Evolution of the s-semantics school

Marco Comini

incidentally taking about the others
Giorgio’s Group
History Description

- Martelli talk: First Age
- Giacobazzi talk: Middle Age
- this talk: New? Age (1993-today)
the Logic Programming Case

- Imperative paradigm canonical behavior: input/output

- Functional paradigm canonical behavior: ground (functional) values

- Logic paradigm canonical behavior: computed answers NOT logical truth
  - s-semantics approach
Denotational Semantics Principles

- Defined by syntax induction
  => Compositional

- Each syntactic category has a denotation
  - Goal-independence

- Can “represent” operational behavior
  - Behavioral program equivalence
  - Correctness & full abstraction
  - Explicitly exploit semantic properties
My s-semantics

“Slice”

My Origins

M. Gabbrielli PhD Thesis

R. Giacobazzi PhD Thesis

standard & non-standard semantics

Goal: build a Framework parametric w.r.t. a chosen (abstract) program property

• correct & fully abstract semantics

• goal-independent program denotation
Taxonomy of Behavior Properties

with Chiara Meo (and Giorgio obviously!)

- Semantic Framework for all abstractions of SLD-trees
- A priori properties of the derived semantics
Theoretical staff that should be useful for applications
Abstract Diagnosis

with Chiara Meo and Giuliana Vitiello: Algorithmic Detection of Program Errors
w.r.t. an abstract behavior property
effective for Nötherian Domains

it is based on one application of the abstract immediate consequences operator
Abstract Verification

- with Roberta Gori: Algorithmic Proof of Abstract Properties
- Abstract Park’s Induction Principle
  \[ T_P^\alpha(S) \leq S \]
- Can reconstruct all had-oc precise (non-effective) proposals
  - type checking is an instance
  - plenty of new hybrid effective (incomplete)
Other Paradigms

- with Moreno Falaschi, Maria Alpuente & others
  - Abstract Diagnosis for Term Rewriting Systems
  - Abstract Diagnosis/Verification for integrated paradigms (FLP) and multi-paradigm declarative languages (Curry, ...)
    - Polyhedral Domains will be essential for numerical language primitives

...we’re getting closer
Giorgio, did we complete “our mission”?
A Curry extension that

- provides programmer not only type specifications but also
  - groundness
  - depth\(k\)
  - ...

- supported by an implementation with mixed analysis/verification
  (inference/checking)
Evolution of the Group