Principles of Abstract Interpretation
MIT press

Ch. **31**, Cartesian congruence analysis

Patrick Cousot

pcousot.github.io

PrAbsInt@gmail.com github.com/PrAbsInt/

These slides are available at
http://github.com/PrAbsInt/slides/slides-31--cartesian-congruences-PrAbsInt.pdf

Chapter 31

Ch. **31**, Cartesian congruence analysis

- The cartesian congruence analysis discovers congruence properties $x = a \pmod{b}$ of values $x \in \mathbb{Z}$ of integer variables $\mathtt{x}$ where the integer coefficients $a, b \in \mathbb{N}$ are automatically inferred by the analysis.
- It generalizes the constancy analysis $x = c \pmod{0}$ and the parity analysis $x = p \pmod{2}$, $p \in \{0, 1\}$.
- Our task is to formalize the congruence abstraction and then derive the value congruence domain

$$\mathbb{D}^{\equiv} \quad \triangleq \quad \langle \mathbb{P}^{\equiv}, \sqsubseteq^{\equiv}, \bot^{\equiv}, \top^{\equiv}, \sqcup^{\equiv}, \sqcap^{\equiv}, 1^{\equiv}, \ominus^{\equiv}, \ominus^{\equiv_1}, \olessthan^{\equiv}, \overline{\olessthan}^{\equiv} \rangle \tag{31.24}$$

  abstracting the collecting domain

$$\langle \wp(\mathbb{V}), \subseteq, \varnothing, \mathbb{V}, \cup, \cap, \{1\}, \ominus, \ominus^{\underset{\triangle}{\times}_1}, \olessthan^{\underset{\triangle}{\times}_1}, \overline{\olessthan}^{\underset{\triangle}{\times}_1} \rangle$$

  by congruences
- This defines an instance of the abstract interpreter performing congruence analysis

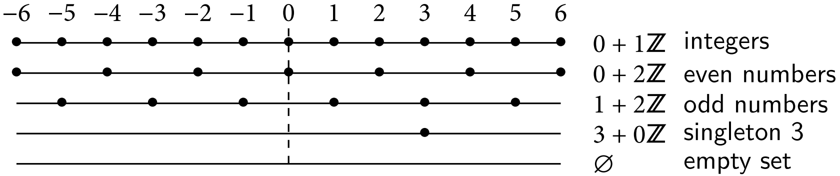
Congruence abstract properties

- After normalization of Section **30.3**, the congruence abstract properties are

$$\mathbb{P}^{\equiv} \quad \triangleq \quad \{\varnothing\} \cup \{c + m\mathbb{Z} \mid c \in \mathbb{Z} \wedge m \in \mathbb{N} \wedge \llbracket m > 0 \mathrel{?} 0 \leqslant c < m \mathrel{:} \mathtt{tt} \rrbracket\}$$

  where $c + m\mathbb{Z} \triangleq c[m] \triangleq \{z \in \mathbb{Z} \mid \exists k \in \mathbb{Z} \,.\, z = c + km\}$.

- Example:



Example of cartesian congruence analysis

- Consider the program

  `while` ℓ_1 `(x < 100) { if` ℓ_2 `(odd(y))` ℓ_3 `x = x + 3 ; else` ℓ_4 `x = x + 6 ;` ℓ_5 `y = y + 1 ; }`ℓ_6

  where variables are initialized to 0.

- Initially $x = y = 0 + 0\mathbb{Z}$ at ℓ_1 and ℓ_2. $y = 0 + 0\mathbb{Z}$ is even so after execution of the iteration body $x = 0 + 0\mathbb{Z}$ or $x = 6 + 0\mathbb{Z}$ and $y = 0 + 0\mathbb{Z}$ or $y = 1 + 0\mathbb{Z}$ at ℓ_1.

- It follows that $x = 0 + 6\mathbb{Z}$ and $y = 0 + 1\mathbb{Z}$ at ℓ_1 hence at ℓ_2 since the test `x < 100` provides no congruence information.

- $y = 0 + 1\mathbb{Z}$ can be odd or even at ℓ_2 so at ℓ_5 we get either $x = 3 + 6\mathbb{Z}$ in the first case and $x = 6 + 6\mathbb{Z} = 0 + 6\mathbb{Z}$ in the second case.

- Therefore $x = 0 + 3\mathbb{Z}$ and $y = 1 + 1\mathbb{Z} = 0 + 1\mathbb{Z}$ at the end of the loop body.

- The join of $x = 0 + 6\mathbb{Z}$ and $x = 0 + 3\mathbb{Z}$ at ℓ_1 yields $x = 0 + 3\mathbb{Z}$ while $y = 0 + 1\mathbb{Z}$ is stable.

- One more iteration yields $x = 3 + 3\mathbb{Z}$ or $x = 6 + 3\mathbb{Z}$ that is $x = 0 + 3\mathbb{Z}$ which is stable.

- We conclude that x is congruent to 3 and y can be any integer value.

Another example:

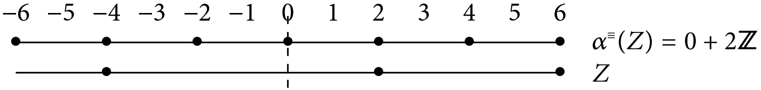
```
while l1: (x < 100) [x:0+3Z; y:0+6Z]
   {
      if l2: (y == 0) [x:0+3Z; y:0+6Z]
         l3: [x:0+3Z; y:0+6Z] x = (x + 3);
       else
         l4: [x:0+3Z; y:0+6Z] x = (x + 6);
      l5: [x:0+3Z; y:0+6Z] y = (y + 6);
   }
l6: [x:0+3Z; y:0+6Z]
```

Congruence abstraction

- Let us define the *congruence abstraction*

$$
\begin{aligned}
\alpha^{\equiv}(\varnothing) &\triangleq \varnothing && (31.2)\\
\alpha^{\equiv}(\{c\}) &\triangleq c + 0\mathbb{Z} && \text{singleton}\\
\alpha^{\equiv}(Z) &\triangleq \text{let } m = \gcd\{|x-y| \mid x, y \in Z \land x \neq y\} && \text{otherwise}\\
&\quad\ \text{and } c = \min\{c' \mid 0 \leqslant c' < m \land \exists k \in \mathbb{Z} \mathrel{.} c' + km \in Z\} \text{ in}\\
&\qquad c + m\mathbb{Z}\\
\gamma^{\equiv}(\varnothing) &\triangleq \varnothing && (31.3)\\
\gamma^{\equiv}(c + m\mathbb{Z}) &\triangleq \{z \in \mathbb{Z} \mid \exists k \in \mathbb{Z} \mathrel{.} z = c + km\}
\end{aligned}
$$

- Example:



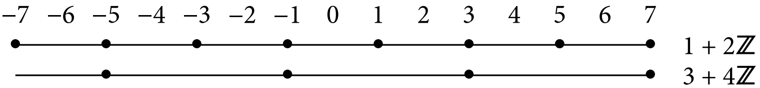
$Z = \{-4, 2, 6\}$. The gcd of 6, 4 and 10 is $m = 2$.

- The abstract inclusion order $\sqsubseteq^{\equiv}$ is

$$P \sqsubseteq^{\equiv} Q \quad \triangleq \quad \gamma^{\equiv}(P) \subseteq \gamma^{\equiv}(Q)$$

Theorem (31.5) [1] $\forall P \in \mathbb{P}^{\equiv} \,.\, \varnothing \sqsubseteq^{\equiv} P$ and

$$c + m\mathbb{Z} \sqsubseteq^{\equiv} c' + m'\mathbb{Z} \quad \Leftrightarrow \quad c \equiv c' \pmod{m'} \wedge m' \,/\, m \qquad \square$$



$$3 + 4\mathbb{Z} \sqsubseteq^{\equiv} 1 + 2\mathbb{Z} \Leftrightarrow 3 \equiv 1 \pmod{2} \wedge 2 \,/\, 4$$

Theorem (31.6) We have the Galois retraction $\langle \wp(\mathbb{Z}),\ \subseteq \rangle \xleftarrow[\alpha^{\equiv}]{\gamma^{\equiv}} \langle \mathbb{P}^{\equiv},\ \sqsubseteq^{\equiv} \rangle$. $\square$

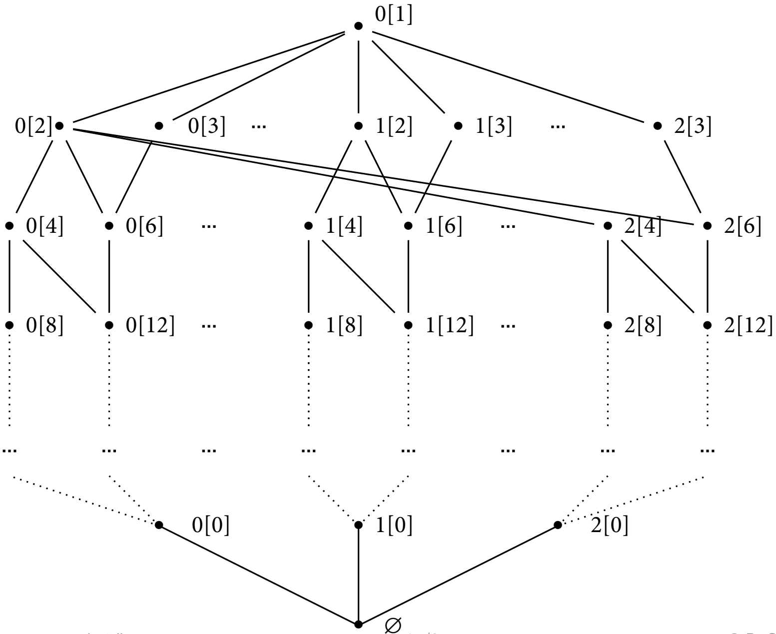
[1] See the proofs in the book.

The congruence complete lattice

- $\langle \wp(\mathbb{Z}),\ \subseteq\rangle \xrightleftharpoons[\alpha^{\equiv}]{\gamma^{\equiv}} \langle \mathbb{P}^{\equiv},\ \sqsubseteq^{\equiv}\rangle$ is a Galoios retraction
- The congruence abstract properties form a complete lattice.

Corollary (31.8) The image of $\langle \wp(\mathbb{Z}),\ \subseteq,\ \varnothing,\ \mathbb{Z},\ \cup,\ \cap\rangle$ by the lower adjoint $\alpha^{\equiv}$ is the complete lattice $\langle \mathbb{P}^{\equiv},\ \sqsubseteq^{\equiv},\ \varnothing,\ 0+1\mathbb{Z},\ \sqcup^{\equiv},\ \sqcap^{\equiv}\rangle$.

(see Section **10.6**)

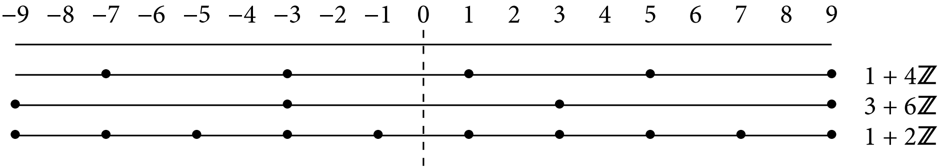

0[1]
0[2]
0[3]
...
1[2]
1[3]
...
2[3]
0[4]
0[6]
...
1[4]
1[6]
...
2[4]
2[6]
0[8]
0[12]
...
1[8]
1[12]
...
2[8]
2[12]
... ...
0[0]
1[0]
2[0]
∅

Congruence lub

Theorem (31.10, Congruence join/lub) [2]

$$c + m\mathbb{Z} \sqcup^{\equiv} c' + m'\mathbb{Z} \quad = \quad c + (m \operatorname{gcd} m' \operatorname{gcd} |c - c'|)\mathbb{Z} \qquad \square$$

Example:



$$1 + 4\mathbb{Z} \sqcup^{\equiv} 3 + 6\mathbb{Z} = 1 + (4 \operatorname{gcd} 6 \operatorname{gcd} |1 - 3|)\mathbb{Z} = 1 + 2\mathbb{Z}$$

[2] See the proofs in the book.

Congruence disjointness

- For the test x == y to be possibly true, the equivalence classes of x and y must not be disjoint.

Theorem (31.12, disjointness) $c + m\mathbb{Z} \sqcap^{\equiv} c' + m'\mathbb{Z} \neq \emptyset$ if and only if $c \equiv c' \pmod{m \gcd m'}$.

Example $1 + 4\mathbb{Z} \sqcap^{\equiv} 0 + 6\mathbb{Z} = \emptyset$ since $1 \not\equiv 0 \pmod{2}$.
$1 + 4\mathbb{Z} \sqcap^{\equiv} 3 + 6\mathbb{Z} \neq \emptyset$ since $1 \equiv 3 \pmod{2}$.

Congruence glb

- When the test `x == y` is true, both `x` and `y` belong to the intersection of their congruence classes.

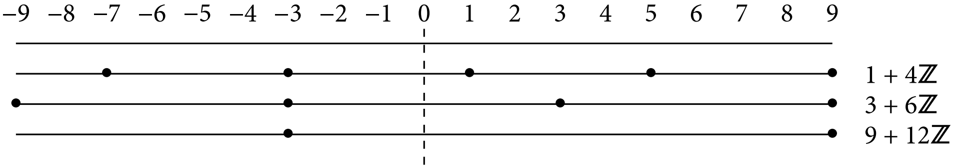
Theorem (31.14, Congruence meet/glb)

$$\begin{array}{rcll} c + m\mathbb{Z} \sqcap^{\equiv} c' + m'\mathbb{Z} & = & c'' + (m \operatorname{lcm} m')\mathbb{Z} & \text{if } c \equiv c' \pmod{(m \gcd m')} \\ & = & \varnothing & \text{otherwise} \end{array}$$

with $c'' \equiv c + \dfrac{c' - c}{m \gcd m'} xm \equiv c' + \dfrac{c - c'}{m \gcd m'} ym' \pmod{m \operatorname{lcm} m'}$ where, by Bachet-Bézout's identity of Theorem 30.7, $x, y \in \mathbb{Z}$ are such that $xm + ym' = m \gcd m'$.

In case $m = m' = 0$, we have $c + 0\mathbb{Z} \sqcap^{\equiv} c' + 0\mathbb{Z} = \varnothing$ when $c \neq c'$ and otherwise $c + 0\mathbb{Z} \sqcap^{\equiv} c + 0\mathbb{Z} = c + 0\mathbb{Z}$.

- **Example:**



We have the Bachet-Bézout's identity $(-1) \times 4 + 1 \times 6 = gcd(4, 6) = 2$ so the intersection is

$$c'' \equiv 1 + \frac{3-1}{4 \operatorname{gcd} 6} \times (-1) \times 4 \equiv -3 \equiv 3 + \frac{1-3}{4 \operatorname{gcd} 6} \times 1 \times 6 \equiv -3 \pmod{(4 \operatorname{lcm} 6)} \equiv 9 \pmod{12}.$$

$\square$

Abstract congruence operations

Theorem (31.18) $\ominus^{\equiv}(c + m\mathbb{Z}) = -c + m\mathbb{Z}$.

Proof of Theorem 31.18

$$
\begin{array}{ll}
& \ominus^{\equiv}\, c + m\mathbb{Z} \\
= & (\alpha^{\equiv}(\ominus\gamma^{\equiv}(c + m\mathbb{Z})) \\
= & \alpha^{\equiv}(\ominus\{z \in \mathbb{Z} \mid \exists k \in \mathbb{Z} \,.\, z = c + km\}) \hfill \text{\textbf{(}def. (31.3) of } \gamma^{\equiv}\text{\textbf{)}} \\
= & \alpha^{\equiv}(\{-z \in \mathbb{Z} \mid \exists k \in \mathbb{Z} \,.\, z = c + km\}) \hfill \text{(def. } \ominus\text{)} \\
= & \alpha^{\equiv}(\{z \in \mathbb{Z} \mid \exists k' \in \mathbb{Z} \,.\, z = -c + k'm\}) \hfill (k' = -k) \\
= & \alpha^{\equiv}(\gamma^{\equiv}(-c + m\mathbb{Z})) \hfill \text{(def. (31.3) of } \gamma^{\equiv}\text{)} \\
= & -c + m\mathbb{Z} \qquad \text{(Theorem 31.6 where } \alpha^{\equiv} \text{ is surjective and Exercise 11.49)} \quad \square
\end{array}
$$

Theorem (31.19)

- $\varnothing \oplus^{\equiv} \varnothing = c + m\mathbb{Z} \oplus^{\equiv} \varnothing = \varnothing \oplus^{\equiv} c' + m'\mathbb{Z} = \varnothing$
- $c + m\mathbb{Z} \oplus^{\equiv} c' + m'\mathbb{Z} = (c + c') + (m \gcd m')\mathbb{Z}$.

- $c + m\mathbb{Z} \ominus^{\equiv} c' + m'\mathbb{Z} = (c - c') + (m \operatorname{gcd} m')\mathbb{Z}$.
- $c + m\mathbb{Z} \otimes^{\equiv} c' + m'\mathbb{Z} = (cc') + (mm' \operatorname{gcd} (mc' \operatorname{gcd} m'c))\mathbb{Z}$.
- $$\begin{array}{rcll} c + m\mathbb{Z} \oslash^{\equiv} c' + m'\mathbb{Z} & = & \varnothing & \text{if } c' + m'\mathbb{Z} = 0 + 0\mathbb{Z} \\ & = & (c/c') + m/|c'|\mathbb{Z} & \text{if } m' = 0,\ c' \neq 0,\ c' \,/\, m, \text{ and } c' \,/\, c \\ & = & 0 + 1\mathbb{Z} & \text{otherwise} \end{array}$$
- [Granger, 1989] proposes a more precise abstract division $\oslash^{\equiv}$ as well as an abstract modulo operation.

The congruence abstract domain

- The value congruence domain (28.42) is

$$\mathbb{D}^{\equiv} \quad \triangleq \quad \langle \mathbb{P}^{\equiv}, \sqsubseteq^{\equiv}, \bot^{\equiv}, \top^{\equiv}, \sqcup^{\equiv}, \sqcap^{\equiv}, 1^{\equiv}, \ominus^{\equiv}, \ominus^{\equiv 1}, \otimes^{\equiv}, \overline{\otimes}^{\equiv} \rangle \tag{31.24}$$

  from which the reachability congruence domain (28.43) is derived as in Chapter **28**.
- By Corollary 31.8, $\langle \mathbb{P}^{\equiv}, \sqsubseteq^{\equiv}, \bot^{\equiv}, \top^{\equiv}, \sqcup^{\equiv}, \sqcap^{\equiv} \rangle$ is a complete lattice where
  - $\sqsubseteq^{\equiv}$ is the order of Theorem 31.5,
  - $\bot^{\equiv} \triangleq \varnothing$,
  - $\top^{\equiv} \triangleq 0 + 1\mathbb{Z}$,
  - the lub $\sqcup^{\equiv}$ is defined in Theorem 31.10,
  - and the glb $\sqcap^{\equiv}$ by Theorem 31.14.
- $1^{\equiv} \triangleq 1 + 0\mathbb{Z}$, $\ominus^{\equiv}$ and $\ominus^{\equiv 1}$ are given above.
- The comparison operators $\otimes^{\equiv}$ and $\overline{\otimes}^{\equiv}$ are the identity but when one argument is $\bot^{\equiv}$, both are constants which can be compared, or the comparison is an equality.

Iteration

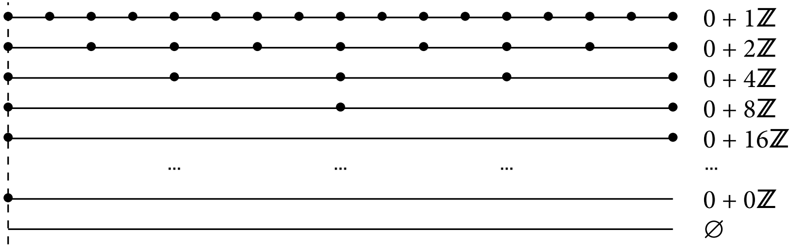
- Increasing iterations always converge in finitely many steps.

Theorem (31.25, No infinite strictly increasing chain) $\langle \mathbb{P}^{\equiv}, \sqsubseteq^{\equiv} \rangle$ has no infinite strictly increasing chain.

Proof of Theorem 31.25 Assume the increasing chain starts with $c + m\mathbb{Z}$ with $m \neq 0$. Any strictly greater congruence class has the form $c + n\mathbb{Z}$ where n divides m and $n \neq m$. It follows that $2|n| \leqslant |m|$ so if the chain is strictly increasing it contains at most $1 + \log_2(m)$ elements $\square$

- $\langle \mathbb{P}^{\equiv}, \sqsubseteq^{\equiv} \rangle$ has infinite strictly decreasing chains.

Example (Infinite strictly decreasing chains)



- It follows that the convergence of the local iterations of Section **29.2** must be enforced by a narrowing such as $\top^{\equiv} \mathbin{\Delta^{\equiv}} \overline{P} \triangleq \overline{P}$ and $\overline{P} \mathbin{\Delta^{\equiv}} \overline{Q} \triangleq \overline{P}$ otherwise.
- More details on narrowings in Sections **33.6** and **34.8**.

Conclusion

- The cartesian integer congruence static analysis is due to Philippe Granger [Granger, 1989] who generalized to rationals [Granger, 1997] and linear congruence equalities [Granger, 1991].
- François Masdupuy introduced interval congruences $[a,b]+m\mathbb{Z}$ [Masdupuy, 1993] and trapezoidal congruences [Masdupuy, 1992].
- Antoine Miné designed zone congruences [Miné, 2002].
- The cartesian congruence analysis is used in Astrée [Bertrane, P. Cousot, R. Cousot, Feret, Mauborgne, Miné, and Rival, 2015] *e.g.* to check that data structures are well-aligned on word boundaries in computer memory *e.g.* to multiples of the word size.

Bibliography I

Bertrane, Julien, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, and Xavier Rival (2015). "Static Analysis and Verification of Aerospace Software by Abstract Interpretation". *Foundations and Trends in Programming Languages* 2.2-3, pp. 71–190.

Granger, Philippe (1989). "Static Analysis of Arithmetical Congruences". *International Journal of Computer Mathematics* 30.3 & 4, pp. 165–190.

— (1991). "Static Analysis of Linear Congruence Equalities among Variables of a Program". In: *TAPSOFT, Vol.1*. Vol. 493. Lecture Notes in Computer Science. Springer, pp. 169–192.

— (1997). "Static Analyses of Congruence Properties on Rational Numbers (Extended Abstract)". In: *SAS*. Vol. 1302. Lecture Notes in Computer Science. Springer, pp. 278–292.

Bibliography II

Masdupuy, François (1992). "Array abstractions using semantic analysis of trapezoid congruences". In: *ICS*. ACM, pp. 226–235.

— (1993). "Semantic Analysis of Interval Congruences". In: *Formal Methods in Programming and Their Applications*. Vol. 735. Lecture Notes in Computer Science. Springer, pp. 142–155.

Miné, Antoine (2002). "A Few Graph-Based Relational Numerical Abstract Domains". In: *SAS*. Vol. 2477. Lecture Notes in Computer Science. Springer, pp. 117–132.

Home work

Read Ch. **31** "Cartesian congruence analysis" of

Principles of Abstract Interpretation
Patrick Cousot
MIT Press

The End, Thank you