Palestine Journal of Mathematics

Vol. ??(?)(20??) , 1–5

WHICH (SUB)DIRECT DECOMPOSITIONS ARE USEFUL?

Manuel Kauers and Günter Pilz

Communicated by Kent Neuerburg

MSC 2010 Classifications: Primary 13C05; Secondary 68W30 20M99.

Keywords and phrases: Chinese remainder theorem, polynomial interpolation, Discrete Fourier Transform.

Abstract Algebraists like decompositions. One goal is that one then can do parallel computations. Some decompositions are particularly useful, and we want to find out when a decomposition is useful (for parallel computations). Other decompositions will also be useful, for the theory and for our knowledge. We will go through some examples.

Instance 1: Vector spaces

Every math student knows: If ${}_KV$ is a finite dimensional vector space over a field K then

$$\dim({}_KV) = n \iff V \simeq K^n$$

How useful this is can be seen in the case when V is the vector space of all equivalence classes of "arrows" in the 3D-space. Imagine how ugly it would be to determine things like linear combinations.

So: $V \longrightarrow K^n$ works by introducing coordinates—after choosing a basis $B = (b_1, b_2, \ldots, b_n)$.

And $V \longleftarrow K^n$ is done by the map $(\lambda_1, \lambda_2, \ldots, \lambda_n) \to \lambda_1 b_1 + \lambda_2 b_2 + \ldots + \lambda_n b_n$

Observe that this is not a "natural isomorphism" in the sense of category theory, because $\longrightarrow$ and $\longleftarrow$ depend on the selection of a basis!

Instance 2: Integers

If we want to do parallel computations in $\mathbb{Z}$, we will not succeed because $(\mathbb{Z}, +)$ is indecomposable. But we can do additions, subtractions, multiplications and taking powers in $\mathbb{Z}_n$, provided that n is sufficiently large.

If $n = p_1^{t_1} \ldots p_r^{t_r}$ then $\mathbb{Z}_n \simeq \mathbb{Z}_{p_1^{t_1}} \oplus \ldots \oplus \mathbb{Z}_{p_r^{t_r}}$ by the map $x \to (x_1, \ldots, x_r)$, where x_i is the residue class of x modulo $p_i^{t_i}$. We can do all computations with $x, y, \ldots$ in each component, and we will end up with an element in $\mathbb{Z}_{p_1^{t_1}} \oplus \ldots \oplus \mathbb{Z}_{p_r^{t_r}}$.

How about the way back: $\mathbb{Z}_n \longleftarrow \mathbb{Z}_{p_1^{t_1}} \oplus \ldots \oplus \mathbb{Z}_{p_r^{t_r}}$? It is the Chinese Remainder Theorem!

And the nice thing: One only has to deal with integers which are $< \bar{n} := \max(p_1^{t_1}, \ldots, p_r^{t_r})$!

So we want $\bar{n}$ to be small, and n should be big. It is better to start with $\bar{n}$. If we choose $\bar{n} = 50$, for instance, we get the maximal n as $n = 2^5 \cdot 3^3 \cdot 5^2 \cdot 7^2 \cdot 11 \cdot 13 \cdots 47 > 10^{21}$. For computations on an actual computer, a more realistic choice is $\bar{n} = 2^{64}$, for which we get a maximal n with $n = 2^{64}3^{40}5^{27} \cdots 4294967291 > 10^{4000000000000000000}$.

If we need to recover larger integers from homomorphic images, we may still get along without increasing $\bar{n}$ if we have some knowledge about the range in which the target integer lives. For example, if we know that it is somewhere between 10000000000 and 10000002000, then it suffices to choose some $\bar{n}$ with $n > 2000$, because then, every equivalence class in $\mathbb{Z}_n$ contains exactly one integer in the range $\{10000000000, \ldots, 10000002000\}$. In particular, we do not need to restrict to nonnegative integers. If we know that we are looking for an integer x with $|x| < M$, it suffices to choose $\bar{n}$ such that $n > 2M$ in order to recover x from its image in $\mathbb{Z}_n$.

See [3] for a more detailed discussion.

Instance 3: Polynomials

Most things which can be done in $\mathbb{Z}$, can also be done in $K[x]$. Let K be again a field, and we work in $K[x]/(f)$, where $f = p_1^{t_1} \dots p_r^{t_r}$ is a polynomial of sufficiently high degree. Recall that this is done, e.g., in coding theory, where f is of the form $f = x^m - 1$ in the case of cyclic codes (see, e.g., [4]). Again, we get $K[x]/(f) \simeq K[x]/(p_1^{t_1}) \oplus \dots \oplus K[x]/(p_r^{t_r})$ by an isomorphism $\longrightarrow$ which takes the equivalence class $[p]_{(f)}$ of $p \in K[x]$ w.r.t. (f) to the r-tuple $([p]_{p_1^{t_1}}, \dots, [p]_{p_r^{t_r}})$

What would be a good choice for f? We have a lot of freedom. Why not simply take $f = (x-k_1)(x-k_2)\dots(x-k_r)$, where the k_i are pairwise different elements of K? We get the components $[p]_{p_j^{t_j}}$ = remainder of p after division by $(x-k_j)$ = $p(k_j)$. Hence $[p]_{(f)}$ is mapped to $(p(k_1), p(k_2), \dots, p(k_r))$, if we identify the constant polynomial k with the element $k \in K$. And what is the way $\longleftarrow$ back? It is again the Chinese Remainder Theorem!

But what does $p(k)$ mean? We are sliding away from polynomials to polynomial functions.

Instance 4: Polynomial functions

To every polynomial $p \in F[x]$, there comes a polynomial function $\bar{p}\colon F \to F$ in the familiar way. Let $P(F)$ be the collection of all polynomial functions on F. One easily sees that $P(F)$ is a ring, in fact a subring of F^F, and that $\varphi\colon F[x] \to P(F)$, $p \mapsto \bar{p}$ is an epimorphism. Is φ always an isomorphism? If F is finite, this cannot happen, because $F[x]$ is infinite, while $P(F)$ is finite. As an example, take $p = x^5 - x \in \mathbb{Z}_5[x]$. Then p induces the zero function.

For fields F, the situation is easy:

Fact 1. If F is a field then $\varphi : F[x] \to P(F),\ p \to \bar{p}$ is an isomorphism $\iff$ F is infinite.

If F is "only" a ring, the dividing line between "iso / not iso" is still unknown. For example, an old result of Áczel [1] says that if the additive group $(F, +)$ is torsion-free then φ is an isomporphism.

So for polynomial functions, one cannot apply the "Principle of Comparing Coefficients" in general. What is a "principle"? Mathematicians do not have principles...

Can we decompose $P(F)$? Well, this is done already, since $P(F)$ is a subring of F^F, by the map $i\colon \bar{p} \to (\dots, \bar{p}(r), \dots)$, where r runs through all elements of F. Applying i makes a lot of sense, since it needs more time to compute $\bar{p} \cdot \bar{q} = \overline{p \cdot q}$, which involves convolutions, where $i(\bar{p}) \cdot i(\bar{q})$ can be done in a component-wise manner! Some additional care is needed for constant coefficients.

Do we have a "formula" to go back again, from $\mathrm{im}(i)$ to $P(F)$, so $P(F) \longleftarrow \mathrm{im}(i)$? Yes, it is the interpolation of polynomials.

Instances 3 and 4: The speed-up

Let us go back to the end of Instance 3. We want to evaluate $p \in F[x]$ at places $k_1, \dots, k_r$, and we now write $\bar{p}(k)$, of course. We still can choose these places.

Let us do this in an intelligent way and look at an example, $p = a_0 + a_1x + a_2x^2 + \dots + a_5x^5$, and $k_1 = 1$, $k_2 = -1$. If we put $a_{ev} := a_0 + a_2 + a_4$ and $a_{od} := a_1 + a_3 + a_5$ then $\bar{p}(1) = a_{ev} + a_{od}$ and $\bar{p}(-1) = a_{ev} - a_{od}$.

So $\bar{p}(1)$ and $\bar{p}(-1)$ can be computed very quickly. If $F = \mathbb{C}$, we might continue with i and $-i$. In general, we might continue with roots ω of unity, if there are enough of them.

Suppose we have a field F which, for some $n \in \mathbb{N}$, has n roots of unity. All finite fields fulfill this, as well as $\mathbb{C}$. Then there is also a "primitive" n-th root ω of unity, and all other roots of unity are powers of ω. So, if ω is a primitive n-th root of unity, we evaluate p at $\omega^0 = 1, \omega, \omega^2, \dots, \omega^{n-1}$.

Then $p = \mathbf{a} = (a_0, a_1, \dots, a_{n-1}) = \sum_{i=0}^{n-1} \hat{a}_i x^i$ is transferred to

$$(\overline{p}(\omega^0), \overline{p}(\omega), \dots, \overline{p}(\omega^{n-1})) =: (\hat{a}_0, \hat{a}_1, \dots, \hat{a}_{n-1}) =: \hat{\mathbf{a}} = \sum_{i=0}^{n-1} \hat{a}_i x^i.$$

The map $\mathbf{a} \to \hat{\mathbf{a}}$ is called the Discrete Fourier Transform (DFT). In applications, $\mathbf{a}$ is called the "signal vector", while $\hat{\mathbf{a}}$ is the "spectral vector". Sometimes, the re-writing of $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ as $\sum_{i=0}^{n-1} a_i x^i$ is called the "z-transform", which is a bit strange since nothing is transformed: $(a_0, a_1, \ldots, a_{n-1})$ and $\sum_{i=0}^{n-1} a_i x^i$ are THE SAME.

What is the transform $\mathbf{a} \to \hat{\mathbf{a}}$ good for? The DFT can be made easier. Take a primitive n-th root ω of unity in F and form the matrix

$$D_{n,\omega} := \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \omega & \omega^2 & \ldots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \ldots & \omega^{2(n-1)} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \ldots & \omega^{(n-1)^2} \end{pmatrix}.$$

$D_{n,\omega}$ is called the DFT-matrix. We write D_n or simply D if the context is clear. It has marvellous properties (see [3]).

Theorem 2. (i) $\hat{\mathbf{a}} = \mathbf{a} \cdot D_{n,\omega}$ (so simultaneous polynomial evaluation becomes matrix-vector multiplication)

(ii) $D_{n,\omega}^{-1} = \frac{1}{n} D_{n,\omega^{-1}}$ (so D is "essentially self-inverse", and multiplying by D^{-1} amounts to interpolation)

(iii) If $p, q \in F[x]$ then $p \cdot q = D^{-1}(Dp \cdot Dq)$ (so D translates convolutions into component-wise products)

In general, the multiplication of an $n \times n$ matrix with a vector requires $O(n^2)$ operations. A famous feature of D_n is that matrix-vector multiplication can be done with only $O(n \log n)$ operations. This is known as *Fast Fourier Transform* (FFT) and rests on the decomposition (in which we assume that n is even)

$$D_{n,\omega} = \begin{pmatrix} I_{n/2} & \Delta \\ I_{n/2} & -\Delta \end{pmatrix} \begin{pmatrix} D_{n/2,\omega^2} & 0 \\ 0 & D_{n/2,\omega^2} \end{pmatrix} P,$$

where $\Delta = \mathrm{diag}(1, \omega, \ldots, \omega^{n/2}) \in F^{n/2 \times n/2}$ and $P \in F^{n \times n}$ is the permutation matrix that maps $(x_0, x_1, \ldots, x_{n-1})$ to $(x_0, x_2, \ldots, x_{n-2}, x_1, x_3, \ldots, x_{n-1})$. If n is a power of two, repeated application of this decomposition allows us to write D_n as a product of $O(\log n)$ many matrices each of which is so simple that matrix-vector multiplication can be done in linear time. This is the essence of the Cooley-Tuckey algorithm [3]. Gilbert Strang said in 1994: "The FFT is the most important numerical algorithm in our lifetime"! We agree with "most important" but not with the restriction to "numerical". The FFT is an important algebraic algorithm as well.

Also, the largest component $\hat{a}_i$ (for $F = \mathbb{C}$) gives the most important information on $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$. This can be used for data compression. In the theory of signal processing and mechatronics, it is customary to say that one works in the "time domain" if data a are studied. In contrast, when working with their Fourier transforms, one says one is in the "phase domain".

So, we have a diagram

$$F[x] \longrightarrow F[x]/(f) \longrightarrow F^n$$

with return map

$$F[x]/(f) \longleftarrow F^n,$$

where $\longleftarrow$ is either Chinese Remainder or interpolation or D^{-1}.

Observe that $\longleftarrow$ does not go to $F[x]$ ("wrong address")! Never mind; there might be many $q \in F[x]$ which have the same equivalence class as p, but then q differs from p by a polynomial of higher degree than $\deg(f)$, which we had excluded.

Instance 4 – Generalized

This can be even further generalized (for details see [2]). Recall that $F[x]/(x^n - 1)$ is isomorphic to the group ring $F[C_n]$ over the cyclic group C_n, and $F[\mathbb{Z}]$ is isomorphic to the ring of Laurent

polynomials in x. So one might shift the interest to group rings. Maschke's Theorem yields a direct decomposition into simple ideals, and representation theory can switch on its powerful machinery to make everything constructive. For example, we have

- $\mathbb{Q}[C_3] \cong \mathbb{Q} \oplus \mathbb{Q}[x]/(x^2+x+1)$,
- $\mathbb{C}[S_3] \cong \mathbb{C} \oplus \mathbb{C} \oplus M_{2\times 2}(\mathbb{C})$.

In these group rings, again a "general Discrete Fourier Transform" is available.

Instance 5: Solving equations of degree ≥ 5?

The successes of parallel computations so far might give rise to another hope: Can we get around the "Abel-Galois-barrier 5" that there cannot be "formulas" for solving equations of degree ≥ 5?

A zero z of a polynomial $p \in F[x]$ is an element $z \in F$ such that $\bar{p}(z) = 0$. In other words, we have $p \circ z = 0$, where $\circ$ denotes the composition of polynomials.

So we "expand" the polynomial ring $(F[x], +, \cdot)$ to the "composition ring" $(F[x], +, \cdot, \circ)$. This means that $(F[x], +, \cdot)$ is a ring and $(F[x], +, \circ)$ is a "near-ring" (addition is not necessarily abelian, and we have just one distributive law), see [6]. Can we find a "full ideal" (= kernel of a composition ring homomorphism), say (f) again, such that $(F[x], +, \cdot, \circ)/(f)$ can be decomposed so that we can work in factors with equivalence classes of polynomials of low degree, find zeroes there, and use the Chinese Remainder Theorem again to get a zero in F?

The following result puts a stop to these considerations:

Theorem 3 (Nöbauer [5]). : The composition ring $(F[x], +, \cdot, \circ)$ is simple iff F is an infinite field.

Game over!

Instance 6: Other decompositions

What about other decomposition results? Two examples:

Theorem 4. (i) The group $(\mathbb{R}/\mathbb{Z}, +)$ is the direct sum of copies of $\mathbb{Q}$ and of groups of the type $\mathbb{Z}_p^\infty$.

(ii) Every lattice is a subdirect product of subdirectly irreducible lattices.

In these cases, it seems to be much harder to retrieve concrete information about the structure which was decomposed. In particular, we do not have a convenient map $\longleftarrow$ from the direct product back into the algebra.

Summary

If a decomposition of an algebra $\mathscr{A}$ into algebras $\mathscr{A}_i$ should be "useful" (for parallel computing, and the like), there should be:

- a "good" map $\longrightarrow$ from $\mathscr{A}$ to the algebras $\mathscr{A}_i$,
- a "sufficiently good and constructive" knowledge of the components $\mathscr{A}_i$, AND
- a "convenient" map $\longleftarrow$ from the decomposition back into $\mathscr{A}$.

References

[1] Janos Aczel. *On Applications and Theory of Functional Equations.* Academic Press, 2014.

[2] Thomas Beth. *Verfahren der schnellen Fourier-Transformation.* Teubner, 1984.

[3] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra.* Cambridge University Press, 1999.

[4] Rudolf Lidl and Harald Niederreiter. *Finite Fields.* Cambridge University Press, 1996.

[5] Hans Lausch and Winfried Nöbauer. *Algebra of Polynomials.* North-Holland, 1973.

[6] Günter Pilz. *Near-Rings.* North-Holland, 1983.

Author information

Manuel Kauers and Günter Pilz, Institute for Algebra, Johannes Kepler University, 4040 Linz, Austria.
E-mail: `manuel.kauers@jku.at, guenter.pilz@jku.at`