Principles of Abstract Interpretation

MIT press

Ch. **38**, Linear equality analysis

Patrick Cousot

pcousot.github.io

PrAbsInt@gmail.com   github.com/PrAbsInt/

These slides are available at
http://github.com/PrAbsInt/slides/slides-38--linear-equality-analysis-PrAbsInt.pdf

Chapter 38

Ch. **38**, Linear equality analysis

Relational versus cartesian properties I

- Cartesian analyzes cannot infer relations between variables so the analysis of `x=0; y=0; while (x<10) {x=x+1; y=y+2;}` cannot infer an upper bound for $\mathtt{y}$.
- The linear equality analysis aims at discovering linear equality relations $\mathbf{A} \times \vec{x} = \vec{b}$ between values $\vec{x}$ of the program variables. In the above example, $2\mathtt{x} - \mathtt{y} = 0$, so $\mathtt{x} \leqslant 10$ implies $\mathtt{y} \leqslant 20$.
- Linear equality analysis was introduced by Michael Karr [Karr, 1976].

Affine properties, Section 38.1

Affine properties I

- A property of program variables $\mathbb{V}$ is a set of environments in $\mathbb{P} = \wp(\mathbb{V} \to \mathbb{F})$ (where $\mathbb{F}$ is the set $\mathbb{Q}$ of rationals (including integers and floats) or $\mathbb{R}$ of reals).
- Let $m = |\mathbb{V}|$ be the cardinality of $\mathbb{V}$ *i.e.* the finite number of program variables.
- If $\mathbb{V} = \{\mathsf{x}_1, \ldots, \mathsf{x}_m\}$, we let $\vec{x}$ be the column vector of values $\rho(\mathsf{x}_1)$, …, $\rho(\mathsf{x}_m)$ of these variables in environment $\rho \in \mathbb{V} \to \mathbb{F}$.
- So, up to the isomorphism $\rho \mapsto \vec{x}$, a program property $P \in \mathbb{P}$ is a set of points $\vec{x}$ in $P \in \mathbb{F}^m$ (and we write $\vec{x} \in P$ for $\rho \in P$ up to this isomorphism).
- The affine space abstract domain consists of those subsets of $\wp(\mathbb{F}^m)$ (*i.e.* $\mathbb{P} = \wp(\mathbb{V} \to \mathbb{F})$) which are affine subspaces of the affine space $\langle\mathbb{F}^m, \langle\vec{\mathbb{F}}^m, \langle\mathbb{F}, +, -, \times, /\rangle, +, -, \times, /\rangle, \vec{+}\rangle$ of finite dimension $m > 0$:

$$\vec{\mathbb{P}} \quad \triangleq \quad \{\vec{\bot}\} \cup \{A + \vec{U} \mid A \in \mathbb{F}^m \wedge \vec{U} \text{ is a vector subspace of } \overrightarrow{\mathbb{F}^m}\}, \qquad \vec{\bot} = \varnothing.$$

Affine properties II

- The infimum $\vec{\bot}$ encodes any system of linear equalities without a solution, and otherwise, we have two representations of the affine subspaces $P \in \vec{\mathbb{P}}$:
  - by a unique $n \times m+1$ matrix $(\mathbf{A}|\vec{b})$ in reduced row echelon form with no zero row encoding $\gamma_{\vec{\equiv}}(\mathbf{A}|\vec{b}) = P$ (where the empty matrix with $n = 0$ encodes the supremum $(\mathbf{0}|\vec{0})$, and otherwise there are no zero rows so $n \leqslant m$ by Theorem 37.9);
  - by a frame or system of generators $\langle \vec{x}_0, \mathbf{B} \rangle$ where $\mathbf{A}\vec{x}_0 = \vec{b}$, $\mathbf{B} = \langle \vec{v}_i, i \in [1, m] \rangle$ is a basis such that $\mathsf{Span}(\mathbf{B}) = \mathsf{Ker}(\mathbf{A})$ so that $\vec{x}_0 \vec{+} \mathsf{Ker}(\mathbf{A}) = \gamma_{\vec{\equiv}}(\mathbf{A}|\vec{b}) = P$.
- The system of generators is computed from $(\mathbf{A}|\vec{b})$ by finding a solution $\vec{x}_0$ using Exercise 37.7 and the basis of the kernel of $\mathbf{A}$ as determined by Lemma 37.12 and Exercise 37.13.
- The matrix $(\mathbf{A}|\vec{b})$ is computed from the system of generators by transformation in reduced row echelon form and elimination of the zero row.

Affine properties III

- The implementation uses one of the two representations at a time and lazily switches to the other one if needed by the next operation to be performed.

Affine abstraction

Affine abstraction I

- Up to the isomorphism between $\wp(\mathbb{V} \to \mathbb{F})$ and $\wp(\mathbb{F}^m)$, a property of the m variables is an element $P \in \wp(\mathbb{F}^m)$ is a variable property, its affine abstraction is

$$\begin{array}{rcl} \alpha_{\mathbb{A}}(\varnothing) & \triangleq & \vec{\bot} \\ \alpha_{\mathbb{A}}(P) & \triangleq & \bigcap\{A + \vec{U} \mid A \in \mathbb{F}^m \wedge \vec{U} \text{ is a vector subspace of } \overrightarrow{\mathbb{F}^m} \wedge P \subseteq A + \vec{U}\} \end{array}$$

i.e. the least affine subspace of $\mathbb{F}^m$ that contains P.

- This is well-defined (Moore family).
- This is a definition by an upper closure condition so that by Exercise 11.87,

$$\langle \wp(\mathbb{F}^m), \subseteq \rangle \xrightleftharpoons[\alpha_{\mathbb{A}}]{\mathbb{1}} \langle \{A + \vec{U} \mid A \in \mathbb{F}^m \wedge \vec{U} \text{ is a vector subspace of } \overrightarrow{\mathbb{F}^m}\}, \subseteq \rangle \qquad (38.2)$$

where $\alpha_{\mathbb{A}}$ is an upper closure operator.

Affine abstract domain

Affine abstract domain I

The affine abstract domain is

$$\vec{\mathbb{D}} \triangleq \langle \vec{\mathbb{P}}, \vec{\sqsubseteq}, \vec{\bot}, \vec{\sqcup}, \overrightarrow{\mathsf{assign}}[\![x, \mathsf{A}]\!], \overrightarrow{\mathsf{test}}[\![\mathsf{B}]\!], \overline{\overrightarrow{\mathsf{test}}}[\![\mathsf{B}]\!]\rangle.$$

- The supremum $\vec{\top} = \mathbb{F}^m$ is represented by the null matrix $(\mathbf{0}|\vec{0})$ with zero row.
- Equality $P \vec{=} P'$ is the equality $(\mathbf{A}|\vec{b}) = (\mathbf{A}'|\vec{b}')$ of the matrices in reduced row echelon form without zero row encoding P and P' since $P = \gamma_{\vec{\equiv}}(\mathbf{A}|\vec{b})$, $P' = \gamma_{\vec{\equiv}}(\mathbf{A}'|\vec{b}')$, and unicity of the reduced row echelon form without zero row.
- The meet $P \vec{\sqcap} P'$ of P and P' represented by $(\mathbf{A}|\vec{b})$ and $(\mathbf{A}'|\vec{b}')$ is the conjunction of the linear equalities that is $\left(\begin{array}{c|c} \mathbf{A} & \vec{b} \\ \mathbf{A}' & \vec{b}' \end{array}\right)$ normalized in reduced row echelon form without zero row.

Affine abstract domain II

- The inclusion $P \mathrel{\vec{\sqsubseteq}} P'$ is $P \mathrel{\vec{\sqcap}} P' \mathrel{\vec{=}} P$ (or the system of generators of P satisfies the linear equalities of P').
- For the join $P \mathrel{\vec{\sqcup}} P'$ of P and P' represented by their systems of generators $\langle \overrightarrow{x_0}, \mathbf{B} \rangle$ and $\langle \overrightarrow{x_0}', \mathbf{B}' \rangle$ has system of generators $\langle \overrightarrow{x_0}, (\mathbf{B}, \mathbf{B}', \vec{x}_0' - \vec{x}_0) \rangle$ (or $\langle \overrightarrow{x_0}, (\mathbf{B}, \mathbf{B}') \rangle$ when $\vec{x}_0' = \vec{x}_0$).

Operations of the affine abstract domain

Affine abstract assignment, Section 38.4 I

- An affine assignment `x = A ;` has the form

$$\mathtt{x}_i \quad = \quad v_1\mathtt{x}_1 + \ldots + v_i\mathtt{x}_i + \ldots + v_m\mathtt{x}_m + v_{m+1}$$

  where $v_1, \ldots, v_i, \ldots, v_m \in \mathbb{F}$ are scalars and $\mathtt{x}_1, \ldots, \mathtt{x}_i, \ldots, \mathtt{x}_m$ are program variables.
- The affine assignment is said to be *invertible* if and only if $v_i \neq 0$.
- Let $(\mathbf{A}'|\vec{b}')$ be the affine abstraction of the reachable values of variables before the assignment (as defined by the forward reachability semantics of Chapter **19**).
- Let x_i' and x_i denote the value of the $\mathtt{x}_i$ before and after the assignment.
- So we have $\mathbf{A}'\vec{x}' = \vec{b}'$ and we must compute $(\mathbf{A}|\vec{b}) \triangleq \overrightarrow{\mathsf{assign}}[\![\mathtt{x}_i, \mathtt{A}]\!]\,(\mathbf{A}'|\vec{b}')$ such that $\mathbf{A}\vec{x} = \vec{b}$ after the affine assignment.
- We have $x_j = x_j'$ for $j \in [0, m] \setminus \{i\}$ since the values of all other variables but $\mathtt{x}_i$ are unchanged.

Invertible affine abstract assignment I

For invertible assignments $v_i \neq 0$, so we can express the old value x_i' of variable $\mathbf{x}_i$ before the assignment in terms of its new value x_i after the assignment. Therefore

$$x_i' \quad = \quad \frac{x_i}{v_i} - \frac{v_1}{v_i}x_1 - \ldots - \frac{v_{i-1}}{v_i}x_{i-1} - \frac{v_{i+1}}{v_i}x_{i+1} \ldots - \frac{v_m}{v_i}x_m - \frac{v_{m+1}}{v_i}.$$

Since $\forall j \in [0, m] \setminus \{i\} \,.\, x_j' = x_j$, the equality constraint before the assignment for each line ℓ of $\mathbf{A}'\vec{x}' = \vec{b}'$ was of the form

$$a_\ell^1 x_1 + \ldots + a_\ell^i x_i' + \ldots + a_\ell^m x_m = a_\ell^{m+1}.$$

Replacing x_i' by its value in terms of the values of the variables after the assignment, we get

$$a_\ell^1 x_1 + \ldots + a_\ell^{i-1} x_{i-1} + a_\ell^i\left(\frac{x_i}{v_i} - \frac{v_1}{v_i}x_1 - \ldots - \frac{v_{i-1}}{v_i}x_{i-1} - \frac{v_{i+1}}{v_i}x_{i+1} \ldots - \frac{v_m}{v_i}x_m - \frac{v_{m+1}}{v_i}\right)$$
$$+ a_\ell^{i+1} x_{i+1} + \ldots + a_\ell^m x_m = a_\ell^{m+1}.$$

Invertible affine abstract assignment II

Grouping the coefficients per variable, we get the line ℓ of $\mathbf{A}\vec{x} = \vec{b}$ after the affine assignment.

$$(a_\ell^1 - \frac{a_\ell^i.v_1}{v_i})x_1 + \ldots + (a_\ell^{i-1} - \frac{a_\ell^i.v_{i-1}}{v_i})x_{i-1} + \frac{a_\ell^i}{v_i}x_i + (a_\ell^{i+1} - \frac{a_\ell^i.v_{i+1}}{v_i})x_{i+1} + \\ \ldots + (a_\ell^m - \frac{a_\ell^i.v_m}{v_i})x_m = a_\ell^{m+1} + \frac{a_\ell^i.v_{m+1}}{v_i}.$$

Non-invertible affine abstract assignment I

- If $v_i = 0$ in the assignment

$$\mathsf{x}_i \quad = \quad v_1\mathsf{x}_1 + \ldots + 0.\mathsf{x}_i + \ldots + v_m\mathsf{x}_m + v_{m+1}$$

  the assignment is non-invertible.
- There is no relationship between the old value x_i' of variable x_i in $(\mathbf{A}'|\vec{b}')$ before the assignment and the new value x_i in $(\mathbf{A}|\vec{b})$ after the assignment.
- It follows that $\mathbf{A}\vec{x} = \vec{b}$ after the affine assignment is given by

$$\exists v \in \mathbb{F} \mathrel{.\,.} \mathbf{A}'\vec{x}[i \leftarrow v] = \vec{b}' \wedge x_i = v_1x_1 + \ldots + v_{i-1}x_{i-1} + v_{i+1}x_{i+1} + \ldots + v_mx_m + v_{m+1}.$$

- So we first eliminate variable x_i by Lemma 37.19 and then add the constraint $x_i = v_1x_1 + \ldots + v_{i-1}x_{i-1} + v_{i+1}x_{i+1} + \ldots + v_mx_m + v_{m+1}$.

Abstraction of an expression into an affine expression, Section **38.4.3** I

- The program assignments $\mathtt{x = A\ ;}$ in Section **4.1** are not necessarily in affine form.
- So we use the following affine abstraction of the arithmetic expression $\mathtt{A}$ in Section **3.4**.
- The idea is that for non-linear expressions like $\mathtt{x} \star \mathtt{y}$ the static analysis may have determined that the value of $\mathtt{x}$ is a scalar c so we can use the linear form $c.\mathtt{y}$.

$$
\begin{array}{rcllr}
\alpha(\mathtt{1}) & \triangleq & 1 & & \\
\alpha(\mathtt{x}) & \triangleq & 1.\mathtt{x} & & \\
\alpha(\mathtt{A}_1 - \mathtt{A}_2) & \triangleq & \alpha(\mathtt{A}_1) - \alpha(\mathtt{A}_2) & & \\
\alpha(\mathtt{c} \star \mathtt{A}) & \triangleq & \alpha(\mathtt{A} \star \mathtt{c}) \ \triangleq \ c.\alpha(\mathtt{A}) & & c \text{ is the value of constant } \mathtt{c} \\
\alpha(\mathtt{x} \star \mathtt{A}) & \triangleq & \alpha(\mathtt{A} \star \mathtt{x}) \ \triangleq \ c.\alpha(\mathtt{A}) & & \text{the analysis has determined that } \rho(\mathtt{x}) = c \\
\alpha(\mathtt{A}) & \triangleq & \top & & \text{otherwise}
\end{array}
$$

Abstraction of an expression into an affine expression, Section **38.4.3** II

- If the abstraction returns $\top$ then the assignment is handled by Lemma 37.19 for eliminating the assigned variable.
- Otherwise the coefficients of the variables are summed up to get a linear assignment of Section **38.4**.

Affine abstract test

- But for linear equality tests, $\overrightarrow{\mathsf{test}}[\![\mathsf{B}]\!](P) = \overrightarrow{\overline{\mathsf{test}}}[\![\mathsf{B}]\!](P) = P$.
- If $\mathtt{B}$ can be put in linear form $\vec{a}\vec{x} = b$ *i.e.*

$$a_1 x_1 + \ldots + a_{i-1} x_{i-1} + a_{i+1} x_{i+1} + \ldots + a_m x_m = b$$

  by transformation of $\mathtt{B}$ as in Section **38.4.3**, then the expression is conjuncted with P.
- $\overrightarrow{\mathsf{test}}[\![\mathsf{B}]\!](\mathbf{A} \mid \vec{b}) = \left(\begin{array}{c|c} \mathbf{A} & \vec{b} \\ \vec{a} & b \end{array}\right)$ and put in reduced row echelon form.

Fixpoint computation

Fixpoint computation I

- The abstract domain has no infinite ascending chain so no widening/narrowing is needed.

Conclusion

Conclusion I

- Michael Karr [Karr, 1976] represents affine spaces as the solution set of linear equation systems $\mathbf{A}\vec{x} = \vec{b}$ represented by the matrix $(\mathbf{A} \mid \vec{b})$ (the number of affine relations between values of variables will be small hence the dimension of the affine space and the size of the system of generators will be large).
- Following what is traditionally done for polyhedral analysis [Cousot and Halbwachs, 1978], Markus Müller-Olm and Helmut Seidl [Müller-Olm and Seidl, 2004a] introduced the use of systems of generators [Müller-Olm and Seidl, 2004a,b].
- [Elder, Lim, Sharma, Andersen, and Reps, 2014] studies variations on matrix representations of affine domains.
- Integers can be analyzed using rationals with a concretization in the integers.
- A better solution is to generalize affine equalities to affine congruences [Granger, 1991].

Bibliography I

Cousot, Patrick and Nicolas Halbwachs (1978). "Automatic Discovery of Linear Restraints Among Variables of a Program". In: *POPL*. ACM Press, pp. 84–96.

Elder, Matt, Junghee Lim, Tushar Sharma, Tycho Andersen, and Thomas W. Reps (2014). "Abstract Domains of Affine Relations". *ACM Trans. Program. Lang. Syst.* 36.4, 11:1–11:73.

Granger, Philippe (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.

Karr, Michael (1976). "Affine Relationships Among Variables of a Program". *Acta Inf.* 6, pp. 133–151.

Müller-Olm, Markus and Helmut Seidl (2004a). "A Note on Karr's Algorithm". In: *ICALP*. Vol. 3142. Lecture Notes in Computer Science. Springer, pp. 1016–1028.

Bibliography II

Müller-Olm, Markus and Helmut Seidl (2004b). "Precise interprocedural analysis through linear algebra". In: *POPL*. ACM, pp. 330–341.

Home work

Read Ch. **38** “Linear equality analysis” of

Principles of Abstract Interpretation
Patrick Cousot
MIT Press

The End, Thank you