Variable Ranges in Linear Constraints

Salvatore Ruggieri
Dipartimento di Informatica,
Università di Pisa, Italy
ruggieri@di.unipi.it

Fred Mesnard
Lim-Iremia,
Université de la Réunion, France
frederic.mesnard@univ-reunion.fr

ABSTRACT

We introduce an extension of linear constraints, called linear-range constraints, which allows for (meta-)reasoning about the approximation width of variables. Semantics for linear-range constraints is provided in terms of parameterized linear systems. We devise procedures for checking satisfiability and for entailing the maximal width of a variable. An extension of the constraint logic programming language CLP($\mathbb{R}$) is proposed by admitting linear-range constraints.

Categories and Subject Descriptors

D.3.2 [**Language Classifications**]: Constraint and logic languages; D.3.3 [**Language Constructs and Features**]: Constraints

Keywords

Linear constraints, Constraint Logic Programming, Interval Arithmetic, Parameterized Polyhedra.

1. INTRODUCTION

Interval arithmetic over the reals has been the subject of extensive studies in the literature since the seminal work of [9], with practical implications in computing with approximations. Programming with intervals has been proposed in the constraint logic programming framework in languages such as CLP(BNR), Newton and CLIP [1, 3, 4]. In this paper, we are interested in linear constraints. As an example, consider the following constraint c:

$$\texttt{100 - R/10 - H/5} \leq \texttt{C, C} \leq \texttt{100 - R/10,}$$
$$\texttt{0} \leq \texttt{R, R} \leq \texttt{500, 0} \leq \texttt{H, H} \leq \texttt{100,}$$

where R and H are distances in meters, and C is a Celsius temperature. The constraint 100 - R/10 - H/5 $\leq$ C, C $\leq$ 100 - R/10 is intended to provide an estimate of the temperature at a distance R and at an height H from the center of a heating source. The constraint 0 $\leq$ R, R $\leq$ 500, 0 $\leq$ H, H $\leq$ 100 provides the intervals of R and H for which the estimate holds. The linear constraint c can be used to derive the temperature at a given distance and height from the center of the heating source. For instance, the solved form of c, R = 160, H = 20 includes 80 $\leq$ C, C $\leq$ 84, or, with a different notation, C = 82 $\pm$ 2, or, in the interval syntax, C $\in$ [80, 84]. Often, also the input variables are known with some approximation. For instance, due to the measurement error of a GPS device, distance and height could be known with an approximation of ± 20 meters and ± 10 meters respectively. In the above example, we could model such a case by solving the constraint c, 140 $\leq$ R, R $\leq$ 180, 10 $\leq$ H, H $\leq$ 30 to derive 72 $\leq$ C $\leq$ 84, namely that the temperature is 78 ± 6. In general, can we conclude that the approximation of the computed C is up to ± 6 when R is known with an approximation of ± 20 and H is known with an approximation of ± 10? The answer is negative. In fact, by solving c, 140 $\leq$ R, R $\leq$ 180, 80 $\leq$ H, H $\leq$ 100 we derive 72 $\leq$ C $\leq$ 86, namely that the temperature is 79 ± 7.

In this paper, we add range (meta-)variables to linear constraints as a means to reason on the approximation of variables. For a variable x, the range variable δ_x denotes the maximum approximation of x. For instance, the inequality $\delta_x \leq 1$ is interpreted as "x is known with an approximation of up to ± 1". *Differently from interval arithmetic, we are not asserting a specific interval for x but a constraint on the interval width for x.* Linear-range constraints mix linear inequalities with upper and lower bounds on range variables. Reconsidering the above example, the linear-range constraint $c, \delta_{\mathtt{R}} \leq 20, \delta_{\mathtt{H}} \leq 10$ models the case that R is known with an approximation of up to ± 20 and H is known with an approximation of up to ± 10. The question we posed can then be restated as follows: does $c, \delta_{\mathtt{R}} \leq 20, \delta_{\mathtt{H}} \leq 10$ entail $\delta_{\mathtt{C}} \leq 6$? The semantics of linear-range constraints is formulated by compiling them into parameterized linear systems and the problem of entailing $\delta_x \leq s$, for $s \in \mathbb{R}$ given a linear-range constraint is solved by reasoning over parameterized linear systems. On these basis, we propose a conservative extension of CLP($\mathbb{R}$), constraint logic programming over the reals, to include linear-range constraints.

1.1 Notation and Background

We adhere to standard notation of linear algebra [10]. $\mathbb{R}$ is the set of real numbers. Small capital letters ($\mathbf{a}$, $\mathbf{b}$, ...) denote column vectors, while capital letters ($\mathbf{A}$, $\mathbf{B}$, ...) denote matrices. $\mathbf{0}$ is the column vector with all elements equal to 0. $\mathbf{a}_i$ denotes the i^{th} element in $\mathbf{a}$, and $row(\mathbf{A}, i)$ the row vector consisting of the i^{th} row of $\mathbf{A}$. $\mathbf{c}^T\mathbf{x}$ denotes the inner product of column vectors $\mathbf{c}$ and $\mathbf{x}$. $\mathbf{Ax} \leq \mathbf{b}$ denotes a system of linear inequalities over the variables in $\mathbf{x}$. We assume that the dimensions of vectors and matrices in inner products and linear systems are of the appropriate size. An equivalent formulation of linear systems is provided in terms of logic formulas. A primitive linear constraint is an expression $a_1 \cdot x_1 + \dots a_n \cdot x_n \leq a_0$, where $a_0, \dots, a_n$ are constants in $\mathbb{R}$ and $x_1, \dots, x_n$ are variables. We will also use the inner product form by rewriting it as $\mathbf{c}^T\mathbf{x} \leq \alpha$. A linear constraint c is a sequence of primitive constraints, whose interpretation is their conjunction. Inequalities $\mathbf{c}^T\mathbf{x} \geq \alpha$ and equalities $\mathbf{c}^T\mathbf{x} = \alpha$ can be reduced to linear constraints. We write $vars(c)$ to denote the set of variables occurring in the linear constraint c. A *polyhedron* is the set of solution points that satisfy a linear system: $Sol(\mathbf{Ax} \leq \mathbf{b}) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{Ax} \leq \mathbf{b}\}$. Next, we define the width of a variable.

Definition 1.1 (width). *Let $S = Sol(\mathbf{Ax} \leq \mathbf{b})$ be a non-empty polyhedron, and x a variable in $\mathbf{x}$. We define: $width(S, x) = max\{x \mid \mathbf{x} \in S\} - min\{x \mid \mathbf{x} \in S\}$ if both min and max exist. Otherwise, $width(S, x) = \infty$. For an empty polyhedron $S = \emptyset$, we define: $width(S, x) = 0$.*

The radius of a variable is the half of its width. Notice that $\infty \geq s$ and $\infty \neq s$ hold for every $s \in \mathbb{R}$. Also, we set $max\ S = \infty$ if $\infty \in S$, and $max\ S = 0$ if $S = \emptyset$.

2. LINEAR-RANGE CONSTRAINTS

Let us first introduce some syntactic means for denoting variable ranges.

Definition 2.1. *Let Var be the set of linear constraint variables. We define $\Delta = \{\delta_x \mid x \in Var\}$ as the set of range variables.*

Intuitively, for a variable x appearing in a linear constraint c, the interpretation of δ_x is that "the range of x is $\pm\delta_x$", or, more formally, that the radius of x in the solutions of c is δ_x. Range constraints are now introduced.

Definition 2.2. *A primitive range constraint is an inequality $\delta_x \simeq s$, where $\simeq$ is in $\{\leq, \geq\}$ and $s \geq 0 \in \mathbb{R}$.*

A range constraint d is a sequence of primitive range constraints. A linear-range constraint $c \wedge d$ is a sequence of linear constraints and range constraints.

We write $\delta_x = s$ as a shorthand for $\delta_x \leq s, \delta_x \geq s$.

What is the semantics of a linear-range constraint? Intuitively, $0 \leq x \leq 2$ denotes a set of values for x ranging from 0 to 2, and, under this interpretation, one would conclude $\delta_x = 1$. The linear-range constraint $1 \leq x \leq 4, \delta_x \leq 1$ denotes a collection of intervals for x of the form $[max(1, a - \delta), min(4, a + \delta)]$ with $a \in \mathbb{R}$ and $0 \leq \delta \leq 1$. Finally, given $x = 2y, \delta_y \leq 1, \delta_x \geq 3$, we should conclude its unsatisfiability since the approximation of x is at most 2, namely the double of the approximation of y, which is in contradiction with $\delta_x \geq 3$.

Definition 2.3 (semantics). *A model of a linear-range constraint $c \wedge d$ is a non-empty polyhedron:*

$$S = Sol(c \wedge \bigwedge_{\delta_x \in vars(d)} \underline{x} \leq x \leq \overline{x})$$

where $\underline{x}, \overline{x} \in \mathbb{R}$ such that for every $\delta_x \simeq s$ in d with $\simeq$ in $\{\leq, \geq\}$, we have: $width(S, x) \simeq 2s$.

$c \wedge d$ is satisfiable if there exists a model of it.

Finally, $c \wedge d$ entails $\delta_x \leq s$ if every model of $c \wedge d$ is a model of $\delta_x \leq s$.

When d is empty, satisfiability conservatively boils down to the condition $Sol(c) \neq \emptyset$. Towards providing a procedure for checking satisfiability in the general case, we first compile linear-range constraints into parameterized linear systems. A parameterized linear system of inequalities (or, parameterized linear constraint) is a linear system $\mathbf{Ax} \leq \mathbf{b} + \mathbf{Ba}$ where $\mathbf{a}$ is a vector of *parameters*. Parameterized linear systems trace back to late 60's in the context of (multi)parametric linear programming [2].

Definition 2.4. *For a linear-range constraint $c \wedge d$, we define the parameterized linear system:*

$$\mathcal{S}(c \wedge d) = c \wedge d \wedge \bigwedge_{\delta_x \in vars(d)} (a_x - \delta_x \leq x \leq a_x + \delta_x, 0 \leq \delta_x),$$

where the a_x, δ_x's are parameters.

Example 2.5. *Consider the three linear-range constraints previously introduced.*

$\mathcal{S}(0 \leq x \leq 2)$ is $0 \leq x \leq 2$, that is a zero-parameter linear system or, equivalently, a linear system.

$\mathcal{S}(1 \leq x \leq 4, \delta_x \leq 1)$ is $1 \leq x \leq 4, a_x - \delta_x \leq x \leq a_x + \delta_x, 0 \leq \delta_x \leq 1$. Intuitively, here a_x models an approximatively known value for x, and δ_x models its radius.

Finally, $\mathcal{S}(x = 2y, \delta_y \leq 1, \delta_x \geq 3)$ is $x = 2y, a_y - \delta_y \leq y \leq a_y + \delta_y, 0 \leq \delta_y \leq 1, a_x - \delta_x \leq x \leq a_x + \delta_x, \delta_x \geq 3$.

The notion of parameterized polyhedra [8] models the solutions of a parameterized linear system.

Definition 2.6. *A* parameterized polyhedron *is the collection of polyhedra defined by fixing the values of the parameters in a parameterized linear system: $Sol(\mathbf{Ax} \leq \mathbf{b} + \mathbf{Ba}, \mathbf{u}) = \{\mathbf{x} \mid \mathbf{Ax} \leq \mathbf{b} + \mathbf{Bu}\}$, where $\mathbf{u} \in \mathbb{R}^{|\mathbf{a}|}$ is an instance of the parameters $\mathbf{a}$.*

$Sol()$ is now a binary function. In addition to a system of parameterized linear inequalities, an assignment to parameters is required.

Example 2.7. *Consider Ex. 2.5. The parameterized polyhedron of $\mathcal{S}(0 \leq x \leq 2)$ boils down to the polyhedron $Sol(0 \leq x \leq 2)$, as one would expect.*

For $\mathcal{S}(1 \leq x \leq 4, \delta_x \leq 1)$, we enumerate below the polyhedra obtained by fixing $\delta_x = 1$, namely by assuming the largest approximation possible for x:

$$\begin{array}{rl} Sol(1 \leq x \leq a_x + 1) & \text{when } a_x \leq 2 \\ Sol(a_x - 1 \leq x \leq a_x + 1) & \text{when } 2 \leq a_x \leq 3 \\ Sol(a_x - 1 \leq x \leq 4) & \text{when } 3 \leq a_x. \end{array}$$

Finally, $\mathcal{S}(x = 2y, \delta_y \leq 1, \delta_x \geq 3)$ is, unexpectedly, non-empty. For instance, by fixing $\delta_x = 3, \delta_y = 0, a_x = 0, a_y = 0$, we have that $x = 0, y = 0$ is a solution.

From Def. 2.3, when d contains only upper bounds to range variables, the set of models of $c \wedge d$ coincides with the set of non-empty polyhedra obtained by instantiating the parameters of $\mathcal{S}(c \wedge d)$. However, the last example points out that this is not the general case. The problem is that lower bounds are not monotonic in the precise sense that $Sol(\mathcal{S}(\delta_x \geq 3), (a_x = 0, \delta_x = 3))$ is a model of $\delta_x \geq 3$, and $Sol(\mathcal{S}(x = 2y, \delta_y \leq 1), (a_y = 0, \delta_y = 0))$ is a model of $x = 2y, \delta_y \leq 1$, but $Sol(\mathcal{S}(x = 2y, \delta_y \leq 1, \delta_x \geq 3), (a_x = 0, \delta_x = 3, a_y = 0, \delta_y = 0))$ is not a model of $x = 2y, \delta_y \leq 1, \delta_x \geq 3$. Then, we explicitly restrict to parameter instances that satisfy the lower bounds in a range constraint.

Definition 2.8. *Let $S = \mathcal{S}(c \wedge d)$. We say that a parameter instance $\mathbf{u}$ is a solution of d in S if for every $\delta_x \geq s$ in d, we have: $width(Sol(S,\mathbf{u}),x) \geq 2s$.*

Also, we extend the *width*() function in the context of parameterized polyhedra.

Definition 2.9 (parwidth). *Let $S = \mathbf{Ax} \leq \mathbf{b} + \mathbf{Ba}$ be a parameterized linear system. The width of x in S is defined as: $parwidth(S,x) = max_{\mathbf{u}}\ width(Sol(S,\mathbf{u}),x)$, if it exists. Otherwise, $parwidth(S,x) = \infty$.*

We postpone to the next section the problem of effectively computing *parwidth*() and checking the existence of a solution of d. The next result shows how to check satisfiability of a linear-range constraint. Since $parwidth(\mathcal{S}(c \wedge d),x)$ turns out to be an upper bound to the maximal width of x in any model of a satisfiable $c \wedge d$, it can be adopted in the entailment problem.

Theorem 2.10. *A linear-range constraint $c \wedge d$ is satisfiable iff c is satisfiable, and $d \wedge \bigwedge_{\delta_x \in vars(d)} 0 \leq \delta_x$ is satisfiable as a linear constraint, and there exists a solution of d in $\mathcal{S}(c \wedge d)$.*

For $c \wedge d$ satisfiable and $s = parwidth(\mathcal{S}(c \wedge d),x)/2 \neq \infty$, we have that $c \wedge d$ entails $\delta_x \leq s$.

Notice that when d contains only upper bounds to range variables, satisfiability of $c \wedge d$ reduces to satisfiability of c. In fact, by setting $\delta_x = 0$ for every x, we readily have that $d \wedge \bigwedge_{\delta_x \in vars(d)} 0 \leq \delta_x$ is satisfiable.

Example 2.11. *Let $c \wedge d$ be $0 \leq x \leq 10, 0 \leq y \leq x, \delta_x = 3, \delta_y \geq 4$. The parameterized linear system $S = \mathcal{S}(c \wedge d)$ is:*

$$0 \leq x \leq 10, 0 \leq y \leq x, \delta_x = 3, 4 \leq \delta_y,$$
$$a_x - \delta_x \leq x \leq a_x + \delta_x, a_y - \delta_y \leq y \leq a_y + \delta_y,$$

where $a_x, a_y, \delta_x, \delta_y$ are parameters. Since c and $\delta_x = 3, \delta_y \geq 4$ are readily satisfiable, in order to use Thm. 2.10, we have to find a parameter instance $\mathbf{u}$ such that $width(Sol(S,\mathbf{u}),x) \geq 6$, since $\delta_x \geq 3$ is in d, and $width(Sol(S,\mathbf{u}),y) \geq 8$, since $\delta_y \geq 4$ is in d. By defining $\mathbf{u}$ as:

$$a_x = 7, \delta_x = 3, a_y = 5, \delta_y = 5,$$

we have $width(Sol(S,\mathbf{u}),x) = 6$ and $width(Sol(S,\mathbf{u}),y) = 10$, namely $\mathbf{u}$ is a solution for d in S.

3. COMPUTING WIDTHS

We recall an intermediate notion to capture the maximum absolute value r, if it exists, of a linear expression over the solutions of a (non-parameterized) polyhedron.

Definition 3.1. *For $S = Sol(\mathbf{Ax} \leq \mathbf{b}) \neq \emptyset$, we define $abs(S, \mathbf{c}^T\mathbf{x} + \alpha) = max\{|\mathbf{c}^T\mathbf{x}_0 + \alpha| \mid \mathbf{x}_0 \in S\}$ if it exists. Otherwise, $abs(S, \mathbf{c}^T\mathbf{x} + \alpha) = \infty$.*

A direct implementation of the *abs*() function relies on standard linear programming problems. For $M = max\{\mathbf{c}^T\mathbf{x} + \alpha \mid \mathbf{x} \in S\}$ and $m = min\{\mathbf{c}^T\mathbf{x} + \alpha \mid \mathbf{x} \in S\}$, we have that $abs(S, \mathbf{c}^T\mathbf{x} + \alpha) = r \in \mathbb{R}$ iff $M \in \mathbb{R}, m \in \mathbb{R}$ and $max\{M, -m\} = r$. Let us now tackle the problem of computing *parwidth*() by reasoning on the Minkowski's form of parameterized linear systems, a generalization due to [8] of the well-known Minkowski's theorem. An implementation of the following result is provided in the `polylib` library [7].

Theorem 3.2 ([8]). *For a parameterized linear system $\mathbf{Ax} \leq \mathbf{b} + \mathbf{Ba}$ there exist:*

- *a generating matrix $\mathbf{R}$,*
- *and finitely many pairs $(\mathbf{v^a}(1), \mathbf{C}_1\mathbf{a} \leq \mathbf{c}_1)$, ..., $(\mathbf{v^a}(k), \mathbf{C}_k\mathbf{a} \leq \mathbf{c}_k)$ where, for $i = 1..k$, $\mathbf{v^a}(i)$ is a vector parametric in $\mathbf{a}$ and $Sol(\mathbf{C}_i\mathbf{a} \leq \mathbf{c}_i) \neq \emptyset$,*

such that:

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

A column of $\mathbf{R}$ is called a *ray*. The set $Cone(\mathbf{R}) = \{\mathbf{x} \mid \mathbf{x} = \mathbf{R}\boldsymbol{\lambda}, \boldsymbol{\lambda} \geq 0\ \}$ is the cone generated by the rays.

A vector $\mathbf{v^a}(i)$ is called a *parameterized vertex*. The set $ConvexHull(\mathbf{V}) = \{\mathbf{x} \mid \mathbf{x} = \mathbf{V}\boldsymbol{\gamma}, \boldsymbol{\gamma} \geq 0, \Sigma\boldsymbol{\gamma} = 1\ \}$, where $\mathbf{V}$ is a finite set of vectors, is the convex hull of the vertices, namely the smallest convex set which contains all vertices.

A system $\mathbf{C}_i\mathbf{a} \leq \mathbf{c}_i$ is called the *validity domain* of the vertex $\mathbf{v^a}(i)$. For a parameter instance $\mathbf{u}$, the convex hull set is built from the (instantiated) vertices whose validity domain includes $\mathbf{u}$. The special case $k = 0$ models *empty parameterized polyhedra*, which are empty for every instance of the parameters.

Example 3.3. *Consider $\mathcal{S}(c \wedge d)$ from Ex. 2.11. The generating matrix of its Minkowski's form has no ray. Fig. 1 shows the set of parameterized vertices and domains. The parameter instance $a_x = 7, \delta_x = 3, a_y = 5, \delta_y = 5$ belongs to the domains of vertices: 1, 2, 4, 5, 7, 8, 9, 10, 12, 13, 14, 15.*

The next result states that $parwidth(S,x)$ can be computed by scanning the pairs of parameterized vertices of S. For each pair, we calculate the maximum distance of the x component of the two vertices over the intersection of their domains, if not empty. The maximum value over all the pairs of vertices is then the result of $parwidth(S,x)$.

Theorem 3.4. *Consider the Minkowski's form as stated in Thm. 3.2 of the parameterized linear system $S = \mathcal{S}(c \wedge d)$. We have: $parwidth(S, \mathbf{x}_i) = r \in \mathbb{R}$ iff $row(\mathbf{R}, i) = \mathbf{0}$; and,*

$$r = max(\{0\} \cup \{s \mid 1 \leq m < n \leq k, Sol(P_{m,n}) \neq \emptyset,$$
$$s = abs(Sol(P_{m,n}), \mathbf{v^a}(m)_i - \mathbf{v^a}(n)_i)\}),$$

where $P_{m,n} = \mathbf{C}_m\mathbf{a} \leq \mathbf{c}_m, \mathbf{C}_n\mathbf{a} \leq \mathbf{c}_n$.

Moreover, there exists a solution of d in S iff the following constraint over parameters:

$$\bigwedge_{\substack{\delta_{\mathbf{x}_i} \geq s \in d,\\ s > 0,\\ row(\mathbf{R},i) = \mathbf{0}}} \ \bigvee_{1 \leq m < n \leq k} (P_{m,n} \wedge |\mathbf{v^a}(m)_i - \mathbf{v^a}(n)_i| \geq 2s)$$

is satisfiable.

Summarizing, Thm. 3.4 provides us with a decision procedure for the existence of solutions of d, and with a procedure for computing $parwidth(\mathcal{S}(c \wedge d), x)$.

Example 3.5. *Consider $c \wedge d$ from Ex. 2.11, and the parameterized vertices and domains in Fig. 1. The intersection of the domains 2 and 14, namely $P_{2,14}$ is:*

$$0 \leq a_y - \delta_y \leq a_x + 3, a_x + 3 \leq 10, 4 \leq \delta_y, \delta_x = 3,$$
$$7 \leq a_x \leq 13, 10 \leq a_y + \delta_y, a_y - \delta_y \leq 10,$$

$$\mathbf{x} = (x, y) \qquad \mathbf{a} = (a_x, a_y, \delta_x, \delta_y)$$

$\mathbf{v}^{\mathbf{a}}(1) = (a_x + 3, a_y + \delta_y)$
when $0 \le a_y + \delta_y \le a_x + 3$,
$a_x + 3 \le 10$

$\mathbf{v}^{\mathbf{a}}(2) = (a_x + 3, a_y - \delta_y)$
when $0 \le a_y - \delta_y \le a_x + 3$,
$a_x + 3 \le 10$

$\mathbf{v}^{\mathbf{a}}(3) = (a_x - 3, a_y + \delta_y)$
when $0 \le a_y + \delta_y \le a_x - 3$,
$a_x - 3 \le 10$

$\mathbf{v}^{\mathbf{a}}(4) = (a_x - 3, a_y - \delta_y)$
when $0 \le a_y - \delta_y \le a_x - 3$,
$a_x - 3 \le 10$

$\mathbf{v}^{\mathbf{a}}(5) = (a_y + \delta_y, a_y + \delta_y)$
when $0 \le a_y + \delta_y \le a_x + 3$,
$a_x - 3 \le a_y + \delta_y \le 10$

$\mathbf{v}^{\mathbf{a}}(6) = (a_y - \delta_y, a_y - \delta_y)$
when $0 \le a_y - \delta_y \le 10$,
$a_x - 3 \le a_y - \delta_y \le a_x + 3$

$\mathbf{v}^{\mathbf{a}}(7) = (a_x + 3, a_x + 3)$
when $0 \le a_x + 3 \le 10$,
$a_y - \delta_y \le a_x + 3 \le a_y + \delta_y$

$\mathbf{v}^{\mathbf{a}}(8) = (a_x - 3, a_x - 3)$
when $0 \le a_x - 3 \le 10$,
$a_y - \delta_y \le a_x - 3 \le a_y + \delta_y$

$\mathbf{v}^{\mathbf{a}}(9) = (a_x + 3, 0)$
when $0 \le a_x + 3 \le 10$,
$0 \le a_y + \delta_y, a_y - \delta_y \le 0$

$\mathbf{v}^{\mathbf{a}}(10) = (a_x - 3, 0)$
when $0 \le a_x - 3 \le 10$,
$0 \le a_y + \delta_y, a_y - \delta_y \le 0$

$\mathbf{v}^{\mathbf{a}}(11) = (0, 0)$
when $-3 \le a_x \le 3$,
$0 \le a_y + \delta_y, a_y - \delta_y \le 0$

$\mathbf{v}^{\mathbf{a}}(12) = (10, a_y + \delta_y)$
when $7 \le a_x \le 13$,
$0 \le a_y + \delta_y \le 10$

$\mathbf{v}^{\mathbf{a}}(13) = (10, a_y - \delta_y)$
when $7 \le a_x \le 13$,
$0 \le a_y - \delta_y \le 10$

$\mathbf{v}^{\mathbf{a}}(14) = (10, 10)$
when $7 \le a_x \le 13$,
$10 \le a_y + \delta_y, a_y - \delta_y \le 10$

$\mathbf{v}^{\mathbf{a}}(15) = (10, 0)$
when $7 \le a_x \le 13$,
$0 \le a_y + \delta_y, a_y - \delta_y \le 0$

Figure 1: Parameterized vertices for $\mathcal{S}(c \wedge d)$ from Ex. 2.11. The additional constraint $4 \le \delta_y, \delta_x = 3$ must be added to the domain of every vertex. It is omitted for lack of space and to enhance readability.

which readily simplifies to:

$$a_x = 7, 0 \le a_y - \delta_y \le 10, 10 \le a_y + \delta_y, 4 \le \delta_y, \delta_x = 3.$$

For $i = 2$, $\mathbf{x}_i$ is y and $\mathbf{v}^{\mathbf{a}}(2)_i - \mathbf{v}^{\mathbf{a}}(14)_i$ is $(a_y - \delta_y) - 10$. The absolute value of such an expression over $P_{2,14}$ is 10, and it can be obtained by the parameter instance $\mathbf{u}$ defined as $a_x = 7, \delta_x = 3, a_y = 5, \delta_y = 5$. Notice this is the parameter instance used in Ex. 2.11.

Since the absolute value over all other pairs of vertices cannot be greater than 10 (due to the original constraint $0 \le x \le 10, 0 \le y \le x$), by Thm. 3.4, we can conclude:

$$parwidth(\mathcal{S}(c \wedge d), y) = 10.$$

In addition to $|\mathbf{v}^{\mathbf{u}}(2)_2 - \mathbf{v}^{\mathbf{u}}(14)_2| = 10 \ge 8$ covering $\delta_y \ge 4$, we also observe that the parameter instance $\mathbf{u}$ above is a solution of $P_{10,14}$ and $|\mathbf{v}^{\mathbf{u}}(10)_1 - \mathbf{v}^{\mathbf{u}}(14)_1| = 6 \ge 6$, covering $\delta_x \ge 3$. From Thm. 3.4, there exists a solution of d, and, as noticed in Ex. 2.11, $\mathbf{u}$ actually turns out to be one of them.

4. RANGE CONSTRAINTS FOR CLP($\mathbb{R}$)

4.1 Operational Semantics of CLP($\mathbb{R}$)

The operational semantics of a CLP($\mathbb{R}$) program P consists of *derivations* from states to states [5, 6]. Here, we consider derivations via the leftmost selection rule. A state $\langle Q \| c_s \rangle$ is a pair of a query Q and a linear constraint c_s called the constraint store. A query $Q = [c,]A_1, \dots, A_n$ is a sequence of $n \ge 0$ atoms, with possibly a linear constraint c before them. A state $\langle [c,]A_1, \dots, A_n \| c_s \rangle$ is reduced to another state, called a *resolvent*, as follows:

$\mathcal{R}1$ If a linear constraint c appears at the left of the query, and $c_s \wedge c$ is satisfiable, the resolvent is:

$$\langle A_1, \dots, A_n \| c_s \wedge c \rangle.$$

$\mathcal{R}2$ If an atom $A_1 = p(x_1, \dots, x_h)$ appears at the left of the query, and $p(y_1, \dots, y_h) \leftarrow c, B_1, \dots, B_k$ is a renamed apart clause from P, then the resolvent is:

$$\langle c, B_1, \dots, B_k, A_2, \dots, A_n \| c_s \wedge \bigwedge_{i=1..h} x_i = y_i \rangle.$$

A *derivation* from an initial state state $S_0 = \langle Q \| \mathtt{true} \rangle$ is a (finite or infinite) maximal sequence of states $S_0, S_1, \dots S_n, \dots$ such that there is a reduction from S_i to S_{i+1}, for $i \ge 0$. A derivation for a query Q is a derivation from $\langle Q \| \mathtt{true} \rangle$. The last state of a finite derivation is of the form $\langle Q' \| c_s \rangle$. If Q' is not empty, the derivation is *failed*. Otherwise, it is *successful*, or a *refutation*: $\exists_{-\mathbf{v}} c_s$ is called the answer constraint, where the final constraint store is projected over the set $\mathbf{v}$ of variables appearing in the initial state S_0.

4.2 Adding Linear-Range Constraints

Assume now that range constraints are admitted in queries and program clauses.

We extend the operational semantics of CLP($\mathbb{R}$) by assuming that a constraint store is now a linear-range constraint $c_s \wedge d_s$. We have to specify further: (1) the notion of satisfiability of a constraint store, which is used in rule $\mathcal{R}1$; (2) a new rule dealing with assertion of range constraints; and (3) the definition of the answer constraint in presence of the variable range constraint d_s.

Concerning (1), we resort to Def. 2.3 for the notion of satisfiability of linear-range constraints, and to Thm. 2.10 for a checking procedure. The transition rule $\mathcal{R}1$ becomes:

$\mathcal{R}1'$ If a linear constraint c appears at the left of the query, and $c_s, c \wedge d_s$ is satisfiable, the resolvent is:

$$\langle A_1, \dots, A_n \| c_s, c \wedge d_s \rangle.$$

Concerning (2), the following intuitive rule can be defined:

$\mathcal{R}1''$ If a range constraint d appears at the left of the query, and $c_s \wedge d_s, d$ is satisfiable, the resolvent is:

$$\langle A_1, \ldots, A_n \| c_s \wedge d_s, d \rangle.$$

Finally, consider (3). The answer constraint $\exists_{-\mathbf{v}} c_s$ projects the linear constraint store c_s over the variables $\mathbf{v}$ of the initial query. How does this extend to linear-range constraints $c_s \wedge d_s$? Two issues must be taken into account:

- The linear-range constraint $c_s \wedge d_s$ may not explicitly contain all range constraints entailed from it. As an example, if $c_s \wedge d_s$ is $y \leq x \leq y+1, \delta_y \leq 1$, one would expect that $\delta_x \leq 1.5$ is in the answer constraint.
- The projection should consider also range variables, not just linear constraint variables.

We tackle the first issue by adding to d_s the range constraint $d' = \mathit{entail}(c_s \wedge d_s, \mathbf{v})$ defined as the conjunction of entailed inequalities $\delta_x \leq s$, for every $x \in \mathbf{v}$, where, as shown in Thm. 2.10, $s = \mathit{parwidth}(\mathcal{S}(c_s \wedge d_s), x)/2 \neq \infty$. In the example, $d' = \delta_y \leq 1, \delta_x \leq 1.5$. For the second issue, we project the final range constraint over the appropriate set of range variables. Summarizing, the answer constraint of a refutation with final constraint store $c_s \wedge d_s$ is defined as:

$$(\ \exists_{-\mathbf{v}} c_s\ ) \wedge (\ \exists_{-\delta_{\mathbf{v}}}(d_s, \mathit{entail}(c_s \wedge d_s, \mathbf{v}))\ ),$$

where $\mathbf{v}$ is the set of variables appearing in the initial state and $\delta_{\mathbf{v}} = \{\delta_x \mid x \in \mathbf{v}\}$.

As an example, consider the classic `Mortgage` program.

```
(m1)  mortgage(P,T,R,B) :-
          T = 0, B = P.
(m2)  mortgage(P,T,R,B) :-
          T >= 1,
          NP = P + P * 0.05 - R,
          NT = T - 1,
          mortgage(NP,NT,R,B).
```

A query $\delta_{\mathtt{R}} \leq$ `5, mortgage(P, 3, R, 0)` is intended to calculate the principal `P` one could be granted for a 3 years mortgage with final balance of zero and annual repay of `R`, where `R` is known with an approximation of 5 units. The constraint in the final store is `NP1 = P * 1.05 - R, NP2 = NP1 * 1.05 - R, NP3 = NP2 * 1.05 - R, NP3 = 0,` $\delta_{\mathtt{R}} \leq$ `5`. By projecting over `R` and `P`, we get the answer constraint `P = R * 2.723,` $\delta_{\mathtt{R}} \leq$ `5,` $\delta_{\mathtt{P}} \leq$ `13.615`. Notice that $13.615 = 5 \cdot 2.723$. Summarizing, under the stated conditions, the granted mortgage can vary up to $\pm$ 13.615 units.

An example with linear-range constraints in programs can be devised for the sum of a list of measures, restricted to those that are known with an approximation of at most ± 1.

```
(s1)  sum([], 0).
(s2)  sum([X|Xs], S) :-
          δX ≤ 1, S = S1 + X, sum(Xs, S1).
(s3)  sum([X|Xs], S) :-
          sum(Xs, S).
```

The query `sum([X, Y], S)` returns the answer constraints:

- `S = X + Y,` $\delta_{\mathtt{X}} \leq$ `1,` $\delta_{\mathtt{Y}} \leq$ `1,` $\delta_{\mathtt{S}} \leq$ `2`, stating that when `X` and `Y` have an approximation of up to ± 1, then the sum is `S = X + Y`, and with an approximation up to ± 2;
- `S = X,` $\delta_{\mathtt{X}} \leq$ `1,` $\delta_{\mathtt{S}} \leq$ `1`, stating that when `Y` has an unknown approximation and `X` has an approximation up to ± 1, then the sum is `S = X`, and with an approximation up to ± 1;
- `S = Y,` $\delta_{\mathtt{Y}} \leq$ `1,` $\delta_{\mathtt{S}} \leq$ `1`, is symmetric to the previous case;
- `S = 0,` $\delta_{\mathtt{S}} =$ `0`, stating that when both `X` and `Y` have an unknown approximation then the sum is zero, and it is a definite value.

The query $\delta_{\mathtt{X}} \geq$ `2,` $\delta_{\mathtt{Y}} \leq$ `0.5, sum([X, Y], S)` returns the answer constraint `S = Y,` $\delta_{\mathtt{X}} \geq$ `2,` $\delta_{\mathtt{Y}} \leq$ `0.5,` $\delta_{\mathtt{S}} \leq$ `0.5`.

5. CONCLUSIONS

To some extent, linear-range constraints are a form of interval (linear) constraints, where intervals $s_0 \leq \delta_x \leq s_1$ refer to the minimal (s_0) and the maximal (s_1) radius of values (i.e., approximation) that a variable x can assume. We adopted a controlled form of parameterized linear systems to devise correct and complete algorithms for satisfiability and entailment. Also, we extended CLP($\mathbb{R}$) with linear-range constraints, hence providing a form of meta-level reasoning about the range of variables.

Future work includes an experimental evaluation of the approach, an enhancement of the syntax of range constraints to admit disequalities (i.e., $\delta_x \neq s$) and generic inequalities (e.g., $\delta_x \leq \delta_y$), and the extension of the entailment procedure to lower bounds on range variables.

6. REFERENCES

[1] F. Benhamou and W. J. Older. Applying interval arithmetic to real, integer, and boolean constraints. *Journal of Logic Programming*, 32(1):1–24, 1997.

[2] T. Gal. *Postoptimal Analyses, Parametric Programming, and Related Topics.* de Gruyter, Berlin, Germany, 2nd edition, 1995.

[3] P. V. Hentenryck, L. Michel, and F. Benhamou. Newton - Constraint programming over nonlinear constraints. *Science of Computer Programming*, 30(1-2):83–118, 1998.

[4] T. J. Hickey. CLIP: A CLP(Intervals) dialect for metalevel constraint solving. In *Practical Aspects of Declarative Languages*, volume 1753 of *LNCS*, pages 200–214. Springer, 2000.

[5] J. Jaffar and M. Maher. Constraint logic programming: A survey. *Journal of Logic Programming*, 19,20:503–581, 1994.

[6] J. Jaffar, M. Maher, K. Marriott, and P. J. Stuckey. The semantics of constraint logic programs. *Journal of Logic Programming*, 37(1-3):1–46, 1998.

[7] V. Loechner. Polylib: a library for manipulating parameterized polyhedra, 2007. Available at `http://icps.u-strasbg.fr/polylib`, Version 5.22.3.

[8] V. Loechner and D. K. Wilde. Parameterized polyhedra and their vertices. *International Journal of Parallel Programming*, 25:525–549, 1997.

[9] R. E. Moore. *Interval Analysis.* Prentice Hall, 1966.

[10] A. Schrijver. *Theory of Linear and Integer Programming.* J. Wiley & Sons, 1986.