Decompositions in algebra

Gent, D.M.H. van

Publications and preprints

Parts of this thesis are based on the following publications and preprints.

D. M. H. van Gent. Nonabelian flows in networks. *Journal of Graph Theory*, 104(1):245–256, 2023. https://doi.org/10.1002/jgt.22958

D. M. H. van Gent. Indecomposable algebraic integers, 2021. https://arxiv.org/abs/2111.00499

H. W. Lenstra Jr., A. Silverberg, and D. M. H. van Gent. Realizing orders as group rings. *Journal of Algebra*, 2023. https://doi.org/10.1016/j.jalgebra.2023.11.017

The following paper is not part of this thesis.

M. J. H. van den Bergh, S. T. Castelein, and D. M. H. van Gent. Order versus chaos. In *2020 IEEE Conference on Games (CoG)*, pages 391–398, 2020. https://doi.org/10.1109/CoG47356.2020.9231895

Contents

Preliminaries

Definitions

In this section we will list some definitions and notations that we assume a reader of the coming chapters be familiar with.

Rings We write $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$ for the rings of integers, rationals, reals and complex numbers respectively. We will avoid the use of the term 'natural numbers' and instead write $\mathbb{Z}_{\geq 0} = \{n \in \mathbb{Z} \mid n \geq 0\}$ for the set of non-negative integers and $\mathbb{Z}_{>0} = \{n \in \mathbb{Z} \mid n > 0\}$ for the set of positive integers. Like in any modern mathematical text, our rings include a multiplicative identity element.

Let R be a ring and let $x \in R$. We say x is *regular* if the maps $R \to R$ given by $y \mapsto xy$ and $y \mapsto yx$ are injective, while x is a *zero-divisor* if it is not regular. We say x is a *unit* if both maps are bijective, and we write R^* for the group of units of R. For a positive integer n, we say x is an *n-th root of unity* if $x^n = 1$, and write $\mu_n(R)$ for the subgroup of R^* of n-th roots of unity when R is commutative. We say x is a *root of unity* if it is an n-th root of unity for some n, and analogously write $\mu(R)$ for the group of roots of unity when R is commutative. We say x is *idempotent* if $x^2 = x$ and write $\mathrm{Id}(R)$ for the set of idempotents of R. We say x is *nilpotent* if $x^n = 0$ for some positive integer n. We write $\mathrm{nil}(R)$ for the set of nilpotents of R, which we call the *nilradical* when R is commutative. We write $\mathrm{Jac}(R) = \{x \in R \mid 1 + RxR \subseteq R^*\}$ for the *Jacobson radical* of R.

Suppose now that R is a commutative ring. We say R is *connected* if it has exactly two idempotents, i.e. the only idempotents are 0 and 1 and $R \neq 0$. We say R is *reduced* if 0 is the only nilpotent of R.

A *number field* is a field of characteristic 0 which has finite dimension as a $\mathbb{Q}$-module. An *order* is a commutative ring whose additive group is

isomorphic to $\mathbb{Z}^n$ for some $n \in \mathbb{Z}_{\geq 0}$. Note that we do not require, as some authors do, that an order be contained in a number field, i.e. that it is an integral domain. Instead, we say R is an order of a number field K if $R \subseteq K$ is an order such that R generates K as $\mathbb{Q}$-module.

Modules Let R be a ring and M and N be (left) R-modules. For a morphism $f\colon M \to N$ of R-modules we write $\ker(f) = \{m \in M \mid f(m) = 0\}$ for the *kernel* of f, write $\operatorname{im}(f) = \{f(m) \mid m \in M\}$ for the *image* of f and write $\operatorname{coker}(f) = N/\operatorname{im}(f)$ for the *cokernel* of f. For a set I, an *I-indexed decomposition* of M is a family $\{M_i\}_{i\in I}$ of R-submodules of M such that the natural map $\bigoplus_{i\in I} M_i \to M$ is an isomorphism, a condition we abbreviate by $\bigoplus_{i\in I} M_i = M$. A *decomposition* of M is an I-indexed decomposition for any set I. We say M is *indecomposable* if $M \neq 0$ and for all $M_1, M_2 \subseteq M$ such that $M_1 \oplus M_2 = M$ we have $M_1 = 0$ or $M_2 = 0$. We make the class of decompositions of M into a (locally small) category, where the morphisms from $\{M_i\}_{i\in I}$ to $\{N_j\}_{j\in J}$ are the maps $f\colon I \to J$ such that $N_j = \bigoplus_{i\in f^{-1}\{j\}} M_i$ for all $j \in J$. For commutative R and $r \in R$ we write $M[r] = \{m \in M : rm = 0\}$ for the *r-torsion submodule* and $M[r^\infty] = \bigcup_{n\geq 0} M[r^n]$. For a set S we write $M^S = \prod_{s\in S} M$ and $M^{(S)} = \bigoplus_{s\in S} M$.

Graphs A *simple graph* is a pair (V, E) where V is a (potentially infinite) set and E is a set of size-2 subsets of V. Let $G = (V, E)$ be a graph. We call the elements of V the *vertices* of G and those of E its *edges*. A *subgraph* of G is a simple graph (W, F) with $W \subseteq V$ and $F \subseteq E$. For $W \subseteq V$ we call $(W, \{\{u, v\} \in E \,|\, u, v \in W\})$ the subgraph of G *induced by* W. For $u \in V$ we call $v \in V$ a *neighbor* of u if $\{u, v\} \in E$. A *connected component* of G is a non-empty set $S \subseteq V$ such that for all $e \in E$ we have $e \subseteq S$ or $e \subseteq V \setminus S$ and which is minimal with respect to inclusion given these properties. We say a simple graph is *connected* if it has precisely one connected component; in particular V is non-empty.

Algorithms

In this thesis we will encounter several algorithms, ranging from theoretical to computational in nature. The computational algorithms [19, 21] are programmed in either Sage [41] or GAP [15]. For our theoretical algorithms it would be important to specify our model of computation and the encoding of our mathematical objects, were it not that in our complexity analyses we will at best prove that the algorithm in question terminates in polynomial time. Assuming we remain sensible, polynomial runtime is invariant

under choice of model and encodings. Thus we allow ourselves in the coming chapters the luxury to reason about these algorithms in an informal and conceptual manner, rather than worry about implementation details. Nonetheless, in this section we will state some basic encodings and results that we will later implicitly use.

We let k be either $\mathbb{Z}$ or $\mathbb{Q}$ and assume there to be some sensible encoding of its elements.

Modules We encode a *linear map* $\alpha\colon k^m \to k^n$ as a matrix, i.e. a k-valued $(n \times m)$-tuple, in the obvious way. We encode a *finitely generated k-module* by a linear map $\alpha\colon k^m \to k^n$, where the corresponding module is $\operatorname{coker}(\alpha)$. The elements of $\operatorname{coker}(\alpha)$ are encoded by representatives in k^n. For $\alpha\colon k^m \to k^n$ and $\beta\colon k^r \to k^s$, a *morphism of finitely generated k-modules* $f\colon \operatorname{coker}(\alpha) \to \operatorname{coker}(\beta)$ is encoded as a linear map $\varphi\colon k^n \to k^s$ on the underlying representations of the elements of $\operatorname{coker}(\alpha)$ and $\operatorname{coker}(\beta)$. Note that every linear map $\alpha\colon k^m \to k^n$ encodes a valid k-module. However, with α and β as above, not every $\varphi\colon k^n \to k^s$ encodes a valid morphism $\operatorname{coker}(\alpha) \to \operatorname{coker}(\beta)$, as it does so if and only if $\operatorname{im}(\varphi\alpha) \subseteq \operatorname{im}(\beta)$. This is generally not a problem however, since our algorithms are only required to work for legal input. Regardless, we can test whether φ encodes a morphism. A submodule of A is encoded as a module A' together with an injective morphism $A' \to A$.

If $k = \mathbb{Q}$, then using linear algebra one can answer most questions about finitely generated k-modules in polynomial time. This also includes computing kernels and images of linear maps, computing hom-sets and splitting exact sequences. For $k = \mathbb{Z}$ one can find an exposition on algorithms for basic questions and constructions involving finitely generated k-modules in Chapter 2 of [5].

Rings A *k-algebra* structure on a finitely generated k-module represented by some map $k^m \to k^n$ is encoded as a k-valued $(n \times n \times n)$-tuple $(e_{hij})_{h,i,j}$, where the multiplication is given by

$$(a_h)_h \cdot (b_i)_i = \Big(\sum_{h,i} a_h b_i e_{hij} \Big)_j .$$

In particular, we can encode finite rings, orders and number fields. We may also compute quotient rings. For some commutative ring R with a given encoding we encode the elements of the *polynomial ring* $R[X]$ as a sequence $(n, f_0, \ldots, f_n)$ with $n \in \mathbb{Z}_{\geq 0}$ and $f_0, \ldots, f_n \in R$, where the corresponding polynomial is given by $\sum_{i=0}^{n} f_i X^i$.

Number fields Suppose for some set S we have an encoding of its elements. We say a map $\varphi\colon S \to \mathbb{C}$ is *computable* if there exists an algorithm A that takes as input an $s \in S$ and $n \in \mathbb{Z}_{\geq 0}$ and computes some $a \in \mathbb{Q}(\mathrm{i})$ such that $|\varphi(s) - a| \leq 2^{-n}$. We represent computable maps by such an algorithm. We say a complex number is computable if the map $\{0\} \to \mathbb{C}$ given by $0 \mapsto z$ is computable. It is undecidable whether two computable complex numbers are equal.

For any k-algebra A which is finitely generated and free as a module one can find for each element of A its minimal polynomial using linear algebra. For a number field K it is possible to factor polynomials over K into irreducibles in polynomial time [31]. Repeated application allows us to compute the splitting field of such a polynomial, although not generally in polynomial time. The roots (with multiplicity) of $f \in \mathbb{Q}[X]$ in $\mathbb{C}$ are computable [25]. Using this, each ring homomorphism $K \to \mathbb{C}$ is computable and we may compute the set of ring homomorphisms $K \to \mathbb{C}$.