Using Math Partner Environment in Algorithmic Mathematics Course

Sergei Pozdniakov

Abstract. One of the fundamental problems in the formation of computational thinking is the inclusion of computing on a computer as an element of human intellectual activity. As a rule, meaningful operations with polynomials cannot be illustrated with simple examples performed "by hand". Therefore, it is advisable to combine the theoretical explanation using tools for performing operations with polynomials. However, the calculations performed on specific examples do not always correspond to mental operations that reflect the essence of the algorithm and allow you to control its execution. The problem arises of comparing the theoretical model of the algorithm for its implementation on specific examples. The talk shows how the parallel use of several representations of polynomials solves this problem. The work was supported by the RFBR grant No. 19-29-14141

Introduction

As shown in [1], a computer experiment does not provide a conceptual understanding of the model hidden behind it. So, for example, working with dynamic geometry, a student can easily check that three medians intersect at one point, but this fact will not provide the emergence of intelligent mechanisms that can be used to solve other problems. Knowledge of this fact will be isolated, analogous to descriptive knowledge in subjects such as geography. Therefore, Papert, explaining the use of a computer tool in explaining the invariance of the value of an inscribed angle that subtends the same arc on the circle, is not limited to a computer experiment, but introduces reasoning based on observations of the rotation of a computer turtle [2].

1. Algorithms over polynomials

Let us consider the problem of mastering algorithms with polynomials using the example of an algorithm for decomposing a polynomial into square-free factors. Despite the fact that the algorithm is quite transparent, it causes difficulties for students. Let's analyze the source of the difficulties. In the process of explaining the square-free factorization algorithm, the factorization into irreducible factors [3] is used, while this algorithm, according to the logic of presentation, precedes the "full" decomposition algorithms. Students have a cognitive dissonance, they subconsciously assume that the factorization into square-free factors is preceded by the factorization into irreducible factors. The MathPartner program [4] has all the necessary operations to implement the decomposition algorithm into square-free factors: the derivative of a polynomial, division of polynomials, finding the GCD of polynomials. This allows us to compare the conceptual model of a polynomial in the form of a complete decomposition into irreducible factors to the canonical representation of the polynomial, which does not carry information about the desired representation. A tabular comparison (see Table 1, 2) of the actions of the steps of the algorithm on the conceptual model and the results of actions on a specific example removes part of the above contradiction. It becomes clear to students that the algorithm works with polynomials in expanded form, but they want to check whether these polynomials really correspond to the factorizations described in the conceptual model. This can be obtained by adding another column to the table (for ease of reading this is presented here in Table 3), in which intermediate polynomials are decomposed into irreducible factors. Using a calculator to work with polynomials, you can pose problems that require a conceptual answer, but which can be verified with examples. For example, "explain the meaning of the $GCD(P; P'/GCD(P'; P))$ result in terms of factoring a polynomials".

2. Features of using Math Partner

As part of the Algorithmic Mathematics course, Math Partner was used to study factorization algorithms. In addition to the above-mentioned decomposition algorithm into square-free factors, they were the algorithms of Kronecker, Berlekamp and Hensel [5]. Note that these algorithms required auxiliary algorithms, such as the linear representation of the GCD of polynomials or finding the kernel of a linear operator. These algorithms themselves are not presented in the Math Partner program, what may be is good, since the students implemented them in stages, illustrating the steps with examples. Similarly, the program was used to study Gröbner bases (Buchberger's algorithm). In this case, the program had an implementation of the algorithm. This allowed, along with a step-by-step illustration of Buchberger's algorithm (reduction of polynomials, construction of s-polynomials), to carry out experiments with the construction of various Gröbner bases, for example, by changing the order of variables.

Algorithm	Conceptual model
$P(x)$	$P_1(x)^{k_1}P_2(x)^{k_2}\dots P_n(x)^{k_n}$
$P'(x)$	$P_1(x)^{k_1-1}P_2(x)^{k_2-1}\dots P_n(x)^{k_n-1}Q(x)$
$T(x) = GCD(P(x), P'(x))$	$P_1(x)^{k_1-1}P_2(x)^{k_2-1}\dots P_n(x)^{k_n-1}$
$R(x) = P(x)/T(x)$	$P_1(x)P_2(x)\dots P_n(x)$
$H(x) = GCD(T(x), R(x))$	$\prod_{i=1,k_i\neq 1}^{n} P_i(x)$
$R(x)/H(x)$	$\prod_{i=1,k_i=1}^{n} P_i(x)$

TABLE 1. Conceptual model

Algorithm	Computational model (example)
$P(x)$	$3x^6 + 7x^5 + 18x^4 + 19x^3 + 21x^2 + 8x + 4$
$P'(x)$	$18x^5 + 35x^4 + 72x^3 + 57x^2 + 42x + 8$
$T(x) = GCD(P(x), P'(x))$	$x^2 + x + 2$
$R(x) = P(x)/T(x)$	$3x^4 + 4x^3 + 8x^2 + 3x + 2$
$H(x) = GCD(T(x), R(x))$	$x^2 + x + 2$
$R(x)/H(x)$	$3x^2 + x + 1$

TABLE 2. Computational model (example)

Algorithm	Checking intermediate steps
$P(x)$	$(3x^2 + x + 1)(x^2 + x + 2)^2$
$P'(x)$	$(x^2 + x + 2)(18x^3 + 17x^2 + 19x + 4)$
$T(x) = GCD(P(x), P'(x))$	$x^2 + x + 2$
$R(x) = P(x)/T(x)$	$(x^2 + x + 2)(3x^2 + x + 1)$
$H(x) = GCD(T(x), R(x))$	$x^2 + x + 2$
$R(x)/H(x)$	$3x^2 + x + 1$

TABLE 3. Checking intermediate steps

Conclusion

An important characteristic of technical thinking is its three-component structure: conceptual-figurative-practical. Traditional teaching of mathematics involves two components: conceptual and figurative. Using the MathPartner allows you to add a practical component that meets the goals of engineering education. The use of computational capabilities is used here not only and not so much to save time on calculations with polynomials, but for the parallel use of several representations, one of which allows you to monitor the internal structure of the polynomial in the process of its transformations, the second shows the real form of the polynomial. Using multiple views in parallel shows techniques for shaping computational thinking - the computer is no longer a "black box".

References

[1] S.Pozdniakov, E.Tolkacheva. *The interaction of algorithms and proofs in the discrete mathematics course for future engineers.* "Polynomial Computer Algebra '2020", October 12-17 2020, Euler International Mathematical Institute, St. Petersburg, RUSSIA, ISBN 978-5-9651-0568-7.

[2] S.Papert, *An Exploration in the Space of Mathematics Educations*, Int J Comput Math Learning, vol. 1, no. 1, pp. 95–123, 1996; doi: 10.1007/BF00191473.

[3] Devenport, Dzh. *Komp'yuternaya algebra: simvol'nye i algebraicheskie vychisleniya*: per. s angl. / Dzh. Devenport, I. Sire, E. Turn'e. – M. : Mir, 1991. – 350 s.

[4] G. I. Malaschonok, *Primenenie servisa MathPartner v obrazovanii [Application of the Mathpartner Service in Education]*, Computer tools in education, no. 3, pp. 29–37, 2017 (in Russian).

[5] Akritas, A. *Osnovy komp'yuternoj algebry s prilozheniyam*: per. s angl. / A. Akritas. – M. : Mir, 1994. – 544 s.

Sergei Pozdniakov
Algorithmic Mathematics Department
St.Petersburg Electrotechnical University "LETI"
Saint-Petersburg, Russia
e-mail: pozdnkov@gmail.com