25 Years of Abstract Interpretation
The German Perspective

Andreas Podelski
University of Freiburg

Who?

Reinhard	Wilhelm	Saarbrücken
Bernhard	Steffen	Dortmund
Jens	Knoop	Wien
Markus	Müller-Olm	Münster
Helmut	Seidl	München
Andreas	Podelski	Freiburg

What?

1. Grammar flow analysis
2. Worst-Case Execution Time (WCET)
3. Data flow analysis as model checking
4. Partial Redundancy Elimination (PRE)
5. Decidability frontiers for abstraction
6. Set-based analysis, set constraints
7. Abstraction refinement-based model checking (ARMC)
8. Transition invariants, termination analysis

1. Grammar Flow Analysis

- abstract one-step derivation ("N → E")
- concrete lattice of sets of derivation trees, abstract lattice of graphs
- first/follow, emptiness, "restraints" between attributes, …
- 1982: Möncke & Wilhelm (2006: C & C)

2. Worst-Case Execution Time (WCET)

- sum up time that all instructions can take on path
  (take maximum over all execution paths)
- cost of instruction depends on state

  predicted cache hit ⇒ tighter WCET bound

- analysis of reachable configurations: cache, pipelines, ...
- SAS'96: Wilhelm, Ferdinand, Martin, ...
  company: AbsInt (Airbus, Bosch, ...)

3. Data Flow Analysis as Model Checking

- data flow facts
  = temporal-logic properties
  = fixpoint expressions in mu-calculus
- fixpoints in finite abstract lattice? - model checking!
- conceptual, practical (company: MetaFrame)
- 1991: Steffen (1998: Schmidt, 2000: C & C)

4. Partial Redundancy Elimination (PRE)

- compiler optimization (code motion)
- extension of total redundancy
  (availability for all/some paths)
- rigorous proof of correctness
- unidirectional
- Knoop, Steffen, Rüthing, 1992 - ...

5. Decidability Frontiers for Abstraction

- When, or how far, do we have to abstract?
  - Ignore guards!
- What abstractions lead to decidability?
  - Intervals!
- 2002 - ... : Müller-Olm, Seidl

6. Set-based Analysis of Programs over Trees

- set-based analysis of program P over lists or other trees:
  1. transform P ⇨ $P^{\#}$ such that: $post_P^{\#}$ = post of $P^{\#}$
  2. solve $P^{\#}$ ($P^{\#}$ is a set constraint)
- set constraint = fixpoint equation over tree grammars !
- *greatest solution characterizes non-termination*
- Reynolds'69, Jones'79, Heintze'89, C & C'92
- 1990 - ... : Frühwirth, Ganzinger, Seidl, Podelski, ...

7. Abstraction Refinement-based Model Checking

- completeness relative to widening
- ARMC: logic-based implementation
  (Andrey Rybalchenko's Model Checker)
- abstraction by linear-arithmetic constraint solver (and theory extensions)
- verification of networked train control systems
- 2000 - ... : Veith, Podelski, Rybalchenko, ...

8. Transition Invariants, Termination Analysis

- abstract least fixpoint not good for termination !
- transition invariant
  = summary
  = abstract least fixpoint
- transition invariant can prove termination
  ⇒ abstract least fixpoint good for termination !
- 2004 - ... : Cook, Podelski, Rybalchenko

What?

1. Grammar flow analysis
2. Worst-Case Execution Time (WCET)
3. Data flow analysis as model checking
4. Partial Redundancy Elimination (PRE)
5. Decidability frontiers for abstraction
6. Set-based analysis, set constraints
7. Abstraction refinement-based model checking (ARMC)
8. Transition invariants, termination analysis

Who?

Reinhard	Wilhelm	Saarbrücken
Bernhard	Steffen	Dortmund
Jens	Knoop	Wien
Markus	Müller-Olm	Münster
Helmut	Seidl	München
Andreas	Podelski	Freiburg

Ongoing

- Oldenburg - Saarbrücken - Freiburg
  AVACS *(DFG)*
  networked embedded systems
- Saarbrücken - Freiburg - Aachen MSRC - Redmond MSR
  Verisoft *(BMBF)*
  Microsoft Hypervisor
- München
  PUMA *(Graduiertenschule)*
  AI & types & theorem proving
- Saarbrücken
  hierarchical shape analysis
- Münster, München
  decidable abstractions of parallel programs
- Freiburg
  symbolic shape analysis, thread-modular analysis, ...
- …

Challenges in AI

… in: Concurrency, Search, Control Theory

Concurrency

- state explosion
- thread-modular verification
  … *Cartesian abstraction*
- synchronization
  … *intersection of automata*
- confluence

AI for AI Planning

- abstraction for good estimates of distance between states
- abstraction is good if search is fast
- abstract interpretation for artificial intelligence?

Control Theory

- system with non-linear dynamics (e.g. pendulum)
- safety
- stability

Towards a Conclusion …

What is specific about German AI research?

Sets vs. Set Theory

- mathematicians use sets
  (notation for formulating ideas)
- logicians do set theory
  (study of the infinite)

AI vs. AI Theory

- program analysis researchers use AI
  (notation for formulating ideas)
- semantics researchers do AI theory
  (study of the infinite)

- German AI researchers are perhaps rather among the AI users