

# Лекция «Арифметические операции с многочленами»

Поздняков Сергей Николаевич

запись конспекта: Кацер Евгений

дата лекции: 7 апреля 2018 г.

## §18 Арифметические операции с многочленами

00:33

Все многочлены (или полиномы) можно представить в виде очень похожем на представление числа в позиционной системе счисления:

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0,$$

где  $a_n \neq 0$ . Если заменить  $x$  на основание  $p$ -ичной системы счисления, мы получим число в этой системе счисления. Но в представлении числа в  $p$ -ичной системе на коэффициенты  $a$  накладывается ограничение по значению (от 0 до  $p - 1$ ), в многочленах такого ограничения нет.

Мы можем представлять многочлен набором коэффициентов, тогда он будет выглядеть как число.

$$a_n a_{n-1} \dots a_0$$

Более того, мы можем взять те же алгоритмы, что и для чисел, и применять их чуть-чуть скорректированные версии. Например, сложение многочленов. При сложении чисел мы начинаем с конца, перенос берем на ноль, складываем цифры текущих разрядов, записываем остаток и запоминаем перенос равный частному от деления суммы цифр на основание системы счисления, осуществляем переход к следующему разряду. При сложении многочленов нам не нужно вычислять остаток и перенос, нужно просто складывать.

Также существуют некоторые теоремы, которые помогают разложить многочлен на множители легче, чем целое число, так как многочлены можно рассматривать как функции и дифференцировать, что дает новые возможности.

*Замечание.* Всегда нужно оговаривать над каким полем коэффициентов рассматривается многочлен.

Мы будем считать, что коэффициенты многочлена всегда принадлежат какому-то полю.

*Примечание.* Поле — множество, для элементов которого выполнены все арифметические операции (кроме деления на 0) и они удовлетворяют известным свойствам, сформулированным, например, при введении рациональных или вещественных чисел. Поля, которые вы знаете, —  $\mathbb{Q}$ ,  $\mathbb{R}$ ,  $\mathbb{C}$ ,  $\mathbb{Z}_p$ , если  $p$  — простое. Операции  $\mathbb{Z}_p$  выполняются гораздо легче, так как количество чисел конечно.

04:51

**Обозначения:**

- $n = \deg p(x)$  — степень многочлена (аналог модуля числа).
- $a_n = l(p(x))$  — старший коэффициент многочлена.

**Пример.** Возьмем два многочлена  $p_1(x) = x^5 + 2x - 1$  и  $p_2(x) = 2x + 1$  над  $\mathbb{Z}_3$  (значит коэффициенты будут из поля чисел по модулю 3 — 0, 1, 2. Из-за этого мы можем не писать, например,  $-1$ , так как  $-1 \equiv 2 \pmod{3}$ ). Представим многочлены, как набор коэффициентов:  $p_1(x) = 100022$  (заменим  $-1$  на 2), а  $p_2(x) = 21$ . Перемножим их, используя скорректированный алгоритм умножения чисел "столбиком":

$$\begin{array}{r}
 \times \quad 100022 \\
 \quad 21 \\
 \hline
 \quad 100022 \\
 200011 \\
 \hline
 2100102
 \end{array}$$

Заметим, что мы не переносим единичку при умножении  $2 \cdot 2$  или при сложении  $2 + 1$  как в троичной системе счисления, так как выполняем операции с многочленами, где нет связи между соседними коэффициентами.

Переведем полученный многочлен в нормальный вид:

$$p_1(x) \cdot p_2(x) = 2x^6 + x^5 + x^2 + 2$$

Теперь рассмотрим деление многочленов «уголком». Разделим  $p_1(x)$  на  $p_2(x)$ , при этом учтем, что для деления мы берем старшие коэффициенты многочленов и делим их, и коэффициенты многочленов вычитаются

независимо друг от друга:

$$\begin{array}{r}
 100022 \overline{) 21} \\
 \underline{12} \phantom{00000} \\
 10 \phantom{00000} \\
 \underline{12} \phantom{00000} \\
 10 \phantom{00000} \\
 \underline{12} \phantom{00000} \\
 12 \phantom{00000} \\
 \underline{12} \phantom{00000} \\
 2 \phantom{00000} \\
 \underline{0} \phantom{00000} \\
 2
 \end{array}$$

Тогда ( $2x^4 = -x^4$  в  $\mathbb{Z}_3$ ):

$$p_1(x) = p_2(x) \cdot (-x^4 - x^3 - x^2 - x) + 2$$

19:23

**Определение.** Будем говорить, что многочлен  $M(x)$  поделен на многочлен  $N(x)$  с остатком, если:

$$M(x) = N(x) \cdot Q(x) + R(x), \quad (*)$$

где  $Q(x)$  — частное, а  $R(x)$  — остаток, причем  $\deg(R(x)) < \deg(N(x))$ .

*Примечание.* Определение перенесено с чисел, изменена лишь последняя часть (о сравнении многочленов). Числа сравниваются по абсолютной величине, а многочлены — по степеням.

**Теорема.** *Представление многочлена в форме (\*) — существует и единственно.*

*Примечание.* Для чисел мы это доказывали с помощью числовой оси: на ней откладывали нужное число раз  $N$ , и оказывалось, что когда мы приблизимся к  $M$  максимально близко, разница между полученной точкой и  $M$  будет очевидно меньше длины  $N$ . Для многочленов мы так сделать не можем, многочлен на прямой нарисовать нельзя, но алгоритм «откладывания» можно преобразовать в алгоритм деления многочленов друг на друга.

*Доказательство.* Для доказательства существования напишем алгоритм, аналогичный алгоритму деления чисел «уголком», а потом докажем, что он для всех многочленов будет работать корректно.

---

**Algorithm 1** Алгоритм построения представления многочлена вида (\*)

---

```

1:  $R(x) := M(x)$ 
2:  $Q(x) := 0$ 

3: while  $\deg(R(x)) \geq \deg(N(x))$  do
4:    $T(x) := \frac{l(R(x))}{l(N(x))} \cdot x^{\deg(R(x)) - \deg(N(x))}$             $\triangleright$  Текущее частное
5:    $Q(x) := Q(x) + T(x)$ 
6:    $R(x) := R(x) - N(x) \cdot T(x)$ 
7: end while
```

---

*Примечание.* Циклическую программу надо стараться написать так, чтобы доказательство корректности программы соответствовало её структуре. В данном случае инвариантом цикла будет выражение:

$$M(x) = N(x) \cdot Q(x) + R(x)$$

База индукции (всегда связана с инициализацией):

$$R(x) = N(x) \cdot 0 + R(x) \text{ — выполнено}$$

Индукционный переход:

Пусть утверждение верно для  $k$ -го шага, тогда:

$$M(x) = N(x) \cdot Q_k(x) + R_k(x)$$

Докажем, что равенство сохранится при переходе к  $k + 1$  шагу:

$$\begin{aligned} Q_{k+1}(x) &= Q_k(x) + T_k(x) \\ R_{k+1}(x) &= R_k(x) - N_k(x) \cdot T_k(x) \end{aligned}$$

После подстановки вместо  $Q_k(x)$  и  $R_k(x)$  получаем:

$$N(x) \cdot Q_{k+1}(x) + R_{k+1}(x) = N(x) \cdot (Q_k(x) + T_k(x)) + R_k(x) - N_k(x) \cdot T_k(x) = M(x)$$

Равенство сохраняется на  $k + 1$  шаге, следовательно, выражение — инвариант цикла.

На каждом шаге цикла мы уменьшаем степень многочлена, так как у  $T(x)$  степень  $= \deg(R(x)) - \deg(N(x))$ , тогда:

$$\deg(T(x)) \cdot \deg(N(x)) = \deg(R(x)) - \deg(N(x)) + \deg(N(x)) = \deg(R(x))$$

И в результате вычитания  $R_k(x) - N_k(x) \cdot T_k(x)$  степень многочлена уменьшится (коэффициенты подбираются при нахождении  $T(x)$ ).

**Упражнение.** Проверить, что на каждом шаге алгоритма степень  $R(x)$  уменьшается.

Из-за того, что степень многочлена на каждом шаге уменьшается, бесконечно работать алгоритм не может (степень — натуральное число). Алгоритм остановится только тогда, когда степень  $R(x)$  будет меньше, чем степень  $N(x)$ , что соответствует определению.

Теперь докажем единственность. Для чисел мы доказывали единственность двумя способами. На той лекции был доказан второй способ, так как первый не подходит для многочленов.

**Обязательное упражнение:** Доказать единственность деления многочленов с остатком.

□

Теперь следующий вопрос: работая с числами, мы говорили о переходе к другому основанию. Как перенести метод перехода к другому основанию на многочлены и что вообще означает в этом случае фраза «переход к другому основанию»?

## §19 Схема Горнера

37:49

Предположим, что у нас многочлен представлен в каноническом виде (разложение по степеням  $x$ ):

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \quad (1)$$

А нам интересно представить его по степеням  $x - a$ . Коэффициенты изменятся:

$$p(x) = b_n (x - a)^n + b_{n-1} (x - a)^{n-1} + \dots + b_1 (x - a) + b_0 \quad (2)$$

*Примечание.* Сразу можно догадаться, что коэффициенты  $a_n$  и  $b_n$  будут равны, так как в обоих представлениях  $x^n$  будет главным.

Как из представления (1) перейти к представлению (2)? Это не совсем аналог перехода к другому основанию. Здесь очень важно знать теорему Безу.

39:25

**Теорема (Безу).** Остаток от деления многочлена  $P(x)$  на многочлен  $(x - a)$  равен  $P(a)$  (значению многочлена в точке  $a$ ).

*Примечание.* Эта теорема связывает алгебру и анализ: алгоритм деления многочленов — алгебра, а подстановка числа в функцию (многочлен можно рассматривать как функцию, которая каждому  $x$  сопоставляет значение  $P(x)$ ) — анализ.

Такие связи очень важны. Примером может послужить только что установленная связь между целыми числами и многочленами, мы почти без изменений перенесли все алгоритмы с целых чисел на многочлены, что позволило сократить время на их изучение.

*Доказательство.* По теореме о делении с остатком представим  $P(x)$  единственным образом:

$$P(x) = (x - a) \cdot Q(x) + R(x), \quad (3)$$

причем:

$$\begin{cases} \deg(R(x)) < \deg((x - a)) \\ \deg((x - a)) = 1 \end{cases} \implies R(x) = R - \text{константа}$$

Если  $R(x)$  — константа, то подставим в обе части (3) вместо  $x - a$ :

$$P(a) = (a - a) \cdot Q(a) + R \implies P(a) = R$$

□

Теперь рассмотрим так называемую схему Горнера, которая даст ответ на исходный вопрос: «Как из представления (1) перейти к представлению (2)?». Сначала найдем коэффициенты многочлена  $Q(x)$  (то есть пока что не коэффициенты (2), потому что мы просто поделили на  $(x - a)$ ). При этом делении мы определим последний из искомых коэффициентов  $b_0$ ). Таким образом за первый (он же типовой) шаг мы найдем коэффициент  $b_0$  и многочлен  $Q(x)$ , который мы дальше будем делить на  $(x - a)$ . При этом степень многочлена  $Q(x)$  будет на единицу меньше чем  $P(x)$ . Далее, если  $Q(x)$  поделим на  $(x - a)$ , то получим  $b_1$ . И так дальше будем повторять нахождение остатка и  $Q(x)$ , применяя деление. В конце концов мы получим все коэффициенты (2).

Давайте сначала разберем, как находятся коэффициенты  $Q(x)$ , а потом напомним общий вывод.

Давайте искать коэффициенты  $Q(x)$  методом неопределенных коэффициентов (мы точно знаем, что степень  $Q(x)$  меньше на единицу степени  $P(x)$ ):

$$Q(x) = c_{n-1}x^{n-1} + \dots + c_1x + c_0$$

Подставим  $P(x)$  и  $Q(x)$  в (3):

$$a_nx^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0 = (x - a) \cdot (c_{n-1}x^{n-1} + \dots + c_1x + c_0)$$

Будем разбираться с коэффициентами по очереди. Начнем со старших коэффициентов:

$$a_n = c_{n-1}$$

$$a_{n-1} = c_{n-2} - ac_{n-1}$$

Можно догадаться, что из-за того, что  $(x - a)$  — двучлен, в правой части любого коэффициента, кроме первого, будут стоять только два коэффициента, дающие нужную степень. Тогда мы можем записать  $k$ -ый коэффициент:

$$a_k = c_{k-1} - ac_k$$

Отсюда мы можем выразить  $c_{k-1}$ :

$$c_{k-1} = a_k + ac_k$$

Может быть немного непривычно, что мы меньший коэффициент выражаем через больший, но это естественно, если мы идём слева направо. Сначала находим коэффициент при наибольшей степени, потом при нахождении следующего коэффициента мы можем использовать найденный и т.д. Таким образом у нас получилась рекуррентная формула для получения этих коэффициентов, которая называется схемой Горнера:



$$\begin{cases} a_n = c_{n-1} \\ c_{k-1} = a_k + ac_k \end{cases}$$

50:04

**Таблица для схемы Горнера:** Рассмотрим процесс построения новых коэффициентов в таблице. Сначала запишем все коэффициенты  $P(x)$  и число  $a$  слева от таблицы:

$$a \mid a_n \mid a_{n-1} \mid \dots \mid a_k \mid \dots \mid a_0 \mid$$

Теперь нужно записать коэффициенты  $Q(x)$ :  $c_{n-1} = a_n$  по доказанному равенству, дальше  $c_{n-2} = a_{n-1} + ac_{n-1}$  (все время смотрим на делитель и на два соседних коэффициента) и так далее до остатка равного  $b_0$ :

$a$	$a_n$	$a_{n-1}$	$\dots$	$a_k$	$\dots$	$a_0$	
		$c_{n-1}$	$c_{n-2} = a_{n-1} + ac_{n-1}$	$c_k = a_{k+1} + ac_{k+1}$		$c_1 = a_1 + ac_1$	$b_0 = a_0 + ac_0$

Теперь представим, что мы те же самые действия применяем уже к полученному многочлену  $Q(x)$ , получая тем самым новый многочлен  $Q'(x)$  с коэффициентами  $c'_i$ . В таблице ещё будет сдвиг, так как мы еще делим на  $(x - a)$ . Остаток теперь будет равен  $b_1$ :

$a$	$a_n$	$a_{n-1}$	$\dots$	$a_k$	$\dots$	$a_0$	
		$c_{n-1} = a_n$	$c_{n-2} = a_{n-1} + ac_{n-1}$	$c_k = a_{k+1} + ac_{k+1}$	$\dots$	$c_0 = a_1 + ac_1$	$b_0 = a_0 + ac_0$
			$c'_{n-2}$	$c'_k$	$\dots$	$c'_0$	$b_1 = c_0 + ac'_0$

И так далее. Если мы будем применять эту схему много раз, то мы получим разложение по степеням  $(x - a)$ .

57:15

**Пример.** Разложить  $P(x) = x^3$  по степеням  $x - 1$  ( $a = 1$ ):

$1$	$1$	$0$	$0$	$0$	
		$1$	$1$	$1$	$1$

Давайте проверим, правильно ли мы выполнили вычисления:

$$x^3 = \underbrace{(x-1) \cdot (x^2 + x + 1)}_{x^3-1} + 1 = x^3$$

Вычисления верны. Продолжим:

1	1	0	0	0	
		1	1	1	1
			1	2	3

Давайте проверим, правильно ли мы выполнили вычисления:

$$x^2 + x + 1 = (x - 1) \cdot (x + 2) + 3 = x^2 + x - 2 + 3 = x^2 + x + 1$$

Вычисления верны. Продолжаем:

1	1	0	0	0	
		1	1	1	1
			1	2	3
				1	3
					1

Теперь, если вычисления сделаны правильно, у нас получились коэффициенты  $b_0 = 1$ ,  $b_1 = 3$ ,  $b_2 = 3$ ,  $b_3 = 1$ , дающие искомое разложение. Проверим результат:

$$\begin{aligned} (x - 1)^3 + 3 \cdot (x - 1)^2 + 3 \cdot (x - 1) + 1 = \\ = x^3 - 3x^2 + 3x - 1 - 3x^2 + 6x + 3 + 3x - 3 + 1 = x^3 \end{aligned}$$

Мы получили правильный ответ, значит наш алгоритм работает.

## §20 Алгоритм Евклида для многочленов и его приложения

64:19

Первый вопрос: «Что называть НОД двух многочленов?». Когда мы рассматривали числа, мы говорили, что можно одну сущность убрать (когда мы сравниваем числа у нас появляется два отношения: первое — делится или не делится, второе — больше или меньше). Мы могли заменить на определение с одним отношением: НОД — делитель, который делится на все остальные делители. В этом случае сравнивать числа с помощью отношения больше-меньше не нужно. Это определение легче перенести на многочлены, следовательно:

*НОД многочленов* — это такой многочлен, который делится на все общие делители этих многочленов.

Тут есть одна тонкость:

**Пример.**  $\text{НОД}(2x - 1; x - \frac{1}{2}) = ?$

Можно сказать, что НОД будет равен  $2x - 1$ , или  $x - \frac{1}{2}$ , или  $4x - 2$ . Все эти ответы будут правильными, потому что все эти многочлены будут делиться на все остальные, поэтому НОД определяется с точностью до коэффициента (если мы будем умножать многочлены на одно какое-либо число, то от этого НОД не меняется).

*Замечание.* НОД многочленов определяется с точностью до числового ( $\neq 0$ ) коэффициента. Поэтому, когда вы рассматриваете алгоритм нахождения НОД, например, деление в столбик, то на каждом шаге мы имеем право умножать на какое-нибудь число (например, чтобы избавиться от дробных коэффициентов):

$$\begin{array}{c|c} P & Q \\ \hline k_1 \cdot R & k_2 \cdot Q \end{array}$$

Если рассматривать протокол в виде таблицы:

$P$	$Q$	$k_0 \cdot R_0$	$k_1 \cdot R_1$	$\dots$	$k_n \cdot \text{НОД}(P; Q)$	$0$

Коэффициенты можно добавить и к  $P$ , и к  $Q$ . Если дальше мы применяем расширенный алгоритм Евклида, то на предпоследнем шаге мы получаем какие-то коэффициенты  $A$  и  $B$ :

$P$	$Q$	$k_0 \cdot R_0$	$k_1 \cdot R_1$	$\dots$	$k_n \cdot \text{НОД}(P; Q)$	$0$
					$A$	
					$B$	

Которые, соответственно, дают нам:

$$A \cdot P + B \cdot Q = \text{НОД}(P; Q)$$

Соответственно, если мы будем домножать, то коэффициенты будут меняться, поэтому нельзя домножать в процессе вычислений. Нужно сначала домножить, а потом посчитать. Здесь  $A$ , и  $B$  — многочлены, причем:

$$\begin{cases} \deg(A) < \deg(Q) \\ \deg(B) < \deg(P) \end{cases} \quad \text{— доказать самостоятельно и перенести на числа}$$

(у чисел будет аналогичное неравенство, только там будет не степень, а абсолютная величина).

*Примечание.* В дальнейшем мы не рассматриваем многочлены на множестве вещественных чисел, так как работа с конечными представлениями вещественных чисел нетривиальна и требует разработки алгоритмов для работы с определенными классами вещественных чисел (например, с квадратичными иррациональностями).

Разберем теперь использование выведенных алгоритмов для работы с многочленами и перенесем китайскую теорему об остатках на многочлены. Переформулируем ее в терминах многочленов, но не для общего случая, а для некоторого интересного частного случая.

73:11

**Задача интерполяции:** Задача интерполяции изучается в математическом анализе, но у нас есть методы решения задач, которые мы можем применить. Оказывается, можно перенести некоторые результаты дискретной математики на классические задачи математического анализа).

Когда мы проводим какие-то эксперименты, мы обычно интерполируем их результаты, то есть, находим какую-то кривую по точкам, которая хорошо приближает нашу картину эксперимента. Если у нас две точки, то проводят прямую:



Рис. 1: Приближение по двум точкам

Если у нас три точки, то через них можно единственным образом построить параболу:

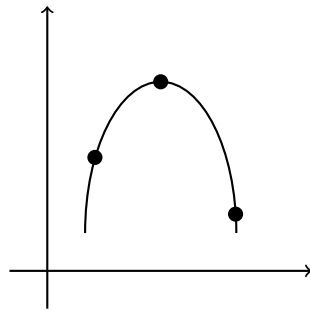


Рис. 2: Приближение по трем точкам

В общем случае задача интерполяции формулируется так: пусть есть  $n + 1$  точка. Когда у нас было две точки, мы интерполировали многочленом первой степени, когда было три — многочленом второй степени, когда  $n + 1$  точек — многочленом  $n$ -ой степени. Нам нужно его найти:

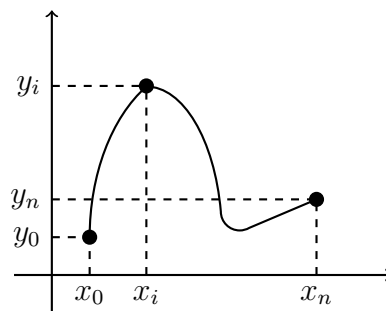


Рис. 3: Общая задача интерполяции

Обозначим этот многочлен, как  $P_n(x)$ :

$$P_n(x_i) = y_i$$

Теперь воспользуемся теоремой Безу, которая говорит, что если поделить многочлен на  $x - x_i$ , то остаток будет  $y_i$ . Воспользуемся теоремой Безу в обратную сторону (мы ее доказали только в одну сторону, но ее просто доказать и в обратную сторону: если значение многочлена равно  $y_i$ , то остаток от деления по  $x$  на  $x - x_i$  будет равен ему же).

**Упражнение.** Доказать теорему Безу в обратную сторону.

76:26

В терминах деления с остатком мы можем написать, что:

$$P_n(x) \equiv y_i \pmod{(x - x_i)},$$

где  $i$  меняется от 0 до  $n$ . Попробуем применить китайскую теорему об остатках. Для этого нужно выяснить: являются  $x - x_i$  и  $x - x_j$  взаимно простыми (при  $i \neq j$ )? Такие вопросы часто ставят студентов в тупик, но на самом деле это очень просто. Мы знаем алгоритм Евклида, вот у нас два многочлена:

$$x - x_i \mid x - x_j$$

Если мы один многочлен заменим разностью двух, множество делителей не изменится. Заменим первый многочлен на разность первого и второго, получится:

$$\begin{array}{c|c} x - x_i & x - x_j \\ \hline x_j - x_i & x - x_j \end{array}$$

$x - x_i$  — многочлен первой степени, а  $x_j - x_i$  — число. А если многочлен константа, то НОД не может быть равен многочлену первой степени, он может быть только нулевой степени.

Обратите внимание, что это очень естественно для задач интерполяции, ведь не бывает, что в одной точке два значения разные.

Мы выяснили, что китайская теорема применима. Давайте вспомним формулу и вычислим её элементы для рассматриваемого нами случая многочленов:

$$P_n(x) \equiv c_1(x) \cdot d_1(x) \cdot r_1(x) + \dots + c_n(x) \cdot d_n(x) \cdot r_n(x) \quad (4)$$

В этой формуле мы знаем остатки:  $r_1(x) = y_1, \dots, r_n(x) = y_n$ .

С  $c_i$  немного сложнее:

$$c_i(x) = (x - x_1) \cdot \dots \cdot (x - x_{i-1}) \cdot (x - x_{i+1}) \cdot \dots \cdot (x - x_n) \quad (\diamond)$$

Теперь нам нужно найти такое  $d_i(x)$ , которое при умножении на  $c_i(x)$  дает остаток 1 по модулю  $x - x_i$ :

$$c_i(x) \cdot d_i(x) = 1 \pmod{x - x_i}$$

По определению сравнения это эквивалентно тому, что разность  $c_i(x) \cdot d_i(x)$  и 1 делится на  $x - x_i$ . Значит есть такое  $Q(x)$ , что:

$$c_i(x) \cdot d_i(x) - 1 = (x - x_i) \cdot Q(x)$$

Перенесем в одну часть:

$$c_i(x) \cdot d_i(x) - (x - x_i) \cdot Q(x) = 1 \quad (*)$$

При анализе расширенного алгоритма Евклида, который является основой решения этого уравнения, мы выяснили, что степень  $d_i(x)$  меньше степени  $x - x_i$ , а степень  $x - x_i = 1$ , следовательно,  $d_i(x)$  — константа ( $d_i(x) = d_i \cdot C$ , где  $C$  — константа). Если  $d_i$  — константа, то мы ее можем очень просто найти из уравнения (\*), подставив в него  $x_i$ :

$$c_i(x_i) \cdot d_i = 1 \implies d_i = \frac{1}{c_i(x_i)}$$

Посмотрите, как интересно, в многочлене  $c_i(x)$  отсутствует  $x_i$  и, если его подставить в (◇) получится некоторое (довольно длинное) число, обратное к которому и будет  $d_i$ .

84:03

В заключение перепишем формулу (4) в форме так называемой теоремы об интерполяционном многочлене Лагранжа:

$$P_n(x) = \sum_{i=0}^n c_i(x) \cdot \underbrace{d_i(x)}_{=\frac{1}{c_i(x_i)}} \cdot \underbrace{r_i(x)}_{=y_i} = \sum_{i=0}^n \frac{c_i(x)}{c_i(x_i)} \cdot y_i$$