Computing Numerical Invariants using Abstract Interpretation based Static Analysis

Khalil Ghorbal
Laboratory for the **M**od**e**lling and **A**nalysis of **I**nteracting **S**ystems
khalil.ghorbal@cea.fr

Doctoriales ParisTech 2009
20 - 26 Sep, 2009
Fréjus, FRANCE

1. *Software Validation and Verification*

Software Testing
[Myers 1979]

- Widely used (Verification and Qualification)
- derived from the specification, model-based testing, ...
- code coverage metrics
- Still *Program testing can be used to show the presence of bugs, but never to show their absence!* (E. Dijkstra (1972))

Computer-assisted Proof

- Property of interest is seen as a Theorem (logically valid) using Predicates calculus
- Powerful, Complete formal proof of the 4 colors theorem (1976)
- Tools: Coq, SMV ...
- Issue: Decidability, termination, hardly automated and not scalable

Model Checking
[Clarke, Emerson, and Sifakis 1981]

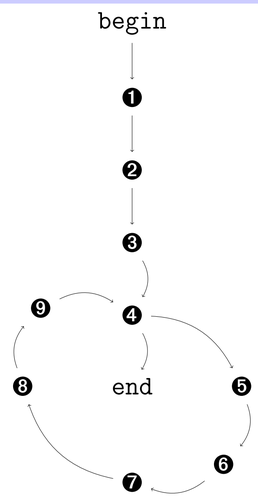
- Derive a formal model from the real program (temporal logic automaton ...)
- Prove the needed properties on the model
- Tools: BLAST, SPIN, ...
- Issue : State explosion problem

Abstract Interpretation
[Cousot and Cousot 1976]

- Semantic formalized as a fix-point of a monotonic operator in a partially ordered structure,
- Fully automated,
- Industrial tools exists : Polyspace Verifier (MathWorks), Astrée (ENS), Fluctuat (CEA), aIT (ABSINT), ...
- Issue: find the *suitable* abstract domain for the properties of interest.

2. *Concrete Semantics*

```
#Second order Filter
begin
En = [-1,1]; ❶
Sn1 = 0; ❷
Sn2 = 0; ❸
i = 0; ❹
while (i<100) ❺ do
  Sn = h(En, Sn1, Sn2); ❻
  Sn2 = Sn1; ❼
  Sn1 = Sn; ❽
  En = [-1,1]; ❾
  i = i+1; ❹
done;
end
```



Control Flow Graph

Equations System

$$\begin{cases} \mathcal{X}_0 = [\![\mathcal{V} \to \mathbb{I}]\!], \text{ where }, \mathbb{I} = \{\mathbb{N}, \mathbb{Q}, \mathbb{R}\} \\ \mathcal{X}_1 = [\![E_n \leftarrow [-1,1]]\!](\mathcal{X}_0) \\ \mathcal{X}_2 = [\![S_{n-1} \leftarrow 0]\!](\mathcal{X}_1) \\ \mathcal{X}_3 = [\![S_{n-2} \leftarrow 0]\!](\mathcal{X}_2) \\ \mathcal{X}_4 = [\![i \leftarrow i+1]\!](\mathcal{X}_9) \\ \mathcal{X}_5 = [\![i < 100]\!](\mathcal{X}_4) \\ \mathcal{X}_6 = [\![S_n \leftarrow h(E_n, S_{n-1}, S_{n-2})]\!](\mathcal{X}_5) \\ \mathcal{X}_7 = [\![S_{n-2} \leftarrow S_{n-1}]\!](\mathcal{X}_6) \\ \mathcal{X}_8 = [\![S_{n-1} \leftarrow S_n]\!](\mathcal{X}_7) \\ \mathcal{X}_9 = [\![E_n \leftarrow [-1,1]]\!](\mathcal{X}_8) \end{cases}$$

- $D = (\wp(\mathcal{V} \to \mathbb{I}), \subseteq, \cup, \cap, \emptyset, (\mathcal{V} \to \mathbb{I}))$ is a **complete lattice**
- each operator $\mathcal{X}_l \mapsto \mathcal{F}(\mathcal{X})$ is monotonic

➙ **Tarski Theorem** ensures the existence of a least fixpoint for $\mathcal{F}$

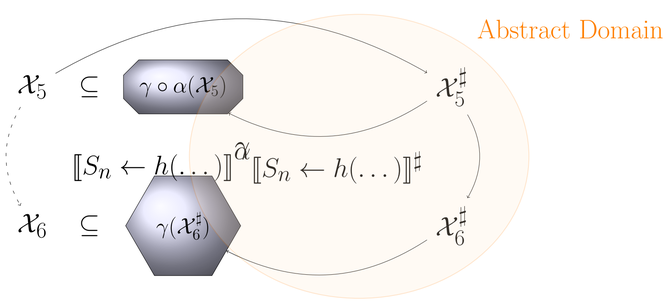
- If the transfer functions ($[\![.]\!]$) are continuous

➙ **Kleene Iteration Technique** reaches the least fixpoint

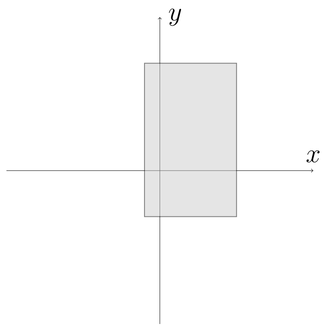
Issues :

☢ $\wp(\mathcal{V} \to \mathbb{I})$ is non representable in finite memory,
☢ $[\![.]\!]$ are non computable,
☢ Iterations over the lattice may be transfinite.

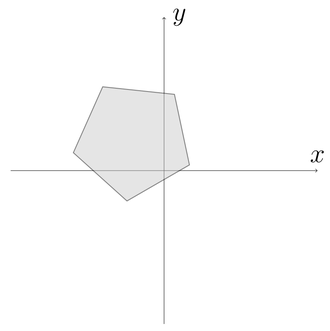
3. *Abstract Interpretation Based Static Analysis*



Abstract the concrete semantics to get a computable over approximation



Non Relational Domain
$x \in I_x$, $y \in I_y$



Relational Domain
$\lambda x + \mu y \leq c$

4. *Affine Forms Abstract Domain* [Goubault and Putot: 06, 08]

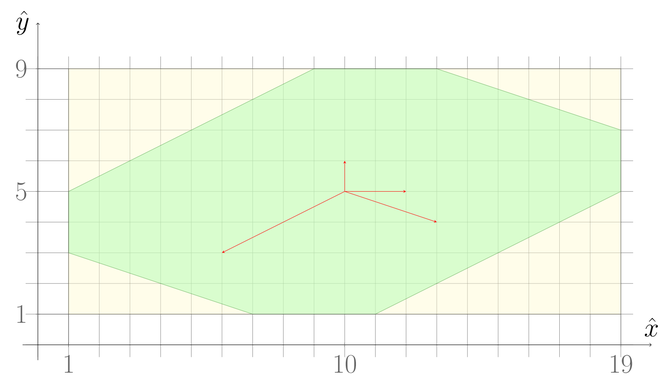
[J. L. D. Comba and J. Stolfi: 93] :

$$\hat{x} = \alpha_0^x + \sum_{i=1}^{n} \alpha_i^x \epsilon_i$$

➙ **shared** noise symbols express **implicit dependencies** between variables

example :

$$\begin{pmatrix} \hat{x} = 10 & -4\epsilon_1 & & +2\epsilon_3 & +3\epsilon_4 \\ \hat{y} = 5 & -2\epsilon_1 & +1\epsilon_2 & & -1\epsilon_4 \end{pmatrix}$$

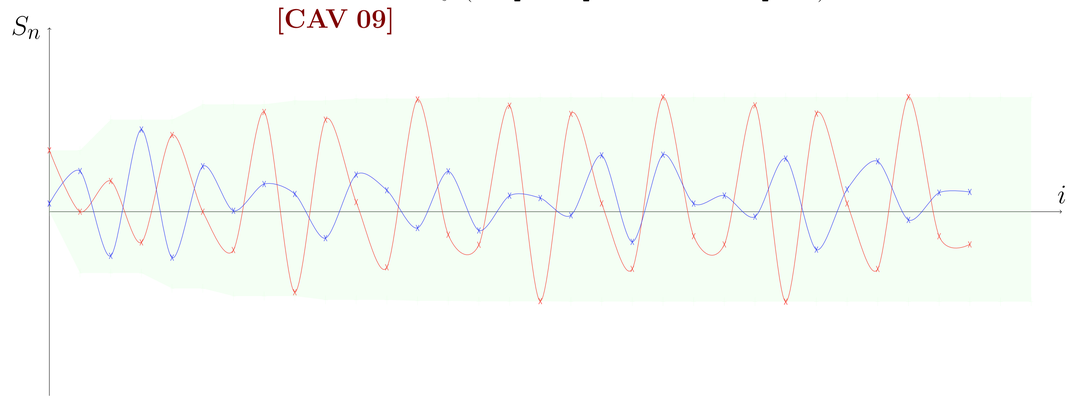


5. *Contributions*

➙ ATV (Astrium ST) Case Study **[DASIA 09]**

➙ Use of optimization techniques (SemiDefinite Programming) in abstract transfer functions,

➙ Development of an abstract domain, *Taylor1+* (Licence GPL), based on affine forms, as a new domain of APRON Library (`http://apron.cri.ensmp.fr`) **[CAV 09]**



Ongoing work: Design of an abstract domain with support of constraints over noise symbols in order to improve the abstraction of the *test* transfer functions.