Typing Linear Constraints for Moding CLP($\mathcal{R}$) Programs

Salvatore RUGGIERI and Fred MESNARD

Dipartimento di Informatica, Università di Pisa, Italy
IREMIA, université de la Réunion, France

- Modes for logic programs assign to every predicate argument an input/output behavior.
  - Input: the predicate argument is ground on calls.
  - Output: the predicate argument is ground on answers.
  - Example: `:- mode append(in, in, out).`
- Modes can be seen as *lightweight* specifications.
- Groundness is restrictive in the CLP($\mathcal{R}$) context. Based on types, we want to extend the notion of moding to upper and/or lower bounds as well.

Definition (types)

A *type* is an element of $\mathcal{BT} = \{\star, \sqcup, \sqcap, \square, !\}$.

- $!$ is intended to type variables that show at most one single value in every solution, a property known as *definiteness*;
- $\square$ is intended to type variables that assume a range of values (hence, lower and upper bounds exist);
- $\sqcup$ (resp., $\sqcap$) is intended for variables that have a lower bound (resp., an upper bound);
- $\star$ is to be used when no upper or lower bound can be stated.

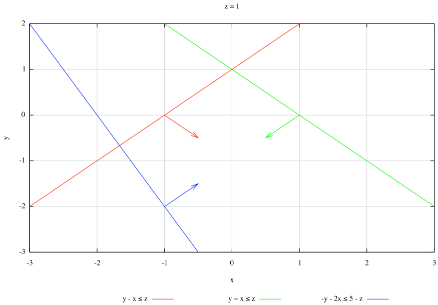
Definition (types assertions)

- An *atomic type declaration* (atd) is an expression $x : \tau$, where x is a variable and $\tau \in \mathcal{BT}$.
- We define $vars(x : \tau) = \{x\}$, and say that x is typed as τ.
- A *type declaration* is a sequence of atd's $d_1, \ldots, d_n$, with $n \geq 0$. We define $vars(d_1, \ldots, d_n) = \cup_{i=1..n} vars(d_i)$.
- A *type assertion* is an expression $\mathbf{d}_1 \vdash c \rightarrow \mathbf{d}_2$, where $\mathbf{d}_1, \mathbf{d}_2$ are type declarations and c is a linear constraint.

Example

$z :! \vdash y - x \leq z, y + x \leq z, -y - 2x \leq 5 - z \rightarrow y : \sqcap, x : \sqcup$
states that if z has a fixed value then either the set of solutions of the involved constraint is empty or the set of solutions is such that y has an upper bound and x has a lower bound.

The set of solutions for $z = 1$:

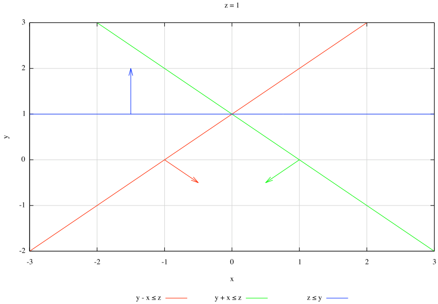


Example

$z :! \vdash y - x \leq z, y + x \leq z, z \leq y \rightarrow y :!, x :!$

states that if z has a fixed value then either the set of solutions of the involved constraint is empty or both x and y assume a unique value in it.

The set of solutions for $z = 1$:



Definition (semantics)

We associate to an atd $d = x : \tau$ a formula $\phi(d)$ over fresh variables $\upsilon(d)$, called parameters, as follows:

$$\phi(x :!) = x = a \qquad \upsilon(x :!) = \{a\}$$
$$\phi(x : \Box) = a \leq x \wedge x \leq b \qquad \upsilon(x : \Box) = \{a, b\}$$
$$\phi(x : \sqcup) = a \leq x \qquad \upsilon(x : \sqcup) = \{a\}$$
$$\phi(x : \sqcap) = x \leq b \qquad \upsilon(x : \sqcap) = \{b\}$$
$$\phi(x : \star) = \mathtt{true} \qquad \upsilon(x : \star) = \emptyset.$$

ϕ and υ extend to type declarations as follows:

$$\phi(d_1, \ldots, d_n) = \wedge_{i=1..n}\phi(d_i) \qquad \upsilon(d_1, \ldots, d_n) = \cup_{i=1..n}\upsilon(d_i).$$

A type assertion $\mathbf{d}_1 \vdash c \rightarrow \mathbf{d}_2$ is valid if for $\mathbf{v} = vars(c) \cup vars(\mathbf{d}_1) \cup vars(\mathbf{d}_2)$, the following formula is true in $\mathcal{R}$:

$$\forall\upsilon(\mathbf{d}_1)\exists\upsilon(\mathbf{d}_2)\forall\mathbf{v}.(\phi(\mathbf{d}_1) \wedge c) \rightarrow \phi(\mathbf{d}_2).$$

Salvatore RUGGIERI and Fred MESNARD

Example

For the type assertion

$$z :! \vdash y - x \leq z, y + x \leq z, z \leq y \rightarrow y :!, x :!$$

the formula to be proved is:

$$\forall a \; \exists b, c \; \forall x, y, z. \; (z = a \wedge y - x \leq z \wedge y + x \leq z \wedge z \leq y) \\ \rightarrow (y = b \wedge x = c).$$

- Such formulas can be checked by real quantifier elimination methods.
- It allows for generalizing to the *non-linear* case!
  For instance: Mathematica, QEPCAD, Redlog.
- But we observe that our formulas represent a quite restricted class.
- Our approach switches from

  the *logical* view of constraints-as-formulas
  to
  a *geometric* view of constraints-as-polyhedra.

Consider a linear constraint c and a type declaration $\mathbf{d}$.

- c can be equivalently represented as a linear system of inequalities $\mathbf{A}_c\mathbf{v} \leq \mathbf{b}_c$ where $\mathbf{v} = \mathit{vars}(c) \cup \mathit{vars}(\mathbf{d})$.
- The linear constraint $\phi(\mathbf{d})$ can be represented as $\mathbf{A_d}\mathbf{v} \leq \mathbf{B_d}\mathbf{a_d}$, where $\mathbf{a_d}$ is the symbolic vector of parameters in $\upsilon(\mathbf{d})$.

The resulting system $\phi(\mathbf{d}) \wedge c$ is a parameterized system of linear inequalities $\mathcal{P}$, where variables in $\upsilon(\mathbf{d})$ play the role of parameters:

$$\begin{pmatrix} \mathbf{A}_c \\ \mathbf{A_d} \end{pmatrix} \mathbf{v} \leq \begin{pmatrix} \mathbf{b}_c \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{B_d} \end{pmatrix} \mathbf{a_d}$$

Definition (Parameterized polyhedron)

A *parameterized polyhedron* is a collection of polyhedra defined by fixing the value for parameters in a parameterized system of linear inequalities: $Sol(\mathbf{Ax} \leq \mathbf{b} + \mathbf{Ba}, \mathbf{u}) = \{\mathbf{x} \mid \mathbf{Ax} \leq \mathbf{b} + \mathbf{Bu}\}$.

Example

Let $\mathbf{d}$ be z :! and c be $y - x \leq z, y + x \leq z, -y - 2x \leq 5 - z$. We have that $\phi(\mathbf{d})$ is $z = a$, and $\phi(\mathbf{d}) \wedge c$ is:

$$\begin{pmatrix} -1 & 1 & -1 \\ 1 & 1 & -1 \\ -2 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 5 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{pmatrix} a$$

Let $x = \mathbf{v}_i$.
$\mathbf{d} \vdash c \rightarrow x : \tau$ is valid iff
for every $\mathbf{u}$, $\mathcal{S}_{\mathbf{u}} = Sol(\mathcal{P}, \mathbf{u})$ either is empty or:

- if $\tau = !$ then $max\{\mathbf{v}_i \mid \mathbf{v} \in \mathcal{S}_{\mathbf{u}}\} = min\{\mathbf{v}_i \mid \mathbf{v} \in \mathcal{S}_{\mathbf{u}}\} \in \mathcal{R}$, namely x assumes a single value;
- if $\tau = \Box$ then $max\{\mathbf{v}_i \mid \mathbf{v} \in \mathcal{S}_{\mathbf{u}}\} \in \mathcal{R}$ and $min\{\mathbf{v}_i \mid \mathbf{v} \in \mathcal{S}_{\mathbf{u}}\} \in \mathcal{R}$ namely both an upper and a lower bound exist for x;
- if $\tau = \sqcup$ then $min\{\mathbf{v}_i \mid \mathbf{v} \in \mathcal{S}_{\mathbf{u}}\} \in \mathcal{R}$, namely a lower bound exists for x;
- if $\tau = \sqcap$ then $max\{\mathbf{v}_i \mid \mathbf{v} \in \mathcal{S}_{\mathbf{u}}\} \in \mathcal{R}$, namely an upper bound exists for x;
- if $\tau = \star$ then we have nothing to show!

From now on, we only consider satisfiable linear constraint (else one can infer assertions of the form $\mathbf{d} \vdash c \rightarrow x :!$, for every x).

Lemma

Consider a parameterized polyhedron $\mathcal{P}$. Let $\mathcal{H}$ be its homogeneous version: $\mathbf{A_c v} \leq \mathbf{0}$, $\mathbf{A_d v} \leq \mathbf{0}$.
$max\{\mathbf{c}^T\mathbf{v} \mid \mathbf{v} \in Sol(\mathcal{H})\} = 0$ iff for every parameter instance $\mathbf{u}$, $Sol(\mathcal{P}, \mathbf{u}) = \emptyset$ or $max\{\mathbf{c}^T\mathbf{v} \mid \mathbf{v} \in Sol(\mathcal{P}, \mathbf{u})\} \in \mathcal{R}$.

- When $\mathbf{c}$ is always 0 except for the i^{th} position where it is 1, we have $\mathbf{c}^T\mathbf{v} = \mathbf{v}_i$.
- The previous lemma solves the problem of deciding whether $\mathbf{d} \vdash c \rightarrow \mathbf{v}_i : \sqcap$, without having to take into account parameters.
- By reasoning similarly for types $\sqcup$ and $\square$, we can state an effective procedure, called LPCHECK

Input: a type assertion $\mathbf{d}_1$, a constraint c and a seq. of vars $\mathbf{x}$.

Step 0 Define $\mathbf{v} = vars(c)$, $\mathbf{n} = nf(\mathbf{d}_1)$, $\mathbf{d} = \mathbf{n}|_{\mathbf{v}}$.

Step 1 Let $\mathbf{A}_c\mathbf{v} \leq \mathbf{b}_c$ be the geometric rep. of c.
Let $\mathbf{A}_\mathbf{d}\mathbf{v} \leq \mathbf{B}_\mathbf{d}\mathbf{a}_\mathbf{d}$ the geometric rep. of $\phi(\mathbf{d})$.

Step 2 If $Sol(\mathbf{A}_c\mathbf{v} \leq \mathbf{b}_c) = \emptyset$ Then for every x in $\mathbf{x}$, output "x :!"

Else

Step 3 for every x in $\mathbf{x} \setminus \mathbf{v}$ and $x : \tau$ in $\mathbf{n}$, output "$x : \tau$";

Step 4 for every x in $\mathbf{x} \cap \mathbf{v}$:

(a) Let $M = max\{x \mid \mathbf{A}_c\mathbf{v} \leq \mathbf{0}, \mathbf{A}_\mathbf{d}\mathbf{v} \leq \mathbf{0}\}$, $m = max\{-x \mid \mathbf{A}_c\mathbf{v} \leq \mathbf{0}, \mathbf{A}_\mathbf{d}\mathbf{v} \leq \mathbf{0}\}$.
(b) Output "$x : \square$" if $M = 0$ and $m = 0$;
(c) Output "$x : \sqcup$" if $M = \infty$ and $m = 0$;
(d) Output "$x : \sqcap$" if $M = 0$ and $m = \infty$;
(e) Else output "$x : \star$".

Figure: LPCheck, sound and complete w.r.t. $\{\star, \sqcup, \sqcap, \square\}$.

Minkowski, Motzkin, 1953:

Theorem (Minkowski's decomposition thm)

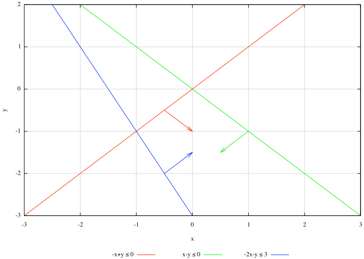
There exists an effective procedure that given $\mathbf{Ax} \leq \mathbf{b}$ *decides whether or not the polyhedron* $Sol(\mathbf{Ax} \leq \mathbf{b})$ *is empty and, if not, it yields a generating matrix* $\mathbf{R}$ *and a vertex matrix* $\mathbf{V}$ *such that:*

$$Sol(\mathbf{Ax} \leq \mathbf{b}) = \{\mathbf{x} \mid \mathbf{x} = \mathbf{R}\boldsymbol{\lambda}, \boldsymbol{\lambda} \geq 0\ \} + \{\mathbf{x} \mid \mathbf{x} = \mathbf{V}\gamma, \gamma \geq 0, \Sigma\gamma = 1\ \}$$

$$Sol(\mathbf{Ax} \leq \mathbf{0}) = \{\mathbf{x} \mid \mathbf{x} = \mathbf{R}\boldsymbol{\lambda}, \boldsymbol{\lambda} \geq 0\ \}$$

Example

$$\begin{pmatrix} -1 & 1 \\ 1 & 1 \\ -2 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix}$$



$$\mathbf{R} = \begin{pmatrix} 1 & 1 \\ -2 & -1 \end{pmatrix} \qquad \mathbf{V} = \begin{pmatrix} 0 & -1 \\ 0 & -1 \end{pmatrix}$$

Loechner and Wilde, 1997:

Theorem (Minkowski's thm for parameterized polyhedra)

Every parameterized polyhedron can be expressed by a generating matrix $\mathbf{R}$ *and finitely many pairs*

$$(\mathbf{v^a}(1), \mathbf{C}_1\mathbf{a} \leq \mathbf{c}_1), \ldots, (\mathbf{v^a}(k), \mathbf{C}_k\mathbf{a} \leq \mathbf{c}_k)$$

where, for $i = 1..k$, $\mathbf{v^a}(i)$ *is a vector parametric in* $\mathbf{a}$ *and* $Sol(\mathbf{C}_i\mathbf{a} \leq \mathbf{c}_i) \neq \emptyset$, *as follows:*

$$Sol(\mathbf{Ax} \leq \mathbf{b} + \mathbf{Ba}, \mathbf{u}) = \{\mathbf{x} \mid \mathbf{x} = \mathbf{R}\boldsymbol{\lambda}, \boldsymbol{\lambda} \geq 0 \}$$
$$+ ConvexHull(\{\mathbf{v^u}(i) \mid i = 1..k, \mathbf{C}_i\mathbf{u} \leq \mathbf{c}_i \})$$

and

$$Sol(\mathbf{Ax} \leq \mathbf{0}) = \{\mathbf{x} \mid \mathbf{x} = \mathbf{R}\boldsymbol{\lambda}, \boldsymbol{\lambda} \geq 0 \}$$

.

Example

$a + b \geq y, y \geq a, y \geq b, x = a$

$$\begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$

$\mathbf{R} = \mathbf{0}$

$$(\begin{pmatrix} a \\ b \end{pmatrix}, b \geq a \geq 0)(\begin{pmatrix} a \\ a \end{pmatrix}, a \geq b \geq 0)(\begin{pmatrix} a \\ a+b \end{pmatrix}, a, b \geq 0)$$

Lemma

Consider the Minkowski's form of a non-empty parameterized polyhedron.
Every $\mathcal{S}_{\mathbf{u}} = \{\mathbf{c}^T\mathbf{x} \mid \mathbf{x} \in Sol(\mathbf{Ax} \leq \mathbf{b} + \mathbf{Ba}, \mathbf{u})\}$ *is empty or a singleton iff*

- $\mathbf{c}^T\mathbf{R} = \mathbf{0}$
- *for* $1 \leq m < n \leq k$, $\mathbf{C}_m\mathbf{a} \leq \mathbf{c}_m, \mathbf{C}_n\mathbf{a} \leq \mathbf{c}_n \models \mathbf{c}^T\mathbf{v}^{\mathbf{a}}(m) = \mathbf{c}^T\mathbf{v}^{\mathbf{a}}(n)$.

Example

Consider $a + b \geq y, y \geq a, y \geq b, x = a$, and the corresponding parameterized polyhedron defined by $\mathbf{R} = \mathbf{0}$ and

$$(\left(\begin{array}{c} \mathrm{a} \\ \mathrm{b} \end{array}\right), b \geq a \geq 0)(\left(\begin{array}{c} \mathrm{a} \\ \mathrm{a} \end{array}\right), a \geq b \geq 0)(\left(\begin{array}{c} \mathrm{a} \\ \mathrm{a+b} \end{array}\right), a, b \geq 0).$$

Let us reason about definiteness of variables x and y by using the previous lemma

- x is definite, since $a = a$ over any polyhedron.
- y is not definite:
  - fine for the first and second vertex: $b \geq a \geq 0, a \geq b \geq 0 \models a = b$
  - but for the first and third vertex: $b \geq a \geq 0, a, b \geq 0 \not\models b = a + b$

Input: a type assertion $\mathbf{d}_1$, a constraint c and a seq. of vars $\mathbf{x}$.

Step 0 Define $\mathbf{v} = vars(c)$, $\mathbf{n} = nf(\mathbf{d}_1)$, $\mathbf{d} = \mathbf{n}|_{!}$.

Step 1 Let $\mathbf{A}_c\mathbf{v} \leq \mathbf{b}$ be the geometric rep. of c, and $\mathbf{A_d}\mathbf{v} \leq \mathbf{B_d}\mathbf{a_d}$ the geometric rep. of $\phi(\mathbf{d})$.

Step 1 For the parametric polyh. $\mathbf{A}_c\mathbf{v} \leq \mathbf{b}, \mathbf{A_d}\mathbf{v} \leq \mathbf{B_d}\mathbf{a_d}$ build the generating matrix $\mathbf{R}$ and the sequence $(\mathbf{v}^{\mathbf{a}}(1), \mathbf{C}_1\mathbf{a} \leq \mathbf{c}_1), \ldots, (\mathbf{v}^{\mathbf{a}}(k), \mathbf{C}_k\mathbf{a} \leq \mathbf{c}_k)$

Step 2 For every $x : \tau$ as output from $\mathrm{LPC{\scriptsize HECK}}$

Step 3 If $\tau \neq \Box$ or $x \notin \mathbf{v}$ then output "$x : \tau$";

Step 4 Else let i such that $x = \mathbf{v}_i$:

(a) Output "x :!" if $row(\mathbf{R}, i) = \mathbf{0}$ and for $1 \leq m < n \leq k$, $\mathbf{C}_m\mathbf{a} \leq \mathbf{c}_m$, $\mathbf{C}_n\mathbf{a} \leq \mathbf{c}_n \models \mathbf{v}^{\mathbf{a}}(m)_i = \mathbf{v}^{\mathbf{a}}(n)_i$;

(b) Output "$x : \Box$" otherwise.

Figure: $\mathrm{POLYC{\scriptsize HECK}}$, sound and complete w.r.t. $\{\star, \sqcup, \sqcap, \Box, !\}$.

Definition (well-moding)

Let P be a CLP($\mathcal{R}$) program. A clause of P:
$p_0(\mathbf{x}_0 : \boldsymbol{\mu}_0 \times \boldsymbol{\tau}_{n+1}) \leftarrow c, p_1(\mathbf{x}_1 : \boldsymbol{\tau}_1 \times \boldsymbol{\mu}_1), \ldots, p_n(\mathbf{x}_n : \boldsymbol{\tau}_n \times \boldsymbol{\mu}_n)$
is well-moded if for $i = 1..n+1$, the type assertion

$$\mathbf{x}_0 : \boldsymbol{\mu}_0, \ldots, \mathbf{x}_{i-1} : \boldsymbol{\mu}_{i-1} \vdash c \longrightarrow \mathbf{x}_i : \boldsymbol{\tau}_i$$

is valid. P is well-moded if all its clauses are well-moded.

A well-moded program well behaves at run-time.

program	C	A	modes	time
ack	3	6	ack($\star \times \square$, $! \times !$, $\star \times !$)	0.0011
ack	3	6	ack($\star \times \square$, $\square \times \square$, $\star \times \square$)	0.0007
fib	2	4	fib($! \times !, \star \times !$)	0.0011
fib	2	4	fib($\star \times \star, \star \times !$)	0.0007
mc91	2	4	mc($\sqcap \times \square, \star \times \square$)	0.0005
mc91	2	4	mc($! \times !, \star \times !$)	0.0006
mortgage	2	3	mortgage($! \times !, \sqcap \times !, ! \times !, \star \times !$)	0.0010
schedule	10	21	schedule($! \times !, \star \times \sqcap, \star \times \sqcup$)	0.0021
tak	3	8	tak($\star \times !$)	0.0015

Table: Time in seconds, Xeon 2.8Ghz, C: # clauses, A: # atoms.

- sources and extended technical report at:
  `http://www.di.unipi.it/~ruggieri/software/`
- extensions:
  - $\Box_r, r \in \mathcal{R}^+$
  - type inference and terminating modes
  - CLP($Term + \mathcal{R}$)