Reverse Mathematics Is Computable for Interval Computations

Martine Ceberio
University of Texas at El Paso, mceberio@utep.edu

Olga Kosheleva
University of Texas at El Paso, olgak@utep.edu

Vladik Kreinovich
University of Texas at El Paso, vladik@utep.edu

Reverse Mathematics Is Computable for Interval Computations

Martine Ceberio[1], Olga Kosheleva[2], and Vladik Kreinovich[1]

Department of [1]Computer Science and [2]Teacher Education
University of Texas at El Paso
500 W. University, El Paso, TX 79968, USA
mceberio@utep.edu, olgak@utep.edu, vladik@utep.edu

Abstract. For systems of equations and/or inequalities under interval uncertainty, interval computations usually provide us with a box whose all points satisfy this system. Reverse mathematics means finding necessary and sufficient conditions, i.e., in this case, describing the set of *all* the points that satisfy the given system. In this paper, we show that while we cannot always exactly describe this set, it is possible to have a general algorithm that, given $\varepsilon > 0$, provides an ε-approximation to the desired solution set.

1 Formulation of the Problem

What is reverse mathematics. In mathematics, whenever a new theorem is proven, often, it later turns out that this same conclusion can be proven under weaker conditions.

For example, first, it was proven that if for a continuous function $f(x)$ from real numbers to real numbers, we have $f(a+b) = f(a) + f(b)$ for all a and b, then this function $f(x)$ is linear , i.e., $f(a) = k \cdot a$ for some k, Later on, it turned out that the same is true not only for continuous functions, but also for all measurable functions.

Because of this phenomenon, every time a new result is proven, researchers start analyzing whether this result can be proven under weaker conditions. I the past, usually, weaker and weaker conditions were found. Lately, however, in some problems, it has become possible to find the weakest possible conditions under which the given conclusion is true.

From the logical viewpoint, the fact that the condition A in the implication $A \Rightarrow B$ cannot be weakened means that A is equivalent to B, i.e., that also $B \Rightarrow A$ – in other words, that we can *reverse* the implication. Because of this, the search for such weakest possible condition is known as the *reverse mathematics*; see, e.g., [6].

What does reverse mathematics means for interval computations? The main problem of interval computations (see, e.g., [3–5]) is, given an algorithm $f(x_1, \ldots, x_n)$ and ranges $[\underline{x}_i, \overline{x}_i]$, to find the range $[\underline{y}, \overline{y}]$ of possible values of $y = f(x_1, \ldots, x_n)$ when $x_i \in [\underline{x}_i, \overline{x}_i]$ for all i.

In practice, this can be used, e.g., to check that under all possible values of the parameters x_i from the corresponding intervals, the system is stable, when stability is described by an inequality $f(x_1, \ldots, x_n) \le y_0$ for some value y_0. Once we know the range, this checking is equivalent to simply checking whether $\overline{y} \le y_0$.

Similarly, if $f(x_1, \ldots, x_n)$ is the amount of potentially polluting chemical released by a plant under conditions x_i, checking whether the level of this chemical never exceeds the desired threshold y_0 is also equivalent to simply checking whether $y \le y_0$.

In addition to knowing that $x_i \in [\underline{x}_i, \overline{x}_i]$, we often have additional constraints on the values x_i, which make the problem more complex.

We may also need to check more complex conditions. For example, in solving system of equations under interval uncertainty, we are often interested in finding all the values $x = (x_1, \ldots, x_n)$ for which, for all possible values $a_1, \ldots, a_m$ from the corresponding intervals, there exist appropriate controls $c_1, \ldots, c_p$ from the given intervals for which a desired inequality $f(x, a, c) \le y_0$ holds. Since all physical quantities are bounded, we can safely assume that all variables in the quantifiers are bounded.

In general, we have a property $P(x_1, \ldots, x_n)$ which can be either a simple inequality like $f(x_1, \ldots, x_n) \le y_0$, or it can be a complex formula obtained from simply inequalities by using logical connectives "and" (&) "or" ($\vee$), and "not" ($\neg$), and quantifiers $\forall x$ and $\exists x$ over real numbers. By using interval methods, we find a box $B = [\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_n, \overline{x}_n]$ for which the desired property $P(x)$ holds for all points $x \in B$. In this context, reverse mathematics means trying to find not just this box, but also the whole set of all the tuples x for which the property $P(x)$ holds.

What we do in this paper. In this paper, we show that such a set can be indeed computed – maybe not exactly, but at least with any possible accuracy.

2 Definitions and the Main Result

Computable numbers, sets, etc.: reminder. According to computable mathematics (see, e.g., [1, 7]), a real number x is *computable* if there exists an algorithm that, given a natural number n, generates a rational number r_n for which $|r_n - x| \le 2^{-n}$. A tuple of computable number is called a *computable tuple.*

A bounded set S is called *computable* if there exists an algorithm, that given a natural number n, generates a finite list S_n of computable tuples for which $d_H(S, S_n) \le 2^{-n}$, where $d_H(A, B)$ is the Hausdorff distance: the smallest $\varepsilon > 0$ for which:

- every element $a \in A$ is ε-close to some element $b \in B$, and
- every element $b \in B$ is ε-close to some element $a \in A$.

A *computable function* is a function $f(x_1, \ldots, x_n)$ for which two algorithms exist:

– the main algorithm that, given rational values $r_1, \ldots, r_n$, returns a computable number $f(r_1, \ldots, r_n)$, and
– an auxiliary algorithm that, given a rational number $\varepsilon > 0$, returns a rational number $\delta > 0$ for which $d(x, x') \le \delta$ implies $d(f(x), f(x')) \le \varepsilon$.

Most arithmetic and elementary functions are everywhere computable. (The only exceptions are discontinuous functions like sign or tangent.)

It is known (and it can be easily proven):

– that min and max are computable,
– that composition of two computable functions is computable, and
– that the maximum and minimum of a computable function over a computable set are also computable.

It is also known that for every computable function f on a computable set S, and for every two values $y^- < y^+$ for which $\min\limits_{x \in S} f(x) < y^-$, there exists a value $y_0 \in [y^-, y^+]$ for which the set $\{x : f(x) \le y_0\}$ is computable [1].

There are also known negative results: e.g.,

– that it is not possible, given two computable numbers x and x', to check whether $x \le x'$, and,
– as a consequence, that it is, in general, not possible, given a computable function f and a number y, to produce a computable set $\{x : f(x) \le y_0\}$ – otherwise, for a constant $f(x) = c$, we would get an algorithm for checking whether $c \le y_0$.

Definition 1. *Let $v_1, \ldots$ be real-valued variables. For each of these variables, we have bounds $\underline{V}_i \le v_i \le \overline{V}_i$.*

– *By a* term, *we mean an expression of the type $f(v_{i_1}, \ldots, v_{i_m})$, where f is a computable function and v_i are given variables.*
– *By an* elementary formula, *we means an expression of one of the types $t_1 < t_2$, $t_1 \le t_2$, or $t_1 = t_2$, where t_1 and t_2 are terms.*
– *By a* property $P(x_1, \ldots, x_n)$, *we mean any formula with free variables $x_1, \ldots, x_n$ which is obtained from elementary formulas by using logical connectives &, $\vee$, $\neg$, and quantifiers $\forall v_{i_{\in [\underline{V}_i, \overline{V}_i]}}$ and $\exists v_{i_{\in [\underline{V}_i, \overline{V}_i]}}$.*

Comment. To simplify the further description, let us represent each equality $t_1 = t_2$ and two inequalities $t_1 \le t_1$ and $t_2 \le t_1$.

Definition 2. *Let $\varepsilon > 0$ be a real number.*

– *We say that elementary formulas $t_1 \le t_2$ (or $t_1 < t_2$) and $t_1 \le t_2 + \varepsilon'$ (or $t_1 < t_2 + \varepsilon'$) are ε-*close *if $|\varepsilon'| \le \varepsilon$.*
– *We say that the formulas $P(x_1, \ldots, x_n)$ and $P(x_1', \ldots, x_n')$ are ε-*close *if P' is the result of replacing, in the formula P, each elementary formula with an ε-close one.*

Comment. In practice, all the values are measured with some accuracy. Thus, if ε is sufficiently small, the two ε-close elementary formulas are practically indistinguishable – and thus, in general, ε-close properties are indistinguishable as well.

Proposition 1. *Let $P(x_1,\ldots,x_n)$ be a property which is satisfied for all the tuples x from a given box. Then, based on the property and the box, we can compute the set $\{x : P'(x)\}$ for some property P' which is ε-close to P.*

Comment. This result can be further strengthened.

Proposition 2. *Let $P(x_1,\ldots,x_n)$ be a property which is satisfied for all the tuples x from a given box. Then, based on the property and the box, we can compute the set $S = \{x : P'(x)\}$ for some property P' which is ε-close to P and for which $S'' \stackrel{\text{def}}{=} \{x : P''(x)\} \subseteq S$ for all properties P'' which are $(\varepsilon/2)$-close to P.*

3 Proof

Proof of Proposition 1. First, let us transform the original property $P(x)$ into a prenex normal form, i.e., into the form (see, e.g., [2]) in which we first have quantities, and then the quantifier-free part. Indeed, if we have a logical connective outside quantifires, we can move the quantifier out by using the equivalent transformations $\neg\forall xP \to \exists x\neg P$, $\forall xP \vee Q \to \forall x(P \vee Q)$, $\forall xP \,\&\, Q \to \forall x(P \,\&\, Q)$, $\neg\exists xP \to \forall x\neg P$, $\exists xP \vee Q \to \exists x(P \vee Q)$, and $\exists xP \,\&\, Q \to \exists x(P \,\&\, Q)$.

Then, we can use de Morgan rules

$$\neg(A \,\&\, B) \to (\neg A) \vee (\neg B) \text{ and } \neg(A \vee B) \to (\neg A) \,\&\, (\neg B)$$

to move all negations inside. When applied to an elementary formulas $t_1 \le t_2$ or $t_1 < t_2$, negation simply means a change in the inequality sign: $\neg(t_1 \le t_2) \to t_2 < t_1$ and $\neg(t_1 < t_2) \to t_2 \le t_1$.

In the resulting formula, let us replace all $<$ with $\le$ – this will not change ε-closeness. Let us now describe the resulting property P_0 in the equivalent form $F(x_1,\ldots,x_n) \le 0$, for some computable function F.

– Each elementary formula $t_1 \le t_2$ can be equivalently reformulated as

$$t_1 - t_2 \le 0.$$

– Each formula $(F_1 \le 0) \vee (F_2 \le 0)$ can be equivalently reformulated as

$$\min(F_1, F_2) \le 0.$$

– Each formula $(F_1 \le 0) \,\&\, (F_2 \le 0)$ can be equivalently reformulated as

$$\max(F_1, F_2) \le 0.$$

– Each formula $\exists v_{i_{\in[\underline{V}_i,\overline{V}_i]}} F(v_i,\ldots)\le 0$ can be equivalently reformulated as

$$\min_{v_i\in[\underline{V}_i,\overline{V}_i]} F(v_i,\ldots)\le 0.$$

– Each formula $\forall v_{i_{\in[\underline{V}_i,\overline{V}_i]}} F(v_i,\ldots)\le 0$ can be equivalently reformulated as

$$\max_{v_i\in[\underline{V}_i,\overline{V}_i]} F(v_i,\ldots)\le 0.$$

For the resulting function $F(x_1,\ldots,x_n)$, for $y^-=0$ and $y^+=\varepsilon$, there exists a number $\varepsilon_0\in(0,\varepsilon)$ for which the set $S_0\stackrel{\text{def}}{=}\{x:F(x)\le\varepsilon_0\}$ is computable.

The corresponding inequality $F(x)\le\varepsilon_0$ is equivalent to $F'(x)\le 0$, where $F'(x)\stackrel{\text{def}}{=}F(x)-\varepsilon_0$. This inequality can be obtained if we replace, in the formula P_0, each elementary formula $t_1\le t_2$ with a formula $t_1\le t_2+\varepsilon_0$. Since $\varepsilon_0<\varepsilon$, this transformation keeps all elementary formulas ε-close to the original ones. So, the resulting formula P'_0 is ε-close to the formula P_0 and we have $S_0=\{x:P'_0(x)\}$.

When we went from P to P_0, all we did was changed the sign of some inequalities. This, in turn, can be obtained by appropriately changing the elementary formulas from the original property P to ε-close ones. Thus, indeed, the set S_0 can be represented as $S_0=\{x:P'(x)\}$ for the resulting formula P' which is ε-close to P.

Proof of Proposition 2 is similar, except that instead of $y^-=0$ we take $y^-=\varepsilon/2$. Then, for every property P'' which is $(\varepsilon/2)$-close to P, on each level of designing a function $F(x_1,\ldots,x_n)$, we will have $F\le F''+\varepsilon/2$ for the function F'' corresponding to the property P''. Thus, at the end, we conclude that $F\le F''+\varepsilon/2$ and, since now $\varepsilon/2<\varepsilon_0$, we conclude that $F(x)\le F''(x)+\varepsilon_0$ and $F'(x)=F(x)-\varepsilon_0\le F''(x)$. Thus. $F''(x)\le 0$ implies that $F'(x)\le 0$. So, indeed, $P''\subseteq S'$. The proposition is proven.

Acknowledgments

This work was supported in part by the US National Science Foundation grant HRD-1242122.

References

1. E. Bishop, *Foundations of Constructive Analysis*, McGraw-Hill, New York, 1967.
2. P. Hinman, *Fundamentals of Mathematical Logic*, A. K. Peters, Natick, Massachusetts, 2005.
3. L. Jaulin, M. Kiefer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control, and Robotics*, Springer, London, 2001.
4. G. Mayer, *Interval Analysis and Automatic Result Verification*, de Gruyter, Berlin, 2017.

5. R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.
6. J. Stillwell, *Reverse Mathematics: Proofs from the Inside Out*, Princeton University Press, Princeton, New Jersey, 2018.
7. K. Weihrauch, *Computable Analysis*, Springer Verlag, Berlin, 2000.