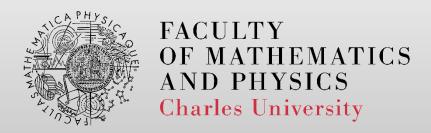
Abstract Interpretation

http://d3s.mff.cuni.cz



Pavel Parízek



Abstract interpretation

- Theoretical framework unifying different program analyses
- Formal underpinning of sound and correct static analyses

Practice: Code Contracts, Astree, Polyspace



Reusing concepts from static analysis

0.0.0

- Control-flow graphs
- Finite lattices
- Transfer functions
- Fixed points
 - Iterative computation
 - Work list algorithm

New concepts

Explicit abstraction

- Concrete domain
- Abstract domain

Galois connections

Purpose: constructing sound abstractions



Example program

```
int compute(int x, int y) {
Requires (x >= 0 \&\& y >= 0);
y = y + 1;
z = x + y;
Assert (z >= 1);
return z;
```

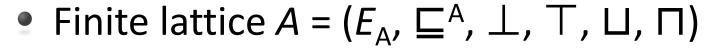
Concrete domain



- Finite lattice $C = (E_C, \sqsubseteq^C)$
 - Set of concrete elements E_C
 - Partial order \sqsubseteq^{c} on E_{c}
- Notation abuse
 - Symbol C means both the finite lattice and the set of concrete elements
- Example
 - Possible values of an integer variable
 - $E_C = 2^{\mathbb{N}}$ (all possible subsets)
 - $\blacksquare \subseteq^{C} = \subseteq$ (plain subset ordering)



Abstract domain



- Set of abstract elements E_A
- Partial order \sqsubseteq^A on E_A
- Least abstract element ⊥
- Greatest abstract element T
- Join operator
- Meet operator Π



Abstract domain: Intervals

Definition

- ⊥ is empty interval
- \blacksquare \top = $\{-\infty, +\infty\}$

• Examples

- \blacksquare [0, 2] \sqsubseteq^{A} [0, 4]
- \blacksquare [0, 2] $\not\sqsubseteq$ ^A [1, 3]



Relation between domains



Computes the most precise abstract representation

• Concretization function $\gamma: A \mapsto C$

- Example: interval domain
 - $\alpha(S) = [\min(S), \max(S)], S = \{s_1, ..., s_N\}$



Galois connection

- Necessary conditions
 - \blacksquare Both functions lpha and γ are monotone
 - $\blacksquare \ \forall \ a \in A, \ c \in C : \alpha(c) \sqsubseteq^{A} a \iff c \sqsubseteq^{C} \gamma(a)$

Relation between partially ordered sets A and C

- Characterizes sound abstraction
 - We can lose precision (over-approximating)

$$c \sqsubseteq^{\mathsf{C}} \gamma \circ \alpha(c) \qquad \alpha \circ \gamma(a) \sqsubseteq^{\mathsf{A}} a$$



Transfer functions



- Concrete transfer function $\tau_c : C \mapsto C$
 - Expresses concrete semantics of program statements
- Abstract transfer function $\tau_A : A \mapsto A$
 - Expresses abstract semantics of program statements
- Relation: $\forall a \in A : \tau_{C} \circ \gamma(a) \subseteq \gamma \circ \tau_{A}(a)$
- Concrete program P_C
- Abstract program P_A



How to compute solution

- Input problem
 - Concrete program P_C
 - Abstract domain A
 - Functions α and γ
 - lacksquare Transfer function au_{A}

- Representing information
 - Separate analysis value for each program variable
 - One large set with values for all program variables

12

How to compute solution

- Input analysis problem
- Representing information

- Approach: find $fp(P_A)$
 - Symbol Ifp ~ least fixed point
 - Using the work-list algorithm

• Result: $lfp(P_C) \sqsubseteq^C \gamma(lfp(P_A))$

Divergence

- Problem: fixpoint computation may diverge
 - Why: infinite increasing chains (sequences)

- Ascending Chain Condition (ACC)
 - Strictly ascending sequence of elements reaches some fixed point (terminates)
 - **Example:** $a_1 \sqsubseteq a_2 \sqsubseteq a_3 \sqsubseteq ... \sqsubseteq a_n = a_{n+1} = a_{n+2}$

Solution: Widening operator



Widening

- Widening operator ∇ : A × A → A

- Sequence $w_0 = a_0, ..., w_{i+1} = w_i \nabla a_{i+1}$ not strictly increasing
 - fixed point computation will terminate

Other benefit: faster convergence



Widening: Intervals

"Intervals" does not satisfy ACC

- Option 1
 - Keep stable bounds (preserve the value)
 - Extrapolate unstable bounds $(\{-\infty, +\infty\})$
- Option 2
 - Keep valid bounds from the first operand
 - Extrapolate bounds otherwise ({-∞,+∞})

16

Widening



- Practice
 - Use widening operator
 - on backward edges in CFG
 - for really too big intervals
- Remedy: Narrowing
 - Complementary operator
 - Goal: improving precision



17

Numerical abstract domains

- Non-relational domains
 - Program variables treated separately
 - Examples: signs, intervals

- Relational domains
 - Consider relations between variables
 - Example: predicate abstraction



Cartesian abstraction

- Very important special case
- Key idea
 - \blacksquare α : flattening the analysis information
 - Y: restores all possible combinations
- Pros: better scalability
- Cons: loses precision
- Example: predicate abstraction



Other abstract domains

- Octagon
 - Values represented as constraints $\pm x \pm y \le c$

- Polyhedral
 - Values represented as constraints $\sum_i a_{ij} * x_i \le c_j$
- Linear equations

Strings: Prefix, Suffix, Character inclusion, ...

Using abstract interpretation

1) Design the abstract domains

2) Define abstraction functions

3) Design widening operators

4) Define all transfer functions



Multiple abstract domains

Combination

- Two abstract domains A_1 , A_2
- Cartesian product: $A_1 \times A_2$

Composition

- One concrete domain C
- Abstract domains A_1 , A_2
- Galois connection
 - α_1 , γ_1 between C and A_1
 - α_2 , γ_2 between A_1 and A_2
- We get the connection between C and A_2
 - Functions: $\alpha_2 \circ \alpha_1$, $\gamma_1 \circ \gamma_2$



Tools

Clousot

- Program analyzer for Code Contracts (C#/.NET)
- Verifies method contracts and low-level errors
- https://www.microsoft.com/en-us/research/project/code-contracts/

Astrée

- Static analyzer for programs written in C
 - Programs without dynamic memory allocation and recursion
- Industrial applications (Airbus A340 SW)
- http://www.astree.ens.fr/

Polyspace

- Static analysis toolset for programs in C/C++/Ada
- http://www.mathworks.com/products/polyspace/

23

Further reading

- F. Nielson, H. R. Nielson, and Chris Hankin.
 Principles of Program Analysis. Springer, 2005
- P. Cousot, R. Cousot, J. Feret, L. Mauborgne, A. Mine, and X. Rival. Why does Astrée Scale Up? Formal Methods in System Design, 35(3), 2009
- P. Ferrara, F. Logozzo, and M. Fahndrich. Safer
 Unsafe Code for . NET. OOPSLA 2008

