The team:

- Patrick Cousot  Prof. ENS
- Radhia Cousot  DR2 CNRS
- Jérôme Feret  CR1 INRIA
- Antoine Miné  CR2 CNRS
- Xavier Rival  CR1 INRIA
- Julien Bertrane  ATER ENS

Recent former permanent members:
- David Monniaux (Vérimag, Grenoble)
- Bruno Blanchet (INRIA Cascade, ENS)
- Laurent Mauborgne (IMDEA, Madrid, Spain)

Recent former non-permanent members:
- Liqian Chen (NUDT, Changsha, China)
- Pietro Ferrara (ETH Zürich)
- Élodie-Jane Sims (CMU-Silicon Valley)

The problem and our approach

Safety and security analysis and verification of complex systems

- Complex systems are hard to design and verify
- Computer-aided analysis and verification of system designs and implementations can help
- We consider automatic and sound methods (providing correctness guarantees) for infinite state systems
Abstract interpretation: from theory to practice

- Design **formal semantics** of discrete / continuous / hybrid systems (describing the possible executions / evolutions of the system over time)
- Formalize the **specification** and mathematical inference of properties of such semantics
- Develop precise, efficient, and scalable **abstractions** of these semantic properties
- Apply (maybe with approximation) to the automatic inference of these semantic properties
- Develop automatic **static analyzers**
- Disseminate these analyzers (e.g. licenses)

### Abstract Domains

**Abstract domain**

- **Algebraic structure** (implemented as an analyzer module)

\[\langle A, \subseteq, \perp, \sqcup, \sqcap, \forall, \Delta, \bar{f}, \bar{b}, \bar{p}, \ldots \rangle\]

such that

- \(P, Q, \ldots \in A\) abstract properties
- \(\subseteq \in A \times A \rightarrow \mathcal{B}\) abstract partial order
- \(\perp, \top \in A\) infimum, supremum
- \(\sqcup, \sqcap, \forall, \Delta \in A \times A \rightarrow A\) abstract join, meet, widening, narrowing
- \(\bar{f} \in (x \times \mathcal{E}(x, f, p)) \rightarrow A \rightarrow A\) abstract forward assignment transformer
- \(\bar{b} \in (x \times \mathcal{E}(x, f, p)) \rightarrow A \rightarrow A\) abstract backward assignment transformer
- \(\bar{p} \in \mathcal{C}(x, f, p) \rightarrow A \rightarrow A\) abstract condition transformer
New numerical abstract domains (cont’d)

- Current research:
  - For efficiency, use the double description with floats while remaining sound
  - For precision, non-linear and sound abstract domains

Abstract Domains

(I) Relational Numerical Domains

New numerical abstract domains

- Challenges: the classical linear/polyhedral abstract domains are implemented with the double description method, using rationals, sound but inefficient
  - Inequalities: \( P = \{x|Ax = b, Cx \geq d\} \)
  - Generators: \( P = \{x|x = L\lambda + R\mu + V\nu, \mu, \nu \geq 0, \sum \nu = 1\} \)
- Results: sound and efficient algorithms using floats (with the inequality representation only) and generalizations (to non-convex relational domains)
Abstract Domains

(II) Symbolic domains

(a) Array content analysis

- **Challenge:** existing abstractions are *imprecise* or *do not scale up*
- **Results:** a new *parametric, efficient, expressive & scalable* abstract domain functor based on the idea of symbolic segmentation with dynamic bounds

- Has been combined with decision trees

(b) Shape analysis

- **Challenges:**
  - dynamic structures with destructive updates
  - complex structural invariants (red-black trees, call-stack, ...)
- **Results:** relational and expressive parametric abstract domains based on inductive properties and disjointness of regions
- **Ongoing work:** standardization of abstract domain interfaces to build composite abstractions (shape-shape and shape-numerical)
Abstract Domains

(II) Symbolic domains

(c) Combining algebraic and logical abstract domains

Logical & algebraic abstraction

- **Algebraic abstractions**: in inference tools, based on the iterated reduced product of a combination of abstract domains
- **Logical abstraction**: in verification tools, based on the Nelson-Oppen satisfiability procedure for combination of theories (SMT solvers)
- The approaches can be combined by
  - generalizing logical abstractions (widenings, …)
  - understanding Nelson-Oppen satisfiability procedure as an **iterated reduced product**

Temporal abstract domains

- **Constraints** to relate values and time
- **Integral bounding** to express quantitative properties (average...)
- **Fast abstract information propagation algorithms**

- **Value changes counting** to express stability specifications

Application: Static Analysis of Dynamic Systems

Static Analysis and Verification

(1) Quasi-synchronous systems

Example

- Two communicating redundant units, imperfect clocks & voter:

![Diagram]

- Result of the static analysis:

Counter-example ??? Specification proved

J. Bertrane. Static Analysis by Abstract Interpretation of the Quasi-synchronous Composition of Synchronous Programs. VMCAI 2005

Static Analysis and Verification

(II) Parallel programs
Static analysis of parallel software

- Critical embedded software is now parallel (e.g., IMA [Integrated Modular Avionics])
- Semantic challenges:
  - implicit communications through shared memory
  - weak memory consistency
  - strict priorities of real-time schedulers
- Abstraction challenges, to scale up while being precise:
  - history-sensitive abstractions of interleaved control flows
  - fixpoint strategies to compute interferences
- Ongoing work on the AstréeA prototype

Reachability analysis (of signaling pathways)

- Focus: models of signaling pathways described by collating biochemical interactions
- Challenge: such systems suffer from a huge combinatorial complexity in the number of different chemical species which may be formed at run-time
- Results: new scalable and precise analyses of the reachable chemical species
- Applications:
  - debug models during modeling
  - precompute properties so as to enable fast stochastic simulation
  - automatic simplification of the model
  - compute an idiomatic description of the systems

Model reduction

- Challenge: quantitative (ODEs or stochastic) semantics are hard to compute, due to the combinatorial complexity (∼10^20 variables).
- Results: introduction of an approximation of the control flow between regions of chemical species, so as to compute exact projections of these semantics.

Static Analysis (III) Biological systems

- Reachability analysis of signaling pathways
- Model reduction
- Static analysis of parallel software
Selected software

APRON Library

- Freely available (LGPL) library of numerical domains developed with INRIA PopArt, including Interproc, a web-based sample analyzer for demonstration, teaching and prototyping common AI-based API for numerical abstract domains
- Reference implementations of classical domains (intervals, polyhedra, octagons, ...)
- Easy to prototype new analyses or domains (e.g., interval polyhedra) with support for
  - integer, rational, machine-integer and float data-types
  - linear, non-linear and float expressions
  - C, C++, Java, OCaml

Astrée

- Routinely used by Airbus France on large software
- Exploitation license (AbsInt Angewandte Informatik GmbH)

AstréeA

- In progress
- Challenging application to a large parallel software at Airbus France
OpenKappa

http://www.kappalanguage.org

- Modelization environment for rule-based modeling
- Qualitative / differential / stochastic semantics
- Automatic and exact model reduction
- Static analysis of reachable chemical species
- Collaborative platform with other tools (simulation, causal structure, ...)

Publications (2010 – 2011)

- J. Bertrane, P. Cousot, R. Cousot, J. Feret, L. Mauborgne, A. Miné, X. Rival. Static Analysis and Verification of Aerospace Software by Abstract Interpretation. ASABE Interdisciplinary Aerospace 2010. (Best Paper Award)

Dissemination

- In the academic world (tutorials, summer schools, invited talks, conference organization, ...)
- In the industrial world (conferences, training, ...)

- In the academic world (tutorials, summer schools, invited talks, conference organization, ...)
- In the industrial world (conferences, training, ...)
Services to the community

General Chairs

SAS 2010
& workshops: NSAD, SASB, TAPAS, 2010

Program Chairs

ACM-POPL 2013, SAS 2010
& workshops: NSAD 2010, SASB 2010/11, TAPAS 2010

Program Committees

ACM-PLDI 10, ACM-POPL 11, APLAS 11, ESOP 10/11/12
HSCC 09/11, IEEE-SEFM 09, SAS 11, VMCAI 09/10/11
CMSB 11,… & workshops: LOLA 12, SSCPS 11, CS2Bio 10,…

Ongoing contracts

- FRAE
- Ascert: Analyses Statiques Certifiées, with INRIA (Celtique, Gallium, Pop Art), 2009–12
- Sardanes: Sémantique, analyse et transformation des applications numériques embarquées synchrones, with Perpignan & Brest Univ., 2009–12

- ANR
- Industry (Airbus France)
- Anastasy: Analyse statique et dynamique, 2010–14
- Royal Society (UK)
- Obfuscation by abstract interpretation: Imperial College, 2010–12

Starting contracts

- EUROPE (ARTEMIS)
Main ongoing collaborations

Main ongoing academic collaborations

INRIA
- GALLIUM (X. Leroy), CELTIQUE (D. Pichardie, S. Blazy), Ascert project
- POP ART (B. Jeannet), Apron library

France
- Brest Univ. (D. Massé), Perpignan Univ. (M. Martel), Sardanes project

Europe
- Imperial College (C. Hankin), Royal Society grant
- IMDEA (L. Mauborgne)
- National Univ. of Defense Techn. (L. Chen, J. Wang)

China

USA
- Harvard University (W. Fontana)
- CMU, Cornell U., NYU, U. of Maryland, Stony Brook, NSF CMACS (Computational Modeling and Analysis for Complex Systems)
- Microsoft Research Redmond
  - Verification of code contracts in CCCheck (aka Clousot) (now distributed with Visual Studio Ultimate)
  - Design of scalable abstract domains for
    - Array content analysis
    - Contract precondition inference from assertions

Main ongoing industrial collaborations

AbsInt Angewandte Informatik

• Exploitation license of ASTRÉE  www.absint.com/astree/

• 3 full-time engineers, everyday collaboration

Astrium

• Verification of space software

Sagem

• Analysis of inertial unit software

Airbus

• Verification of parallel programs, mainly for a large program developed by Airbus France (and next many other parallel programs)
Positioning with respect to industry

- **Strong cooperation** with Airbus software production (started 1999)
- **Swarming cooperation**: Microsoft research (started 2009) on verification of actual software (by CCCheck),
- **Lightweight cooperations** on limited projects (Astrium, Sagem, Daimler,...) and many more to come with the starting MBAT project
- **Technological transfer**: AbsInt
- Cooperation on contracts is relatively easy, effective influence is more difficult and long term

Foundations

- More complex **semantics** e.g.
  - **Fairness** in concurrency
  - **Discrete/differential/hybrid semantics**
- More **complex properties** of infinite state systems, e.g. on-going work on
  - **Liveness** (e.g. termination)
  - **Language-based security**
  - **Quantitative properties**
- More **infinite state systems** e.g.
  - **Parallel/distributed systems**
  - **Closed-loop analysis** of computer-controlled discrete/continuous cyber-physical systems

Follow-up of Astrée industrial exploitation

- **Support for** the Astrée analyzer
- **New abstract domains** for new application domains
- **New abstractions** for alarm investigation

Objectives for the next four years
Numerical abstract domains

• Support of the APRON library
• Robust implementations of prototype numerical abstract domains

Symbolic abstract domains

• New abstract domains for analyzing properties of dynamic data structures (segmentation, partitioning, shape, ...)
• Combination of partitioning & shape analysis abstract domains
• Design and implementation of a library of abstract domains and combination operations
• Implementation in static analyzers (AstréeA)

Abstraction of biological systems

• Study the relation between the differential and stochastic semantics (infinite set of polymers)
• Improve model reduction
• Extend the rule-based language (diffusion, compartments)
• Design semantics and their abstractions for multiscale systems (large/small populations, slow and rapid reactions)
• Modeling of more complete composite systems (cooperating pathways)

Abstraction of parallelism

• Flow-sensitive, history-sensitive, or relational inter-thread static analysis for proving the absence of runtime errors in parallel embedded software
• Consistent memory model (supporting for volatile accesses)
• Scalable & precise analysis of complex data structures
• Scalable & precise interference analysis
• Real-time, scheduling with dynamic priorities (e.g. priority ceiling protocols), ...
• Long term goal: achieve a maturity level similar to that of Astrée and start the industrialization of AstréeA.
More details

• see


The End