Departmental Technical Reports (CS)   Department of Computer Science

11-2015

On the Importance of Duality and Multi-Ality In Mathematics Education

Mourat Tchoshanov
University of Texas at El Paso, mouratt@utep.edu

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

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

ON THE IMPORTANCE OF DUALITY AND MULTI-ALITY IN MATHEMATICS EDUCATION

Tchoshanov, Mourat, PhD, Professor
Kosheleva, Olga, PhD, Associate Professor
Kreinovich, Vladik, PhD, Professor
mouratt@utep.edu, olgak@utep.edu , vladik@utep.edu

Abstract: For each mathematical object, there are usually several different equivalent representations: for example, a spatial object can be represented either in geometric terms, or by a function that describes its shape. The need for several representations comes from the fact that each of these representations is useful in solving some problems for which the use of other representations is less helpful. Thus, the more representations a student knows, the more capable this student is of solving mathematical problems. In this paper, we propose a general formal description of the corresponding notion of duality (and, more generally, "multi-ality"), and we explain the importance of duality and multi-ality in mathematics education.
Keywords: duality, multi-ality, objects vs. processes, NP-hard problems

Duality and multi-ality: examples. Many mathematical objects allow several equivalent representations. Let us give several examples; for more examples, see, e.g., [1].

A definable real number can be represented either by its definition, or by a corresponding infinite decimal fraction. For example, a fraction 1 / 3 can be naturally interpreted as one third of a pie, one third of an interval, etc. On the other hand, the same real number 1 / 3 can be equivalently represented by an infinite decimal fraction:

$$1 / 3 = 0.33333…$$

Similarly, the square root of two can be naturally interpreted in geometric terms – e.g., as the length of the diagonal of the unit square. On the other hand, this same number can be also represented by an infinite decimal fraction 1.41421356237…

Both representations are useful:
- If we want to *prove* results about the corresponding real numbers, then it is more advantageous to use their definitions. For example, if we want to prove that the fourth power of the square root of 2 is equal to 4, we can easily do it by using the definition of the square root. However, if, instead, we use an infinite decimal expansion, then, since in practice, we can only process finitely many digits, we will have only an approximate equality – so, no matter how many digits we take, we will never be sure that the result is indeed equal to 4.
- On the other hand, if we want to use the corresponding numbers in *computations*, then the decimal fraction representation is much more appropriate. Indeed, it is not a priori clear how, based on the definition of the square root -- as the number x for which $x^2 = 2$ -- we can compute x, and even if we know how to compute x, we still need to spend some time performing these auxiliary computations. On the other hand, if we use a decimal fraction representation, then computing x with any given accuracy is straightforward: we just take the corresponding digits of this expansion.

Similarly, when we solve geometric problems, we have two possible representations for the corresponding spatial objects:
- We can represent the spatial object as a combination of several standard geometric shapes, such as lines, circles, ellipses, etc.
- Alternatively, we can represent a spatial object by the function that describes its shape.

For example, a unit circle can be represented by the algebraic equation $x^2 + y^2 = 1$ that describes its shape.

Both representations are useful:
- A geometric representation usually leads to nice and succinct results about areas, volumes, intersections, etc.
- On the other hand, the purely geometric approach requires ingenuity even when solving simple problems. In contrast, the algebraic representation (while often more cumbersome) allows straightforward algorithms that help us solve many problems in a routine way – and sometimes even problems for which no purely geometric solution is known.

For example, sometimes, we have three lines that intersect in a single point: e.g., three heights, three medians, or three bisectors in a triangle. For each of these cases, the geometric proof is simple but requires a lot of ingenuity, while an algebraic approach provides us with an algorithmic solution – namely, the famous logician Alfred Tarski came up with an algebraic algorithm for solving all problems of elementary geometry; see, e.g., [2, 3].

Similar "duality" can be observed beyond elementary mathematics. For example, in solving partial differential equations, it is often beneficial to use different coordinate systems; see, e.g., [4]. For example, in the analysis of rotation-invariant equations, it is often advantageous to use radial coordinates instead of the usual Cartesian ones.

Another important example is signal processing. We can view each signal the way we observe it, as a sequence of values $x(t)$ observed at different moments of time. Sometimes, however, it is beneficial to use a different representation – as a linear combinations of sinusoids with different frequency.

This representation dates back to Isaac Newton, who famously showed that, by placing a prism in the path of a light ray, we can decompose every light into lights of different basic colors. It turned out that, in mathematical terns, each basic-color light is a sinusoid, with frequency representing color. So, the possibility of such a decomposition means that any signal can be represented as a linear combination of sinusoids.

This alternative representation --- developed by a French mathematician Jean-Baptiste Fourier – is indeed very helpful in solving linear differential equations, in filtering, and in signal (and image) processing in general.

Yet another example comes from the mathematical analysis of quantum phenomena. It is known that in quantum physics there are two possible representations of quantum objects:
- We can view the elementary quantum object (electron, photon, etc.) as a particle.
- Alternatively, we can view this same object as a wave.

Both representations are useful:
- For example, in the analysis of photo-effect (and, more generally, to explain why certain processes lead to light of a certain color, which is the basis of spectroscopy), it is useful to treat light as consisting of particles (photons).
- On the other hand, in the analysis of light propagation, interference, diffraction, etc., it is more convenient to view light as a wave.

It is interesting to mention that historically, this duality led to the appearance of Schrodinger equations, the main mathematical model of quantum physics [4]. Specifically, after Louis de Broglie came up with an informal idea of a dual particle-wave character of quantum objects, Einstein – who was not very fond of quantum ideas -- advised Schroedinger to formalize de Broglie's idea and to show that the resulting equations do not work. To Scroedinger's (and

Einstein's) big surprise, this idea worked perfectly: Schroedinger equations describe all the observed quantum phenomena perfectly well!

Pedagogical problem. In many areas of mathematics and applications of mathematics, it is beneficial to know several equivalent representations of the corresponding mathematical object – since in different problems, different representations can lead to a solution. The problem is that in many cases, while the students are familiar with each of these representations, they have trouble translating from one representation to another – and, as a result, they are not able to solve problems for which such a translation can be helpful.

For many students, this inability to translate emerges on the very basic level, of translation between fractions and their decimal representations. For example, it is typical for many students not to realize that the infinite decimal expansions 0.4999… and 0.5000… represent the same fraction: $1/2$.

What we do in this paper. At first glance, this may sound like a minor problem. However, we show that this impression is somewhat misleading. Specifically, our analysis shows that the ability to translate is crucial for the students' ability to solve mathematical problem.

To start with our analysis, we need to describe the corresponding notions of duality and multi-ality in precise terms.

Towards a precise definition of duality and multi-ality. To come up with such a precise definition, we need to recall what is a *problem* and what it means to be able to *solve* a problem.

Ability to solve a problem means an algorithm. In precise terms, the ability to solve problems from a certain class means that the students have mastered an *algorithm* for solving this problem.

There are many precise definitions of an algorithm. In a nutshell, an algorithm is a well-defined sequence of steps that is always guaranteed to stop and to produce the correct result; see, e.g., [5].

Not all algorithms are practical. It is important to emphasize that the above definition of an algorithm lack an importance nuance: for an algorithm to be practical, this algorithm needs to stop not just in principle, it needs to stop in reasonable time. In theoretical computer science, this additional requirement is captured by the notion of a *feasible* algorithm.

Most practical algorithm require a number of computational steps which is bounded by a polynomial of the length n of the input. For example, we have algorithms that require n * log(n) steps, we have algorithms that require n^2 steps, n^3 steps, n^4 steps, etc. Such algorithms are known as *polynomial-time* algorithms.

On the other hand, some algorithms require exponential time, e.g., 2^n steps. One can easily check that even for reasonable sizes n, e.g., for n = 1000, the resulting number of steps exceeds the lifetime of the Universe, so these algorithms are clearly not practical.

To capture this difference, theoretical computer science defines an algorithm to be feasible if it is polynomial-time.

Comment. It is important to mention that this definition of feasible algorithms is not perfect. For example, an algorithm that takes time $10^{1000} * n$ is *not* practically feasible, but since f(n) = $10^{1000} * n$ is a linear function (hence a polynomial), it is feasible in the sense of the above definition.

Vice versa, an algorithm that takes time $2^{0.000000000001 * n}$ is *not* feasible according to the above definition, but for all practical purposes, it is feasible.

In most cases, however, the above definition works well – and it is the best definition we have – so it is normally used as a definition of a feasible algorithm.

What is a problem: towards a definition. The notion of a problem differs from discipline to discipline. In humanities, there are problems – like problems of ethics or the problem of meaning of life – which are very important, but for which there is no easy-to-use well-accepted criterion for checking whether a proposed solution indeed solves the problem. In contrast, in mathematics, we usually deal with problems for which it may be hard to come up with a solution, but once a solution is given – with all its justifications – it is relatively easy to check that we have indeed solved the problem.

For example, if we want to solve a system of equations, then it may be difficult to compute its solution. However, once a solution is proposed, it is easy to check that this is indeed a solution – we just plug in the values into the corresponding formulas and check that the left-hand side of each equation is indeed equal to the right-hand side.

Similarly, it may be difficult to find a solution to a system of differential equations. However, once we have a proposed solution, we can simply plug it in into the equations and check that in all these equations, we indeed have an equality.

Another example: it is usually very difficult to come up with a proof of the new mathematical result. However, once the detailed proof is presented, it is straightforward to check the correctness of this detailed proof: for that, it is sufficient to check that each step of this proof follows the rules.

In all these cases:
- we are given an object x: a system of equations, a statement to prove, etc., and
- we want to find an object y: a solution to the system x, a proof of the statement x – that satisfies an easy-to-check (i.e., feasible) property C(x, y).

In the example of a proof, it is important that the proof y be not too long. Indeed, if the proof is too long – and historically, some computer-generated proofs were very long -- it is impossible to check the proof by hand, and thus, the proof may contain mistakes – and historically, computer-generated proofs did contain mistakes.

How can we formulate the condition that the length len(y) should not be too large? Similarly to the formalization of the idea that the algorithm's running time should be too long, it is natural to formalize this condition by requiring that the length len(y) should not exceed some polynomial of the length of x. Thus, we arrive at the following definition.

What is a problem: a precise definition. A *generic problem* is a pair (C, P_l), in which:
- C(x, y) is a feasible algorithm that, given two strings x and y, returns "true" or "false"
- $P_l(n)$ is a polynomial.

By an *instance* of the general problem (C, P_l), we mean the following problem:
- *Given:* a string x.
- *Find:* a string y for which C(x, y) is true, and len(y) does not exceed $P_l(\text{len}(x))$.

Generic problems are also known as *non-deterministic polynomial problems*, or *NP-problems*, for short. This means that once we have guessed a solution y, we can check, in polynomial time, whether y is indeed a solution – i.e., this means that the algorithm C(x, y) is feasible.

Can all problems be solved by a feasible algorithm? Some problems can be solved by a feasible algorithm. The class of such problems is usually denoted by P (short for *polynomial-time*). A natural question is: can all problems be solved by a feasible algorithm? In terms of classes NP and P, this question can be formulated as follows: is P equal to NP?

This is a known open problem. While most computer scientists believe that P is different from NP, no one has so far been able to prove (or disprove) this belief.

What *is* known is that some NP-problems are the hardest to solve, in the following precise sense: we say that a generic problem (C, P_l) is *NP-hard* if every NP-problem can be reduced to the problem (C, P_l).

Here, by reduction, we mean reduction is the usual sense: for example,
- once we know how to solve quadratic equations $a * x^2 + b * x + c = 0$,
- we can solve equations of the type $p + q * x + r / x = 0$.

Namely, we can multiply both sides of the second equation by x, and get an equation of the first type $p * x + q * x^2 + r = 0$, which corresponds to $(a, b, c) = (q, p, r)$.

Examples of NP-hard problems. Many generic problems are NP-hard. Historically the first example of an NP-hard problem is the following *Propositional Satisfiability problem*:
- *Given:* a propositional formula, i.e., a Boolean combination of the Boolean (true-false) variables $v_1, \ldots, v_n$, such as $(v_1 \lor \text{not } v_2 \lor v_3)$ & $(\text{not } v_1 \& v_2)$
- *Find:* the truth values of the variables v_i that make the formula true.

In the above example, v_1 = false and v_2 = false makes the corresponding formula true.

Another example of an NP-hard problem is the following *Ali-Baba problem* (also known as the *knapsack problem*). According to *1001 Nights*, Ali-Baba stumbles upon a cave where 40 thieves have been keeping their loot. In this case, there are several items, with weights $w_1, \ldots, w_n$, and prices $p_1, \ldots, p_n$. There is a limit W on how much weight a donkey can carry. Within this limitation, he would like to maximize the price of what he takes away – or at least make sure that the overall price of selected items exceeds a certain threshold P.

A natural way to describe possible choices is to introduce n 0-1 variables $x_1, \ldots, x_n$, so that $x_i = 1$ means that we take the i-th item, and $x_i = 0$ means that we did not take this item. In terms of these variables x_i, the overall weight of selected items is equal to $x_1 * w_1 + \ldots + x_n * w_n$, and the overall price is equal to $x_1 * p_1 + \ldots + x_n * p_n$. So, we arrive at the following problem:
- *Given*: the values $w_1, \ldots, w_n$, W, $p_1, \ldots, p_n$, P
- Find: the values xi from {0, 1} for which the sum $x_1 * w_1 + \ldots + x_n * w_n$ does not exceed W and the sum $x_1 * p_1 + \ldots + x_n * p_n$ is larger than or equal to P.

So how do we solve NP-hard problems: the importance of duality and multi-ality. For each NP-hard problem, there is usually a good heuristic that helps us solve many instances of this problem.

For example, for propositional satisfiability, it makes sense to select v_i = true if the corresponding formula has more occurrences of v_i than of its negation not v_i, and select v_i = false otherwise. In the above example, it makes sense to select v_3 = true.

For the Ali-Baba problem, it makes sense, e.g., to start with the object whose price per weight p_i / w_i is the largest (if it fits on the donkey), then next largest, etc.

These heuristics help to solve many instances of the corresponding general problems, but not all of them. So how can we solve the remaining problems? Here is where NP-hardness can help. According to the definition, NP-hardness means that any problem from the class NP can be reduced to this problem. So, for example, if for some propositional formula, the corresponding heuristic does not work, we can reduce it to an instance of the Ali-Baba problem and try the Ali-Baba heuristic. If this does not work either, we can reduce it to yet another NP-hard problem and use the corresponding heuristic, etc.

The more translations we apply, the higher the chance that we will be able to solve the original problem. Indeed, let us denote the probability of solving the problem by using the corresponding heuristic by p. In these terms, the probability that the problem will not be solved by using this heuristic is equal to $1 - p$. If we cannot solve the problem by using the original heuristic, then we translate it into another problem and apply the other problem's heuristic. Here:

- the conditional probability of this second heuristic to be successful is p, so the probability that the problem will be solved by the second heuristic is p * (1—p),
- the conditional probability of this second heuristic being *not* successful is equal to $1 - p$, so the probability that this problem will not be solved by two heuristics is $(1 - p)^2$.

If the problem is not solved by using two heuristics, we can translate it into the third problem, etc.

In general, if we use h heuristics, the probability that the problem will still not be solved after trying all these heuristics is equal to $(1 - p)^h$. Thus, the probability that the problem will be solved is equal to $1 - (1 - p)^h$.

As h increases, this success probability tends to 1. Thus, the more translations we know, the higher the probability that the original problem will be solved.

From this viewpoint, it is extremely important not only to know how to solve different types of problems, but also to be able to translate each problem from one representation to another. In other words, to be able to solve complex problems, it is vitally important to use duality and multi-ality.

Conclusions. Many mathematical objects have several different representations: a rational number can be represented either as a fraction or by its decimal expansion, a spatial object can be represented either in geometric terms, or by a function describing its shape, etc. In many cases, this duality – and, more generally, multi-ality – helps to solve the related problems.

In this paper, we show that duality and multi-ality are crucially important for solving problems. From this viewpoint, is it extremely important that the students not only master techniques for solving problem in one of the representations, it is important that they are also able to translate each problem from one representation to another – i.e., that they master the *relation* between different representations. Thus, emphasizing such relations will enhance the quality of mathematics education.

References
1.   Sfard, A. On the dual nature of mathematical conceptions: reflectons on processes and objects as different sides of the same coin // Educational Studied in Mathematics. – 1991. – P. 1-36.
2.   Tarski, A. A Decision Method for Elementary Algebra and Geometry. Berkeley an Los Angeles, 1951, 63 P.
3.   Basu, S., Pollack, R., Roy, M.-F. Algorithms in Real Algebraic Geometry. Springer-Verlag, Berlin, 2006
4.   Feynman, R., Leighton, R., Sands, M. The Feynman Lectures on Physics, Addison Wesley, Boston, Massachusetts, 2005.
5.   Papadimitriou, C. Computational Complexity, Addison Welsey, Reading, Massachusetts, 1994.