Research Introduction

Louis Rustenholz[1,2]

[1]Universidad Politécnica de Madrid (UPM), Spain
[2]IMDEA Software Institute, Spain

February 6th, 2025
PLIS Group, Roskilde University

CLIP Lab @ IMDEA Software Institute, Madrid

IMDEA Software Institute

Madrid Institute for Advanced Studies
Focus: *Safe*, *Reliable* and *Efficient* Software

Program Analysis and Verification,
Cryptography and Cybersecurity,
Languages and Compilers, ...

CLIP Lab

The **C**omputational logic, **L**anguages,
Implementation, and **P**arallelism **Lab**oratory

Manuel Hermenegildo

Pedro Lopez-Garcia

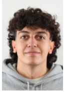
Jose F. Morales

Manuel Carro

Daniel Jurjo

Daniela Ferreiro

Louis Rustenholz



Marco Ciccale

Paula Corral

Kirelys Lugo

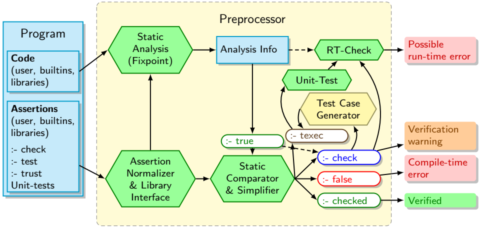
An Extensible Logic Programming Language,
designed to make full use of advanced
Analysis, **Verification** and **Optimisation**

CLIP Lab @ IMDEA Software Institute, Madrid

```
:- module(_, [nrev/2], [assertions,fsyntax,nativeprops]).

:- entry nrev/2 : {list, ground} * var.

:- pred nrev(A,B) : list(A) => list(B)
   + ( not_fails, is_det, steps_o( length(A)) ).

nrev( [] )    := [].
nrev( [H|L] ) := ~conc( ~nrev(L),[H] ).


:- pred conc(A,B,C) + ( terminates, is_det, steps_o(length(A)) ).

conc( [],     L ) := L.
conc( [H|L], K ) := [ H | ~conc(L,K) ].
```

Preprocessor Option Browser

Use Saved Menu Configuration (menu_last_config)	:	none
Menu Level (menu_level)	:	naive
Action (inter_all)	:	check_assertions
Analysis Domain (assert_ctcheck)	:	manual
Modules to Check (ct_modular)	:	curr_mod
Customize Analysis Flags (check_config_ana)	:	on
Analyze Non-Failure (ana_nf)	:	nf
Analyze Numeric (ana_num)	:	none
Analyze Cost (ana_cost)	:	none
Analyze Determinism (ana_det)	:	det
Analysis entry (entry_point)	:	entry
Incremental (incremental)	:	off
Intermodular (intermod)	:	off
Report Non-Verified Assrts (ass_not_stat_eval)	:	warning
Generate Certificate (gen_certificate)	:	off
Generate CT Checking Intervals (ctchecks_intervals)	:	on
Generate Output (menu_output)	:	on
| Output Language (output_lang)	:	source
| Include Program Point (pp_info)	:	off
| Multi-variant Analysis Results (vers)	:	off
| Collapse Versions (collapse_ai_vers)	:	on




```
#pragma check fir(xn, coeffs, state, ELEMENTS) :
          (1 <= ELEMENTS && energy <= 416.0)
#pragma true fir(xn, coeffs, state, ELEMENTS) :
          ( energy >= 3.35*ELEMENTS + 13.96 &&
            energy <= 3.35*ELEMENTS + 14.4 )
#pragma checked fir(xn, coeffs, state, ELEMENTS) :
          (1 <= ELEMENTS && ELEMENTS <= 120 && energy <= 416.1)
#pragma false fir(xn, coeffs, state, ELEMENTS) :
          (121 <= ELEMENTS && energy <= 416.1)

int fir(int xn, int coeffs[], int state[], int ELEMENTS)
{ unsigned int ynl; int ynh;
  ynl = (1<<23); ynh = 0;
  for(int j=ELEMENTS-1; j!=0; j--) {
      state[j] = state[j-1];
      {ynh, ynl} = macs(coeffs[j], state[j], ynh, ynl); }
  state[0] = xn;
  {ynh, ynl} = macs(coeffs[0], xn, ynh, ynl);
  if (sext(ynh,24) == ynh) {
      ynh = (ynh << 8) | (((unsigned) ynl) >> 24);}
  else if (ynh < 0) { ynh = 0x80000000; }
  else { ynh = 0x7fffffff; }
  return ynh; }
```

PhD Research Topic

Abstract Interpretation-based **Static Analysis of Cost Properties** (resource consumption) of software, with a particular focus on **Energy Consumption**

Cost Analysis: Bounds on Resource Consumption

Program $\mathcal{P}$

```
def foo(lst, x) =
    <body>
```

\+

Cost Model

Cost Analysis

Cost Bounds

$$\mathbf{f}_{\mathbf{lb}}(\vec{n}) \leq \mathbf{Cost}_{\texttt{foo}} \leq \mathbf{f}_{\mathbf{ub}}(\vec{n})$$

Size Abstraction

$n_1 = \texttt{len(lst)}$, $n_2 = \texttt{val(x)}$, ...

WCET,
Side-Channel

Parallelisation,
Scheduling

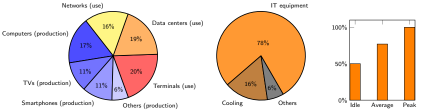
Transparency, Optimisation

PhD Research Topic

Abstract Interpretation-based **Static Analysis of Cost Properties** (resource consumption) of software, with a particular focus on **Energy Consumption**

Key personal motivation, in terms of societal impact: **carbon footprint of IT**.

Share of global emissions:
2.5% $\rightsquigarrow$ **5%** in the last ten years.



Energy Usage in IT. By subfield, and case studies on a data center.

Energy consumption *of programs* is relevant and understudied by carbon audit experts.

Previous collaboration: **ENTRA project**
ENergy **TRA**nsparency

Also in contact with associations on evolution of carbon norms and regulations.

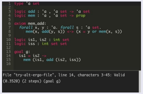
Personal Research Interests and Background

Background

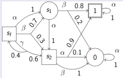
- **2016–2018**, Preparatory Classes, Mathematics / Physics
  (Louis-le-Grand, Paris, France)
- **2018–2022**, École Polytechnique, MSc
  Multidisciplinary school of Engineering and Public Administration
  Major: Theoretical Computer Science and Mathematics
- **2021–2022**, MPRI, Formal Methods and Programming Languages, MSc
  Joint degree ENS/Polytechnique/Université de Paris

Selected Past Projects (pre-PhD)

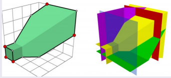
Computer Science



SMT-Solver Alt-Ergo

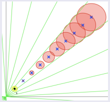


Categorical Approaches
on Model Checking



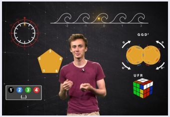
Abstract Interpretation of Neural Networks
using Tropical Geometry [SAS21]

Mathematics Outreach



Undergraduate book on S-unit equations
Geometric and Algebraic Number Theory

High-school
Research Tournaments



High-school MOOCs

Personal Research Interests and Background

Current Research Interests

Static Analysis of Programs and Systems

- Automated discovery of truths, beyond verification: *automated understanding and improvement*
- **Abstract Interpretation**
- **“Bounding the behaviour of systems”,** beyond software: hardware/networks, or even cyberphysical systems, biochemical reaction networks

Applied Semantics

- Denotational Approaches
- Abstraction Methods
- **Order Theory**

Geometric Viewpoints

(Generalised) Recurrence Equations

i.e. functional discrete equations, $f : \mathbb{Z}^k \to \mathbb{R}$.

- Accurate, executable abstractions of systems
- Non-linear invariant inference
- Non-standard equations (CAS vs programs)

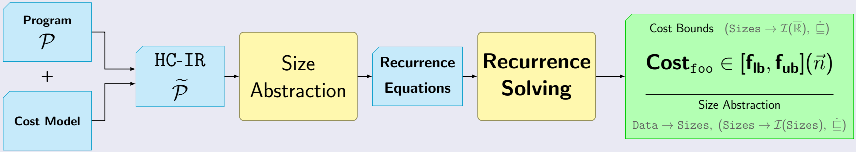
Dynamic/Static Cooperations

- Machine Learning + Formal Methods
- Observation + Reasoning
- (Optimisation/Search) + (Logic/Order)

PhD Research Topic

Abstract Interpretation-based **Static Analysis of Cost Properties** (resource consumption) of software, with a particular focus on **Energy Consumption**

Our cost analysis pipeline



Challenges

- Fine-grained (energy) cost models, hardware behaviours, etc.
  `[WIP]`. C.f. [ENTRA], WCET literature.
- Complex control flows, general programs
  [RustenholzMSc22, TPLP24, SAS24]. `[WIP]`.
- State of the art in recurrence solvers
  [op.cit.].
- Improve size abstractions
  [RidouxMSc24]. `[WIP]`.

Generalised Recurrence Equations

i.e. functional discrete equations, $f : \mathbb{Z}^k \to \mathbb{R}$. (Undecidable: must aim for bounds.)

CAS vs Generalised Equations

$$f(n) = a(n) \cdot f(\mathbf{n-1}) + b(n)$$

complex factors, simple recursive calls

(classical computer algebra)

$$f(\vec{n}) = \begin{cases} \dots \\ a \cdot f(\phi(\vec{n})) + b(\vec{n}) \\ \dots \end{cases}$$

simple factors, complex recursive calls
complex control flow
(program analysis)

Conditionals, non-linear recursion, ...

$$f(\vec{n}) = \begin{cases} \dots \\ \sum_{j=1}^{k_i} \left(a_{i,j}(\vec{n}) \cdot f(\phi_{i,j}(\vec{n}))\right) + b_i(\vec{n}) & \text{if } \varphi_i(\vec{n}) \\ \dots \end{cases}$$

where $a_{i,j} \geq 0$, and $\phi_{i,j}, b_i, \varphi_i$ arbitrary.

(Unbounded) max/min

$$f(n) = \begin{cases} 0 & \text{if } n \leq 2 \\ n + \max_{1 \leq k \leq n-1} f(k) + f(n-k) & \text{if } n > 2 \end{cases}$$

Self-composition

$$f(n) = \begin{cases} f(f(n-1)) + 1 & \text{if } n > 0 \\ 0 & \text{if } n = 0 \end{cases}$$

Equation solving as pre/postfixpoint search [SAS24]

Equations $\leftrightarrow$ Operators   Solutions $\leftrightarrow$ Fixpoints   Bounds $\leftarrow$ Pre/Postfixp

Example

Search $f : \mathbb{N}^2 \to \mathbb{R}$ such that

$$f(n,c) = \boxed{\begin{cases} f(n-1,0)+n+300 & \text{if } n>0 \text{ and } c \geq 100, \\ f(n-1,c+1)+n & \text{if } n>0 \text{ and } c<100, \\ c & \text{if } n=0, \end{cases}}$$

➔

Example

Search $f : \mathbb{N}^2 \to \mathbb{R}$ such that $f = \Phi f$, i.e. $f \in \mathrm{Fixp}(\Phi)$, where

$$\Phi : (\mathbb{N}^2 \to \mathbb{R}) \to (\mathbb{N}^2 \to \mathbb{R})$$

$$f \mapsto (n,c) \mapsto \begin{cases} f(n-1,0)+n+300 & \text{if } n>0 \text{ and } c \geq 100, \\ f(n-1,c+1)+n & \text{if } n>0 \text{ and } c<100, \\ c & \text{if } n=0. \end{cases}$$

For a complete lattice, order $\mathcal{D} \to \mathbb{R}$ pointwise, and extend to $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$

Theorem

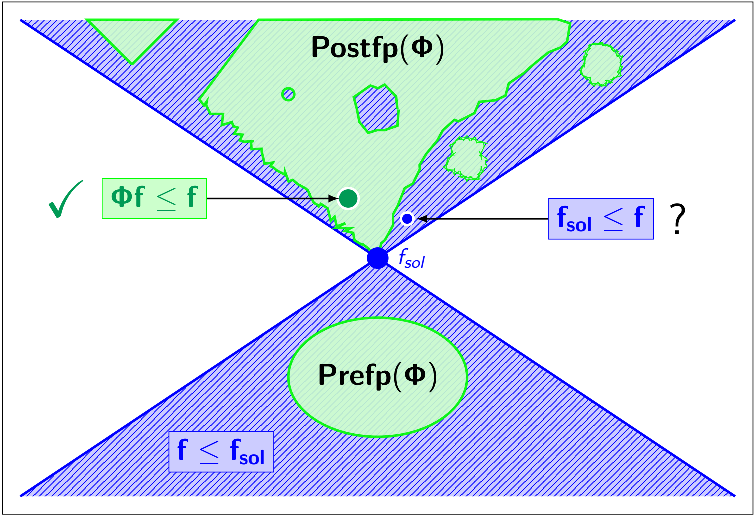
Let $\Phi : (\mathcal{D} \to \overline{\mathbb{R}}) \to (\mathcal{D} \to \overline{\mathbb{R}})$ *be a* **monotone equation**.

- *If* $f \in \mathrm{Postfp}(\Phi)$, *i.e.* $\Phi f \leq f$, *then* $\mathrm{lfp}\,\Phi \leq f$.
- *If* $f \in \mathrm{Prefp}(\Phi)$, *i.e.* $f \leq \Phi f$, *then* $f \leq \mathrm{gfp}\,\Phi$.

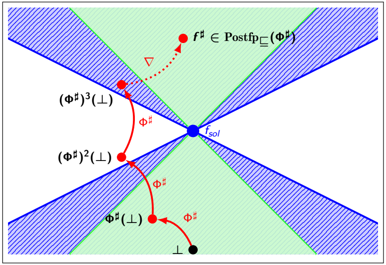
Insight: cost equations are typically monotone for this pointwise order.

Equation solving as pre/postfixpoint search [SAS24]

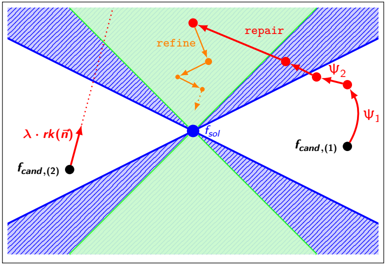
Equations $\leftrightarrow$ Operators   Solutions $\leftrightarrow$ Fixpoints   Bounds $\leftarrow$ Pre/Postfixp



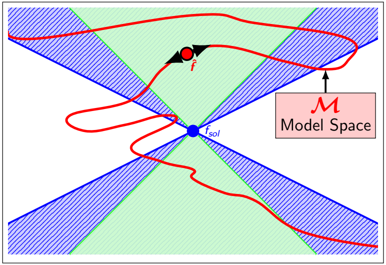
Equation solving as pre/postfixpoint search [SAS24]



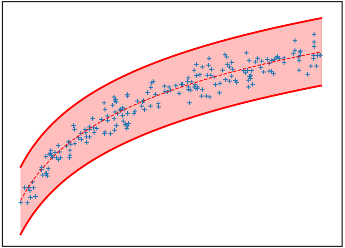
Abstract Interpretation



Geometry-based expression **Repair**



Search on subvarieties: **Templates**, ∀-elim



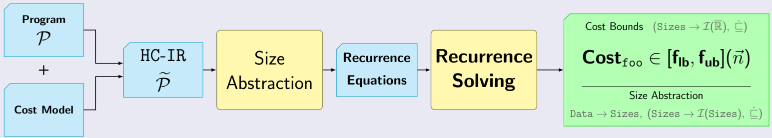
Constrained **Optimisation**,
with *provability constraints*

Interval Recurrence Equations @ Roskilde

Generalised signatures (richer domains/codomains):

- Classical: $\mathbb{Z}^k \to \mathbb{R}$
- **Intervals**: $\mathbb{Z}^k \to \mathcal{I}(\mathbb{R})$, $\quad \mathcal{I}(\mathbb{Z})^k \to \mathcal{I}(\mathbb{R})$, $\quad \mathcal{I}(\mathbb{Z})^k \to \mathcal{I}(\mathbb{Z})^r$
- With environment: $\mathsf{L}^\sharp \to \mathcal{I}(\mathbb{Z})^k \to \mathcal{I}(\mathbb{R})$

Cost analysis pipeline



- Novel size abstractions $\rightsquigarrow$ **interval** equations more appropriate.
- Non-monotonic equations? Set/**interval** approach. (CPS/CRN, virtual counters, ...)

To solve them, we want some **bound separation** technique.

$$\Phi_{\mathcal{I}} \in \big((\mathbb{Z}^k \to \mathcal{I}(\mathbb{R})) \to (\mathbb{Z}^k \to \mathcal{I}(\mathbb{R}))\big) \;\boldsymbol{\rightarrow}\; (\Phi_{lb}, \Phi_{ub}) \in \big((\mathbb{Z}^k \to \mathbb{R}) \to (\mathbb{Z}^k \to \mathbb{R})\big)^2$$

Interval Recurrence Equations @ Roskilde

To solve them, we want some **bound separation** technique.

$$\Phi_{\mathcal{I}} \in \left((\mathbb{Z}^k \to \mathcal{I}(\mathbb{R})) \to (\mathbb{Z}^k \to \mathcal{I}(\mathbb{R}))\right) \ \boldsymbol{\rightarrow}\ (\Phi_{lb}, \Phi_{ub}) \in \left((\mathbb{Z}^k \to \mathbb{R}) \to (\mathbb{Z}^k \to \mathbb{R})\right)^2$$

- Typical approach is conservative (and not yet well-formalised), especially in presence of some non-monotonicity.
- John has discovered a phenomenon where it is possible to preserve more precision.

  $$\Phi_{\mathcal{I}} \ \boldsymbol{\rightarrow}\ (\Phi_{\downarrow}, \Phi_{\uparrow})$$

  However, its soundness is not yet proven, and seems to rely on the structure of $\Phi_{\mathcal{I}}$.
  Monotonicity properties. Synchronised non-determinism?
- We will investigate the principles underlying this phenomenon: find out correct assumptions, and conclude with a proof.

Thank you for hosting me!

Questions?

(Now or later this month)