Applied Mathematical Sciences, Vol. 3, 2009, no. 47, 2343 - 2347

Any Statement Can Be Generalized so that it Becomes Trivial: A Simple Formalization of D. K. Faddeev's Belief

Vladik Kreinovich

Department of Computer Science
University of Texas at El Paso, 500 W. University
El Paso, TX 79968, USA
email vladik@utep.edu

Abstract

In his unpublished lectures on general algebra, a well-known algebraist D. K. Faddeev expressed a belief that every true mathematical statement can be generalized in such a way that it becomes trivial. To the best of our knowledge, this belief has never been formalized before. In this short paper, we provide a simple formalization (and proof) of this belief.

Mathematics Subject Classification: 03A05

Keywords: power of generalization, trivial statements

1 D. K. Faddeev's Belief

In the 1970s, I had a privilege of attending several lectures on general algebra given by a well-known Russian algebraist Dmitry Konstantinovich Faddeev [1, 5]. One of his goals was to help mathematicians and computer scientists appreciate the usefulness of algebraic methods. He started with several examples of specific mathematical and computational problems about matrices, operations, functions, etc., and showed how these problems become easier to solve when reformulated in pure algebraic terms, in terms of the corresponding algebraic structures (groups, rings, fields, algebras, etc.).

Based on these examples, he then expressed a general belief that any true mathematical statement can be generalized in such a way that it becomes practically trivial.

For D. K. Faddeev and his students, this belief justified a useful heuristic: if a problem turns out to be too hard to solve, let us try to find a generalization. However, to the best of our knowledge, this heuristic belief has never before been formalized.

In this paper, we provide a simple formalization of D. K. Faddeev's belief, a formalization which makes it a provable theorem.

2 Towards Formalization

Let us start with a statement F. We believe that this statement is true, and that it is provable in some formal system S (e.g., in ZF set theory). Our objective is to looks for such a proof.

In these terms, D. K. Faddeev's belief states that it is possible to find easy-to-prove ("trivial", in mathematical terms) more general statement from which F follows. Intuitively, the existence of such a "more general statement" means the following.

3 For This Formalization, We Need a General Class of Objects

First, we need to describe a general class of objects t (groups, rings, fields, etc.); such objects are described by a property $P(t)$.

Comment. In practice, we usually have several different conditions $P_1(t), \ldots, P_k(t)$ which characterize a given type of structures. In this case, we take their conjunction $P_1(t)\,\&\,\ldots\,\&\,P_k(t)$ as the desired property $P(t)$.

4 We Need a General Result

Second, we need to have an easy-to-prove general theorem that every object t of the above type has some general property. We will denote this general property by $G(t)$, so the desired formula takes the form $\forall t\,(P(t) \to G(t))$.

5 The Original Statement Must Easily Follow from a Particular Case of the General Result

Third, we need to have an easy proof that the original statement F follows from $G(t_0)$ for an appropriate object t_0.

This object t_0 must satisfy the property P, and we must make sure that the corresponding statement $P(t_0)$ is also easy to prove.

6 What is Trivial: Towards Formalization

To complete our formalization, we need to explain what "trivial" (easy-to-prove) means. Here is where algorithms actually come in (usually, rather implicitly).

Intuitively, a class of statements is "trivial" if, given a statement from this class, we can easily tell whether this statement is true or false. This "easily tell" cannot rely on ingenious ideas: otherwise, it would not be trivial. For example, solving linear equations is trivial, solving quadratic equations is trivial. In other words, this solution must come from an *algorithm* – and this algorithm has to be easy-to-apply.

For example, it is known that there is an algorithm which decides all first order formulas of the theory of real numbers with equality, inequality, addition, and multiplication; see, e.g., [2, 7, 8]. However, since this algorithm is non-trivial (requires doubly exponential time), mathematicians rarely call the corresponding problems trivial.

We will say that an algorithm is trivial, e.g., if it requires linear time, i.e., time bounded by a linear function of the length n of the input formula: $t(n) \le C \cdot n$. Of course, strictly speaking, a linear-time algorithm can require time $t(n) = 10^{40} \cdot n$, in which case it is linear-time but not practically useful (and clearly not trivial). However, as we will see from the proof, in our case, the corresponding linear-time algorithm will indeed be trivial.

Now, we are ready for the formal definitions.

Definition 6.1

- *We say that an algorithm U is an* easy-proof algorithm *if it always finishes in linear time returning "true", "false", or "unknown", with the following two properties:*
  - *if $U(F)$ returns "true", then F is a provable formula;*
  - *if $U(F)$ returns "false", then F is a negation of the provable formula.*
- *We say that a formula F is U-*trivial *if $U(F)$ returns "true".*

Definition 6.2 *Let U be an easy-proof algorithm. We say that a formula F is U-*generalizable to be trivial *if there exist an object t_0 and formulas $P(t)$ and $G(t)$ for which the following three statements are U-trivial: $\forall t\,(P(t) \to G(t))$, $P(t_0)$, and $G(t_0) \to F$.*

Proposition 6.3 *There exists an easy-proof algorithm U for which every formula F is provable if and only if it is U-generalizable to be trivial.*

7 Proof

1°. If F is U-generalizable to be trivial, this means that the above three statements are provable. From $\forall t\,(P(t) \to G(t))$ and $P(t_0)$, we conclude that $G(t_0)$ is provable. From this and from the fact that the implication $G(t_0) \to F$ is provable, we conclude that F is provable.

2°. Vice versa, let us assume that F is provable, i.e., that there exists a formal step-by-step proof t_0 of the statement F. Then:

- for every t, we denote the statement that t is a proof by $P(t)$;
- for every proof t, we denote the statement which is proven in this proof by $G(t)$; and
- we denote the given proof of F by t_0.

Let us show that this selection makes the formula F U-trivial.

2.1°. In a formal step-by-step proof, we start with axioms of the given formal system, and we apply rules to transform previously proven statements into new ones (and the proof must contain detailed explanations of what exactly rule we apply and how).

Given a text, it is easy to check whether this text is indeed a formal step-by-step proof: we just need to check that the first formulas are indeed axioms, and that every consequent formula is indeed obtained from the previous ones by following the rule claimed in this text.

This proof-checking procedure is easy, so checking $P(t_0)$ is easy: it can be definitely done in linear time.

2.2°. The fact $\forall t\,(P(t) \to G(t))$ that every proof leads to a correct result is also easy to prove, in constant time (e.g., by induction over the length of the proof).

2.3°. Finally, the fact that $P(t_0)$ implies F follows from the fact that t_0 is actually the proof of F.

The proposition is proven.

8 Discussion

First, it is important to remember that provability in a theory usually cannot be described within the same theory, so the above generalization requires a theory which is logically stronger than the original one; see, e.g., [3, 4, 6]. This is OK, since a generalization is, in general, stronger than the original theory.

Second, our objective was to formalize and verify D. K. Faddeev's belief. So now, we have a new justification of the natural heuristic originated by this belief: if a statement is too hard to proof, try to generalize it.

It is worth mentioning that while heuristically, this idea is helpful, theoretically – as we can see from the above simple proof – finding a generalization is as difficult as proving the original statement.

ACKNOWLEDGEMENTS. This work was supported in part by the Max Planck Institut für Mathematik, by the NSF grants EAR-0225670, EIA-0080940, HRD-0734825, by Grant 1 T36 GM078000-01 from the National Institutes of Health, by Texas Department of Transportation grant No. 0-5453, and by the Japan Advanced Institute of Science and Technology (JAIST) International Joint Research Grant 2006-08.

The author is thankful to Dima Grigoriev and to the participants of the Conference on the Methods of Proof Theory in Mathematics, Bonn, June 4–10, 2007, for valuable discussions.

References

[1] A.D. Aleksandrov, M.I. Bashmakov, Z.I. Borevich, V.N. Kublanovskaya, M.S. Nikulin, A.I. Skopin, and A.V. Yakovlev, Dmitry Konstantinovich Faddeev, *Russian Math. Surveys*, 1989, Vol. 44, No. 3, pp. 223–231.

[2] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in real algebraic geometry*, Springer-Verlag, Berlin, 2006.

[3] M. Ben-Ari, *Mathematical Logic for Computer Science*, Springer Verlag, Berlin, 2001.

[4] H.B. Enderton, *A Mathematical Introduction to Logic*, Academic Press, 2001.

[5] D.K. Faddeev and V.N. Faddeeeva, *Computational methods of linear algebra*, W. H. Freeman, 1963.

[6] E. Mendelson, *Introduction to Mathematical Logic*, Chapman & Hall/CRC, 1997.

[7] B. Mishra, "Computational real algebraic geometry", in: *Handbook on Discreet and Computational Geometry*, CRC Press, 1997.

[8] A. Tarski, *A decision method for elementary algebra and geometry*, 2nd ed., Berkeley and Los Angeles, 1951.

Received: March, 2009