Constraint-Based Static Analysis of Programs

Zohar Manna
Stanford University

Joint work with
Aaron Bradley, Michael Colón, Sriram Sankaranarayanan, Henny Sipma
Motivation

**Objective:** To extract information about the program behavior from the program text

- invariants
- ranking functions (termination)
- temporal properties
- ...

Trivial Example

integer $i, j$ where $i = 2 \land j = 0$

$\ell_0: \text{ while (...) do}$

\[
\begin{cases}
\text{if (...) then} & \text{ } \\
& i := i + 4 \\
\text{else} & \text{ } \\
& (i, j) := (i + 2, j + 1)
\end{cases}
\]

$i \geq 2$, $j \geq 0$, and $i - 2j \geq 2$ are invariants.

Objective: To obtain such invariants automatically
Buffer Overflow Analysis

1: int *a = malloc( sizeof(int) * n);
2: int i,j,k;
3: for(i=0; i<n; ++i)
4:     for(j=0; 2*j<=i; ++j)
5:         if (a[i] <= a[2*j+1])
6:             ....
7:     ...

Check bounds for each array access.

0 ≤ i < n?

0 ≤ 2j + 1 < n?
Division by Zero

1: double a, b, c
2: ....
3: while ( b > 0 || c >= 0 ) {
4:     a = a + b/(c+b-1);
5: ....
6: }

$c + b - 1 > 0$

Prove every divisor non-zero.
Deadlock Freedom

Is this Petri net deadlock free? [Zhou et al. : 1992]
Termination

\begin{verbatim}
local i, j, k: integer
while ( i \leq 100 \land j \leq k ) do
    (i, j, k) := (j, i + 1, k - 1)
od
\end{verbatim}

Termination is proved by the \underline{ranking function}

\[ -i - j + k \]

which decrements by 2 each iteration.

We need the \underline{supporting invariant}

\[ -i - j + k \geq 0 \]

to establish termination.
Preliminaries: Transition Systems

**integer** $i, j$ where $i = 2 \land j = 0$

$l_0$: while true do

\[
\begin{align*}
  i & := i + 4 \\
  \text{or} \\
  (i, j) & := (i + 2, j + 1)
\end{align*}
\]

Transition system:

\[
\begin{align*}
  \langle L : \{l_0\}, & \quad V : \{i, j\}, \quad T : \{\tau_1, \tau_2\}, \quad \Theta : (i = 2 \land j = 0), \quad L_0 : l_0 \rangle \\
  \text{locations} & \quad \text{variables} & \quad \text{transitions} & \quad \text{initial condition} & \quad \text{initial location}
\end{align*}
\]

with

\[
\begin{align*}
  \tau_1 & = \left\langle l_0, l_0, \rho_{\tau_1} : (i' = i + 4 \land j' = j) \right\rangle \\
  \tau_2 & = \left\langle l_0, l_0, \rho_{\tau_2} : (i' = i + 2 \land j' = j + 1) \right\rangle
\end{align*}
\]

transition relation
Transition System: Execution

\[ \langle L, V, \mathcal{T}, \Theta, L_0 \rangle \]

**Computation:** Infinite sequence of states \( \langle l_i, x_i \rangle \)

\[ \langle l_0, x_0 \rangle \xrightarrow{\tau_1} \langle l_1, x_1 \rangle \xrightarrow{\tau_2} \langle l_2, x_2 \rangle \rightarrow \cdots \]

such that

- Initial Condition satisfied
  \[ l_0 = L_0 \land \Theta(x_0) \]

- Consecutive states \( \langle l_i, x_i \rangle \rightarrow \langle l_{i+1}, x_{i+1} \rangle \) satisfy some transition
  \[ \tau_k : \langle l_i, l_{i+1}, \rho_{\tau_k}(x_i, x_{i+1}) \rangle \]
**Invariants**

Assertion $\psi$ is an **invariant of $P$** iff it is true at all the **reachable states** of $P$.

**Example:**
- **reachable states**: $\langle \ell_0, 2 \rangle, \langle \ell_0, 4 \rangle, \langle \ell_0, 8 \rangle, \langle \ell_0, 16 \rangle, \ldots$
- **invariant**: $at_{\ell_0} \rightarrow x$ is even
An assertion $\varphi$ is **inductive assertion** iff

**Initiation** The assertion is true initially,

$$\Theta \models \varphi$$

**Consecution** For every transition $\tau$, if $\varphi$ is true before $\tau$ is taken, then it is true after taking $\tau$,

$$\varphi \land \rho_\tau \models \varphi'$$
Invariant vs. Inductive Assertion

Inductive Assertion $\Rightarrow$ Invariant
Invariant $\Rightarrow$ Inductive Assertion

To prove invariant $\varphi$, find inductive assertion $\psi$, satisfying initiation + consecution, such that

$$\psi \rightarrow \varphi$$

To find invariants, we really search for inductive assertions.
Example

integer $i, j$ where $i = 2 \land j = 0$

while true do

\[
l_0 : \begin{cases} 
  i := i + 4 \\
  \text{or} \\
  (i, j) := (i + 2, j + 1)
\end{cases}
\]

Is $\varphi : i - 2j \geq 2$ an invariant?
Example: Continued

Show

Initiation: \( (i = 2 \land j = 0) \implies i - 2j \geq 2 \)

Consecution: Two transitions \( \tau_1 \) and \( \tau_2 \).

\[
\begin{align*}
&i - 2j \geq 2 \land (i' = i + 4 \land j' = j) \implies i' - 2j' \geq 2 \\
&i - 2j \geq 2 \land (i' = i + 2 \land j' = j + 1) \implies i' - 2j' \geq 2
\end{align*}
\]

\( \varphi : i - 2j \geq 2 \) is inductive assertion \( \implies \) invariant.
Preliminaries: Assertion Domains

Assertion Domain:
A class of assertions, containing: the initial assertion, the transition relations, and the target invariants.

Common Examples:

1. Linear Equalities over Reals
   \[ 2i + j - 3 = 0, \]

2. Linear Inequalities over Reals
   \[ 2i + 3j - 2 \leq 0, \]
   (convex polyhedra)

3. Integer Arithmetic
   \( (\exists k)[2j = i \land i = j + k], \)
   (Presburger Arithmetic)

4. Multiplication over Reals
   \( (\exists a, b)[ai^3 + bi = j \lor i = j^2]. \)
Linear Invariants

Invariants in the domain of Linear Equalities or Inequalities are called \textit{Linear Invariants}.

All variables and constants are assumed to range over the reals.

\textbf{Example:}

\begin{verbatim}
integer i, j where \( i = 2 \land j = 0 \)
\ell_0 : while (...) do
  if (...) then
    i := i + 4
  else
    (i, j) := (i + 2, j + 1)
\end{verbatim}

\( \varphi_1 : i \geq 2 \) and \( \varphi_2 : i - 2j \geq 2 \) are linear invariants.

“\( i \) is even” is an invariant but not linear.
Linear Relations Analysis
**Static Analysis: Traditional Approach**

**Goal:** Given a program, find invariants

Symbolic forward simulation to obtain an overapproximation of the reachable state space (i.e. invariants)
Forward Propagation

\[ \mathcal{F}_0 : \Theta \]
\[ \mathcal{F}_1 : \mathcal{F}_0 \lor (\bigvee_{\tau \in \mathcal{T}} \text{post}_\tau(\mathcal{F}_0)) \]
\[ \mathcal{F}_2 : \mathcal{F}_1 \lor (\bigvee_{\tau \in \mathcal{T}} \text{post}_\tau(\mathcal{F}_1)) \]
\[ \vdots \]

until \[ \mathcal{F}_{i+1} \rightarrow \mathcal{F}_i \] (use widening operator to force convergence)

where \[ \text{post}_\tau(\varphi) : \exists V_0 \cdot (\varphi (V_0) \land \rho_\tau(V_0, V)) \]
Problems

1. May not converge in finite time
   **Example:**
   
   ```plaintext
   integer i where i = 0
   while true do i := i + 1
   
   \[ F_0 : i = 0 \]
   \[ F_1 : i = 0 \lor i = 1 \]
   \[ F_2 : i = 0 \lor i = 1 \lor i = 2 \]
   
   We never reach: \( i \geq 0 \)
   
   2. May not be able to detect convergence
   
   \[ F_{n+1} \rightarrow F_n \]
**Common Solution**

**Abstract Interpretation** [Cousot&Cousot,77]: perform the symbolic simulation in an abstract domain:

<table>
<thead>
<tr>
<th>Reference</th>
<th>Shape</th>
<th>Invariants (over reals)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karr ’76</td>
<td></td>
<td>(a_1x_1 + \cdots + a_nx_n = b) (linear equalities)</td>
</tr>
<tr>
<td>Müller-Olm, Seidl, ’04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gulwani, Necula ’03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cousot, Halbwachs ’79</td>
<td></td>
<td>(a_1x_1 + \cdots + a_nx_n \leq b) (linear inequalities)</td>
</tr>
<tr>
<td>Cousot, Cousot ’76</td>
<td></td>
<td>(\ell \leq x_i \leq u) (intervals)</td>
</tr>
<tr>
<td>Mine ’01</td>
<td></td>
<td>(x_i - x_j \leq b) (octagons)</td>
</tr>
<tr>
<td>Clarisó, Cortadella ’04</td>
<td></td>
<td>(\sum a_ix_i \leq \ell, a_i \in {-1, 0, 1}) (octahedra)</td>
</tr>
<tr>
<td>Sankaranarayananan, Sipma, Manna ’04</td>
<td></td>
<td>(a_1x_1 + \cdots + a_nx_n \leq b) (octagons)</td>
</tr>
</tbody>
</table>

\(_a_i \) fixed
Example: Forward Propagation

integer $i, j$ where $i = 2 \land j = 0$

$l_0 : \text{ while true do}$

\[
\begin{align*}
  i &:= i + 4 \\
\text{or} \\
(i, j) &:= (i + 2, j + 1)
\end{align*}
\]

**Domain:** Linear Inequalities over Reals
Step 1: Iteration

\[\eta_0 : (j = 0) \land (i = 2)\]
\[\text{post}(\eta_0, \tau_1) : (j = 0) \land (i = 6)\]
\[\text{post}(\eta_0, \tau_2) : (j = 1) \land (i = 4)\]
\[\eta_1 : (0 \leq j \leq 1) \land (2 \leq i - 2j \leq 6)\]
Step 2: Iteration

\[ \eta_1 : \ (0 \leq j \leq 1) \land (2 \leq i - 2j \leq 6) \]
\[ post(\eta_1, \tau_1) : \ (0 \leq j \leq 1) \land (6 \leq i - 2j \leq 10) \]
\[ post(\eta_1, \tau_2) : \ (1 \leq j \leq 2) \land (2 \leq i - 2j \leq 6) \]
\[ \eta_2 : \ (0 \leq j \leq 2) \land (2 \leq i - 2j \leq 10) \]
Step 3: Widening Iteration

\[ \eta_1 : \ (0 \leq j \leq 1) \land (2 \leq i - 2j \leq 6) \]
\[ \eta_2 : \ (0 \leq j \leq 2) \land (2 \leq i - 2j \leq 10) \]
\[ \eta_3 \text{(widening)} : \ (0 \leq j) \land (2 \leq i - 2j) \]
Iteration: Step 4

\[
\begin{align*}
\eta_3 : \quad & (0 \leq j) \land (2 \leq i - 2j) \\
post(\eta_3, \tau_1) : \quad & (0 \leq j) \land (2 \leq i - 2j) \\
post(\eta_3, \tau_2) : \quad & (0 \leq j) \land (2 \leq i - 2j) \\
\eta_4 : \quad & (0 \leq j) \land (2 \leq i - 2j)
\end{align*}
\]

Note: Termination of iteration, \( \eta_4 = \eta_3 \).

The final invariants are \( 0 \leq j \land 2 \leq i - 2j \Rightarrow i \geq 2 \).
Constraint-based Analysis
Constraint-based Analysis: Overview

1. Fix the domain and template of the desired invariant
   Examples:
   - linear invariant over reals
   - polynomial invariant over reals

2. Provide the conditions for the invariant to hold

3. Encode the conditions on the invariant as a system of constraints

4. Solve the constraints

5. Every solution is an invariant of the desired domain and template
Computing Linear Invariants
1. Fix domain and template

- **Domain:** Linear inequalities over reals

- **Template** (target invariant):
  \[ c_1x_1 + c_2x_2 + \ldots + c_nx_n + d \leq 0 \]

where

\{x_1, \ldots, x_n\} are the program variables

and

\{c_1, \ldots, c_n, d\} are unknown coefficients
2. Invariant Conditions

The property

$$\psi : c_1x_1 + c_2x_2 + \ldots + c_nx_n + d \leq 0$$

is an invariant of transition system

$$\Phi : \langle L, \ V : \{x_1, \ldots, x_n\}, \ \Theta, \ \mathcal{T} : \{\tau_1, \ldots, \tau_k\}, \ L_0 \rangle$$

if

$$\Theta \models \psi \quad \text{(initiation)}$$

$$\psi \land \rho_{\tau_1} \models \psi'$$

$$\vdots$$

$$\psi \land \rho_{\tau_k} \models \psi'$$

that is, if

- it is implied by the initial condition, and
- it is preserved by all transitions of the system
**Invariant Conditions: Example**

\[
\text{integer } i, j \text{ where } i = 2 \land j = 0
\]

\[
l_0 : \text{ while true do}
\]

\[
\begin{align*}
  & i := i + 4 \\
  \text{or} \\
  & (i, j) := (i + 2, j + 1)
\end{align*}
\]

Target invariant: \( \psi : c_1i + c_2j + d \leq 0 \)

Conditions:

\[
\begin{align*}
  & i = 2 \land j = 0 \\
  & \psi \\
  & c_1i + c_2j + d \leq 0 \\
  & \rho_{\tau_1, \rho_{\tau_2}} \\
  & i' = i + 4 \land j' = j \\
  & \psi' \\
  & c_1i' + c_2j' + d \leq 0
\end{align*}
\]
Farkas’s Lemma

Let $S$ be a system of linear inequalities over real-valued variables $x_1, \ldots, x_n$,

$$S : \begin{bmatrix}
  a_{11}x_1 + \cdots + a_{1n}x_n + b_1 & \leq 0 \\
  \vdots & \vdots & \vdots \\
  a_{m1}x_1 + \cdots + a_{mn}x_n + b_m & \leq 0
\end{bmatrix}$$

and $\psi$ a linear inequality,

$$\psi : c_1x_1 + \cdots + c_nx_n + d \leq 0$$

If $S$ is satisfiable, $S \models \psi$ iff there exist real multipliers $\lambda_1, \ldots, \lambda_m \geq 0$ such that:

$$c_1 = \sum_{i=1}^{m} \lambda_i a_{i1} \quad \ldots \quad c_n = \sum_{i=1}^{m} \lambda_i a_{in} \quad d \leq \left( \sum_{i=1}^{m} \lambda_i b_i \right)$$
3. Encode the conditions: Initiation

Initiation:

\[ \Theta \models c_1x_1 + \cdots + c_nx_n + d \leq 0 \]

is encoded by

\[
\begin{array}{c|ccccc}
& a_{11}x_1 & + & \cdots & + & a_{1n}x_n & + & b_1 & \leq & 0 \\
\vdots & \vdots & & \ddots & & \vdots & & \vdots & \vdots & \vdots \\
\lambda_m & a_{m1}x_1 & + & \cdots & + & a_{mn}x_n & + & b_m & \leq & 0 \\
\hline
\end{array}
\]

which produces the constraints

\[
S_0 : \exists(\lambda_1 \ldots \lambda_m \geq 0)
\begin{cases}
  c_1 = \sum_{i=1}^{m} \lambda_i a_{i1} & \land \\
  \cdots & \land \\
  c_n = \sum_{i=1}^{m} \lambda_i a_{in} & \land \\
  d \leq \sum_{i=1}^{m} \lambda_i b_i
\end{cases}
\]
**Example: Encoding Initiation**

Target invariant \[ \psi : c_1i + c_2j + d \leq 0 \].

Initial Condition: \[ i = 2 \land j = 0 \models c_1i + c_2j + d \leq 0 \]

\[ \Theta \]

\[ \psi \]

\[ \begin{array}{c|ccc}
\lambda_1 & i & -2 & = 0 \\
\lambda_2 & j & = 0 \\
\end{array} \]

\[ \Theta \]

\[ \exists \lambda_1, \lambda_2 \ [\lambda_1 = c_1 \land \lambda_2 = c_2 \land d \leq -2\lambda_1] \]

No requirement \( \lambda_1, \lambda_2 \geq 0 \)!

The constraint after elimination of the \( \lambda \)'s is

\[ S_0 : 2c_1 + d \leq 0 \]
Example: Encoding Consecution for $\tau_1$

$$c_1i + c_2j + d \leq 0 \quad \land \quad i' = i + 4 \quad \land \quad j' = j \quad \models \quad c_1i' + c_2j' + d \leq 0$$

\[\psi\] \[\rho_{\tau_1}\] \[\psi'\]

\[
\begin{array}{c|c|c}
\mu_1 & c_1i + c_2j + d & \leq 0 \quad \leftarrow \quad \psi \\
\lambda_1 & i - i' & + 4 = 0 \\
\lambda_2 & j - j' & = 0 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c}
\mu_1c_1 + \lambda_1 = 0 & \land & \mu_2c_2 + \lambda_2 = 0 & \land & -\lambda_1 = c_1 & \land & -\lambda_2 = c_2 & \land & d \leq \mu_1d + 4\lambda_1 \\
\end{array}
\]

Constraints:

$$(\exists \mu_1 \geq 0)(\exists \lambda_1, \lambda_2)$$

Eliminating $\mu_1, \lambda_1, \lambda_2$:

$S_1 : (c_1 \leq 0) \lor (c_1 = 0 \land c_2 = 0)$
Example: Combined Constraint

The overall constraint is:

\[
(2c_1 + d \leq 0) \land \\
\bigg[ (c_1 \leq 0) \lor \\
(c_1 = 0 \land c_2 = 0) \bigg] \land \\
\bigg[ (2c_1 + c_2 \leq 0) \lor \\
(c_1 = 0 \land c_2 = 0) \bigg]
\]

which simplifies to

\[
2c_1 + d \leq 0 \land c_1 \leq 0 \land 2c_1 + c_2 \leq 0
\]
4. **Solve the constraints**

**Solve** the constraint systems

\[ S_0 \land S_1 \land \ldots \land S_k \]

for \( \{c_1, \ldots, c_n, d\} \)
**Example: Solving the Constraints**

The basic solutions of

\[ 2c_1 + d \leq 0 \land c_1 \leq 0 \land 2c_1 + c_2 \leq 0 \]

are

<table>
<thead>
<tr>
<th>( c_1 )</th>
<th>( c_2 )</th>
<th>( d )</th>
<th>( c_1 i + c_2 j + d \leq 0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>(-1 \leq 0)</td>
</tr>
<tr>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>(-j \leq 0)</td>
</tr>
<tr>
<td>−1</td>
<td>2</td>
<td>2</td>
<td>(-i + 2j + 2 \leq 0)</td>
</tr>
</tbody>
</table>

which corresponds to the inductive invariants

\[ j \geq 0 \quad \text{and} \quad i - 2j \geq 2 \quad \Rightarrow \quad i \geq 2 \]
5. Solutions

For all solutions of \( \{c_1, \ldots, c_n, d\} \),

\[
c_1x_1 + \ldots + c_nx_n + d \leq 0
\]

is an invariant.

- **Good news:**
  The method is **complete** for linear systems (over reals).
  The solutions of \( \{c_1, \ldots, c_n\} \) represent **all** inductive invariants that are linear inequalities of the given template.
Summary

1. Fix a target invariant with unknown coefficients,
   \[ c_1 i + c_2 j + d \leq 0 \]

2. Encode the invariant conditions (initiation and consecution for each transition)

3. Compute constraints on the unknown coefficients,
   \[ 2c_1 + d \leq 0 \land c_1 \leq 0 \land 2c_1 + c_2 \leq 0 \]

4. Solve these constraints
   \[ \langle c_1, c_2, d \rangle = \langle 0, -1, 0 \rangle \quad \langle c_1, c_2, d \rangle = \langle -1, 2, 2 \rangle \]

5. Generate the invariants
   \[ \langle 0, -1, 0 \rangle \leftrightarrow 0i - 1j + 0 \leq 0 \]
   \[ \langle -1, 2, 2 \rangle \leftrightarrow -1i + 2j + 2 \leq 0 \]

Invariants: \[ \boxed{j \geq 0} \text{ and } \boxed{i - 2j \geq 2} \Rightarrow \boxed{i \geq 2} \]
Pros and Cons

Advantages:
- No widening necessary
- All inductive invariants are generated (or obtained as consequences)
- System structure can be exploited to obtain linear constraints: Petri nets
- Properties other than invariants

Disadvantages:
- The constraint systems $S_1, \ldots, S_k$ are nonlinear and may be hard to solve. Tool: QEPCAD [Hong 93] (Cylindrical Algebraic Decomposition)
- But: $S_1, \ldots, S_k$ are parametric linear ($cx$ okay, but not $x^2$)
  More efficient solution methods:
  - factorization, polynomial root finding
  Tool: REDLOG [Weispfenning 92; Dolzmann, Sturm 97]
Computing Linear Ranking Functions
1. Fix domain and template

- **Domain:** Linear ranking functions over reals

- **Template:**

\[ c_1 x_1 + c_2 x_2 + \ldots + c_n x_n + d \]

where

\( \{x_1, \ldots, x_n\} \) are the program variables

and

\( \{c_1, \ldots, c_n, d\} \) are unknown coefficients
2. Property Conditions

The function
\[ \delta : c_1 x_1 + c_2 x_2 + \ldots + c_n x_n + d \]
is a ranking function of a loop
\[ \Phi : \langle L, V : \{x_1, \ldots, x_n\}, \Theta, T : \{\tau_1, \ldots, \tau_k\}, L_0 \rangle \]
if
\[ \rho_{\tau_1} \models \delta \geq 0 \]
\[ \vdots \]
\[ \rho_{\tau_k} \models \delta \geq 0 \]
under the boundedness condition
\[ \rho_{\tau_1} \models \delta - \delta' \geq \epsilon \]
\[ \vdots \]
\[ \rho_{\tau_k} \models \delta - \delta' \geq \epsilon \]
under the ranking condition
for some \( \epsilon > 0 \); that is, if
- it is bounded from below, and
- it is decreased by each transition.
3. **Encode the conditions**

\[ \delta : c_1 x_1 + \cdots + c_n x_n + d \]

Use Farkas’s Lemma:

- **Bounded:**
  \[ B_i : \rho_{\tau_i} \models \delta \geq 0 \]

- **Ranking**
  \[ R_i : \rho_{\tau_i} \models \delta - \delta' \geq \epsilon \]

for some \( \epsilon > 0 \)
4. Solve the constraints

Solve the constraint systems

\[ B_1 \land \ldots \land B_k \land R_1 \land \ldots \land R_k \]

for \( \{c_1, \ldots, c_n, d\} \)
5. Solutions

The function
\[ c_1 x_1 + c_2 x_2 + \ldots + c_n x_n + d \]
is a ranking function for all solutions of \( \{c_1, \ldots, c_n, d\} \).

- **Good news:**
  The method is complete for linear systems over reals.
  The solutions represent all linear ranking functions of the given (uninitialized) loop.

- **Good news:**
  Constraints are all linear: can be solved efficiently

- **Bad news:**
  Most ranking functions require supporting invariants to prove boundedness (combine!)

\[ I \land \rho_{\tau_i} \models \delta \geq 0 \]
Computing Nonlinear Invariants
**GCD-LCM Example**

\[ \text{GCD}(x_1, x_2) - \text{Greatest Common Divisor of } x_1 \text{ and } x_2, \]
\[ \text{GCD}(14, 21) = 7, \quad \text{GCD}(13, 21) = 1 \]

\[ \text{LCM}(x_1, x_2) - \text{Least Common Multiple of } x_1 \text{ and } x_2, \]
\[ \text{LCM}(14, 21) = 42, \quad \text{LCM}(13, 21) = 273 \]

\[ \text{GCD}(x_1, x_2) \cdot \text{LCM}(x_1, x_2) = x_1 \cdot x_2 \]
GCD-LCM Example

integer $x_1, x_2, y_1, y_2, y_3, y_4$ where

( $y_1 = x_1$ \& $y_2 = y_3 = x_2$ \& $y_4 = 0$ )

$\ell_0 : \text{while } (y_1 \neq y_2) \text{ do}$

\[
\begin{align*}
\ell_1 : & \text{ while } (y_1 > y_2) \text{ do} \\
& (y_1, y_4) := (y_1 - y_2, y_4 + y_3) \\
\ell_2 : & \text{ while } (y_2 > y_1) \text{ do} \\
& (y_2, y_3) := (y_2 - y_1, y_3 + y_4)
\end{align*}
\]

\{ $y_1 = \text{GCD}(x_1, x_2)$, $y_3 + y_4 = \text{LCM}(x_1, x_2)$ \}
Template

\[ c_1x_1x_2 + c_2y_1x_2 + c_3y_2x_1 + c_4y_1y_3 + c_5y_2y_4 + c_6 = 0 \]

**Question:** For what values of \( c_1, \ldots, c_6 \) is \( p = 0 \) invariant at \( \ell_0 \)?

**Goal:** Find values of \( c_1, \ldots, c_6 \) such that

- **Initiation:** \( \emptyset \models p = 0 \),

- **Consecution:** \( p = 0 \land \rho_{\tau_1} \models p' = 0 \),
  \( p = 0 \land \rho_{\tau_2} \models p' = 0 \).
The Goal

To find $c_1, \ldots, c_6$ such that

$$p = 0,$$

construct $c_1, \ldots, c_6$ such that

initiation + consecution

are satisfied.

The Problem:
How do we encode

$$p_1 = 0 \land p_2 = 0 \land \cdots \land p_m = 0 \models p = 0 \quad \cdots (A)$$

where $p_i, p$ are polynomials?

We shall use Gröbner bases.
Linear Algebra

Linear Equalities (over reals):

\[ e : 2x + 3y + \frac{4}{5}z - 4 \]

**Problem:** When does

\[ (e_1 = 0 \land e_2 = 0 \land \cdots \land e_m = 0) \models e = 0? \quad \cdots (A) \]

\[ \text{Linear Equalities} \quad \text{Linear Equality} \]

**Answer:** \[ e = \lambda_1 e_1 + \lambda_2 e_2 + \cdots + \lambda_m e_m \quad \cdots (B) \]

for some \( \lambda_1, \ldots, \lambda_m \), real multipliers.

\( (A) \Leftrightarrow (B) \)

**Algorithm:** *Gaussian Elimination*
Linear Programming

Linear inequalities (over reals):

\[ e : 2x + 3y + \frac{4}{5}z - 4 \]

**Problem:** When does

\[
\begin{aligned}
(e_1 \leq 0 \land e_2 \leq 0 \land \cdots \land e_m \leq 0) \quad \Rightarrow \quad e \leq 0 \quad ? \quad \cdots (A)
\end{aligned}
\]

**Answer** (Farkas’s Lemma): \( e = \lambda_1 e_1 + \lambda_2 e_2 + \cdots + \lambda_m e_m \quad \cdots (B) \)

for some \( \lambda_1, \ldots, \lambda_m \geq 0 \), real multipliers.

\( (A) \Leftrightarrow (B) \)

**Algorithm:** *Cylindrical Algebraic Decomposition*
Polynomials (over reals):

\[ 3x^3y^3 + \frac{7}{9}x^2y^2z^2 + \frac{3}{4}y^2 \]

**Problem:** When does

\[ p_1 = 0 \land p_2 = 0 \land \cdots \land p_m = 0 \models p = 0 \quad ? \quad \cdots (A) \]

\[ \text{polynomial equalities} \quad \text{polynomial equality} \]

**Answer:** \( p = g_1p_1 + g_2p_2 + \cdots + g_mp_m \quad \cdots (B) \)

for some \( g_1, \ldots, g_m \), arbitrary polynomial multipliers.

\( (B) \implies (A) \)

**Algorithm:** *Gröbner bases and normal form reduction.*
Ideals

Ideal: The ideal generated by

\[ P = \{ p_1, \ldots, p_m \} \]

is the set of all polynomials of the form

\[ \text{Ideal}(P) = \{ g_1 p_1 + \ldots + g_m p_m \mid g_1, \ldots, g_m \text{ polynomials} \} \]

Example: Let \( P = \{ x^2 - y, y - z, x + z \} \).

\[ \text{Ideal}(P) = \left\{ \begin{array}{c}
g_1(x^2 - y) + g_2(y - z) + g_3(x + z) \\
g_1, g_2, g_3 \text{ are polynomials over } x, y, z
\end{array} \right\} \]

\[ -zx - z = \underbrace{1}_{g_1} \cdot (x^2 - y) + \underbrace{1}_{g_2} \cdot (y - z) + \underbrace{-x}_{g_3} (x + z) \]

Therefore, \(-zx - z \in \text{Ideal}(P)\).
**Ideal Membership**

**Goal:** Given polynomials \( P = \{p_1, \ldots, p_m\} \) and \( p \),

Decide

\[
(p_1 = 0 \land p_2 = 0 \land \cdots \land p_m = 0) \implies (p = 0) \quad \cdots \quad (A)
\]

**Solution:** Test if

\[
p \in \text{Ideal}(\{p_1, p_2, \ldots, p_m\}) \quad \cdots \quad (B)
\]

i.e, \( p = g_1p_1 + \cdots + g_mp_m \) for some polynomials \( g_1, \ldots, g_m \).

- We cannot efficiently test \((A)\).
- We know from Algebraic Geometry that \((B) \implies (A)\).

**Question:** How do we test \((B)\) efficiently?
Testing Ideal Membership

Given any set of polynomials \( P = \{p_1, \ldots, p_m\} \) and \( p \),

**How do we test if \( p \in \text{IDEAL}(P) \)?**

1. Compute Gröbner basis \( G \) of \( P \) (independent of \( p \)),
i.e, set of polynomials \( G = \{p'_1, \ldots, p'_k\} \), such that
   - \( \text{IDEAL}(G) = \text{IDEAL}(P) \),
   - \( G \)-rules \( \xrightarrow{G} \) are confluent and terminating.

Use Büchberger’s Algorithm + Refinements.

2. Apply the \( G \)-rules to \( p \). It leads to
   unique normal form \( \text{NF}_G(p) \), \( p \xrightarrow{G} \cdots \xrightarrow{G} \text{NF}_G(p) \).

**Theorem:** \( p \in \text{IDEAL}(P) \) \( \cdots \) (B)

iff

\( \text{NF}_G(p) = 0 \). \( \cdots \) (C)
Testing Ideal Membership: Example

Let \( P = \{p_1 : x^2 - y, \ p_2 : y + z, \ p_3 : x - z\} \).
Can we find out if
\[
x^2 - z \in \text{IDEAL}(P)
\]
using \( \frac{p}{P} \)? No!

Gröbner basis of \( P \) is
\[
G = \{z^2 - z, y - z, x + z\}
\]
Can we find out using \( \frac{G}{G} \)? Yes!

Any sequence of \( G \)-reductions \( \frac{G}{G} \) from
\[
p : x^2 - z
\]
has normal form 0. Therefore
\[
x^2 - z \in \text{IDEAL}(P)
\]
Template Constraints

Let $P = \{x^2 - y, \ y + z, \ x - z\}$.

Problem: For what values of $c_1, c_2, \ldots, c_5$ does

$$p_1 = 0 \land p_2 = 0 \land p_3 = 0 \models c_1 x^2 + c_2 y^2 + c_3 z^2 + c_4 z + c_5 = 0 ?$$

Solution:

1. Compute the Gröbner basis of $P$,

$$G = \{z^2 - z, \ y - z, \ x + z\}$$

2. Compute normal form of $p$ using $G$-rules,

$$\text{NF}_G(p) = (c_1 + c_2 + c_3 + c_4)z + c_5$$

3. Set every coefficient to be zero,

$$(c_1 + c_2 + c_3 + c_4 = 0) \land (c_5 = 0)$$
Template Constraints (Cont)

Note: For solutions to \( c_1, \ldots, c_5 \) that satisfy

\[
(c_1 + c_2 + c_3 + c_4 = 0) \land (c_5 = 0)
\]

it follows that \( \text{NF}_G(p) = 0 \)

therefore, \( c_1 x^2 + c_2 y^2 + c_3 z^2 + c_4 z + c_5 \in \text{Ideal}(P) \) \( \cdots (B) \)

therefore, \( p_1 = 0 \land p_2 = 0 \land p_3 = 0 \models p = 0 \) \( \cdots (A) \)

Example: Consider a solution

\[
\langle c_1, \ldots, c_5 \rangle = \langle 1, -1, 0, 0, 0 \rangle
\]

Then,

\[
p_1 = 0 \land p_2 = 0 \land p_3 = 0 \models 1 x^2 + -1 y^2 + 0 z^2 + \cdots + 0 = 0
\]

\[
x^2 - y^2 = 0
\]
The condition
\[ \Theta \models p = 0 \]
is encoded by reducing \( p \) wrt to the Gröbner basis \( G \) of \( \{\Theta\} \):
\[ p \xrightarrow{G} \cdots \xrightarrow{G} NF(p) \]
and setting
\[ NF(p) \equiv 0 \]
which produces a set \( S_0 \) of linear constraints on \( \{c_1, \ldots c_{10}\} \).
3. Encode the conditions: Consecution

The condition

\[ p = 0 \land p_{\tau_i} \models p' = 0 \]

is not practical to encode. Instead we encode one of

\[ p_{\tau_i} \models p' = 0 \]
\[ p_{\tau_i} \models p' - p = 0 \]

which result in a set \( S_i \) of linear constraints, or more general

\[ \exists \text{ real } \lambda. \quad p_{\tau_i} \models p' - \lambda p = 0 \]
\[ \exists \text{ polynomial } q. \quad p_{\tau_i} \models p' - qp = 0 \]

which result in a set of nonlinear constraints.
4. Solve the constraints

Solve

\[ S_0 \land S_1 \land \ldots \land S_k \]

for \( \{c_1, \ldots, c_{10}\} \)
5. Solutions

For all solutions of \(\{c_1, \ldots, c_{10}\}\),

\[
\begin{align*}
    c_1 x^3 + c_2 x^2 y + c_3 x^2 z + c_4 xy^2 + c_5 xyz + c_6 xz^2 + \\
    c_7 y^3 + c_8 y^2 z + c_9 yz^2 + c_{10} z^3 &= 0
\end{align*}
\]

is an invariant.

- **Good news:**
  Constraints are all linear: can be solved efficiently

- **Bad news:**
  Invariants are missed because of strengthening the conditions
  Trade-off between complexity and generality
**Example: Nonlinear Invariant Generation**

integer $i, j, k, s$ where $(s = 0 \land j = k \land j \geq 0)$

$l_0: \textbf{while } (k \geq 0) \textbf{ do}$

\[ l_1: (s, k) := (s + i, k - 1) \]

\[ l_2: \]

Target Invariant: \[ p = c_1 s + c_2 ik + c_3 ij + c_4 jk + c_5 \]

**Question:** For what values of $c_1, \ldots, c_5$, is $p = 0$ inductive at $l_0$?
Example: Nonlinear Invariant Generation

1. Fix a template (usually a “generic polynomial” of degree $m$),

\[ c_1s + c_2ik + c_3ij + c_4jk + c_5 \]

2. Generate constraints by encoding initiation and consecution,

\[ c_2 + c_3 = 0 \quad \land \quad c_4 = c_5 = 0 \quad \land \quad c_1 - c_2 = 0 \]

3. Solve the constraints,

\[ c_1 = c_2 = -1, \quad c_3 = 1, \quad c_4 = c_5 = 0 \]

4. Generate the invariant

\[ -s - ik + ij = 0 \]

Invariant: $s = i(j - k)$ at $l_0$. 

Example: Back to GCD-LCM

integer $x_1, x_2, y_1, y_2, y_3, y_4$ where

$\quad (\ y_1 = x_1 \land y_2 = y_3 = x_2 \land y_4 = 0 \ )$

$l_0 : \textbf{while } (y_1 \neq y_2) \textbf{ do}$

\[
\begin{array}{c}
\quad l_1 : \textbf{while } (y_1 > y_2) \textbf{ do} \\
\quad \quad (y_1, y_4) := (y_1 - y_2, y_4 + y_3) \\
\quad l_2 : \textbf{while } (y_2 > y_1) \textbf{ do} \\
\quad \quad (y_2, y_3) := (y_2 - y_1, y_3 + y_4)
\end{array}
\]

$\{ y_1 = \text{GCD}(x_1, x_2), \ y_3 + y_4 = \text{LCM}(x_1, x_2) \}$
Example

1. Fix a template (usually a “generic polynomial” of degree \( m \)),

\[
c_1 x_1 x_2 + c_2 y_1 x_2 + c_3 y_2 x_1 + c_4 y_1 y_3 + c_5 y_2 y_4 + c_6 = 0
\]

2. Generate constraints by encoding initiation and consecution,

\[
\begin{align*}
&c_1 + c_2 + c_3 + c_4 = 0 \land c_6 = 0 \quad \cdots \text{Initiation} \\
&c_4 = c_5 \land c_2 = 0 \quad \cdots \text{Consecution } \tau_1 \\
&c_4 = c_5 \land c_3 = 0 \quad \cdots \text{Consecution } \tau_2
\end{align*}
\]

3. Solve the constraints,

\[
\langle c_1, c_2, c_3, c_4, c_5, c_6 \rangle = \langle -1, 0, 0, 1, 1, 0 \rangle
\]

4. Generate the invariant at \( \ell_0 : -x_1 x_2 + y_1 y_3 + y_2 y_4 = 0 \),

\[
y_1 y_3 + y_2 y_4 = x_1 x_2
\]
Example

integer $x_1, x_2, y_1, y_2, y_3, y_4$ where  

\[ \ell_0 : \text{while } (y_1 \neq y_2) \text{ do} \]

\[ \ell_1 : \text{while } (y_1 > y_2) \text{ do} \]
\[ (y_1, y_4) := (y_1 - y_2, y_4 + y_3) \]
\[ \ell_2 : \text{while } (y_2 > y_1) \text{ do} \]
\[ (y_2, y_3) := (y_2 - y_1, y_3 + y_4) \]

\{ $y_1 = \text{GCD}(x_1, x_2), \ y_3 + y_4 = \text{LCM}(x_1, x_2)$\}

Proving Partial Correctness:
Discovered Assertion + “GCD facts” suffice!
Advantages of Constraint-based Approach

- Controlling the complexity of the constraints
  - Strengthen the conditions on the property
    \[
    \Theta \models \psi \\
    \psi \land \rho \tau \models \psi' \quad \Rightarrow \quad \Theta \models \psi \\
    \rho \tau \models \psi'
    \]
  - Constrain the property
    \[
    c_1 x^3 + c_2 x^2 y + c_3 x^2 z + c_4 xy^2 + c_5 xyz + c_6 xz^2 + c_7 y^3 + c_8 y^2 z + c_9 yz^2 + c_{10} z^3
    \]
    \[
    \downarrow
    \]
    \[
    c_1 x^3 + c_2 xy^2 + c_3 xz^2 + c_4 y^2 z
    \]
Constraint-based Approach (Cont)

Advantages:
- Not limited to invariants
  - termination
  - temporal properties (LTL safety)
- Can exploit system structure to simplify the constraint system
  - Petri nets
- Can take advantage of results in constraint solving community:
  - Sophisticated techniques from linear algebra and algebraic geometry
  - Exploits recent advances

Disadvantages:
- Hard to solve constraints exactly.
- Domain is fixed (e.g., fixed degree bound).
Papers

- Termination analysis (TACAS’01, CAV’02, CAV’05)
- Linear invariant generation (CAV’03, SAS’04, VMCAI’05, VMCAI’06)
- Nonlinear invariant generation (POPL’04)
- Nonlinear invariant generation for hybrid systems (HSCC’04)
- Differential equations (HSCC’06)

Related Work

- Abstract interpretation
  - [Cousot, Cousot’77]
  - [Cousot, Halbwachs’79]

- Set-constraint based analysis
  - [Heintze’93]
  - [Aiken’99]

- Termination analysis
  - [Podelski, Rybalchenko, VMCAI’04, LICS’04]
  - [Cousot, VMCAI’05]

- Nonlinear invariants
  - [Bensalem et al, SAS’00]
  - [Müller-Olm, Seidl, SAS’02, POPL’04]
  - [Tiwari et al, TACAS’01, HSCC’03]
  - [Rodriguez-Carbonell, Kapur, ISSAC’04]
  - [Cousot, VMCAI’05]
Current Topics of Investigation

- Classification of systems with simpler constraint systems
- Extension to game properties (ATL*)
- Extension to other domains, in particular nonlinear inequalities
- More efficient constraint solving strategies