \rightarrow \text{Abs}(P) & \gamma : \text{Abs}(P) \rightarrow 2^\Sigma \\ \text{with} & \alpha(S) := \bigsqcup \{Q_\sigma \mid \sigma \in S\} & \gamma(Q) := \{\sigma \in \Sigma \mid \sigma \models Q\} \\ \text{where} & Q_\sigma := \bigwedge (\{p \in P \mid \sigma \models p\} \cup \{\neg p \in \neg P \mid \sigma \not\models p\}) \end{aligned}$$

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 $\alpha(S) = Q_{\sigma_1} \sqcup Q_{\sigma_2}$
$$\begin{aligned} &= (p_1 \wedge \neg p_2 \wedge \neg p_3) \sqcup (p_1 \wedge p_2 \wedge \neg p_3) \\ &= \frac{(p_1 \wedge \neg p_2 \wedge \neg p_3) \sqcup (p_1 \wedge p_2 \wedge \neg p_3)}{\equiv} \\ &\equiv p_1 \wedge \neg p_3 \end{aligned}$$
- If $Q = p_1 \wedge \neg p_2 \in \text{Abs}(P)$, then $\gamma(Q) = \{\sigma \in \Sigma \mid \sigma(x) < \sigma(y)\}$

Recap: Predicate Abstraction

Abstract Semantics for Predicate Abstraction

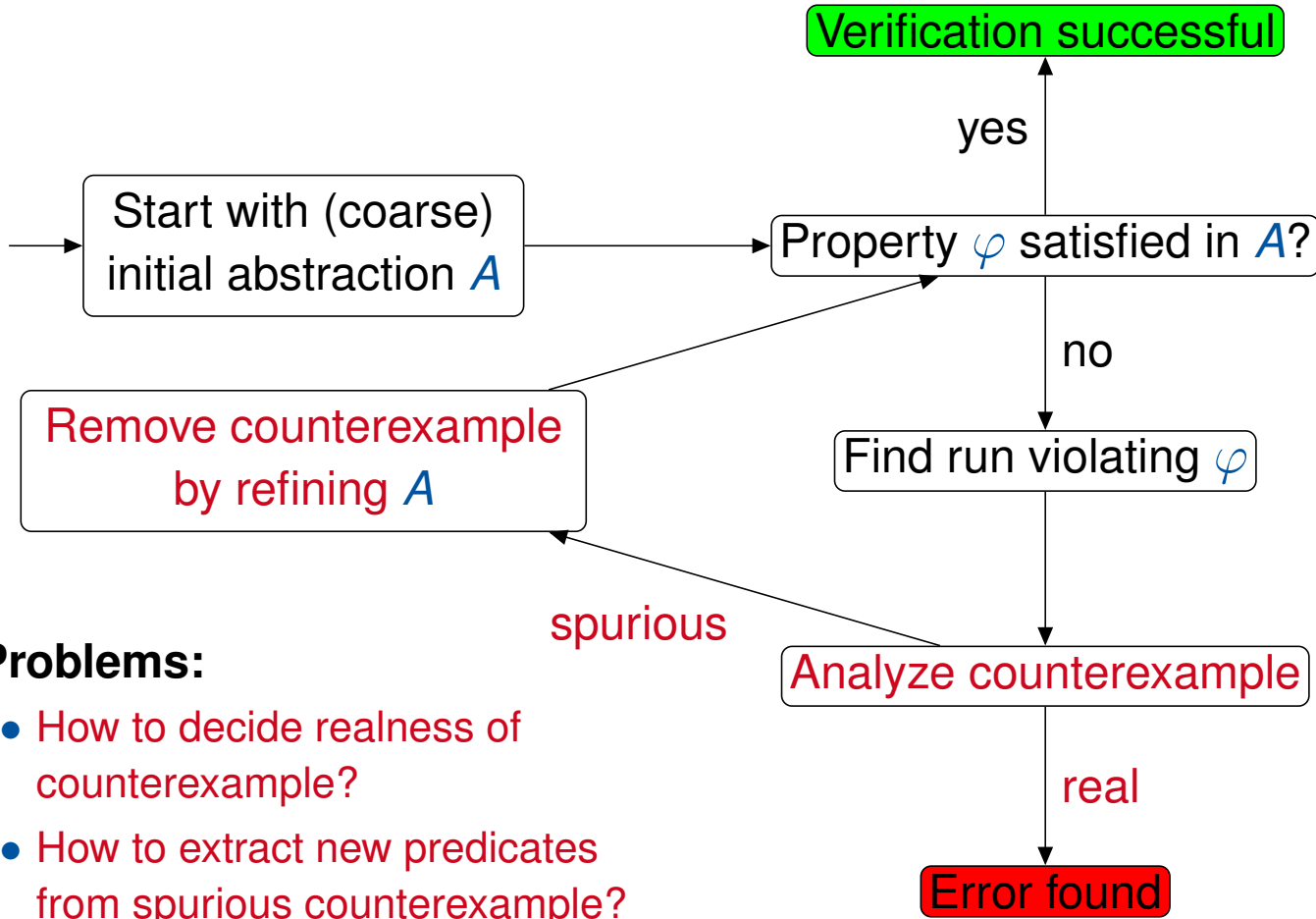
Definition (Execution relation for predicate abstraction)

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:

$$\begin{array}{c} \text{(skip)} \frac{}{\langle \text{skip}, Q \rangle \Rightarrow \langle \downarrow, Q \rangle} \quad \text{(asgn)} \frac{}{\langle x := a, Q \rangle \Rightarrow \langle \downarrow, \bigsqcup \{ Q_{\sigma[x \mapsto \text{val}_{\sigma}(a)]} \mid \sigma \models Q \} \rangle} \\ \text{(seq1)} \frac{\langle c_1, Q \rangle \Rightarrow \langle c'_1, Q' \rangle \quad c'_1 \neq \downarrow}{\langle c_1; c_2, Q \rangle \Rightarrow \langle c'_1; c_2, Q' \rangle} \quad \text{(seq2)} \frac{\langle c_1, Q \rangle \Rightarrow \langle \downarrow, Q' \rangle}{\langle c_1; c_2, Q \rangle \Rightarrow \langle c_2, Q' \rangle} \\ \text{(if1)} \frac{}{\langle \text{if } b \text{ then } c_1 \text{ else } c_2 \text{ end}, Q \rangle \Rightarrow \langle c_1, \overline{Q \wedge b} \rangle} \\ \text{(if2)} \frac{}{\langle \text{if } b \text{ then } c_1 \text{ else } c_2 \text{ end}, Q \rangle \Rightarrow \langle c_2, \overline{Q \wedge \neg b} \rangle} \\ \text{(wh1)} \frac{}{\langle \text{while } b \text{ do } c \text{ end}, Q \rangle \Rightarrow \langle c; \text{while } b \text{ do } c \text{ end}, \overline{Q \wedge b} \rangle} \\ \text{(wh2)} \frac{}{\langle \text{while } b \text{ do } c \text{ end}, Q \rangle \Rightarrow \langle \downarrow, \overline{Q \wedge \neg b} \rangle} \end{array}$$

Counterexample-Guided Abstraction Refinement

Reminder: CEGAR



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

Definition 15.1 (Counterexample)

- A **counterexample** is a sequence of $k \geq 1$ abstract transitions of the form

$$\langle c_0, Q_0 \rangle \Rightarrow \langle c_1, Q_1 \rangle \Rightarrow \dots \Rightarrow \langle c_k, Q_k \rangle$$

where

- $c_0, \dots, c_k \in \text{Cmd}$ (or $c_k = \downarrow$)
- $Q_0, \dots, Q_k \in \text{Abs}(P)$ with $Q_0 \equiv \text{true}$ and $Q_k \not\equiv \text{false}$
- It is called **real** if there exist concrete states $\sigma_0, \dots, \sigma_k \in \Sigma$ such that
$$\forall i \in \{1, \dots, k\} : \sigma_i \models Q_i \quad \text{and} \quad \langle c_{i-1}, \sigma_{i-1} \rangle \rightarrow \langle c_i, \sigma_i \rangle$$
- Otherwise it is called **spurious**.

Counterexample-Guided Abstraction Refinement

Elimination of Spurious Counterexamples I

Lemma 15.2

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

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

Proof (idea).

Inductive definition of b_i as **strongest postconditions**:

1. $b_0 := \text{true}$
2. for $i = 1, \dots, k$: definition of b_i depending on b_{i-1} and on (axiom) transition rule applied in $\langle c_{i-1}, \cdot \rangle \Rightarrow \langle c_i, \cdot \rangle$:

Atomic commands:

- (skip) $b_i := b_{i-1}$
- (asgn) $b_i := \exists x'. (b_{i-1} [x \mapsto x'] \wedge x = a[x \mapsto x'])$
(for “ $x := a$ ”; x' = previous value of x)

Control structures (with guard b):

- (if1) $b_i := b_{i-1} \wedge b$
- (if2) $b_i := b_{i-1} \wedge \neg b$
- (wh1) $b_i := b_{i-1} \wedge b$
- (wh2) $b_i := b_{i-1} \wedge \neg b$

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



Elimination of Spurious Counterexamples II

Example 15.3

- 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:** $P = \emptyset$ ($\implies Abs(P) = \{\text{true}, \text{false}\}$)
- (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) $b_i := \exists x'. (b_{i-1}[x \mapsto x'] \wedge x = a[x \mapsto x'])$
 $\implies b_1 := \exists x'. (b_0[x \mapsto x'] \wedge x = z[x \mapsto x']) \equiv (x = z)$
 - (asgn) $b_i := \exists x'. (b_{i-1}[x \mapsto x'] \wedge x = a[x \mapsto x'])$
 $\implies b_2 := \exists z'. (b_1[z \mapsto z'] \wedge z = z + 1[z \mapsto z'])$
 $= \exists z'. (x = z' \wedge z = z' + 1) \equiv (z = x + 1)$
 - (asgn) $b_i := \exists x'. (b_{i-1}[x \mapsto x'] \wedge x = a[x \mapsto x'])$
 $\implies b_3 := \exists y'. (b_2[y \mapsto y'] \wedge y = z[y \mapsto y']) \equiv (z = x + 1 \wedge y = z)$
 - (if1) $b_i := b_{i-1} \wedge b$
 $\implies b_4 := (b_3 \wedge x = y) \equiv (z = x + 1 \wedge y = z \wedge x = y) \equiv \text{false}$

Counterexample-Guided Abstraction Refinement

Abstraction Refinement

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

Lemma 15.4

After refinement, the spurious counterexample

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

with $Q_k \neq \text{false}$ does not exist any more.

Proof.

omitted □

Examples

A Simple Example

Example 15.5 (cf. Example 15.3)

- Let $c_0 := [x := z]^0; [z := z + 1]^1; [y := z]^2;$
if $[x = y]^3$ then $[skip]^4$ else $[skip]^5$ end
- $P = \emptyset, P' = \{\underbrace{x = z}_{p_1}, \underbrace{z = x + 1}_{p_2}, \underbrace{y = z}_{p_3}, \underbrace{x = y}_{p_4}\}$

- Refined abstract transitions:

$$\begin{aligned} \langle 0, \text{true} \rangle &\Rightarrow \langle 1, p_1 \wedge \neg p_2 \rangle \\ &\Rightarrow \langle 2, \neg p_1 \wedge p_2 \rangle \\ &\Rightarrow \langle 3, \neg p_1 \wedge p_2 \wedge p_3 \wedge \neg p_4 \rangle \\ &\Rightarrow \langle 4, \underbrace{\neg p_1 \wedge p_2 \wedge p_3 \wedge \neg p_4 \wedge p_4}_{\equiv \text{false}} \rangle \end{aligned}$$

Examples

Another Example: Multiplication

Example 15.6

- Let $c_0 := [z := 0]^0$;
 while $[x > 0]^1$ do
 $[z := z + y]^2$;
 $[x := x - 1]^3$
 end;
 if $[z \bmod y = 0]^4$ then
 $[\text{skip}]^5$
 else
 $[\text{skip}]^6$
 end;
- **Global assumption:** $y > 0$
- **Interesting property:** label 6 unreachable (since z multiple of y)
- **Initial abstraction:** $P = \emptyset$ ($\implies \text{Abs}(P) = \{\text{true}, \text{false}\}$)
- **Abstraction refinement:** on the board