Quantified Constraints Under Perturbation

Stefan Ratschan
e-mail: Stefan.Ratschan@risc.uni-linz.ac.at*

January 18, 2001

Abstract

Quantified constraints (i.e., first-order formulae over the real numbers) are often exposed to perturbations: Constants that come from measurements usually are only known up to certain precision, and numerical methods only compute with approximations of real numbers. In this paper we study the behavior of quantified constraints under perturbation by showing that one can formulate the problem of solving quantified constraints as a nested parametric optimization problem followed by one sign computation. Using the fact that minima and maxima are stable under perturbation, but the sign of a real number is stable only for non-zero inputs, we derive practically useful conditions for the stability of quantified constraints under perturbation.

1 Introduction

Let a quantified constraint be a first-order formula over the real numbers. This means that it contains quantifiers ($\exists$, $\forall$), connectives ($\wedge$, $\vee$, $\neg$), predicate symbols (e.g., $=$, $<$, $\leq$), function symbols (e.g., $+$, $-$, $\times$, sin, exp), rational constants and variables ranging over real numbers. Solving such constraints has various applications in studying the stability of differential equations [19, 13], control theory [9, 17, 8], and various other areas [33, 40, 15]. Traditionally, quantified constraints have been solved by symbolic methods [34, 6, 7, 20], but lately (validated) numerical methods appear for special cases [31, 32, 3, 12, 16, 21] and the general case [25, 24, 27].

Such numerical methods introduce perturbations, for example via rounding errors. Furthermore, constants that come from measurements usually are known only up to uncertainties. Thus it is important to know how the solution set of quantified constraints behaves under perturbations. We serve in clarifying this by the following contributions of this paper:

*This work was supported by the Austrian Science Fund FWF in the frame of the project SFB F1303

- We show that the solution set of (closed) quantified constraints is stable under perturbation except for one (unstable) sign computation. More specifically, we formulate quantified constraint solving as a nested parametric optimization problem followed by the computation of the sign of the resulting optimum.
- We provide two practically useful conditions for stability, one considering the behavior under exchanging strict inequalities with non-strict inequalities and one considering the volume of the occurring solution sets.
- We show that, for constraints containing only inequality predicates, stability under perturbation is independent of whether we use information about the continuity of constraints and equality of variables.

The paper is self-contained except for some basic knowledge of mathematical logic [10]. Its structure is as follows: In Section 2, we give a formalization of the problem. In Section 3, we show that for a certain class of quantified constraints their solution set is stable under perturbation except for one sign computation. In Section 4, we present criteria that are useful in checking whether a given constraint is stable. In Section 5, we discuss various more general forms of perturbation. In Section 6, we discuss the generalization of the preceding results to arbitrary constraints, and in Section 7, we conclude the paper.

2 Stability under Perturbation

The motivation for this paper comes from applying numerical computation to the problem of computing the solution set of quantified constraints [25, 24, 27]. For numerical computation to be applicable, our problem needs to fulfill Hadamard's definition of well-posedness:

1. A solution exists
2. The solution is unique
3. The result depends continuously on the data

The first condition is always fulfilled because a quantified constraint always has a (possibly empty) solution set. Also the second condition is always fulfilled because the solution set of a quantified constraint is unique. So it remains to study the third condition: Whether the result depends continuously on the data, that is, whether the solution set of quantified constraints is stable under perturbation.

Let us first fix some preliminaries: We denote by C the set of all (quantified) constraints. We assume that the set of allowed variables is totally ordered, and we allow only function symbols that denote continuous functions. Furthermore we require that all atomic sub-constraints are of the form θ `r 0`, where θ is a term and `r` a predicate symbol. The zero on the right-hand sides will give us a handle for perturbing atomic constraints.

We typeset expressions in the object language (i.e., constraints and all their syntactic components) in typewriter font. The symbol ϕ will function as a meta-variable over constraints, and the symbol θ will function as a meta-variable over terms. For any syntactic entity `s` (i.e., constraint, term, predicate symbol, function symbol) we denote its meaning by $[\![\mathtt{s}]\!]$. Note that in contrast to mathematical logic [10] we assign meaning independent of a certain variable assignment. As a result, the meaning of a constraint is a solution set, the meaning of a term is a function, and the meaning $[\![\mathtt{x}]\!]$ of a variable `x` is the identity function (see [27] for details). Furthermore, the solution set of a closed constraint is either empty (i.e., the constraint is false) or contains the empty tuple (i.e., the constraint is true).

Now we proceed by clarifying the notion of perturbation. We need to know how to measure distance:

Definition 1 *A function* $d : A \times A \to \mathbb{R}$ *is a* pseudo-metric *iff*

- *for all* $x \in A$, $d(x, x) = 0$,
- *for all* $x, y \in A$, $d(x, y) = d(y, x)$, *and*
- *for all* $x, y, z \in A$, $d(x, z) \leq d(x, y) + d(y, z)$.

For solution sets x and y we define their distance $d(x, y)$ as the volume of the symmetric set difference (i.e., $\mathrm{Vol}(x \nabla y)$). In $\mathbb{R}^0$ we assign the volume zero to the empty set and infinite volume to the set containing the empty tuple. For vectors and functions we define their distance as the supremum of the distance of their corresponding elements.

A quantified constraint contains many entities that one could perturb. Observe that one can view a constraint as a mapping that takes the meaning of its constant symbols (i.e., real constants), and its variables (i.e., identity functions) and produces the according meaning (i.e., solution set) of the constraint. These two kinds of inputs often occur in perturbed form: Constants that come from measurements in nature are known only up to a certain accuracy; computation with approximations to identity functions can be cheaper than computation with the exact identity functions [27, 25]; and rounding errors change these objects if we represent them by floating-point numbers. So we study perturbations of constant symbols and variables in this paper.

However, in order to keep the exposition simple, we will first do the main development only with perturbations of the zeros that occur on the right-hand side of predicates in atomic constraints. In Section 5 we will generalize the approach to perturbations of arbitrary constant symbols and variables.

Definition 2 *Given a constraint ϕ in n atomic sub-constraints and a $\delta \in \mathbb{R}^+$, a δ-perturbation of ϕ is a real tuple $(c_1, \dots, c_n)$ such that $d((c_1, \dots, c_n), (0, \dots, 0)) \leq \delta$. A constraint ϕ' is the* result of a δ-perturbation of ϕ *iff it results from replacing the k-th zero on the right-hand side of an atomic sub-constraint of ϕ with c_k.*

Now we can use the notion of continuity from analysis for defining stability:

Definition 3 *A function $f : C \to A$ is* stable for a constraint ϕ *iff for all $\varepsilon > 0$ there exists a $\delta > 0$ such that for all ϕ' resulting from a δ-perturbation ϕ, $d(f(\phi), f(\phi')) < \varepsilon$. A constraint ϕ is* stable *iff $[\![\cdot]\!]$ is stable for ϕ.*

As an example for an unstable constraint ϕ consider $(\exists \mathtt{x})[\mathtt{x}^2 < \mathtt{0}]$, for which $[\![\phi]\!]$ is $\{\}$ (i.e., ϕ is false). Choose $\varepsilon = 1$ and let $\delta > 0$ be arbitrary but fixed. Then the tuple (δ) is a δ-perturbation of ϕ, but for the result $(\exists \mathtt{x})[\mathtt{x}^2 < \delta]$ of this perturbation, which we denote by ϕ', $[\![\phi']\!] = \{()\}$ (i.e., ϕ' is true). Thus $d([\![\phi]\!], [\![\phi']\!]) = \infty$ and $[\![\phi]\!]$ is unstable.

For the rest of the paper we need some additional technical tools: For any constraint we call the exchange of $\leq$ by $<$, $\geq$ by $>$ and vice-versa an *infinitesimal predicate change.* Furthermore we assume that each occurring variable $\mathtt{x}$ ranges over a compact subset of the reals (e.g., a closed, finite interval) which we denote by $R_{\mathtt{x}}$. Such ranges arise naturally in most practical applications. They significantly simplify the exposition in this paper, although most of the results can be straightforwardly generalized to the non-compact case by introducing infinitely small and infinitely large elements. We also assume that all occurring solution sets are measurable.

When doing recursion over the structure of constraints, one often needs to project a tuple of real numbers into the free-variable space of a sub-constraint. For example, given a constraint $xy > 0 \ \wedge \ z < 0$, where the variables are alphabetically ordered, and a tuple $(1, 2, 3)$, we want to project $(1, 2, 3)$ to $(1, 2)$ for the sub-constraint $xy > 0$, and to (3) for the sub-constraint $z < 0$:

Definition 4 *For sets of variables V and V' such that $V \supseteq V'$, we denote by $(a_1, \dots, a_n)_{V \mapsto V'}$ the tuple $(b_1, \dots, b_k)$ such that $b_i = a_j$ where j is the index of the i-th variable in V' among the variables in V. For a constraint ϕ with free variables V, a constraint ϕ' with free variables V', and a tuple $(a_1, \dots, a_n)$, $(a_1, \dots, a_n)_{\phi \mapsto \phi'} := (a_1, \dots, a_n)_{V \mapsto V'}$.*

3 Stability Except for One Sign Computation

In this section we formulate quantified constraint solving as a nested parametric optimization problem followed by the computation of the sign of the resulting optimum. For this we restrict ourselves to a certain class of constraints, and show later, how to extend the results to the stability of general constraints.

Definition 5 *A constraint is* monotonic *iff it does not contain negations and all of its atomic constraints either have the form* $\theta > 0$ *or the form* $\theta \geq 0$.

We can transform arbitrary constraints to monotonic constraints as follows:

1. Push negations inside as far as possible,
2. Replace $\neg\theta < 0$ by $\theta \geq 0$, $\neg\theta \leq 0$ by $\theta > 0$, $\neg\theta > 0$ by $\theta \leq 0$, $\neg\theta \geq 0$ by $\theta < 0$, $\neg\theta = 0$ by $\theta \neq 0$, and $\neg\theta \neq 0$ by $\theta = 0$,
3. Replace atomic constraints of the form $\theta = 0$ by $\theta \leq 0 \ \wedge \ \theta \geq 0$, and $\theta \neq 0$ by $\theta < 0 \ \vee \ \theta > 0$, and
4. Replace atomic constraints of the form $\theta < 0$ by $-\theta > 0$, and $\theta \leq 0$ by $-\theta \geq 0$.

For a constraint ϕ we denote the result of this transformation by $T(\phi)$ (for example, $T(\mathtt{x}^2 = 0 \wedge \mathtt{y} < 0) = -\mathtt{x}^2 \geq 0 \wedge \mathtt{x}^2 \geq 0 \wedge \mathtt{y} > 0)$.

Now one can easily prove by induction over the structure of monotonic constraints that such constraints are monotonic in the following sense:

Lemma 1 *For every monotonic constraint* ϕ *and the result* ϕ' *of a perturbation of* ϕ *with non-negative elements only,* $[\![\phi]\!] \supseteq [\![\phi']\!]$.

For monotonic constraints we can replace the discrete notion of truth by a continuous concept (a similar idea has been used by S. Shary [30]):

Definition 6 *For every monotonic constraint* ϕ *in* n *free variables, let* $\tau(\phi)$, *the* degree of truth, *be the following function in* $\mathbb{R}^n \to \mathbb{R}$:

- $\tau(\theta \ \mathtt{r} \ \mathtt{c})(a_1, \ldots, a_n) = [\![\theta]\!](a_1, \ldots, a_n)$
- $\tau(\phi_1 \wedge \phi_2) = \tau(\wedge, V_1, V_2)(\tau(\phi_1), \tau(\phi_2))$, *where* V_1 *is the set of variables in* ϕ_1 *and* V_2 *is the set of variables in* ϕ_2, *and*
- $\tau(\phi_1 \vee \phi_2) = \tau(\vee, V_1, V_2)(\tau(\phi_1), \tau(\phi_2))$, *where* V_1 *is the set of variables in* ϕ_1 *and* V_2 *is the set of variables in* ϕ_2,
- $\tau(\exists \mathtt{x} \ \phi) = \tau(\exists \mathtt{x})(\tau(\phi))$
- $\tau(\forall \mathtt{x} \ \phi) = \tau(\forall \mathtt{x})(\tau(\phi))$

where

- $\tau(\wedge, V_1, V_2) : (\mathbb{R}^{|V_1|} \to \mathbb{R}) \times (\mathbb{R}^{|V_2|} \to \mathbb{R}) \to (\mathbb{R}^{|V_1 \cup V_2|} \to \mathbb{R})$ *such that for all* $a \in \mathbb{R}^{|V_1 \cup V_2|}$, $\tau(\wedge, V_1, V_2)(f_1, f_2)(a) = \min(f_1(a_{V_1 \cup V_2 \mapsto V_1}), f_2(a_{V_1 \cup V_2 \mapsto V_1}))$, *and*

- $\tau(\vee, V_1, V_2) : (\mathbb{R}^{|V_1|} \to \mathbb{R}) \times (\mathbb{R}^{|V_2|} \to \mathbb{R}) \to (\mathbb{R}^{|V_1 \cup V_2|} \to \mathbb{R})$ *such that for all* $a \in \mathbb{R}^{|V_1 \cup V_2|}$, $\tau(\vee, V_1, V_2)(f_1, f_2)(a) = \max(f_1(a_{V_1 \cup V_2 \mapsto V_1}), f_2(a_{V_1 \cup V_2 \mapsto V_1}))$, *and*

- $\tau(\exists \mathtt{x}) : (\mathbb{R}^{n+1} \to \mathbb{R}) \to (\mathbb{R}^n \to \mathbb{R})$, *such that* $\tau(\exists \mathtt{x})(f)(a_1, \ldots, a_n) = \sup\{f(x, a_1, \ldots, a_n) | x \in R_{\mathtt{x}}\}$, *and*

- $\tau(\forall \mathtt{x}) : (\mathbb{R}^{n+1} \to \mathbb{R}) \to (\mathbb{R}^n \to \mathbb{R})$, *such that* $\tau(\forall \mathtt{x})(f)(a_1, \ldots, a_n) = \inf\{f(x, a_1, \ldots, a_n) | x \in R_{\mathtt{x}}\}$.

One can view this definition as the interpretation of constraints in an infinitely-many-valued logic [28, 35]. Note that the suprema and infima are always attained because every variable $\mathtt{x}$ ranges over a compact set $R_{\mathtt{x}}$ and all function symbols denote continuous functions only. Furthermore note that infinitesimal predicate changes do not change the result of this definition.

Since we can easily transform every constraint to a monotonic constraint, we accordingly extend the definition of τ to non-monotonic constraints (in fact one could also extend the above definition for constraints containing negations by interpreting negation as multiplication with -1, but this would have the danger of confusing intuition).

Example 1

- $\tau(\exists \mathtt{x}\ -\mathtt{x}^2 \geq \mathtt{0}) = \sup\{\tau(-\mathtt{x}^2 \geq \mathtt{0})(a) \mid a \in R_{\mathtt{x}}\} = 0$

- $\tau(\exists \mathtt{x}\ -\mathtt{x}^2 + \mathtt{1} \geq \mathtt{0}) = \sup\{\tau(-\mathtt{x}^2 + \mathtt{1} \geq \mathtt{0})(a) \mid a \in R_{\mathtt{x}}\} = 1$

- $$\begin{aligned}\tau(\exists \mathtt{x}\, \forall \mathtt{y}\ -\mathtt{x}^2 + \mathtt{y}^2 \geq \mathtt{0}) &= \sup\{\tau(\forall \mathtt{y}\ -\mathtt{x}^2 + \mathtt{y}^2 \geq \mathtt{0})(x) | x \in R_{\mathtt{x}}\} = \\ &= \sup\{\inf\{\tau(-\mathtt{x}^2 + \mathtt{y}^2)(x, y) | y \in R_{\mathtt{y}}\} | x \in R_{\mathtt{x}}\} = \\ &= \sup\{\inf\{-x^2 + y^2 | y \in R_{\mathtt{y}}\} | x \in R_{\mathtt{x}}\} = \sup\{-x^2 | x \in R_{\mathtt{x}}\} = 0\end{aligned}$$

In certain cases the degree of truth captures the semantics of constraints:

Lemma 2 *For every monotonic constraint ϕ in n free variables that contains $\geq$ predicates only, for all $(a_1, \ldots, a_n)$,*

$$(a_1, \ldots, a_n) \in [\![\phi]\!] \text{ iff } \tau(\phi)(a_1, \ldots, a_n) \geq 0.$$

Proof. We proceed by recursion over the structure of constraints. The lemma obviously holds for atomic constraints. Now consider constraints of the following forms:

$\phi_1 \wedge \phi_2$: Let $(a_1, \ldots, a_n)$ be arbitrary but fixed. Then

$$(a_1, \ldots, a_n) \in [\![\phi_1 \wedge \phi_2]\!]$$

iff both

$$(a_1, \ldots, a_n)_{\phi_1 \mapsto \phi_1 \wedge \phi_2} \in [\![\phi_1]\!] \text{ and } (a_1, \ldots, a_n)_{\phi_2 \mapsto \phi_1 \wedge \phi_2} \in [\![\phi_2]\!]$$

iff (by induction hypothesis)

$$\tau(\phi_1)(a_1, \ldots, a_n)_{\phi_1 \mapsto \phi_1 \wedge \phi_2} \geq 0 \text{ and } \tau(\phi_2)(a_1, \ldots, a_n)_{\phi_2 \mapsto \phi_1 \wedge \phi_2} \geq 0$$

iff

$$\min\{\tau(\phi_1)(a_1, \ldots, a_n)_{\phi_1 \mapsto \phi_1 \wedge \phi_2}, \tau(\phi_2)(a_1, \ldots, a_n)_{\phi_2 \mapsto \phi_1 \wedge \phi_2}\} \geq 0.$$

$\exists \mathtt{x}\ \phi$: Let $(a_1, \ldots, a_n)$ be arbitrary but fixed. Then $(a_1, \ldots, a_n) \in [\![\exists \mathtt{x}\ \phi]\!]$ iff

$$\text{there is an } x \text{, for which} (a_1, \ldots, a_n) \in [\![\phi]\!],$$

iff (induction hypothesis)

$$\text{there is an } x \text{ such that } \tau(\phi)(x, a_1, \ldots, a_n) \geq 0$$

iff (by compactness of $R_{\mathtt{x}}$ and continuity the supremum is attained)

$$\sup\{\tau(\phi)(x, a_1, \ldots, a_n) | x \in R_{\mathtt{x}}\} \geq 0$$

iff (by definition of τ)

$$\tau(\exists \mathtt{x}\ \phi)(a_1, \ldots, a_n) \geq 0.$$

The cases for disjunction and universal quantification are dual. ■

One can easily adapt the above proof to get the analogous lemma for strict inequalities:

Lemma 3 *For every monotonic constraint ϕ in n free variables that contains $>$ predicates only, for all $(a_1, \ldots, a_n)$,*

$$(a_1, \ldots, a_n) \in [\![\phi]\!] \textit{ iff } \tau(\phi)(a_1, \ldots, a_n) > 0.$$

By combining Lemma 2 and Lemma 3 we know that one can extract the meaning of a constraint from the sign of its degree of truth, as long as this sign is strictly negative of positive:

Lemma 4 *For every monotonic constraint ϕ in n free variables, for all $(a_1, \ldots, a_n)$, if $\tau(\phi)(a_1, \ldots, a_n) > 0$ then $(a_1, \ldots, a_n) \in [\![\phi]\!]$, and if $\tau(\phi)(a_1, \ldots, a_n) < 0$ then $(a_1, \ldots, a_n) \notin [\![\phi]\!]$.*

It is a well-known fact in computable analysis [23, 37], that the minimum and maximum is stable, and so we have by induction:

Lemma 5 *For every monotonic constraint ϕ, τ is stable for ϕ.*

For monotonic constraints we can easily influence the degree of truth:

Lemma 6 *For every monotonic constraint ϕ and result ϕ' of a δ-perturbation $(\delta, \ldots, \delta)$ of ϕ, $\tau(\phi') = \tau(\phi) - \delta$.*

Proof. After moving all the constants representing δ on the right-hand side of atomic sub-constraints to the according left-hand sides, all occurring atomic sub-constraints are perturbed by $-\delta$ and by induction also $\tau(\phi')$ is perturbed by $-\delta$. ∎

Since the sign of a real number is stable only for non-zero inputs, quantified constraints are stable under perturbation up to sign computation:

Theorem 1 *A monotonic constraint ϕ in n free variables is stable iff*

$$\mathrm{Vol}\{x \in R_{\mathtt{x}} | \tau(\phi)(x) = 0\} = 0$$

Proof.

$\Rightarrow$: Assume that $\mathrm{Vol}\{x \in R_{\mathtt{x}} | \tau(\phi)(x) = 0\} > 0$. By Lemma 6, for every $\delta > 0$ there is a δ-perturbation of zeros such that for all $(a_1, \ldots, a_n)$ such that $\tau(\phi)(a_1, \ldots, a_n) = 0$, $\tau(\phi')(a_1, \ldots, a_n) < 0$. In a similar way we can make these elements positive. So, by Lemma 4, the volume of the $(a_1, \ldots, a_n)$ for which the property of being in $[\![\phi]\!]$ changes under small perturbations is greater than zero, and so ϕ is unstable.

$\Leftarrow$: Assume that $\mathrm{Vol}\{x \in R_{\mathtt{x}} | \tau(\phi)(x) = 0\} = 0$. Since τ is stable and the test for positivity and negativity of a real number is stable for non-zero inputs, by Lemma 4, the result ϕ' of a small perturbation of ϕ can only be different at $(a_1, \ldots, a_n)$ for which $\tau(\phi)(a_1, \ldots, a_n) = 0$. The volume of these $(a_1, \ldots, a_n)$ is zero and so $d(\phi, \phi') = \mathrm{Vol}([\![\phi]\!] \nabla [\![\phi']\!])$ is zero under small perturbations and ϕ is stable.

∎

Corollary 1 *A closed monotonic constraint ϕ is stable iff $\tau(\phi) \neq 0$.*

4 Stability Conditions

In this section we provide two theorems that provide conditions when a given constraint is stable. The first considers infinitesimal predicate changes.

Definition 7 *A constraint ϕ in n free variables is* stable under infinitesimal predicate changes *iff the volume of the $(a_1, \ldots, a_n)$ for which $(a_1, \ldots, a_n) \in [\![\phi]\!]$ changes under infinitesimal predicate changes is zero.*

Theorem 2 *A monotonic constraint ϕ is stable under infinitesimal predicate changes iff $\mathrm{Vol}\{x \in R_{\mathtt{x}} | \tau(\phi)(x) = 0\} = 0$.*

Proof. We prove that for an arbitrary but fixed $(a_1, \ldots, a_n)$, $(a_1, \ldots, a_n) \in [\![\phi]\!]$ changes under infinitesimal predicate changes iff $\tau(\phi)(a_1, \ldots, a_n) = 0$. For this replace all inequalities in ϕ by $\geq$. By Lemma 2 this means that $(a_1, \ldots, a_n) \in [\![\phi]\!]$. But if we replace all inequalities by $>$, by Lemma 3 this means that $(a_1, \ldots, a_n) \notin [\![\phi]\!]$. So $(a_1, \ldots, a_n) \in [\![\phi]\!]$ is not stable under infinitesimal predicate changes. ■

Corollary 2 *A monotonic constraint is stable iff it is stable under infinitesimal predicate changes.*

Now we prove a stability condition that uses volume quantifiers [18, 2, 1, 22, 36, 26]. For this we assign semantics also to constraints containing quantifiers that are annotated with an additional non-negative real number v. The intuition is, that $[\![\exists_{\mathtt{v}} \mathtt{x}\ \phi]\!]$ is true iff the volume of the solution set of ϕ is greater than $[\![\mathtt{v}]\!]$ and $[\![\forall_{\mathtt{v}} \mathtt{x}\ \phi]\!]$ iff the volume of the complement of the solution set of ϕ is less or equal $[\![\mathtt{v}]\!]$.

Now let us denote by $Z(\phi)$ the constraint that is obtained from ϕ by replacing all quantifiers by volume quantifiers whose annotation is zero.

Theorem 3 *For every monotonic constraint ϕ in n free variables, for every $(a_1, \ldots, a_n)$ for which $\tau(\phi)(a_1, \ldots, a_n)$ is non-zero, $(a_1, \ldots, a_n) \in [\![\phi]\!]$ iff $(a_1, \ldots, a_n) \in [\![Z(\phi)]\!]$.*

Proof. W.l.o.g. we assume that ϕ has the form $(\mathtt{Q_1 x_1}) \ldots (\mathtt{Q_k x_k})\phi'$, where ϕ' is quantifier-free. We proceed by induction over the quantifiers $\mathtt{Q_k}, \mathtt{Q_{k-1}}, \ldots, \mathtt{Q_1}$. For the base case of a constraint without quantifiers there is nothing to prove.

Now assume that for all $(a_1, \ldots, a_n, b_1, \ldots, b_i)$ such that

$$\tau((\mathtt{Q_{i+1} x_{i+1}}) \ldots (\mathtt{Q_k x_k})\phi')(a_1, \ldots, a_n, b_1, \ldots, b_i)$$

is non-zero,

$$(a_1, \ldots, a_n, b_1, \ldots, b_i) \in [\![(\mathtt{Q_{i+1} x_{i+1}}) \ldots (\mathtt{Q_k x_k})\phi']\!] \text{ iff } (a_1, \ldots, a_n, b_1, \ldots, b_i) \in [\![Z((\mathtt{Q_{i+1} x_{i+1}}) \ldots (\mathtt{Q_k x_k})\phi')]\!]$$

We prove that for all $(a_1, \ldots, a_n, b_1, \ldots, b_{i-1})$ such that

$$\tau((\mathtt{Q_i x_i})(\mathtt{Q_{i+1} x_{i+1}}) \ldots (\mathtt{Q_k x_k})\phi')(a_1, \ldots, a_n, b_1, \ldots, b_{i-1})$$

is non-zero,

$$(a_1, \ldots, a_n, b_1, \ldots, b_{i-1}) \in [\![(\mathtt{Q_i x_i})(\mathtt{Q_{i+1} x_{i+1}}) \ldots (\mathtt{Q_k x_k})\phi']\!] \text{ iff}$$
$$(a_1, \ldots, a_n, b_1, \ldots, b_{i-1}) \in [\![Z((\mathtt{Q_i x_i})(\mathtt{Q_{i+1} x_{i+1}}) \ldots (\mathtt{Q_k x_k})\phi')]\!]$$

$\mathtt{Q_i} = \exists$: Let $(a_1, \ldots, a_n, b_1, \ldots, b_{i-1})$ be arbitrary but fixed.

If $\tau((\mathtt{Q_i x_i})(\mathtt{Q_{i+1} x_{i+1}}) \ldots (\mathtt{Q_k x_k})\phi')(a_1, \ldots, a_n, b_1, \ldots, b_{i-1}) > 0$, then by definition of τ and compactness of $R_{\mathtt{x_i}}$ there is a $b_i \in R_{\mathtt{x_i}}$ such that

$$\tau((\mathtt{Q_{i+1} x_{i+1}}) \ldots (\mathtt{Q_k x_k})\phi')(a_1, \ldots, a_n, b_1, \ldots, b_{i-1}, b_i) > 0.$$

By continuity, b_i has a neighborhood of non-zero volume such that for all b_i' in this neighborhood

$$\tau((\mathtt{Q_{i+1} x_{i+1}}) \ldots (\mathtt{Q_k x_k})\phi')(a_1, \ldots, a_n, b_1, \ldots, b_{i-1}, b_i') > 0,$$

and by the induction hypothesis

$$(a_1, \ldots, a_n, b_1, \ldots, b_{i-1}, b_i') \in [\![Z((\mathtt{Q_{i+1} x_{i+1}}) \ldots (Q_k x_k)\phi')]\!].$$

So by the definition of $\exists_0$,

$$(a_1, \ldots, a_n, b_1, \ldots, b_{i-1}) \in [\![Z((\mathtt{Q_i x_i})(\mathtt{Q_{i+1} x_{i+1}}) \ldots (\mathtt{Q_k x_k})\phi')]\!].$$

If $\tau((\mathtt{Q_i x_i})(\mathtt{Q_{i+1} x_{i+1}}) \ldots (\mathtt{Q_k x_k})\phi')(a_1, \ldots, a_n, b_1, \ldots, b_{i-1}) < 0$, then by definition of τ, for all $b_i \in R_{\mathtt{x_i}}$,

$$\tau((\mathtt{Q_{i+1} x_{i+1}}) \ldots (\mathtt{Q_k x_k})\phi')(a_1, \ldots, a_n, b_1, \ldots, b_{i-1}, b_i) < 0.$$

By the induction hypothesis, for all $b_i \in R_{\mathtt{x_i}}$,

$$(a_1, \ldots, a_n, b_1, \ldots, b_i) \notin [\![Z((\mathtt{Q_{i+1} x_{i+1}}) \ldots (Q_k x_k)\phi')]\!].$$

So by the definition of $\exists_0$,

$$(a_1, \ldots, a_n, b_1, \ldots, b_{i-1}) \notin [\![Z((\mathtt{Q_i x_i})(\mathtt{Q_{i+1} x_{i+1}}) \ldots (\mathtt{Q_k x_k})\phi')]\!].$$

$\mathtt{Q_i} = \forall$: Dual

■

Corollary 3 *For every stable monotonic constraint ϕ in n free variables, the volume of the $(a_1, \ldots, a_n)$ for which not $(a_1, \ldots, a_n) \in [\![\phi]\!]$ iff $(a_1, \ldots, a_n) \in [\![Z(\phi)]\!]$ is zero.*

Corollary 4 *For every stable closed monotonic constraint ϕ, $[\![\phi]\!] = [\![Z(\phi)]\!]$.*

This means that if $[\![\phi]\!] \neq [\![Z(\phi)]\!]$ then we know that that ϕ is unstable. However the other direction does not hold. For example the constraint $(\exists \mathtt{x})\ \mathtt{0} \geq \mathtt{0}$ is unstable although $[\![(\exists \mathtt{x})\ \mathtt{0} \geq \mathtt{0}]\!] = [\![(\exists_{\mathtt{0}} \mathtt{x})\ \mathtt{0} \geq \mathtt{0}]\!]$.

5 Different Forms of Stability

In this section we consider the following more general forms of perturbations:

- Perturbations of all the constants occurring in constraints (instead of only the zeros occurring on the right-hand side of atomic sub-constraints).
- Perturbations of identity functions denoted by the variables in a constraint.

Here there are several possibilities for choosing the perturbations of the identity functions:

- Since the same variable can occur several times in a constraint, we can perturb different occurrences of the same variable either by the same or by different perturbations.
- We can use continuous or discontinuous perturbations.

Observe that each type of perturbation results in a *perturbation function* $P : C \times \mathbb{R} \to 2^C$ such that $P(\phi, \delta)$ is the set of all results of δ-perturbations of ϕ. Each perturbation function P results in an according notion of stability using P.

Let us denote by P_0 the perturbation function used up to this section — it perturbs only the zeros on the right-hand sides of atomic sub-constraints. Using the additional types of perturbations described above we get the new perturbation functions $P_c, P_{c,v/dep/cont}, P_{c,v/dep/disc}, P_{c,v/indep/cont}, P_{c,v/indep/disc}$. For every constraint ϕ and positive real number δ we have the subset relation depicted in Figure 1 (where $\to$ corresponds to $\supseteq$).

$$
\begin{tikzcd}
 & P_{c/indep/disc}(\phi,\delta) \arrow[dl] \arrow[dr] & \\
P_{c/dep/disc}(\phi,\delta) \arrow[dr] & & P_{c/indep/cont}(\phi,\delta) \arrow[dl] \\
 & P_{c/dep/cont}(\phi,\delta) \arrow[d] & \\
 & P_c(\phi,\delta) \arrow[d] & \\
 & P_0(\phi,\delta) &
\end{tikzcd}
$$

Figure 1: Perturbation Functions

Lemma 7 *Let P_1, P_2 be perturbation functions. If for all $\delta \in \mathbb{R}$, $P_1(\phi, \delta) \supseteq P_2(\phi, \delta)$ then stability of ϕ using P_1 implies stability of ϕ using P_2.*

So the arrows in Figure 1 also are valid implications wrt. stability. But also the other direction is valid. For showing this observe that the degree of truth is stable under all perturbation functions under considerations. So we can adapt the right-to-left direction of the proof of Theorem 1 to assure that if a monotonic constraint is stable then it is also stable under general perturbations. Thus all perturbation functions result in equivalent notions of stability for monotonic constraints — using additional information about continuity of functions and equality of variables does not result in more stable constraints. The situation would change when considering variable perturbations without constant perturbation — but we do not discuss this case further in this paper.

6 Stability of General Constraints

Now we extend our considerations to constraints that are not monotonic. For this recall that one can view a quantified constraint as a mapping that takes the meaning of its constant symbols (i.e., real constants), and its variables (i.e., identity functions) and produces the according meaning of the constraint (i.e., solution set). If this mapping is the same for two constraints ϕ_1 and ϕ_2 then ϕ_1 is stable iff ϕ_2 is stable.

Now observe that for any constraint ϕ containing only the predicate symbols $<, \leq, >, \geq$ this mapping is the same for ϕ and its monotonic transformation $T(\phi)$. So we simply can extend the definition of degree of truth to such constraints and keep all the stability theorems in this paper.

Now consider constraints that contain equalities. For these the above mapping is changed by the monotonic transformation. But we can simulate every perturbation of such a general constraint on its monotonic transformation by assigning the same perturbation to the two variables/constants generated by the transformation of an equality. So for any perturbation function, there is an according subset that takes into account that certain variables/constants are the same, and, by Lemma 7, we get:

Lemma 8 *For every constraint ϕ, if $T(\phi)$ is stable wrt. a perturbation function P then also ϕ is stable wrt. P.*

For the other direction we study how one can influence the result of a monotonic transformation of an equality by discontinuous transformations:

Lemma 9 *For every constraint ϕ in n variables and $(a_1, \ldots, a_n)$ such that $\tau(T(\phi))(a_1, \ldots, a_n) = 0$, for every $\delta > 0$ there is a result ϕ' of a δ-perturbation of ϕ using $P_{c,v/dep/disc}$ such that $\tau(T(\phi'))(a_1, \ldots, a_n)$ is negative.*

Proof. We perturb all atomic sub-constraints of $T(\phi)$ that result from inequalities in ϕ according to Lemma 6.

Now consider a sub-constraint of $T(\phi)$ resulting from an equality $\theta = \mathtt{0}$ in ϕ. Its degree of truth is less or equal zero everywhere. Observe that for every perturbation of $\theta = \mathtt{0}$ there is an according perturbation of $T(\theta = \mathtt{0})$. Furthermore for every constraint of the form $\theta = \mathtt{0}$ such that $[\![\theta = \mathtt{0}]\!]$ has no zero, the resulting $\tau(T(\theta = \mathtt{0}))$ is negative. Thus we will perturb $\theta = \mathtt{0}$ in such a way that it has no zeros.

First let c be a constant that is less or equal δ, such that $[\![\theta = \mathtt{c}]\!]$ does not have infinitely many zeros. Now it is easy to find a discontinuous δ-perturbation of the variables in $\theta = \mathtt{c}$ that always jumps over the zero line, and thus has no zeros. ■

Corollary 5 *A constraint ϕ is stable using $P_{c,v/dep/disc}$ iff its transformation $T(\phi)$ is stable.*

For example a constraint of the form $\exists x\ \theta = 0$ is stable using $P_{c,v/dep/disc}$ iff $[\![\theta]\!]$ has no zero. Obviously the above holds also for $P_{c,v/indep/disc}$.

7 Conclusion

In this paper we have studied the behavior of quantified constraints under perturbation. We have shown that the solution set of a quantified constraint is stable under perturbation except for one (unstable) sign computation, we have provided two practically useful conditions for stability, and we have shown that, for constraints containing only inequality predicates, stability under perturbation is independent of whether we use information about the continuity of constraints and equality of variables.

Since the model used in this paper (and in any kind of numerical error analysis) is an abstraction, in rare cases a concrete method might even succeed to compute the solution set of an unstable constraint. For the author's method [25, 24] this can happen when certain floating point operations can be done exactly without rounding errors.

What can be done if one encounters an unstable quantified constraint? One possibility would be to compute the solution of a similar, but stable problem, that is, to regularize the problem. This idea has traditionally been applied to certain ill-posed problems in numerical mathematics [11], but is recently frequently used when applying numerical methods to discontinuous algebraic problems [4, 14, 29]. In our case one could use approximate quantifiers [26] instead of the classical exact quantifiers.

The degree of truth defined in this paper does not only provide a criterion for termination. One can also interpret it as a condition number — its size is a measure for the distance of an input constraint to an ill-posed input. For validated numerical methods the inverse of the condition number usually is a measure for the run-time, and also the inverse of the degree of truth provides a rough estimation for the run-time of numerical solvers. Further work will study the details.

A further observation is, that for certain functions (e.g., low degree polynomials) there are explicit formulas for the minimum and maximum. Thus it might be interesting to explicitly compute the degree of truth by substituting terms provided by such formulas, in a similar way as the method of quantifier elimination by substitution of parametrized test points [20, 38, 39].

The results in this paper are not applicable for checking continuous stability of constraints that contain equalities. This is also further work.

The author thanks Hoon Hong for the initial motivation for studying approximate quantified constraint solving, and Richard Liska for interesting example problems that gave the impetus to writing this paper.

References

[1] F. Bacchus. Lp, a logic for representing and reasoning with statistical knowledge. *Computational Intelligence*, 6:209–231, 1990.

[2] F. Bacchus. *Representing and Reasoning with Probabilistic Knowledge.* MIT Press, 1990.

[3] F. Benhamou and F. Goualard. Universally quantified interval constraints. In *Proceedings of the Sixth International Conference on Principles and Practice of Constraint Programming (CP'2000)*, Lecture Notes in Computer Science, Singapore, 2000. Springer Verlag.

[4] G. Bodnár, P. Pau, and J. Schicho. Exact real computation in computer algebra. Technical Report 00-33, RISC-Linz, 2000.

[5] B. F. Caviness and J. R. Johnson, editors. *Quantifier Elimination and Cylindrical Algebraic Decomposition.* Texts and Monographs in Symbolic Computation. Springer, 1998.

[6] G. E. Collins. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In *Second GI Conf. Automata Theory and Formal Languages*, volume 33 of *Lecture Notes in Computer Science*, pages 134–183. Springer- Verlag, Berlin, 1975. Also in [5].

[7] G. E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *Journal of Symbolic Computation*, 12:299–328, 1991. Also in [5].

[8] P. Dorato. Quantified multivariate polynomial inequalities. *IEEE Control Systems Magazine*, pages 48–58, October 2000.

[9] P. Dorato, W. Yang, and C. Abdallah. Robust multi-objective feedback design by quantifier elimination. *Journal of Symbolic Computation*, 24:153–159, 1997.

[10] H.-D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic*. Springer Verlag, 1984.

[11] H. W. Engl. Regularization methods for the stable solution of inverse problems. *Surv. Math. Ind*, 3:71–143, 1993.

[12] J. Garloff and B. Graf. Solving strict polynomial inequalities by Bernstein expansion. In N. Munro, editor, *The Use of Symbolic Methods in Control System Analysis and Design*, pages 339–352. The Institution of Electrical Engineers (IEE), London, 1999.

[13] H. Hong, R. Liska, and S. Steinberg. Testing stability by quantifier elimination. *Journal of Symbolic Comptutation*, 24(2):161–187, 1997.

[14] Y. Huang, H. J. Stetter, W. Wu, and L. Zhi. Pseudofactors of multivariate polynomials. In *Proc. ISSAC'00*, pages 160–167. ACM Press, 2000.

[15] N. I. Ioakimidis. Quantifier elimination in applied mechanics problems with cylindrical algebraic decomposition. *Int. J. Solids Structures*, 34(30):4037–4070, 1997.

[16] L. Jaulin and É. Walter. Guaranteed tuning, with application to robust control and motion planning. *Automatica*, 32(8):1217–1221, 1996.

[17] M. Jirstrand. Nonlinear control system design by quantifier elimination. *Journal of Symbolic Computation*, 24(2):137–152, 1997.

[18] H. J. Keisler. Probability quantifiers. In J. Barwise and S. Feferman, editors, *Model Theoretic Logics*, chapter XIV. Springer Verlag, New York, 1985.

[19] R. Liska and S. Steinberg. Applying quantifier elimination to stability analysis of difference schemes. *Computer Journal*, 36(5):497–593, 1993.

[20] R. Loos and V. Weispfenning. Applying linear quantifier elimination. *The Computer Journal*, 36(5):450–462, 1993.

[21] S. Malan, M. Milanese, and M. Taragna. Robust analysis and design of control systems using interval arithmetic. *Automatica*, 33(7):1363–1372, 1997.

[22] A. Mostowski. On a generalization of quantifiers. *Fundamenta Mathematicae*, 44:12–36, 1957.

[23] M. B. Pour-El and J. I. Richards. *Computability in Analysis and Physics*. Springer-Verlag, 1989.

[24] S. Ratschan. Approximate quantified constraint solving (AQCS). `http://www.risc.uni-linz.ac.at/research/software/AQCS`, 2000. Software package.

[25] S. Ratschan. Approximate quantified constraint solving by cylindrical box decomposition. Submitted for Publication, 2000.

[26] S. Ratschan. Convergence of quantified constraint solving by approximate quantifiers. Technical Report 00-23, Research Institute for Symbolic Computation (RISC) - Linz, 2000. Submitted for Publication.

[27] S. Ratschan. Uncertainty propagation in heterogeneous algebras for approximate quantified constraint solving. *Journal of Universal Computer Science*, 6(9), 2000.

[28] N. Rescher. *Many-valued Logic*. McGraw-Hill, New York, 1969.

[29] A. Schönhage. Quasi-GCD computations. *J. Complexity 1*, pages 118–137, 1985.

[30] S. P. Shary. A new approach to the analysis of static systems under interval uncertainty. In G. Alefeld, A. Frommer, and B. Lang, editors, *Proceedings of the International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN-95)*, pages 118–132. Akademie Verlag, 1995.

[31] S. P. Shary. Algebraic approach to the interval linear static identification, tolerance, and control problems, or one more application of Kaucher arithmetic. *Reliable Computing*, 2(1):3–33, 1996.

[32] S. P. Shary. Outer estimation of generalized solution sets to interval linear systems. *Reliable Computing*, 5:323–335, 1999.

[33] T. Sturm. Reasoning over networks by symbolic methods. *Applicable Algebra in Engineering Communication and Computing*, 10(1):79–96, September 1999.

[34] A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. Univ. of California Press, Berkeley, 1951. Also in [5].

[35] A. Urquhart. Many-valued logic. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic, Vol. III: Alternatives in Classical Logic*, chapter III.2, pages 71–116. D. Reidel Publishing Company, 1986.

[36] J. Väänänen. Generalized quantifiers. *Bulletin of the European Association for Theoretical Computer Science*, 1997.

[37] K. Weihrauch. *Introduction to Computable Analysis.* Texts in Theoretical Computer Science. Springer Verlag, 2000.

[38] V. Weispfenning. Quantifier elimination for real algebra — the cubic case. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation in Oxford*, pages 258–263. ACM Press, July 1994.

[39] V. Weispfenning. Quantifier elimination for real algebra - the quadratic case and beyond. *Applicable Algebra in Engineering, Communication and Computing*, 8(2):85–101, 1997.

[40] V. Weispfenning. Simulation and optimization by quantifier elimination. *Journal of Symbolic Computation*, 24(2), 1997.