Termination Analysis of Integer Linear Loops

Aaron R. Bradley, Zohar Manna, and Henny B. Sipma

Computer Science Department

Stanford University

Outline

Goal:
Invariant generation and termination analysis of loops
with integer variables.

Outline:

- Motivating example.
- Main technique.
- Empirical observations.

Example

$$\texttt{unsigned int } x,\ y,\ x_0$$

$$\Theta\text{: } x > 0 \ \wedge \ x\%2 = 0$$

$$P_1 :: \left[\begin{array}{l} \textbf{do} \\ \quad x_0 := x; \\ \quad \textbf{if } x\%3 = 0 \\ \quad \textbf{then } x := x - 2 \\ \quad \textbf{else } x := x - \frac{x}{2} \\ \textbf{while } x_0 \neq x \end{array}\right] \quad \| \quad P_2 :: \left[\begin{array}{l} \textbf{while } y > x \textbf{ do} \\ \quad y := y - x \\ \textbf{done} \end{array}\right]$$

Do processes P_1 and P_2 always exit?

(Note: strong assumptions about atomicity.)

Example

$$\texttt{unsigned int } x,\ y,\ x_0$$
$$\Theta\text{: } x > 0 \ \wedge \ x\%2 = 0$$

$$P_1 :: \left[\begin{array}{l} \textbf{do} \\ \quad x_0 := x; \\ \quad \textbf{if } x\%3 = 0 \\ \quad \textbf{then } x := x - 2 \\ \quad \textbf{else } x := x - \frac{x}{2} \\ \textbf{while } x_0 \neq x \end{array}\right] \quad \| \quad P_2 :: \left[\begin{array}{l} \textbf{while } y > x \textbf{ do} \\ \quad y := y - x \\ \textbf{done} \end{array}\right]$$

Abstract program:

$$\begin{array}{ll} \texttt{uint } x,\ y & \\ \Theta : & x > 0 \ \wedge \ x\%2 = 0 \\ \tau_1 : & x\%3 = 0 \ \wedge \ x' = x - 2 \ \wedge \ y' = y \\ \tau_2 : & x\%3 \neq 0 \ \wedge \ x' = x - \frac{x}{2} \ \wedge \ y' = y \\ \tau_3 : & y > x \ \wedge \ y' = y - x \ \wedge \ x' = x \end{array}$$

Example: Invariants

$$
\begin{array}{ll}
\texttt{uint}\ x,\ y & \\
\Theta: & x > 0\ \wedge\ x\%2 = 0 \\
\tau_1: & x\%3 = 0\ \wedge\ x' = x - 2\ \wedge\ y' = y \\
\tau_2: & x\%3 \neq 0\ \wedge\ x' = x - \frac{x}{2}\ \wedge\ y' = y \\
\tau_3: & y > x\ \wedge\ y' = y - x\ \wedge\ x' = x
\end{array}
$$

Invariant: $\boxed{x \geq 1}\ \wedge\ y \geq 0$

Initiation $x > 0\ \Rightarrow\ x \geq 1$

Consecution

- τ_1: $x \geq 1\ \wedge\ x\%3 = 0\ \Rightarrow\ x' = x - 2 \geq 1$
- τ_2: $x > 1\ \Rightarrow\ x' = x - \frac{x}{2} \geq 1$

  $x = 1\ \Rightarrow\ \frac{x}{2} = 0\ \Rightarrow\ x' = x$
- τ_3: $x' = x$

Example: Termination

$\texttt{uint}\ x,\ y$

$\Theta: \quad x > 0 \ \wedge \ x\%2 = 0$

$\tau_1: \quad x\%3 = 0 \ \wedge \ x' = x - 2 \ \wedge \ y' = y$

$\tau_2^a: \quad x\%3 \neq 0 \ \wedge \ x' = x - \frac{x}{2} \ \wedge \ y' = y \ \wedge \ x' \neq x$

$\tau_2^b: \quad x\%3 \neq 0 \ \wedge \ x' = x - \frac{x}{2} \ \wedge \ y' = y \ \wedge \ x' = x \quad$ (ignore)

$\tau_3: \quad y > x \ \wedge \ y' = y - x \ \wedge \ x' = x$

Choose **ranking function** $\boxed{x + y}$

Bounded $x \geq 1 \ \wedge y \geq 0 \ \Rightarrow \ x + y \geq 0$

Ranking

- τ_1: $x' = x - 2 \ \Rightarrow \ x' + y' < x + y$
- τ_2^a: $x > 1 \ \Rightarrow \ \frac{x}{2} \geq 1 \ \Rightarrow \ x' < x \ \Rightarrow \ x' + y' < x + y$

  $x = 1 \ \Rightarrow \ \frac{x}{2} = 0 \ \Rightarrow \ x' = x \ \Rightarrow$ not taken
- τ_3: $x \geq 1 \wedge y' = y - x \ \Rightarrow \ y' < y \ \Rightarrow \ x' + y' < x + y$

Example: Termination

$$\texttt{unsigned int } x,\ y,\ x_0$$
$$\Theta\colon\ x > 0\ \wedge\ x\%2 = 0$$

$$P_1 :: \left[\begin{array}{l} \textbf{do} \\ \quad x_0 := x; \\ \quad \textbf{if } x\%3 = 0 \\ \quad \textbf{then } x := x - 2 \\ \quad \textbf{else } x := x - \frac{x}{2} \\ \textbf{while } x_0 \neq x \end{array}\right] \quad \| \quad P_2 :: \left[\begin{array}{l} \textbf{while } y > x \textbf{ do} \\ \quad y := y - x \\ \textbf{done} \end{array}\right]$$

$\Rightarrow$ Terminates on all input.

Goal:

> Synthesize **invariants** and **linear ranking functions** automatically.

Outline

- **Related work**
- Loop abstraction & definitions
- Synthesis
- Observations
- Conclusion

Related Work: Termination

Colón & Sipma 2001, 2002

Polyhedra-based synthesis of linear ranking functions for linear loops.

Podelski & Rybalchenko 2004

Complete method, over loops with one transition and without an initial condition.

Bradley, Manna & Sipma 2005

Complete method for lexicographic linear ranking functions over linear loops.

Cousot 2005

Incomplete but efficient method for synthesis of polynomial ranking functions over polynomial loops.

All methods analyze loops with **real** variables.

Related Work: Invariant Generation

Cousot & Halbwachs 1978

Abstract interpretation with polyhedra.

Colón, Sankaranarayanan & Sipma 2003

Constraint-based linear invariant generation.

Etc.

All methods analyze loops with **real** variables.

Our Contribution

Loops with **real** variables:

- Polyhedra for invariant generation.
- Strict duality of real linear constraints for constraint-based analysis of linear loops.
- Weak duality of real polynomial constraints for constraint-based analysis of polynomial loops.

No known duality for integer linear constraints.

Our contribution: Complete method for synthesis of

- linear ranking functions
- linear invariants

over linear loops with **integer** variables.

Application: Computer Arithmetic

Müller-Olm & Seidl 2005

- Single fixed modulo. No arbitrary modulo or division.
- Affine invariants (equations) with modular arithmetic.
- Very efficient.

Sound (inequality) invariant generation for computer arithmetic:

- unsigned: $x + y \;\Rightarrow\; (x + y)\%2^n$
- signed: $x + y \;\Rightarrow\; (x + y + 2^{n-1})\%2^n - 2^{n-1}$

$\Rightarrow$ linear inequality invariants sound for computer arithmetic

Outline

- Related work
- **Loop abstraction & definitions**
- Synthesis
- Observations
- Conclusion

Integer Linear Loop

Integer Variable:

- x of type `int`: domain is $\mathbb{Z}$
- x of type `uint`: domain is $\mathbb{Z}^*$

Integer Linear Formula:

Term c, cx, $c_1 \frac{E}{c_2}$, $c_1(E\%c_2)$
for constants $c, c_1 \in \mathbb{Z}$, $c_2 \in \mathbb{Z}^+$
for integer variable x
for expression E

Expression E, the summation of terms

Atom $E_1 \bowtie E_2$, $\bowtie \in \{<, \leq, =, \neq, \geq, >\}$

Formula Boolean combination of atoms

Integer Linear Loop

Loop Abstraction:

$$L : \langle \mathcal{V}_{\mathbb{Z}}, \mathcal{V}_{\mathbb{Z}^*}, \Theta, \mathcal{T} \rangle$$

Variables $\mathcal{V} = \mathcal{V}_{\mathbb{Z}} \cup \mathcal{V}_{\mathbb{Z}^*}$

- $x \in \mathcal{V}_{\mathbb{Z}}$ have type `int`
- $x \in \mathcal{V}_{\mathbb{Z}^*}$ have type `uint`

Initial Condition $\Theta(\mathcal{V})$ is integer linear formula

Transitions $\tau(\mathcal{V}, \mathcal{V}') \in \mathcal{T}$ are integer linear formulae

Presburger Loop

Presburger Formula: integer linear formula without / and %

Presburger Loop: integer linear loop with Presburger formulae

$$\mathcal{P}(\mathcal{A}[c_1 \tfrac{E}{c_2}]) \;=\; (\exists a)(\exists b \in [0..c_2-1]) \left[\begin{array}{ll} & [c_2 a + b = E \wedge E \geq 0 \wedge \mathcal{P}(\mathcal{A}[c_1 a])] \\ \vee & [c_2 a - b = E \wedge E \leq 0 \wedge \mathcal{P}(\mathcal{A}[c_1 a])] \end{array} \right]$$

$$\mathcal{P}(\mathcal{A}[c_1 (E \% c_2)]) \;=\; (\exists a)(\exists b \in [0..c_2-1]) \left[\begin{array}{ll} & [c_2 a + b = E \wedge E \geq 0 \wedge \mathcal{P}(\mathcal{A}[c_1 b])] \\ \vee & [c_2 a - b = E \wedge E \leq 0 \wedge \mathcal{P}(\mathcal{A}[-c_1 b])] \end{array} \right]$$

$$\mathcal{P}(\mathcal{A}) \;=\; \mathcal{A}$$

Example: Presburger Loop

$\texttt{uint}\ x,\ y$

$\Theta: \quad x > 0\ \wedge\ x\%2 = 0$

$\tau_1: \quad x\%3 = 0\ \wedge\ x' = x - 2\ \wedge\ y' = y$

$\tau_2: \quad x\%3 \neq 0\ \wedge\ x' = x - \frac{x}{2}\ \wedge\ y' = y$

$\tau_3: \quad y > x\ \wedge\ y' = y - x\ \wedge\ x' = x$

$\Downarrow$

$\texttt{uint}\ x,\ y$

$\Theta: \quad (\exists a)[x > 0\ \wedge\ 2a = x]$

$\tau_1: \quad (\exists a)[3a = x\ \wedge\ x' = x - 2\ \wedge\ y' = y]$

$\tau_2: \quad (\exists a, b)[3a \neq x \wedge (2b = x \vee 2b + 1 = x) \wedge x' = x - b \wedge y' = y]$

$\tau_3: \quad y > x\ \wedge\ y' = y - x\ \wedge\ x' = x$

Example: Presburger Loop

$\texttt{uint}\ x,\ y$

$\Theta: \quad (\exists a)[x > 0\ \wedge\ 2a = x]$

$\tau_1: \quad (\exists a)[3a = x\ \wedge\ x' = x - 2\ \wedge\ y' = y]$

$\tau_2: \quad (\exists a, b)[3a \neq x \wedge (2b = x \vee 2b + 1 = x) \wedge x' = x - b \wedge y' = y]$

$\tau_3: \quad y > x\ \wedge\ y' = y - x\ \wedge\ x' = x$

$\Downarrow$

$\texttt{uint}\ x,\ y,\ a,\ b$

$\Theta: \quad x > 0\ \wedge\ 2a = x$

$\tau_1: \quad 3a = x\ \wedge\ x' = x - 2\ \wedge\ y' = y$

$\tau_2: \quad 3a \neq x\ \wedge\ 2b = x\ \wedge\ x' = x - b\ \wedge\ y' = y$

$\tau_3: \quad 3a \neq x\ \wedge\ 2b + 1 = x\ \wedge\ x' = x - b\ \wedge\ y' = y$

$\tau_4: \quad y > x\ \wedge\ y' = y - x\ \wedge\ x' = x$

(but do synthesis over x and y)

Linear Notation

Consider $\mathcal{V} = \{x_1, \ldots, x_n\}$.

homogenous vector:

$$\mathbf{x} = (x_1, \ldots, x_n, 1)^{\mathrm{T}}$$

linear assertion:

$$\mathbf{Ax} \geq 0$$

$$\begin{bmatrix} a_{1,1} & \cdots & a_{1,n} & a_{1,n+1} \\ & & \vdots & \\ a_{k,1} & \cdots & a_{k,n} & a_{k,n+1} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ 1 \end{bmatrix} \geq \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$$\bigwedge_{i \in \{1,\ldots,k\}} (a_{i,1}x_1 + \cdots + a_{i,n}x_n + a_{i,n+1} \geq 0)$$

for integer matrix $\mathbf{A}$

Linear Inductive Invariant

Consider $L : \langle \mathcal{V}_{\mathbb{Z}}, \mathcal{V}_{\mathbb{Z}^*}, \Theta, \mathcal{T} \rangle$ with $\mathcal{V} = \{x_1, \ldots, x_n\}$.

$\mathbf{Ix} \geq \mathbf{0}$, for integer matrix $\mathbf{I}$, is a linear inductive invariant if

(Initiation)

$$(\forall \mathbf{x})[\Theta \ \rightarrow \ \mathbf{Ix} \geq \mathbf{0}]$$

(Consecution)

$$(\forall \tau \in \mathcal{T})(\forall \mathbf{x}, \mathbf{x}')[\mathbf{Ix} \geq \mathbf{0} \wedge \tau \ \rightarrow \ \mathbf{Ix}' \geq \mathbf{0}]$$

Linear Ranking Function

Consider $L : \langle \mathcal{V}_{\mathbb{Z}}, \mathcal{V}_{\mathbb{Z}^*}, \Theta, \mathcal{T} \rangle$ with $\mathcal{V} = \{x_1, \ldots, x_n\}$.

$\mathbf{r}^{\mathrm{T}}\mathbf{x}$, for integer vector $\mathbf{r}$, is a **linear ranking function** with supporting invariant $\mathbf{Ix} \geq \mathbf{0}$ if

(Bounded)

$$(\forall \tau \in \mathcal{T})(\forall \mathbf{x})[\mathbf{Ix} \geq \mathbf{0} \wedge \tau \ \rightarrow \ \mathbf{r}^{\mathrm{T}}\mathbf{x} \geq 0]$$

(Ranking)

$$(\forall \tau \in \mathcal{T})(\forall \mathbf{x}, \mathbf{x}')[\mathbf{Ix} \geq \mathbf{0} \wedge \tau \ \rightarrow \ \mathbf{r}^{\mathrm{T}}\mathbf{x}' < \mathbf{r}^{\mathrm{T}}\mathbf{x}]$$

Template Expression & Assertion

Consider $\mathcal{V} = \{x_1, \ldots, x_n\}$.

Template Expression:

$$\mathbf{c}^{\mathrm{T}}\mathbf{x} \;=\; c_1 x_1 + \cdots + c_n x_n + c_{n+1}$$

for unknown integer coefficients $\mathbf{c}$

Template Assertion:

$$\mathbf{Ix} \geq \mathbf{0}$$

for unknown integer coefficients $\mathbf{I}$

Outline

- Related work
- Loop abstraction & definitions
- **Synthesis**
- Observations
- Conclusion

Overview

Given: $L : \langle \mathcal{V}_{\mathbb{Z}}, \mathcal{V}_{\mathbb{Z}^*}, \Theta, \mathcal{T} \rangle$

Goal: Synthesize ranking function and supporting invariants.

Method:

- Construct **synthesis template** $\varphi[P]$.
  - Unknown parameters P.
  - Encodes ranking and invariant conditions.
- Existence of ranking function $\Leftrightarrow (\exists P)(\forall \mathbf{x}, \mathbf{x}')\varphi[P]$.
- Search for instantiation $\widetilde{P}$ of P such that $(\forall \mathbf{x}, \mathbf{x}')\varphi[\widetilde{P}]$.
- All validity checks are decidable (Presburger arithmetic).

Ranking Function Synthesis Template

Consider $L : \langle \mathcal{V}_{\mathbb{Z}}, \mathcal{V}_{\mathbb{Z}^*}, \Theta, \mathcal{T} \rangle$.

$P = \{\text{unknown coefficients } \mathbf{I}, \mathbf{r}\}$

$$
\varphi[P]: \quad
\begin{array}{lll}
 & \Theta \rightarrow \mathbf{Ix} \geq 0 & \text{(initiation)} \\
\wedge & \bigwedge_{\tau \in \mathcal{T}} \left[(\mathbf{Ix} \geq 0 \wedge \tau) \rightarrow \mathbf{Ix}' \geq 0 \right] & \text{(consecution)} \\
\wedge & \bigwedge_{\tau \in \mathcal{T}} \left[(\mathbf{Ix} \geq 0 \wedge \tau) \rightarrow \mathbf{r}^{\mathrm{T}}\mathbf{x} \geq 0 \right] & \text{(bounded)} \\
\wedge & \bigwedge_{\tau \in \mathcal{T}} \left[(\mathbf{Ix} \geq 0 \wedge \tau) \rightarrow \mathbf{r}^{\mathrm{T}}\mathbf{x}' < \mathbf{r}^{\mathrm{T}}\mathbf{x} \right] & \text{(ranking)}
\end{array}
$$

Motivation

Could enumerate P vectors.

Impractical. Instead:

- Examine (infinite) sets of P vectors.
- Eliminate a set if (provably) no member validates $(\forall \mathbf{x}, \mathbf{x}')\varphi[\widetilde{P}]$.
- Test a member of the set.
- If test fails, divide and recurse.

Parameter Region, Instance & Corner

Region: For P, closed hyper-rectangle

$$R = [\mathbf{l}, \mathbf{u}] \in \mathbb{R}^{|P|}$$

Instance: For $\widetilde{P} \in \mathbb{Z}^{|P|}$, $\widetilde{P} \in R$ if

$$(\exists \mathbf{r} \in \mathbb{Q}^{|P|} \cap R)(\exists s > 0)[\widetilde{P} = s \cdot \mathbf{r}]$$

Corner:

- P corresponding to extreme point $\mathbf{r} \in R$.
- Each dimension $\mathbf{r}_i$ is assigned ℓ_i or $\mathbf{u}_i$ from R.
- At **lower** corner, every $\mathbf{r}_i$ is assigned ℓ_i.

TERMINATES

Given $L : \langle \mathcal{V}_{\mathbb{Z}}, \mathcal{V}_{\mathbb{Z}^*}, \Theta, \mathcal{T} \rangle$ and synthesis template $\varphi[P]$.

Search:

1. Initialize set of regions S to $\{[-1, 1]^{|P|}\}$.
2. While S is not empty
   (a) Choose R from S.
   (b) If R does not contain a solution (**infeasible**), prune.
   (c) If corner $\widetilde{P} \in R$ validates $(\forall \mathbf{x}, \mathbf{x}')\varphi[\widetilde{P}]$, return **true**.
   (d) If above maximum depth, bisect R and add pair to S.

Problem: 2(b)

Pruning Regions

How can we prove that a region is infeasible?

> If **relaxed** version of the problem is invalid, then each instance is invalid.

Relaxed problem is in Presburger arithmetic.

Relaxation

Consider

- template $\varphi[P]$
- $R : [\mathbf{l}, \mathbf{u}]$ for P

Relaxation φ_R: Replace each inequality

$$\alpha \geq \beta$$

of $\varphi[P]$ with

$$\overline{\alpha} \geq \underline{\beta}$$

where $\overline{\alpha}$ is obtained by

- replacing each term p with u_p
- replacing each term px with $u_p x$ if $\varphi[P]$ requires $x \geq 0$
- replacing px with $\ell_p x$ otherwise

and $\underline{\beta}$ is similarly obtained.

Quadrant Completion

Works for `uint`. What about `int` variables?

Consider $L : \langle \mathcal{V}_{\mathbb{Z}}, \mathcal{V}_{\mathbb{Z}^*}, \Theta, \mathcal{T} \rangle$, $\varphi[P]$.

Quadrant completion $\widehat{\varphi}[P]$:

$$\left(\bigwedge_{x \in \mathcal{V}_{\mathbb{Z}}} (x \bowtie_x 0 \ \wedge \ x' \bowtie_{x'} 0) \right) \ \rightarrow \ \varphi[P],$$

$\bowtie \in \{\leq, \geq\}$

Forces each $x \in \mathcal{V}_{\mathbb{Z}}$ and $x' \in \mathcal{V}'_{\mathbb{Z}}$ to range over either $\mathbb{Z}^*$ or $-\mathbb{Z}^*$.

Infeasible Region

Lemma

Given $\varphi[P]$, R for P. If for some $\widehat{\varphi}$[P],

$$(\forall \mathbf{x}, \mathbf{x}')\widehat{\varphi}_R$$

is invalid, then R is infeasible.

$\Rightarrow$ Criterion for pruning regions.

Sound & Complete

Theorem

Given $L : \langle \mathcal{V}_{\mathbb{Z}}, \mathcal{V}_{\mathbb{Z}^*}, \Theta, \mathcal{T} \rangle$, supporting invariant template size sz, and maximum search depth $D \cdot |P|$.

Assume

- bisection bisects one of the widest dimensions of an R;
- the **lower** corner $\widetilde{P}$ is checked as a potential solution of an R.

Then L has a linear ranking function
with supporting sz-conjunct linear invariant,
expressed so that all coefficients are integers in $[-2^{D-1},\ 2^{D-1})$

iff

(TERMINATES L isz) returns **true**.

Sound & Complete

What does it mean?

- If $45x + 37y + 253z + 19 \geq 0$ is an invariant, then if $D = 8$, it will be found.
- Incrementing D results in a complete method.

Synthesis Template: Variant

$$\varphi[P]: \quad (\exists \mathbf{i}, r)(\forall \mathbf{x}, \mathbf{x}') \left[\begin{array}{ll} & \Theta \rightarrow \mathbf{I}\mathbf{x} + \mathbf{i} \geq 0 \\ \wedge & \bigwedge_{\tau \in \mathcal{T}} \left[(\mathbf{I}\mathbf{x} + \mathbf{i} \geq 0 \wedge \tau) \rightarrow \mathbf{I}\mathbf{x}' + \mathbf{i} \geq 0 \right] \\ \wedge & \bigwedge_{\tau \in \mathcal{T}} \left[(\mathbf{I}\mathbf{x} + \mathbf{i} \geq 0 \wedge \tau) \rightarrow \mathbf{r}^{\mathrm{T}}\mathbf{x} + r \geq 0 \right] \\ \wedge & \bigwedge_{\tau \in \mathcal{T}} \left[(\mathbf{I}\mathbf{x} + \mathbf{i} \geq 0 \wedge \tau) \rightarrow \mathbf{r}^{\mathrm{T}}\mathbf{x}' < \mathbf{r}^{\mathrm{T}}\mathbf{x} \right] \end{array} \right]$$

Instances still decidable.

Invariant Generation

Similar technique.

Challenge: Avoid regions with weaker assertions than already-discovered invariants.

Solution: Subsumption check.

Outline

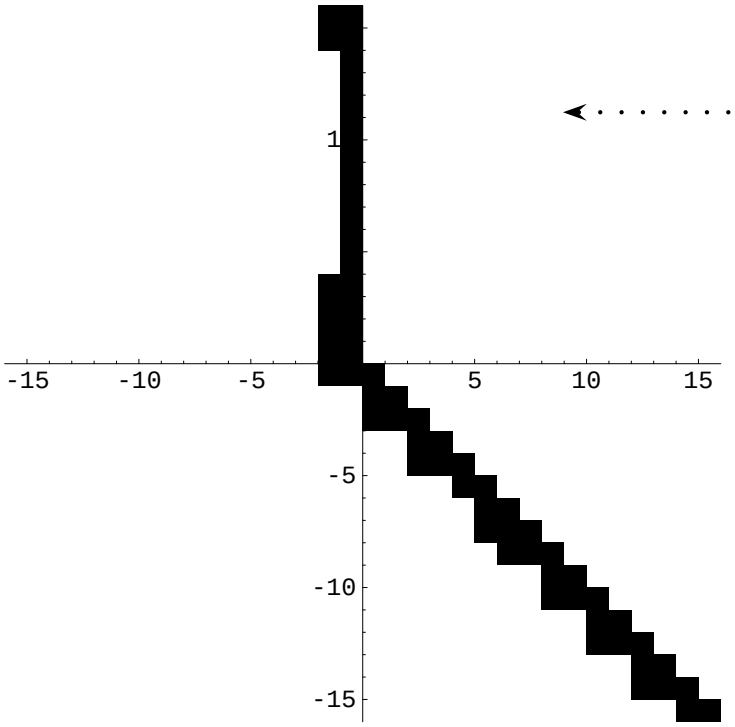
- Related work
- Loop abstraction & definitions
- Synthesis
- **Observations**
- Conclusion

Picture: Invariant Generation

$\texttt{uint}\ x$

$\Theta:\ x > 0$

$\tau_1:\ 2x > 1 \wedge x' = x - \frac{x}{2}$



true

$x \geq 1$

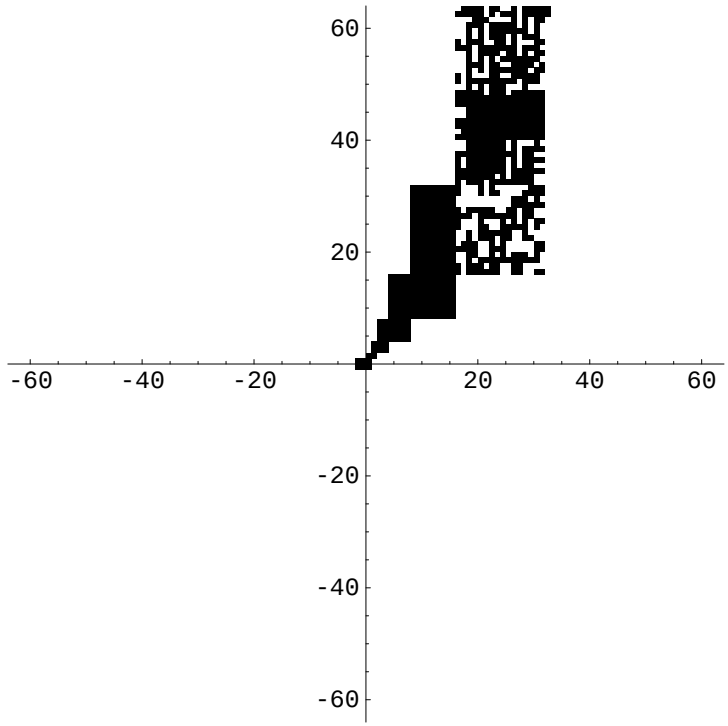
Picture: Termination

$\texttt{int}\ i,\ j,\ k$

$\Theta:\ \top$

$\tau_1:\ i + 17j + 33k > 0 \wedge i' = i + 16 \wedge j' = j - 1 \wedge k' = k$

$\tau_2:\ i + 17j + 33k > 0 \wedge i' = i + 32 \wedge j' = j \wedge k' = k - 1$



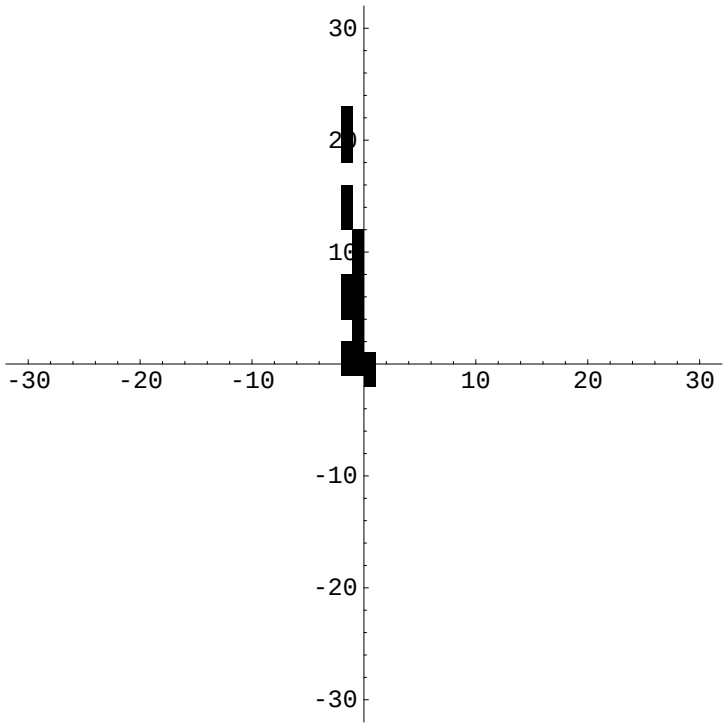
Picture: Termination

$\texttt{uint}\ x, s$

$\Theta:\ s = 1$

$\tau_1:\ x > 100 \wedge s \neq 1 \wedge x' = x - 10 \wedge s' = s - 1$

$\tau_2:\ x \leq 100 \wedge x' = x + 11 \wedge s' = s + 1$



Summary of Experiments

#Vars (I/T)	**Time**	**Depth**	**Regions**
3 (T)	20s	2	—
3 (T)	5s	2	—
2 (I)	1s	5	8%
3 (T)	100s	7	.07%
2 (T)	1s	6	1%
3 (I)	45s	5	5%
8 (I)	300s	2	—
4 (T)	15s	2	—

Outline

- Related work
- Loop abstraction & definitions
- Synthesis
- Observations
- **Conclusion**

Conclusion

- Complete method for synthesis of linear ranking functions over integer linear loops.
- Complete method for linear invariant generation over integer linear loops.
- Future work:
  - Heuristics to guide bisection dimension, to choose region from queue.
  - Strengthen check for infeasible regions.
  - Hybrid analysis with real-variable techniques.

Thank you!