# Finding Polynomial and Array Invariants using Dynamic Analysis

ThanhVu (Vu) Nguyen*
Deepak Kapur*, Westley Weimer†, Stephanie Forrest*

*University of New Mexico, †University of Virginia

June 8, 2012



UNM

# Introduction

Ensuring that a program behaves correctly is critical

# Introduction

Ensuring that a program behaves correctly is critical

Dynamic analysis discovers invariants from traces

- Input: traces (observed values of variables)
- Output: relations among variables

# Introduction

Ensuring that a program behaves correctly is critical

Dynamic analysis discovers invariants from traces

- Input: traces (observed values of variables)
- Output: relations among variables

Our dynamic system discovers two forms of invariants

# Introduction

Ensuring that a program behaves correctly is critical

Dynamic analysis discovers invariants from traces

- Input: traces (observed values of variables)
- Output: relations among variables

Our dynamic system discovers two forms of invariants

- Polynomials
  - Equalities: $x - yq = r, x^{10} = -10.23478$
  - Inequalities: $x^2 - \varepsilon \leq y \leq x^2 + \varepsilon$
- Arrays
  - Simple relations: $A[i][j] = -3B[2i] + C[i][j] + D[8]$
  - Nested relations: $A[i][j] = B[C[i + j]][C[3j]]$

# Daikon

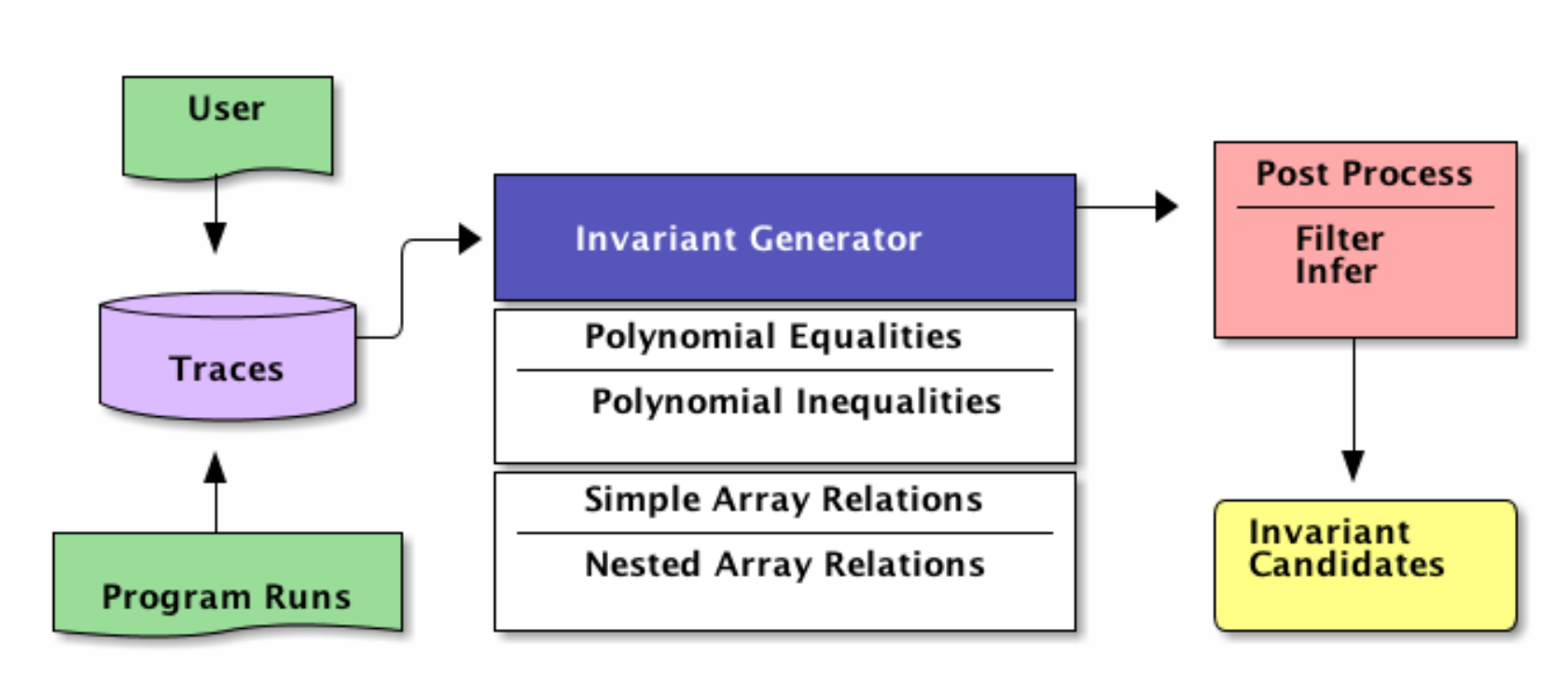

- Comes with a large set of pre-defined templates

$$\text{Polynomials} : x + 2y - 3z + 4 = 0, x = y^2$$

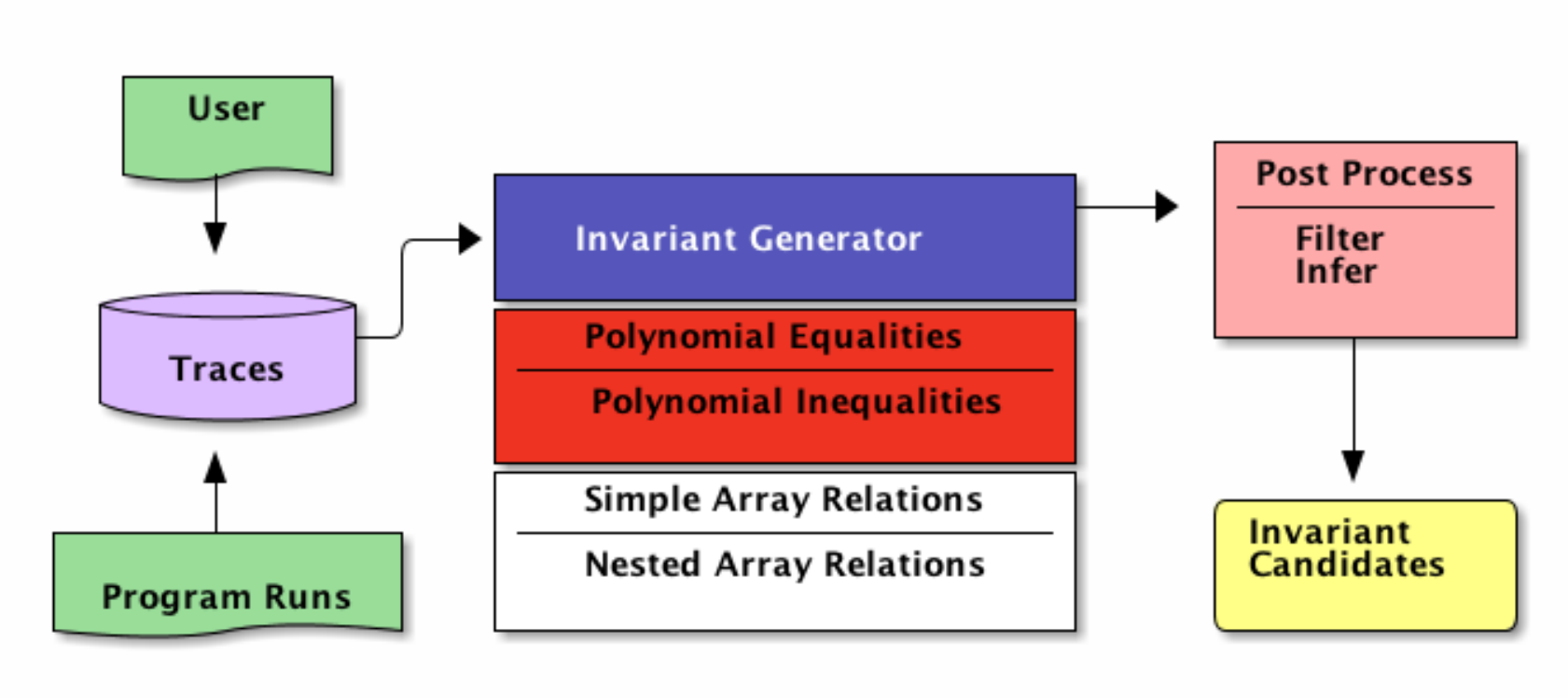
$$\text{Arrays} : \text{sorted}(A), \text{member}(a, A), \text{reverse}(A, B), A = B$$

- User-defined: $x = y^2 + 10, x = y^3$
- Filters out templates from traces
- Cannot find *general* linear or nonlinear relations
- Has limited support for relations among arrays

# Overview of the System



# Polynomial Invariants






UNM

# Example: Cohen Integer Division

```
1  def intdiv(x, y):
2      q = 0
3      r = x
4      while r ≥ y:
5          a = 1
6          b = y
7          while r ≥ 2b:
8              [L]
9              a = 2a
10             b = 2b
11         r = r - b
12         q = q + a
13     return q
```

# Example: Cohen Integer Division

```
1  def intdiv(x, y):
2      q = 0
3      r = x
4      while r ≥ y:
5          a = 1
6          b = y
7          while r ≥ 2b:
8              [L]
9              a = 2a
10             b = 2b
11         r = r - b
12         q = q + a
13     return q
```

| $x$ | $y$ | $a$ | $b$ | $q$ | $r$ |
|---|---|---|---|---|---|
| 15 | 2 | 1 | 2 | 0 | 15 |
| 15 | 2 | 2 | 4 | 0 | 15 |
| 15 | 2 | 1 | 2 | 4 | 7 |

# Example: Cohen Integer Division

```
1  def intdiv(x, y):
2      q = 0
3      r = x
4      while r ≥ y:
5          a = 1
6          b = y
7          while r ≥ 2b:
8              [L]
9              a = 2a
10             b = 2b
11         r = r - b
12         q = q + a
13     return q
```

| $x$ | $y$ | $a$ | $b$ | $q$ | $r$ |
|---|---|---|---|---|---|
| 15 | 2 | 1 | 2 | 0 | 15 |
| 15 | 2 | 2 | 4 | 0 | 15 |
| 15 | 2 | 1 | 2 | 4 | 7 |
| 4 | 1 | 1 | 1 | 0 | 4 |
| 4 | 1 | 2 | 2 | 0 | 4 |

# Example: Cohen Integer Division

```
1  def intdiv(x, y):
2      q = 0
3      r = x
4      while r ≥ y:
5          a = 1
6          b = y
7          while r ≥ 2b:
8              [L]
9              a = 2a
10             b = 2b
11         r = r - b
12         q = q + a
13     return q
```

| $x$ | $y$ | $a$ | $b$ | $q$ | $r$ |
|---|---|---|---|---|---|
| 15 | 2 | 1 | 2 | 0 | 15 |
| 15 | 2 | 2 | 4 | 0 | 15 |
| 15 | 2 | 1 | 2 | 4 | 7 |
| 4 | 1 | 1 | 1 | 0 | 4 |
| 4 | 1 | 2 | 2 | 0 | 4 |

Invariants at L: $\{b = ya, x = qy + r, r \geq 2ya\}$

# Nonlinear Equations

Examples

$$\text{Integer division} : x = qy + r$$
$$\text{Extended gcd} : \gcd_{A,B} = iA + jB$$

**Find equations of the form**

$$c_0 + c_1 x + c_2 y + c_3 xy + \cdots + c_n x^d y^d = 0, \quad c_i \in \mathbb{R}$$

Method

- Generates equations from program traces
- Solves equations using a standard equation solver

# Finding Nonlinear Equations using Equation Solver

- Terms and degrees

$$V = \{r, y, a\}; \quad \deg_{\max} = 2 \quad \Rightarrow \quad T = \{1, r, y, a, ry, ra, ya, r^2, y^2, a^2\}$$

# Finding Nonlinear Equations using Equation Solver

- Terms and degrees

$$V = \{r, y, a\}; \quad \deg_{\max} = 2 \quad \Rightarrow \quad T = \{1, r, y, a, ry, ra, ya, r^2, y^2, a^2\}$$
$$T = \{\ldots, \log(r), a^y, \sin(y), \ldots\}$$

# Finding Nonlinear Equations using Equation Solver

- Terms and degrees

$$V = \{r, y, a\}; \quad \deg_{\max} = 2 \quad \Rightarrow \quad T = \{1, r, y, a, ry, ra, ya, r^2, y^2, a^2\}$$

- Equation template

$$c_1 + c_2 r + c_3 y + c_4 a + c_5 ry + c_6 ra + c_7 ya + c_8 r^2 + c_9 y^2 + c_{10} a^2 = 0$$

# Finding Nonlinear Equations using Equation Solver

- Terms and degrees

$$V = \{r, y, a\}; \;\; \deg_{\max} = 2 \quad \Rightarrow \quad T = \{1, r, y, a, ry, ra, ya, r^2, y^2, a^2\}$$

- Equation template

$$c_1 + c_2 r + c_3 y + c_4 a + c_5 ry + c_6 ra + c_7 ya + c_8 r^2 + c_9 y^2 + c_{10} a^2 = 0$$

- System of linear equations

$$\begin{aligned} \text{trace 1} \quad &: \quad \{r = 15, y = 2, a = 1\} \\ \text{eq 1} \quad &: \quad c_1 + 15c_2 + 2c_3 + c_4 + 30c_5 + 15c_6 + 2c_7 + 225c_8 + 4c_9 + c_{10} = 0 \\ &\vdots \end{aligned}$$

# Finding Nonlinear Equations using Equation Solver

- Terms and degrees

$$V = \{r, y, a\};\ \ \deg_{\max} = 2 \quad \Rightarrow \quad T = \{1, r, y, a, ry, ra, ya, r^2, y^2, a^2\}$$

- Equation template

$$c_1 + c_2 r + c_3 y + c_4 a + c_5 ry + c_6 ra + c_7 ya + c_8 r^2 + c_9 y^2 + c_{10} a^2 = 0$$

- System of linear equations

$$\begin{aligned} \text{trace 1} &: \{r = 15, y = 2, a = 1\} \\ \text{eq 1} &: c_1 + 15c_2 + 2c_3 + c_4 + 30c_5 + 15c_6 + 2c_7 + 225c_8 + 4c_9 + c_{10} = 0 \\ &\vdots \end{aligned}$$

- Solve for coefficients $c_i$

$$V = \{x, y, a, b, q, r\};\ \ \deg_{\max} = 2 \quad \Rightarrow \quad \{b = ya, x = qy + r\}$$

# Nonlinear Inequalities

Example:

Square root : $x + \varepsilon \geq y^2 \geq x - \varepsilon$

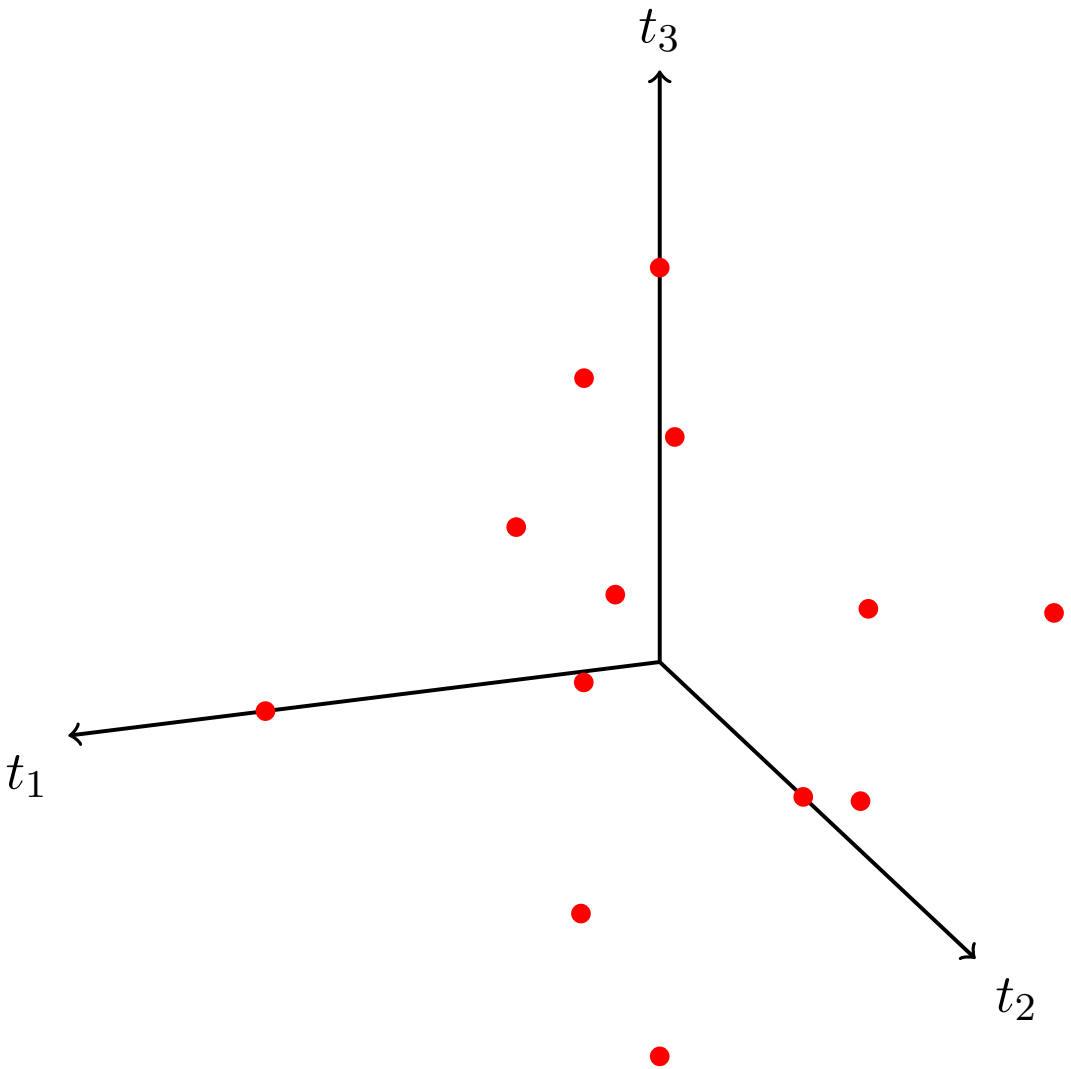
**Find inequalities of the form**

$$c_0 + c_1 x + c_2 y + c_3 xy + \cdots + c_n x^d y^d \geq 0, \quad c_i \in \mathbb{R}$$

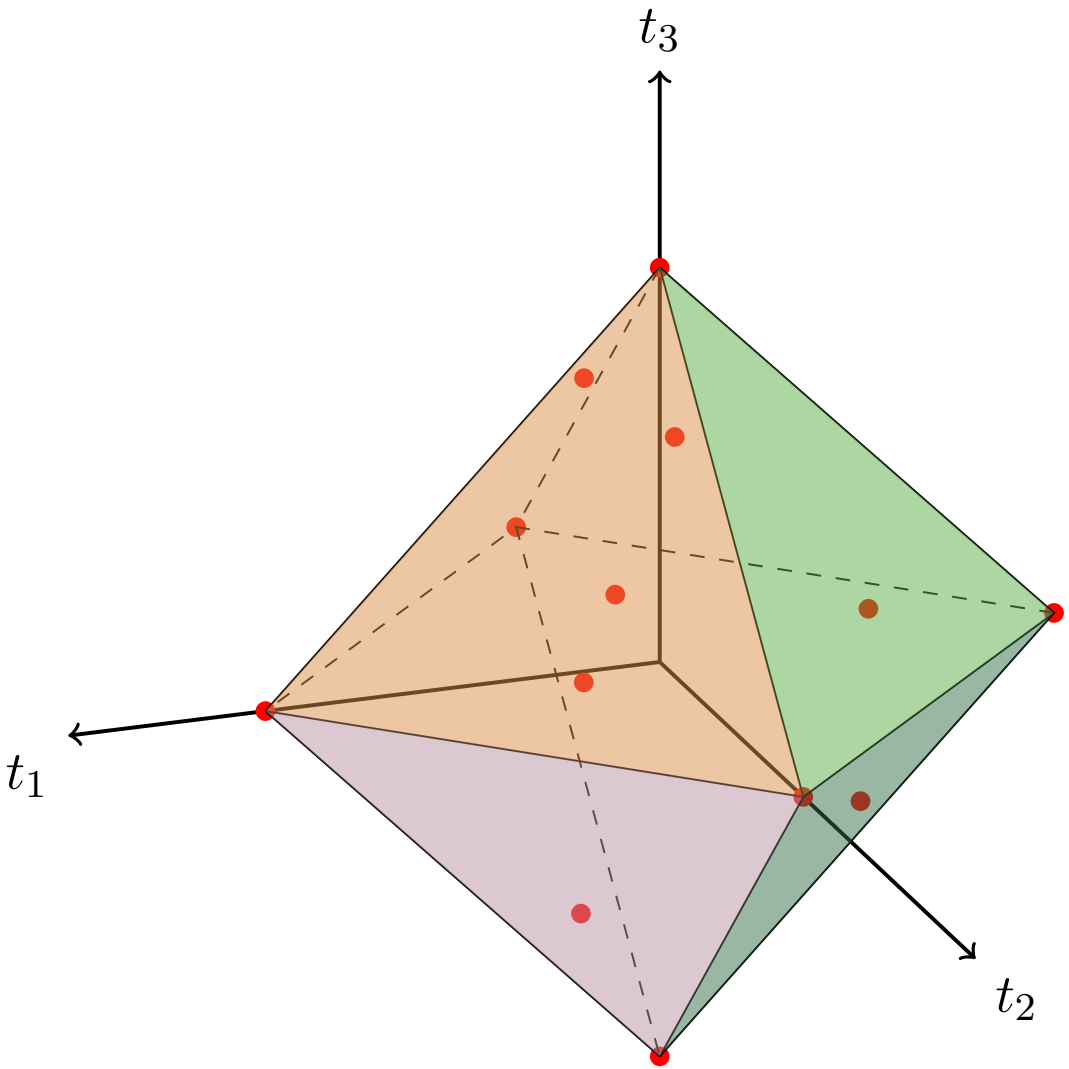
Method

1. Polyhedra
   - Represents trace values as points
   - Builds a bounded convex polyhedron and extracts facets
2. Deduction
   - Deduces invariants when additional information is given

# Finding Nonlinear Inequalities using Polyhedra



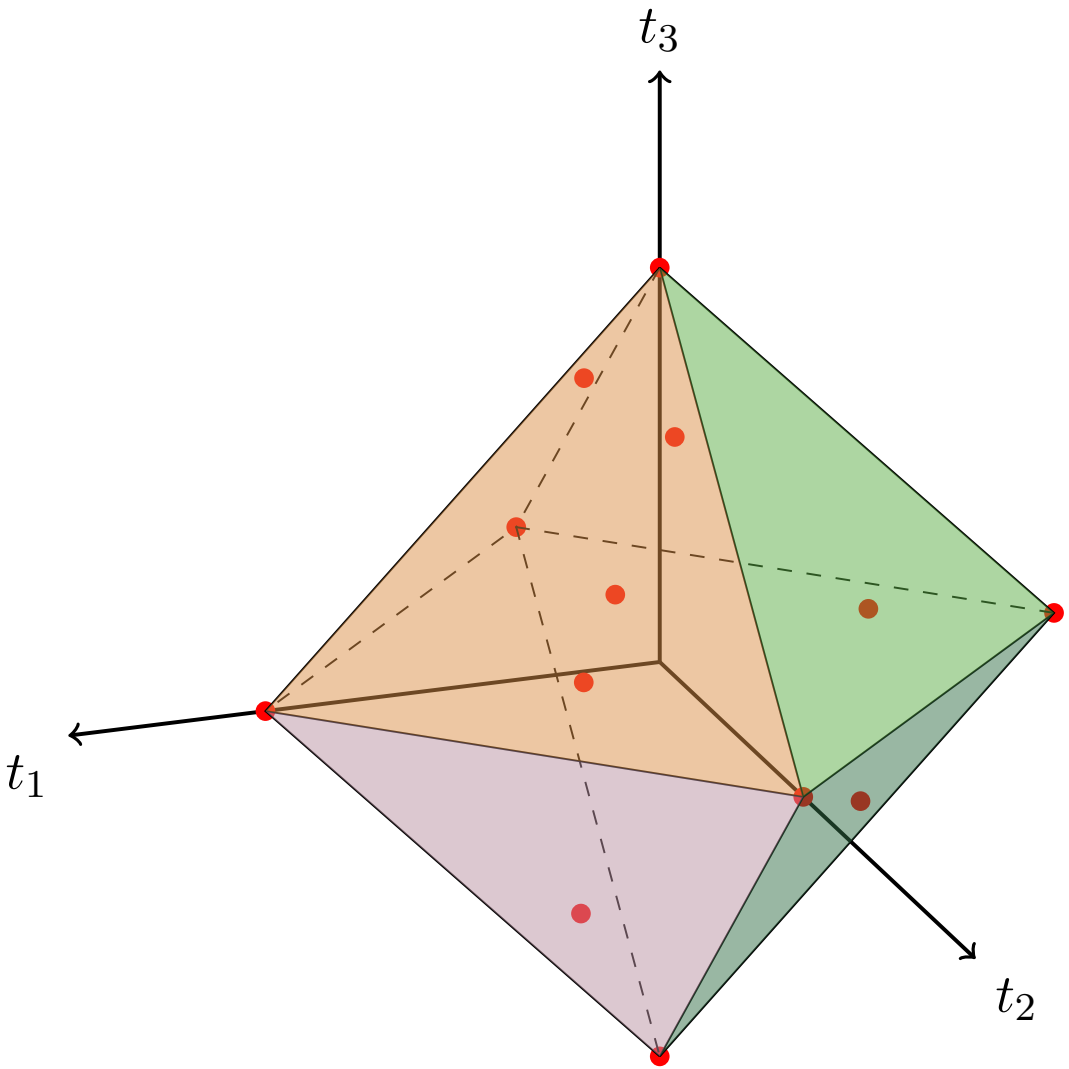
# Finding Nonlinear Inequalities using Polyhedra

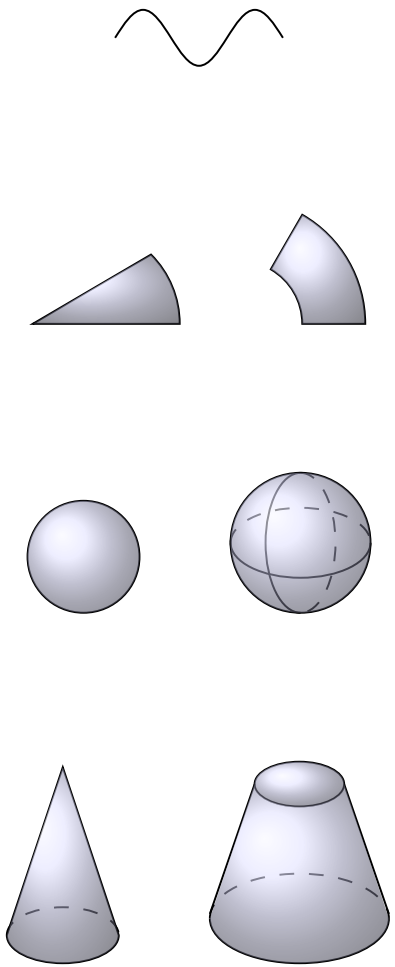





UNM

# Finding Nonlinear Inequalities using Polyhedra



# Finding Nonlinear Inequalities using Deduction

- Use inequality tests $P$ from loop or branch conditions

$$\text{while } (r \geq 2b) \; \{[\mathsf{L}] \ldots\}$$

# Finding Nonlinear Inequalities using Deduction

- Use inequality tests $P$ from loop or branch conditions

$$\text{while } (r \geq 2b)\ \{[\mathsf{L}] \ldots\}$$

- Obtain equality relations $Q$ at L

$$\{b = ay, qy + r = x\}$$

# Finding Nonlinear Inequalities using Deduction

- Use inequality tests $P$ from loop or branch conditions

$$\text{while } (r \geq 2b)\ \{[\mathsf{L}] \ldots\}$$

- Obtain equality relations $Q$ at L

$$\{b = ay, qy + r = x\}$$

- Deduce new, non-trivial inequality relations at L from $P$ and $Q$

$$\begin{aligned} (r \geq 2b \ \wedge\ b = ay) &\quad \Rightarrow \quad r \geq 2ay \\ (r \geq 2b \ \wedge\ qy + r = x) &\quad \Rightarrow \quad x - qy \geq 2b \end{aligned}$$

# Results for Polynomial Invariants

| Program | Desc | Inv Type |
|---|---|---|
| divbin | div | eq |
| cohendiv | div | eq, ieq |
| mannadiv | int div | eq |
| hard | int div | eq |
| sqrt1 | sqr | eq, ieq |
| dijkstra | sqr | eq |
| freire1 | sqr | eq |
| freire2 | cubic root | eq |
| cohencube | cube | eq |
| euclidex1 | gcd | eq |
| euclidex2 | gcd | eq |
| euclidex3 | gcd | eq |
| lcm1 | gcd, lcm | eq |
| lcm2 | gcd, lcm | eq |
| prodbin | product | eq |
| prod4br | product | eq |
| fermat1 | divisor | eq |
| fermat2 | divisor | eq |
| knuth | divisor | eq |
| geo2 | geo series | eq |
| geo3 | geo series | eq |
| ps2 | pow sum | eq |
| ps3 | pow sum | eq |
| ps4 | pow sum | eq |

**24 programs**

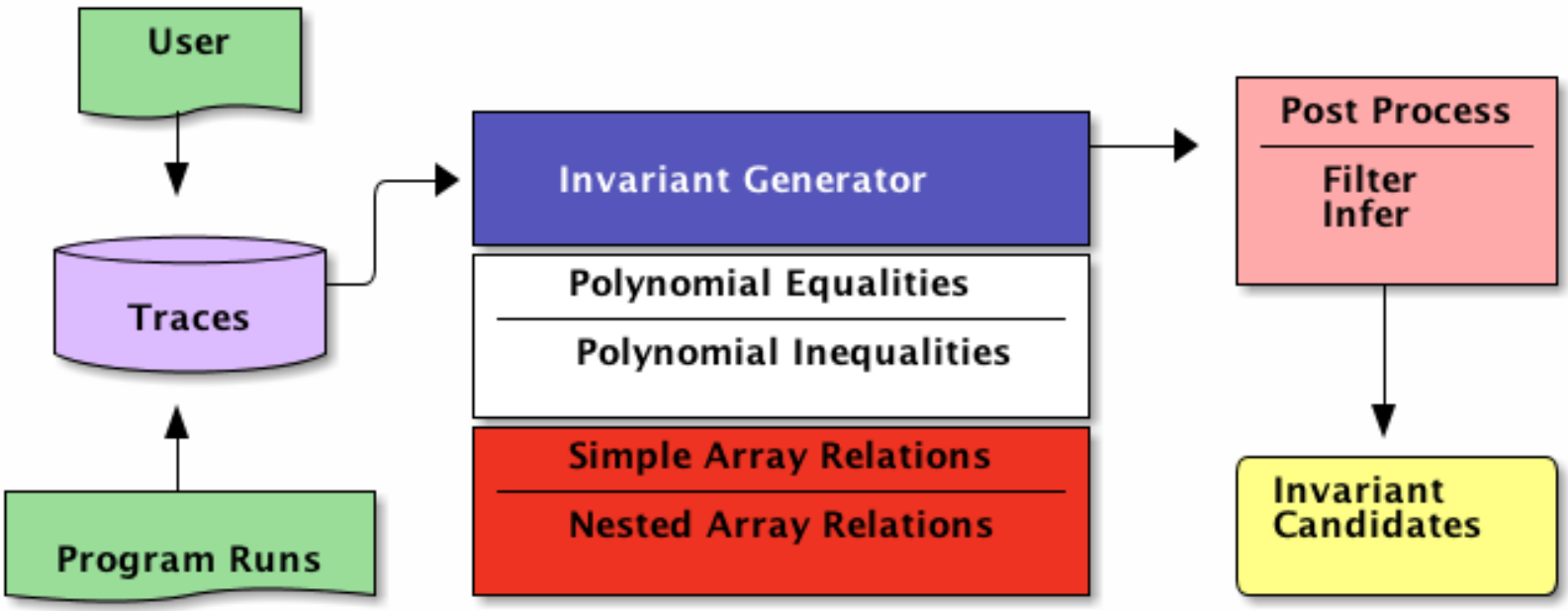
# Results for Polynomial Invariants

| Program | Desc | Inv Type | \|Vars\| | Degree | Annotated Invs |
|---|---|---|---|---|---|
| divbin | div | eq | 5 | 2 | 1 |
| cohendiv | div | eq, ieq | 6 | 2 | 2 |
| mannadiv | int div | eq | 5 | 2 | 1 |
| hard | int div | eq | 6 | 2 | 1 |
| sqrt1 | sqr | eq, ieq | 4 | 2 | 2 |
| dijkstra | sqr | eq | 5 | 2 | 1 |
| freire1 | sqr | eq | 3 | 2 | 1 |
| freire2 | cubic root | eq | 4 | 3 | 2 |
| cohencube | cube | eq | 5 | 3 | 3 |
| euclidex1 | gcd | eq | 10 | 2 | 3 |
| euclidex2 | gcd | eq | 8 | 2 | 2 |
| euclidex3 | gcd | eq | 12 | 2 | 4 |
| lcm1 | gcd, lcm | eq | 6 | 2 | 1 |
| lcm2 | gcd, lcm | eq | 6 | 2 | 1 |
| prodbin | product | eq | 5 | 2 | 1 |
| prod4br | product | eq | 6 | 3 | 1 |
| fermat1 | divisor | eq | 5 | 2 | 1 |
| fermat2 | divisor | eq | 5 | 2 | 1 |
| knuth | divisor | eq | 8 | 3 | 1 |
| geo2 | geo series | eq | 4 | 2 | 1 |
| geo3 | geo series | eq | 5 | 3 | 1 |
| ps2 | pow sum | eq | 3 | 2 | 1 |
| ps3 | pow sum | eq | 3 | 3 | 1 |
| ps4 | pow sum | eq | 3 | 4 | 1 |
| **24 programs** | | | | | **35** |

# Results for Polynomial Invariants

| Program | Desc | Inv Type | \|Vars\| | Degree | Annotated Invs | Discovered Invs | Time (secs) |
|---|---|---|---|---|---|---|---|
| divbin | div | eq | 5 | 2 | 1 | 1 | 0.5 |
| cohendiv | div | eq, ieq | 6 | 2 | 2 | 2 | 1.3 |
| mannadiv | int div | eq | 5 | 2 | 1 | 1 | 0.3 |
| hard | int div | eq | 6 | 2 | 1 | 1 | 0.9 |
| sqrt1 | sqr | eq, ieq | 4 | 2 | 2 | 2 | 0.7 |
| dijkstra | sqr | eq | 5 | 2 | 1 | 1 | 0.5 |
| freire1 | sqr | eq | 3 | 2 | 1 | 1 | 0.2 |
| freire2 | cubic root | eq | 4 | 3 | 2 | 2 | 3.2 |
| cohencube | cube | eq | 5 | 3 | 3 | 3 | 12.6 |
| euclidex1 | gcd | eq | 10 | 2 | 3 | 3 | 6.5 |
| euclidex2 | gcd | eq | 8 | 2 | 2 | 2 | 2.5 |
| euclidex3 | gcd | eq | 12 | 2 | 4 | 4 | 10.1 |
| lcm1 | gcd, lcm | eq | 6 | 2 | 1 | 1 | 0.5 |
| lcm2 | gcd, lcm | eq | 6 | 2 | 1 | 1 | 0.6 |
| prodbin | product | eq | 5 | 2 | 1 | 1 | 0.3 |
| prod4br | product | eq | 6 | 3 | 1 | 1 | 8.1 |
| fermat1 | divisor | eq | 5 | 2 | 1 | 1 | 0.8 |
| fermat2 | divisor | eq | 5 | 2 | 1 | 1 | 0.4 |
| knuth | divisor | eq | 8 | 3 | 1 | 1 | 71.5 |
| geo2 | geo series | eq | 4 | 2 | 1 | 1 | 0.2 |
| geo3 | geo series | eq | 5 | 3 | 1 | 1 | 3.1 |
| ps2 | pow sum | eq | 3 | 2 | 1 | 1 | 0.1 |
| ps3 | pow sum | eq | 3 | 3 | 1 | 1 | 0.3 |
| ps4 | pow sum | eq | 3 | 4 | 1 | 1 | 0.8 |
| **24 programs** | | | | | **35** | **35** | **avg 3.6** |

# Array Invariants



# Simple Array Relations

Examples

$$\text{block2State} : R[i][j] = t[4i + j]$$
$$\text{keySetupEnc8} : R[i][j] = \text{cipherKey}[8i + j]$$

**Find simple array relations of the form**

$$A_1 + c_2 A_2 + \cdots + c_n A_n + c_0 = 0, \quad c_i \in \mathbb{R}$$

Method

- Flattens array elements as new variables
- Infers linear equalities among array elements
- Finds relations among array indices

# Finding Simple Array Relations

- Represent array elements with new variables

trace 1 : $\{A = [-546, -641, 34], B = [-78, 3, -92, -34, 4]\}$

trace 2 : $\{A = [133, -333, -323], B = [19, 96, -48, -80, -47]\}$

trace 3 : ...

$\Downarrow\Downarrow$

| | $A_0$ | $A_1$ | $A_2$ | $B_0$ | $B_1$ | $B_2$ | $B_3$ | $B_4$ |
|---|---|---|---|---|---|---|---|---|
| trace 1 | -546 | -641 | 34 | -78 | 3 | -92 | -34 | 4 |
| trace 2 | -133 | -333 | -323 | -19 | 96 | -48 | -80 | -47 |
| trace 3 | ... | | | | | | | |
| ⋮ | | | | | | | | |

# Finding Simple Array Relations

- Represent array elements with new variables

trace 1 : $\{A = [-546, -641, 34], B = [-78, 3, -92, -34, 4]\}$

trace 2 : $\{A = [133, -333, -323], B = [19, 96, -48, -80, -47]\}$

trace 3 : ...

$\Downarrow$

| | $A_0$ | $A_1$ | $A_2$ | $B_0$ | $B_1$ | $B_2$ | $B_3$ | $B_4$ |
|---|---|---|---|---|---|---|---|---|
| trace 1 | -546 | -641 | 34 | -78 | 3 | -92 | -34 | 4 |
| trace 2 | -133 | -333 | -323 | -19 | 96 | -48 | -80 | -47 |
| trace 3 | ... | | | | | | | |
| ⋮ | | | | | | | | |

- Find linear relations from traces

$$\begin{aligned} A_0 - 7B_0 &= 0 \\ A_1 - 7B_2 &= 3 \\ A_2 - 7B_4 &= 6 \end{aligned}$$

# Finding Simple Array Relations

- Hypothesize

$$A[i] = lB[j] + k, \qquad i \in \{0, 1, 2\}$$

# Finding Simple Array Relations

- Hypothesize

$$A[i] = lB[j] + k, \qquad i \in \{0, 1, 2\}$$

- Find $j$

# Finding Simple Array Relations

- Hypothesize

$$A[i] = lB[j] + k, \qquad i \in \{0, 1, 2\}$$

- Find $j$
  - Express relation between $A[i]$ and $B[j]$ as $j = ip + q$

# Finding Simple Array Relations

- Hypothesize

$$A[i] = lB[j] + k, \qquad i \in \{0, 1, 2\}$$

- Find $j$
  - Express relation between $A[i]$ and $B[j]$ as $j = ip + q$

$$A_0 - 7B_0 = 0 \quad \Rightarrow \quad 0 = 0p + q$$

# Finding Simple Array Relations

- Hypothesize

$$A[i] = lB[j] + k, \qquad i \in \{0, 1, 2\}$$

- Find $j$
  - Express relation between $A[i]$ and $B[j]$ as $j = ip + q$

$$\begin{aligned} A_0 - 7B_0 = 0 \quad &\Rightarrow \quad 0 = 0p + q \\ A_1 - 7B_2 = 3 \quad &\Rightarrow \quad 2 = 1p + q \\ A_2 - 7B_4 = 6 \quad &\Rightarrow \quad 4 = 2p + q \end{aligned}$$

# Finding Simple Array Relations

- Hypothesize

$$A[i] = lB[j] + k, \qquad i \in \{0, 1, 2\}$$

- Find $j$
  - Express relation between $A[i]$ and $B[j]$ as $j = ip + q$

$$\begin{aligned} A_0 - 7B_0 = 0 \quad &\Rightarrow \quad 0 = 0p + q \\ A_1 - 7B_2 = 3 \quad &\Rightarrow \quad 2 = 1p + q \\ A_2 - 7B_4 = 6 \quad &\Rightarrow \quad 4 = 2p + q \end{aligned}$$

  - Solve for $p, q$

$$\begin{aligned} \{q = 0, p = 2\} \quad &\Rightarrow \quad j = 2i \\ &\Rightarrow \quad A[i] = lB[2i] + k \end{aligned}$$

# Finding Simple Array Relations

- Hypothesize

$$A[i] = lB[j] + k, \qquad i \in \{0, 1, 2\}$$

- Find $j$
  - Express relation between $A[i]$ and $B[j]$ as $j = ip + q$

$$\begin{aligned} A_0 - 7B_0 = 0 \quad &\Rightarrow \quad 0 = 0p + q \\ A_1 - 7B_2 = 3 \quad &\Rightarrow \quad 2 = 1p + q \\ A_2 - 7B_4 = 6 \quad &\Rightarrow \quad 4 = 2p + q \end{aligned}$$

  - Solve for $p, q$

$$\begin{aligned} \{q = 0, p = 2\} \quad &\Rightarrow \quad j = 2i \\ &\Rightarrow \quad A[i] = lB[2i] + k \end{aligned}$$

- Find $l, k$

$$A[i] = 7B[2i] + 3i$$

# Nested Array Relations

Examples

invSubBytes : $R[i][j] = S[T[i][j]]$

**Find nested array relations of the grammar**

$$\begin{aligned} A[i_1]\cdots[i_k] &\mapsto e \\ e &\mapsto B[e]\cdots[e] \end{aligned}$$

E.g. $A[i][j] = B[C[j+3]][D[E[2i+j]]$

Method

- Uses reachability analysis to find potential nesting relations
- Reduces to a satisfiability problem and solves with a theorem prover

# Finding Nested Array Relations

- Given

$$A = [7, 1, -3], B = [1, -3, 5, 1, 0, 7, 1], C = [8, 5, 6, 6, 2, 1, 4]$$

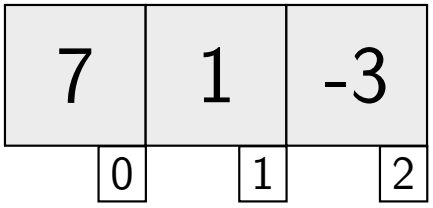
- Generate nestings

$$A[i] = B[\dots], A[i] = C[\dots], \dots, C[i] = A[\dots], A[i] = B[C[\dots]], \dots$$

- Validate nestings

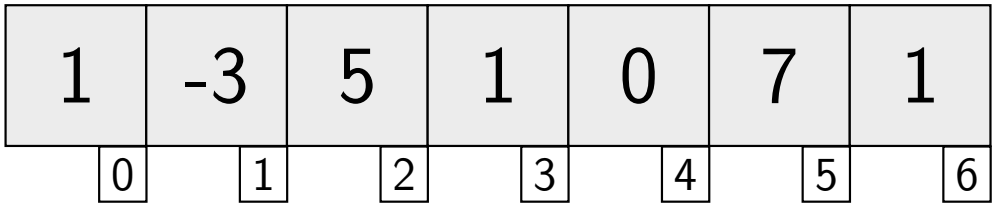
$$\text{Discard } B[i] = C[\dots] \text{ because } B[1] \notin C$$

# Reachability Analysis on $A[i] = B[C[\ldots]]$

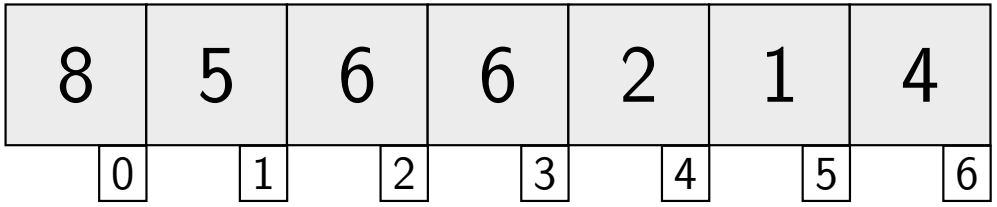



**A**

$$A[0] \stackrel{?}{=} B[C[\ldots]]$$
$$A[1] \stackrel{?}{=} B[C[\ldots]]$$
$$A[2] \stackrel{?}{=} B[C[\ldots]]$$




**B**

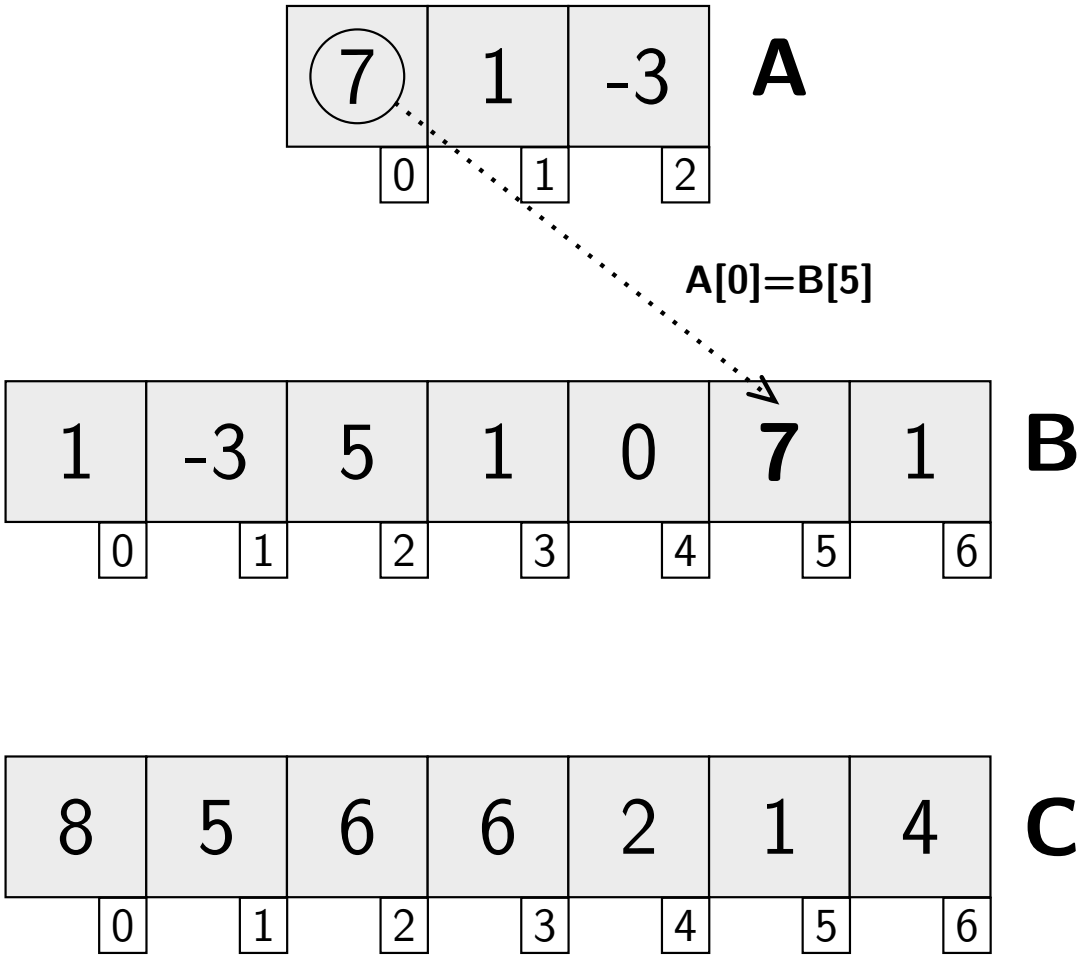



**C**

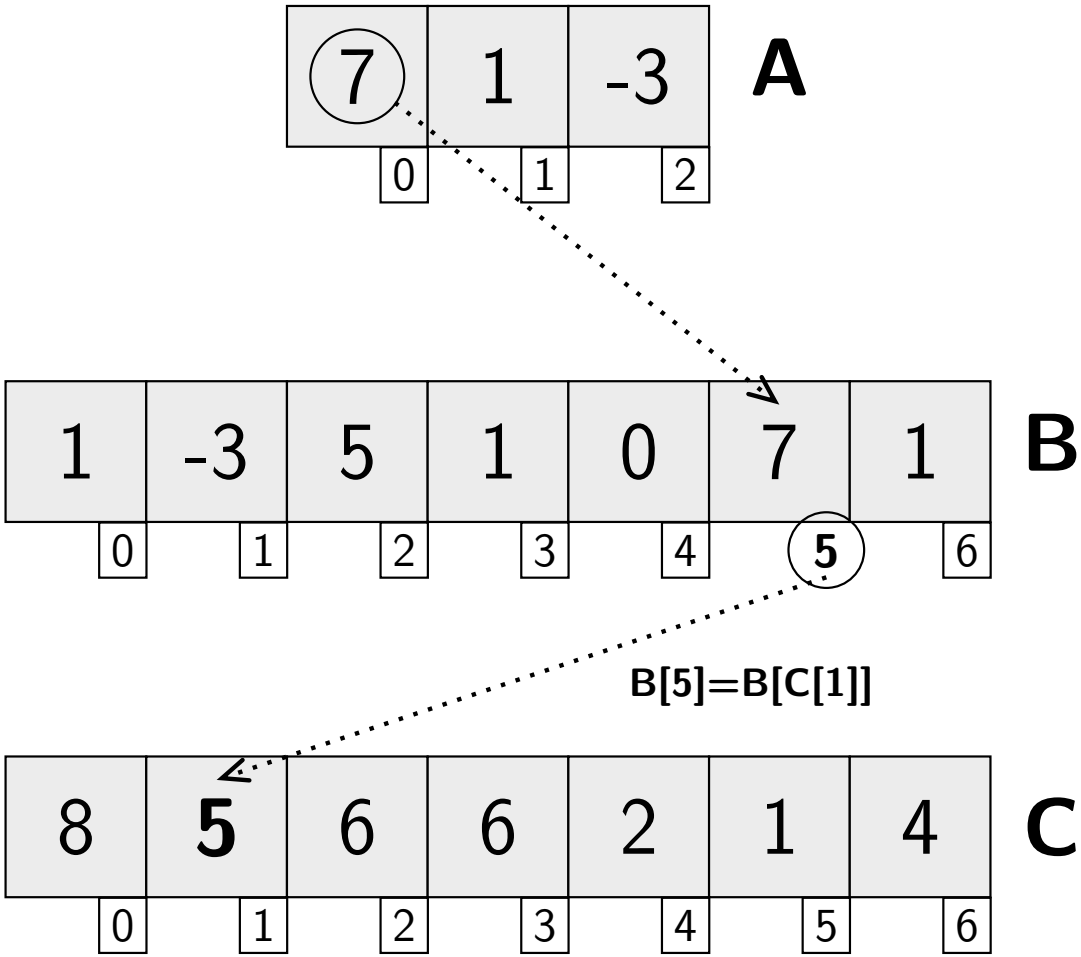


UNM

# Reachability Analysis on $A[i] = B[C[\ldots]]$

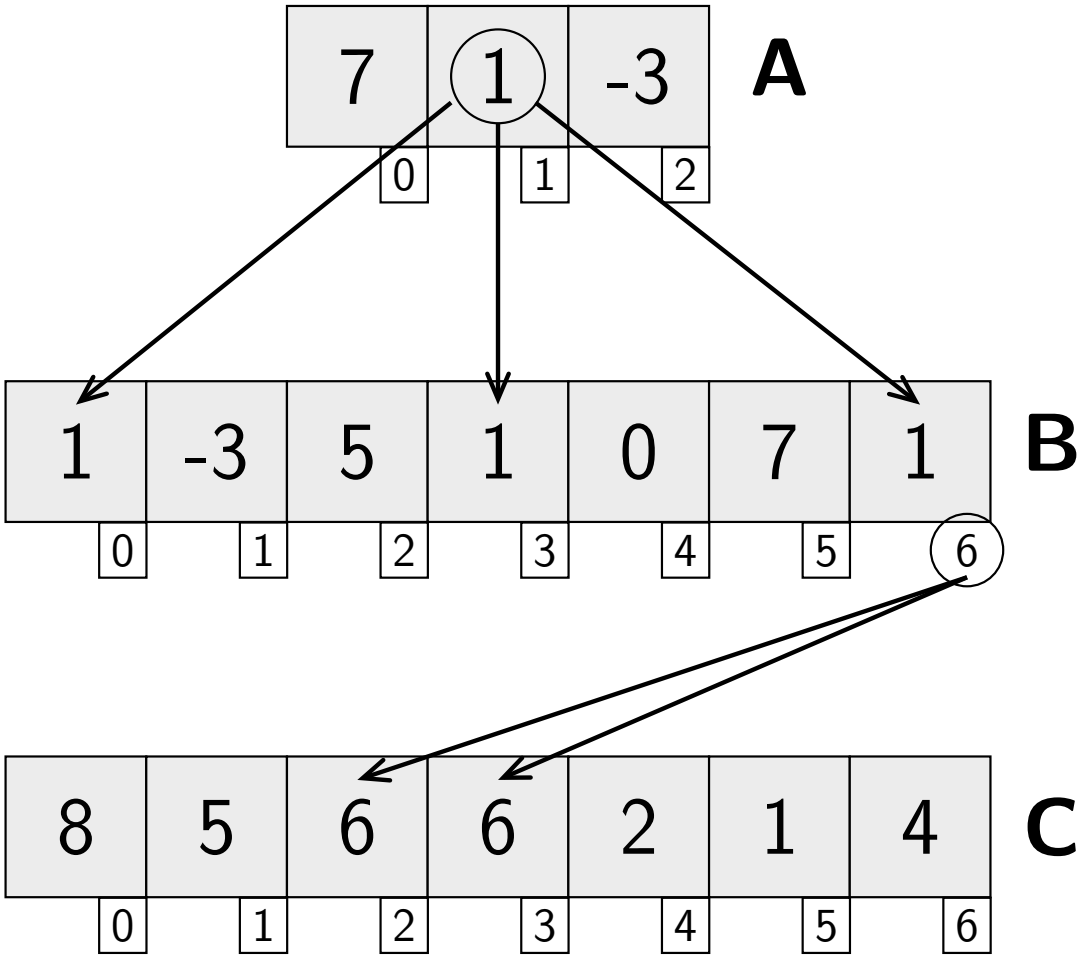


# Reachability Analysis on $A[i] = B[C[\ldots]]$

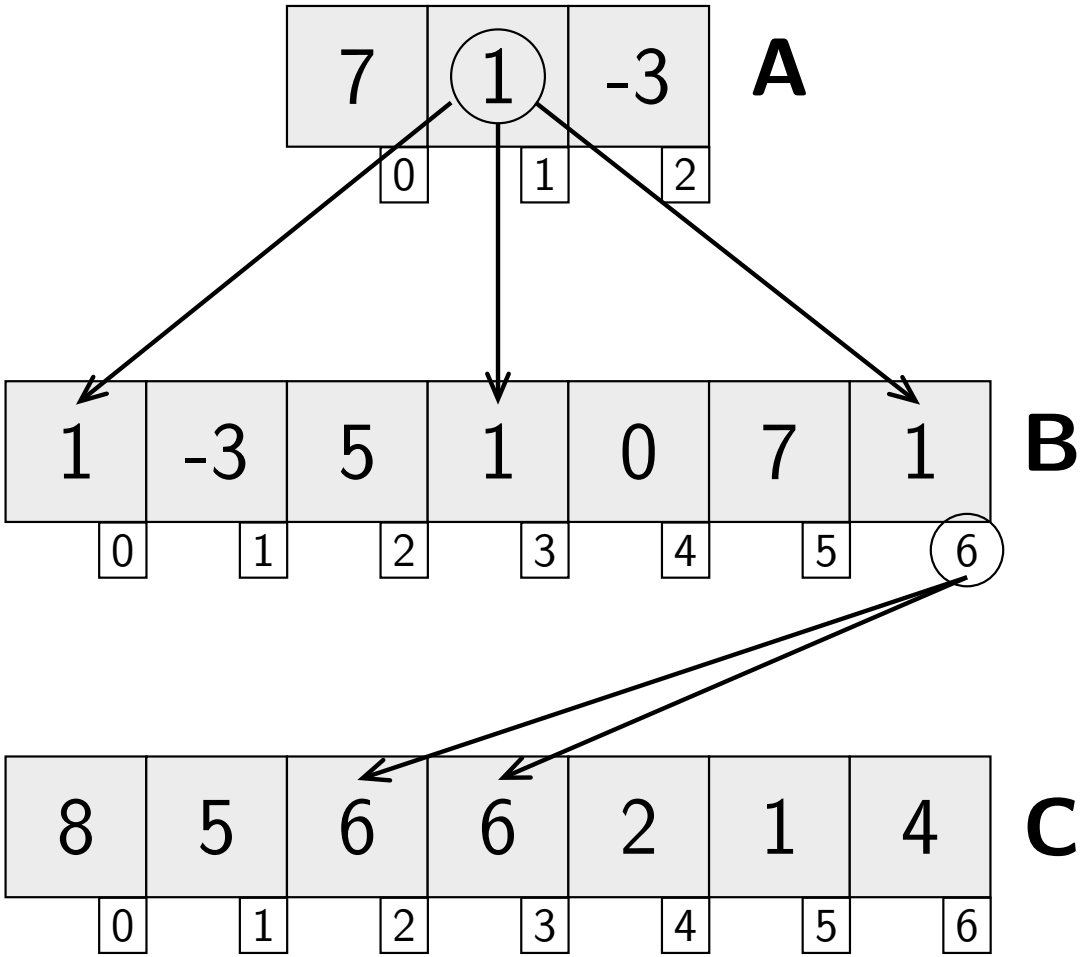



$$A[0] \quad = \quad B[C[1]]$$

# Reachability Analysis on $A[i] = B[C[\ldots]]$

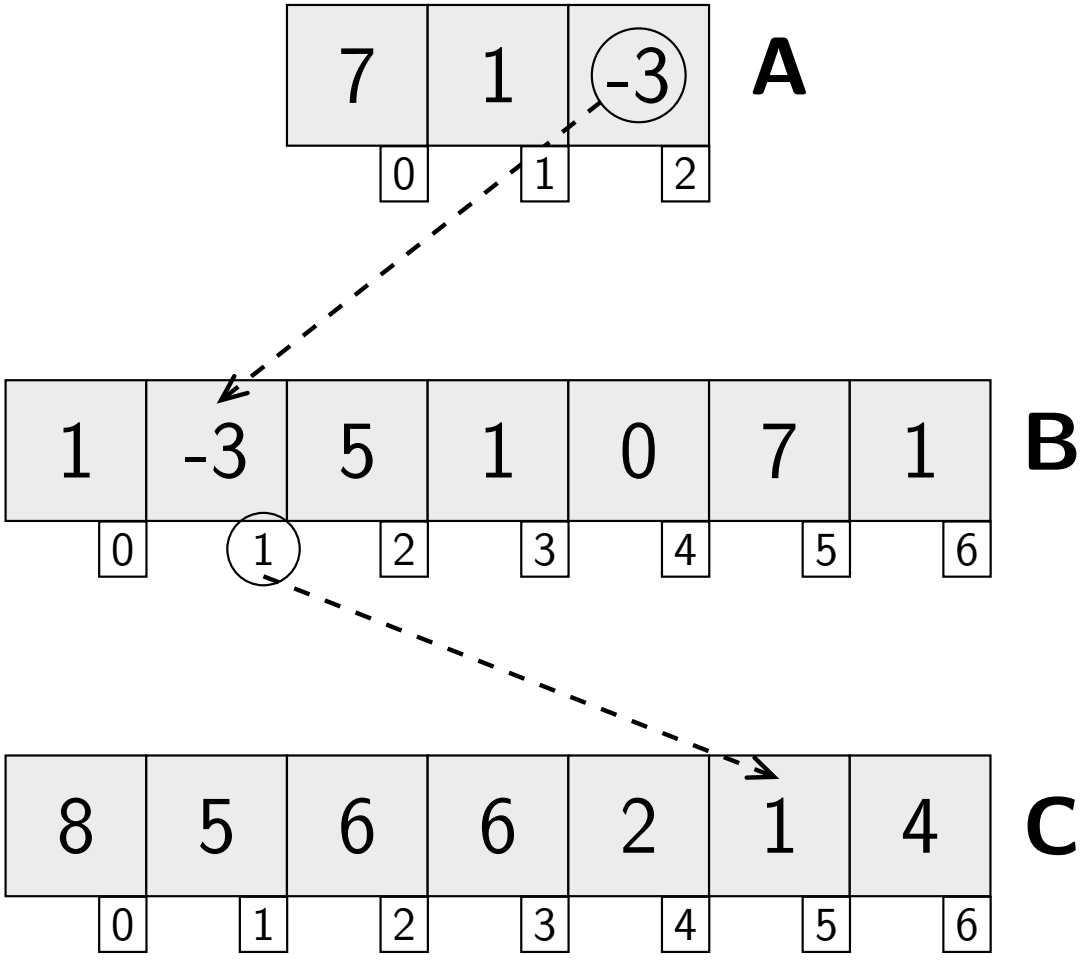


# Reachability Analysis on $A[i] = B[C[\ldots]]$




$$A[0] = B[C[1]]$$
$$A[1] = B[C[2]] \vee B[C[3]]$$

# Reachability Analysis on $A[i] = B[C[\ldots]]$

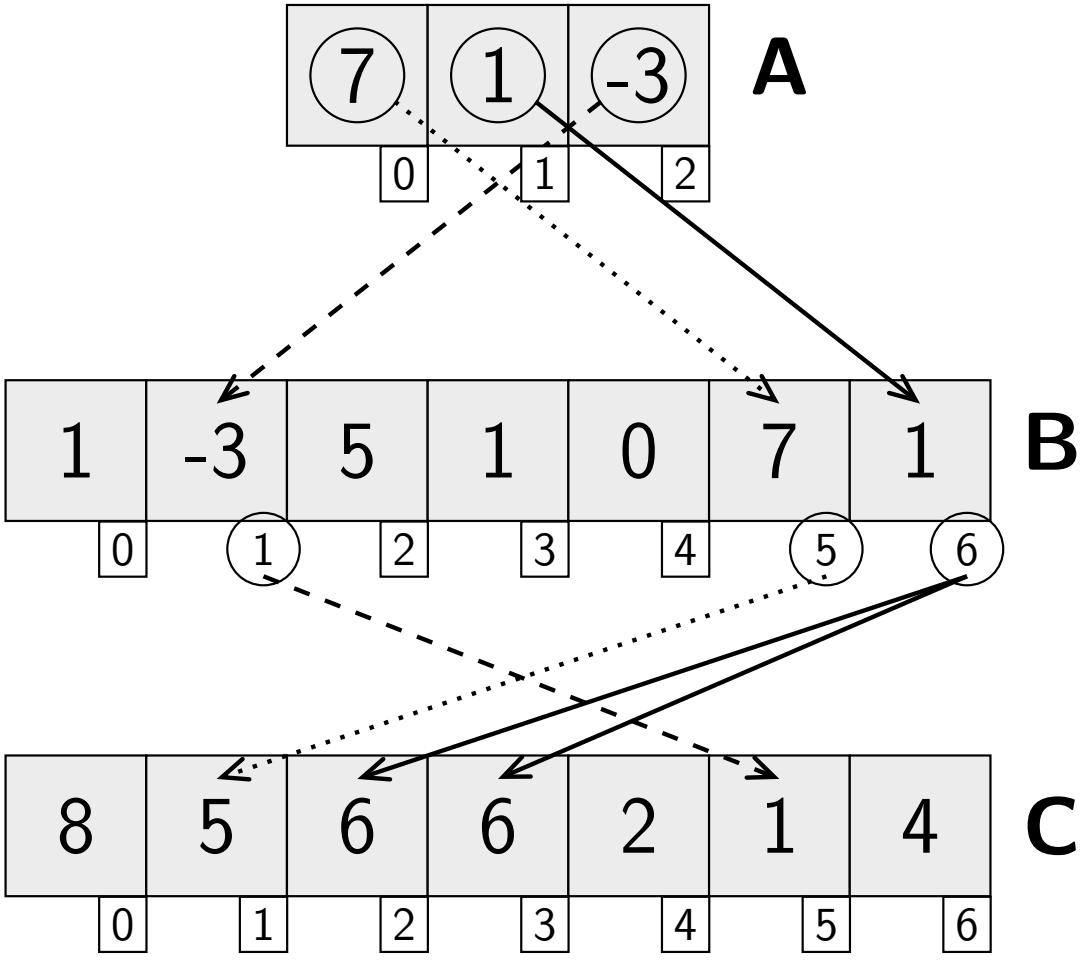



$$A[0] = B[C[1]]$$
$$A[1] = B[C[2]] \vee B[C[3]]$$
$$A[2] = B[C[5]]$$



UNM

# Reachability Analysis on $A[i] = B[C[\ldots]]$




$$
\begin{aligned}
A[0] &= B[C[1]] \\
A[1] &= B[C[2]] \vee B[C[3]] \\
A[2] &= B[C[5]] \\
&\Downarrow \\
A[i] &= B[C[j]]
\end{aligned}
$$

# Equation Solving for $A[i] = B[C[j]]$

$$
\begin{aligned}
A[0] &= B[C[1]] \\
A[1] &= B[C[2]] \vee B[C[3]] \\
A[2] &= B[C[5]]
\end{aligned}
$$

- Express relation between $A[i]$ and $B[C[j]]$ as $j = ip + q$

# Equation Solving for $A[i] = B[C[j]]$

$$\begin{aligned} A[0] &= B[C[1]] \\ A[1] &= B[C[2]] \vee B[C[3]] \\ A[2] &= B[C[5]] \end{aligned}$$

- Express relation between $A[i]$ and $B[C[j]]$ as $j = ip + q$

$$\{1 = 0p + q, 2 = 1p + q, 5 = 2p + q\}$$

# Equation Solving for $A[i] = B[C[j]]$

$$\begin{aligned} A[0] &= B[C[1]] \\ A[1] &= B[C[2]] \vee B[C[3]] \\ A[2] &= B[C[5]] \end{aligned}$$

- Express relation between $A[i]$ and $B[C[j]]$ as $j = ip + q$

$$\{1 = 0p + q, 2 = 1p + q, 5 = 2p + q\}$$

- Solve for $p, q$

No Solution

# Equation Solving for $A[i] = B[C[j]]$

$$\begin{aligned} A[0] &= B[C[1]] \\ A[1] &= B[C[2]] \vee B[C[3]] \\ A[2] &= B[C[5]] \end{aligned}$$

- Express relation between $A[i]$ and $B[C[j]]$ as $j = ip + q$

$$\{1 = 0p + q, 3 = 1p + q, 5 = 2p + q\}$$

# Equation Solving for $A[i] = B[C[j]]$

$$\begin{aligned} A[0] &= B[C[1]] \\ A[1] &= B[C[2]] \vee B[C[3]] \\ A[2] &= B[C[5]] \end{aligned}$$

- Express relation between $A[i]$ and $B[C[j]]$ as $j = ip + q$

$$\{1 = 0p + q, 3 = 1p + q, 5 = 2p + q\}$$

- Solve for $p, q$

$$\begin{aligned} \{q = 1, p = 2\} &\Rightarrow j = 2i + 1 \\ &\Rightarrow A[i] = B[C[2i + 1]] \end{aligned}$$

# Supporting Functions

Examples

$$\text{addRoundKey} : R[i][j] = \text{xor}(T[i][j], H[i][j])$$
$$\text{multWord} : R[i] = T[\text{mod}(L[A[i]] + L[B[i]], 255)]$$

Treat functions as a special type of arrays

$$m(2,3) = 6, \quad m(-1,1) = -1, \quad m(0,0) = 0, \ldots$$
$$\Downarrow\Downarrow$$
$$M[2][3] = 6, \quad M[-1][1] = -1, \quad M[0][0] = 0, \ldots$$

# Using SMT solver

Apply SMT solving to improve reachability analysis

$$A[0] = B[C[1]]$$
$$A[1] = B[C[2]] \vee B[C[3]]$$
$$A[2] = B[C[5]]$$
$$\Downarrow\Downarrow$$
$$(0p + q = 1) \wedge (1p + q = 2 \vee 1p + q = 3) \wedge (2p + q = 5)$$

# Results for Array Invariants

| Function | Desc | Inv Type | \|Arrays\| | Dimension | Annotated Invs |
|---|---|---|---|---|---|
| multWord | mult | N(4) | 7 | 2 | 1 |
| xor2Word | xor | N(1) | 4 | 2 | 1 |
| xor3Word | xor | N(1) | 5 | 3 | 1 |
| subWord | subs | N(1) | 3 | 1 | 1 |
| rotWord | shift | S | 2 | 1 | 1 |
| block2State | convert | S | 2 | 2 | 1 |
| state2Block | convert | S | 2 | 2 | 1 |
| subBytes | subs | N(1) | 3 | 2 | 1 |
| invSubByte | subs | N(1) | 3 | 2 | 1 |
| shiftRows | shift | S | 2 | 2 | 1 |
| invShiftRow | shift | S | 2 | 2 | 1 |
| addKey | add | N(1) | 4 | 2 | 1 |
| mixCol | mult | U | 4 | 2 | 1 |
| invMixCol | mult | U | 4 | 2 | 1 |
| keySetEnc4 | driver | S,U | 2 | 2 | 2 |
| keySetEnc6 | driver | S,U | 2 | 2 | 2 |
| keySetEnc8 | driver | S,U | 2 | 2 | 2 |
| keySetEnc | driver | U | 4 | 1 | 1 |
| keySetDec | driver | U | 4 | 2 | 1 |
| keySched1 | driver | U | 3 | 2 | 1 |
| keySched2 | driver | U | 3 | 2 | 1 |
| aesKeyEnc | driver | eq,U | 7 | 2 | 2 |
| aesKeyDec | driver | eq,U | 7 | 2 | 2 |
| aesEncrypt | driver | U | 8 | 4 | 1 |
| aesDecrypt | driver | U | 8 | 4 | 1 |
| **25 functions** | **N**=Nested, **S**=Simple, **U**=unsupported | | | | **30** |

# Results for Array Invariants

| Function | Desc | Inv Type | \|Arrays\| | Dimension | Annotated Invs | Discovered Invs | Time (secs) |
|---|---|---|---|---|---|---|---|
| multWord | mult | N(4) | 7 | 2 | 1 | 1 | 3.6 |
| xor2Word | xor | N(1) | 4 | 2 | 1 | 1 | 0.1 |
| xor3Word | xor | N(1) | 5 | 3 | 1 | 1 | 0.1 |
| subWord | subs | N(1) | 3 | 1 | 1 | 1 | 0.4 |
| rotWord | shift | S | 2 | 1 | 1 | 1 | 0.5 |
| block2State | convert | S | 2 | 2 | 1 | 1 | 2.0 |
| state2Block | convert | S | 2 | 2 | 1 | 1 | 11.7 |
| subBytes | subs | N(1) | 3 | 2 | 1 | 1 | 0.6 |
| invSubByte | subs | N(1) | 3 | 2 | 1 | 1 | 3.8 |
| shiftRows | shift | S | 2 | 2 | 1 | 1 | 12.2 |
| invShiftRow | shift | S | 2 | 2 | 1 | 1 | 8.3 |
| addKey | add | N(1) | 4 | 2 | 1 | 1 | 0.6 |
| mixCol | mult | U | 4 | 2 | 1 | 0 | - |
| invMixCol | mult | U | 4 | 2 | 1 | 0 | - |
| keySetEnc4 | driver | S,U | 2 | 2 | 2 | 1 | 4.5 |
| keySetEnc6 | driver | S,U | 2 | 2 | 2 | 1 | 6.7 |
| keySetEnc8 | driver | S,U | 2 | 2 | 2 | 1 | 10.6 |
| keySetEnc | driver | U | 4 | 1 | 1 | 0 | - |
| keySetDec | driver | U | 4 | 2 | 1 | 0 | - |
| keySched1 | driver | U | 3 | 2 | 1 | 0 | - |
| keySched2 | driver | U | 3 | 2 | 1 | 0 | - |
| aesKeyEnc | driver | eq,U | 7 | 2 | 2 | 1 | 0.1 |
| aesKeyDec | driver | eq,U | 7 | 2 | 2 | 1 | 0.1 |
| aesEncrypt | driver | U | 8 | 4 | 1 | 0 | - |
| aesDecrypt | driver | U | 8 | 4 | 1 | 0 | - |
| **25 functions** | **N**=Nested, **S**=Simple, **U**=unsupported | | | | **30** | **17** | **tot 65.9** |

# Summary

Dynamic Analysis

- Polynomial Invariants
  - Equalities: solve equations over (nonlinear) terms
  - Inequalities: construct a polyhedron over trace points
    Use deduction when additional information is available
- Array Invariants
  - Simple relations: find relations among individual elements
  - Nested relations: perform reachability analysis and use SMT solving

Results

- Identify 100% of the nonlinear invariants in 24 arithmetic algorithms
- Find 60% of the array relations of an AES implementation
- Current dynamic analysis work cannot find these invariants

# Thank you for your attention !

Project is open source and available at

http://code.google.com/p/invgen/

Ask me about (or read the paper for details)

- refining and dealing with spurious invariants (more theorem proving)
- complexities of the techniques (e.g. finding nested arrays is NP-complete)
- additional invariants (e.g. disjunctive properties)
- using discovered invariants to repair programs (current work)