ALGORITHMS FOR LATTICE GAMES

ALAN GUO AND EZRA MILLER

Abstract. This paper provides effective methods for the polyhedral formulation of impartial finite combinatorial games as lattice games [GMW09, GM09]. Given a rational strategy for a lattice game, a polynomial time algorithm is presented to decide (i) whether a given position is a winning position, and to find a move to a winning position, if not; and (ii) to decide whether two given positions are congruent, in the sense of misère quotient theory [Pla05, PlSi07]. The methods are based on the theory of short rational generating functions [BaWo03].

1. Introduction

In [GM09], we reformulated the theory of finite impartial combinatorial games in the language of combinatorial commutative algebra and convex rational polyhedral geometry. In general, the data provided by a lattice game—a rule set and a game board—do not allow for easy computation. Our purpose in this note is to provide details supporting the claim that *rational strategies* (Definition 3.2) and *affine stratifications* (Definition 4.1), two data structures introduced to encode the set of winning positions of a lattice game [GM09], allow for efficient computation of winning strategies using the theory of short rational generating functions [BaWo03]. For example, given a rational strategy, which is a short rational generating function for the set of winning positions, computations of simple Hadamard products decide, in polynomial time, whether any particular position in a lattice game is a winning or losing position, and which moves lead to winning positions if the starting position is a losing position (Theorem 3.6). Thus the algorithms produce a winning strategy whose time is polynomial in the input complexity (Definition 2.4) once certain parameters, such as dimension of the lattice game, are held constant. Other algorithms extract rational strategies from affine stratifications (Theorem 5.1) in polynomial time. None of these results are hard, given the theory developed by Barvinok and Woods [BaWo03], but it is worth bringing these methods to the attention of the combinatorial game theory community. In addition, the details require care, and a few results of independent interest arise along the way, such as Theorem 4.2, which says that the complement of a set with an affine stratification possesses an affine stratification.

Lattice games that are *squarefree*, [GM09, Erratum], generalizing the well-known heap-based *octal games* [GuSm56] (or see [GM09, Example 6.4]) such as Nim [Bou02]

Date: 5 September 2011.

Key words and phrases. combinatorial game, lattice game, convex polyhedron, generating function, affine semigroup, misère quotient.

and DAWSON'S CHESS [Daw34] (with bounded heap size), have tightly controlled structure in their sets of winning positions under normal play [GM09, Theorem 6.11]. However, our efficient algorithm for computing the set of winning positions in normal-play squarefree games [GM09, Theorem 7.4] fails to extend to misère play, where the final player to move loses. A key long-term goal of this project is to find efficient algorithms for computing winning strategies in misère squarefree games, particularly DAWSON'S CHESS (Remark 3.7). As a first step, we conjecture that every squarefree lattice game—under normal play, ordinary misère play, or the more general misère play allowed by the axioms of lattice games—possesses an affine stratification (Conjecture 4.5). We had earlier conjectured that every lattice game, squarefree or not, possesses an affine stratification [GM09, Conjecture 8.9], but Alex Fink has disproved that by showing general lattice games to be far from behaving so calmly [Fin11].

Other data structures, notably misère quotients [Pla05, PlSi07], that encode winning strategies in families of games suffer as much as rational strategies do in the face of Fink's aperiodicity: sufficiently aperiodic sets of P-positions induce trivial misère equivalence relations. Consequently, the bipartite monoid structure from misère quotient theory [Pla05, PlSi07] results in an easy monoid (a free finitely generated commutative monoid) with an inscrutable bipartition (the aperiodic set of P-positions). Nonetheless, it remains plausible that misère quotients extract valuable information about general squarefree games. As such, it is useful to continue the quest for algorithms to compute misère quotients, which have already led to advances in our understanding of octal games with finite quotients [Pla05, PlSi07]; see [Wei09] for recent progress in some cases. Theorem 6.2 is a step toward computing infinite misère quotients from rational strategies: it gives an efficient test for misère equivalence.

The results in this note reduce the problem of efficiently finding a winning strategy for a given family of finite impartial combinatorial games specified by a single rule set to the problem of efficiently computing an affine stratification, or even merely a rational strategy. Neither of these would complete the solution of, say, DAWSON'S CHESS in polynomial time, because the polynomiality here assumes bounded heap size, but they would be insightful first steps.

Acknowledgements. The seminal idea for this work emerged from extended discussions at Dalhousie University with the participants of Games-At-Dal-6 (July, 2009). Thanks also go to Sasha Barvinok for a discussion of computations with generating functions, and for pointing out his beautiful survey article [Bar06]. Partial funding was provided by NSF Career DMS-0449102 = DMS-1014112 and NSF DMS-1001437.

2. LATTICE GAMES

Precise notions of complexity require a review of the axioms for lattice games from [GM09]. To that end, fix a pointed rational convex polyhedron $\Pi \subset \mathbb{R}^d$ with recession cone C of dimension d. Write $\Lambda = \Pi \cap \mathbb{Z}^d$ for the set of integer points in Π.

Definition 2.1. Given an extremal ray ρ of a cone C, the *negative tangent cone* of C along ρ is $-T_\rho C = -\bigcap_{H\supset\rho} H_+$ where $H_+ \supseteq C$ is the positive closed halfspace bounded by a supporting hyperplane H for C.

Definition 2.2. A finite subset $\Gamma \subset \mathbb{Z}^d \smallsetminus \{\mathbf{0}\}$ is a *rule set* if

1. there exists a linear function on $\mathbb{R}^d$ that is positive on $\Gamma \cup C \smallsetminus \{\mathbf{0}\}$; and
2. for each ray ρ of C, some vector $\boldsymbol{\gamma}_\rho \in \Gamma$ lies in the negative tangent cone $-T_\rho C$.

Definition 2.3. Given the polyhedral set $\Lambda = \Pi \cap \mathbb{Z}^d$, fix a rule set Γ.

- A *game board* $\mathcal{B}$ is the complement in Λ of a finite Γ-order ideal in Λ called the set of *defeated positions*.
- A *lattice game* is defined by a game board and a rule set.

Definition 2.4 (Input complexity of a lattice game). Let $(\Gamma, \mathcal{B})$ be a lattice game with rule set Γ and game board $\mathcal{B}$. Γ may be represented as a $d \times n$ matrix with entries γ_{ij} for $1 \le i \le d$ and $1 \le j \le n$, where $n = |\Gamma|$. The game board $\mathcal{B}$ may be represented by the m generators of the finite Γ-order ideal, hence a $d \times m$ matrix with entries a_{ij} for $1 \le i \le d$ and $1 \le j \le m$. The *input complexity* of the lattice game is the number of bits needed to represent these $d(m+n)$ numbers, namely

$$d(m+n) + \sum_{i=1}^{d} \Big(\sum_{j=1}^{n} \log_2 |\gamma_{ij}| + \sum_{j=1}^{m} \log_2 |a_{ij}| \Big).$$

3. Rational strategies as data structures

Definition 3.1. For $A \subseteq \mathbb{Z}^d$, the *generating function* for A is the formal series

$$f(A;\mathbf{t}) = \sum_{\mathbf{a}\in A} \mathbf{t}^{\mathbf{a}}.$$

Definition 3.2. A *rational strategy* for a lattice game is a generating function for the set of P-positions of the form

$$f(A;\mathbf{t}) = \sum_{i\in I} \alpha_i \frac{\mathbf{t}^{\mathbf{p}_i}}{(1-\mathbf{t}^{\mathbf{a}_{i1}})\cdots(1-\mathbf{t}^{\mathbf{a}_{ik(i)}})},$$

for some finite set I, nonnegative integers $k(i)$, rational numbers α_i, along with vectors $\mathbf{p}_i, \mathbf{a}_{ij} \in \mathbb{Z}^d$ and $\mathbf{a}_{ij} \neq \mathbf{0}$ for all i, j [GM09, Definition 8.1]. A rational strategy is *short* if the number $|I|$ of indices is bounded by a polynomial in the input complexity.

Definition 3.3 (Complexity of short rational generating functions). Fix a positive integer k. Let $A \subseteq \mathbb{Z}^d$ and

$$f(A;\mathbf{t}) = \sum_{i\in I} \alpha_i \frac{\mathbf{t}^{\mathbf{p}_i}}{(1-\mathbf{t}^{\mathbf{a}_{i1}})\cdots(1-\mathbf{t}^{\mathbf{a}_{ik}})}$$

for some vectors $\mathbf{p}_i, \mathbf{a}_{ij} \in \mathbb{Z}^d$ and $\mathbf{a}_{ij} \neq 0$ for all i, j. If $\mathbf{p}_i = (p_{i1}, \ldots, p_{id})$ and $\mathbf{a}_{ij} = (a_{ij1}, \ldots, a_{ijd})$ for all i, j, and α_i is given as a ratio $\pm\frac{\alpha_i'}{\alpha_i''}$ of positive integers, then the *complexity* of $f(A; \mathbf{t})$ is the number

$$|I|(1 + d + kd) + \sum_{i \in I} \Big(\log_2 \alpha_i' + \log_2 \alpha_i'' + \sum_{j=1}^{d} \log_2 |p_{ij}| + \sum_{j=1}^{k} \sum_{r=1}^{d} \log_2 |a_{ijr}| \Big).$$

Definition 3.4 ([BaWo03, Definition 3.2])**.** For Laurent power series

$$f_1(\mathbf{t}) = \sum_{\mathbf{a} \in \mathbb{Z}^d} \beta_{1\mathbf{a}} \mathbf{t}^{\mathbf{a}} \quad \text{and} \quad f_2(\mathbf{t}) = \sum_{\mathbf{a} \in \mathbb{Z}^d} \beta_{2\mathbf{a}} \mathbf{t}^{\mathbf{a}}$$

in $\mathbf{t} \in \mathbb{C}^d$, the *Hadamard product* $f = f_1 \star f_2$ is the power series

$$f(\mathbf{t}) = \sum_{\mathbf{a} \in \mathbb{Z}^d} (\beta_{1\mathbf{a}} \beta_{2\mathbf{a}})\, \mathbf{t}^{\mathbf{a}}.$$

Lemma 3.5. *Fix k. Let $A, B \subseteq \mathbb{Z}^d$ lie in a common pointed rational cone C. If $f(A; \mathbf{t})$ and $f(B; \mathbf{t})$ are rational generating functions with $\leq k$ denominator binomials in each, then there is an algorithm for computing $f(A; \mathbf{t}) \star f(B; \mathbf{t})$ as a rational generating function in polynomial time in the complexity of the generating functions.*

Proof. Choose an affine linear function ℓ that is negative on C. Write

$$\begin{aligned} f(A; \mathbf{t}) &= \sum_{i \in I} \alpha_i \frac{\mathbf{t}^{\mathbf{p}_i}}{(1 - \mathbf{t}^{\mathbf{a}_{i1}}) \cdots (1 - \mathbf{t}^{\mathbf{a}_{ik}})} \\ f(B; \mathbf{t}) &= \sum_{j \in J} \beta_i \frac{\mathbf{t}^{\mathbf{q}_i}}{(1 - \mathbf{t}^{\mathbf{b}_{j1}}) \cdots (1 - \mathbf{t}^{\mathbf{b}_{jk}})}, \end{aligned}$$

where $\mathbf{p}_i, \mathbf{q}_i \in \mathbb{Z}^d$, $\mathbf{a}_{ir}, \mathbf{b}_{jr} \in C$ for all i, j, r. Since $\langle \ell, \mathbf{a}_{ir} \rangle < 0$ and $\langle \ell, \mathbf{b}_{jr} \rangle < 0$ for all i, j, r, by Lemma 3.4 of [BaWo03] we can compute

$$\frac{\mathbf{t}^{\mathbf{p}_i}}{(1 - \mathbf{t}^{\mathbf{a}_{i1}}) \cdots (1 - \mathbf{t}^{\mathbf{a}_{ik}})} \star \frac{\mathbf{t}^{\mathbf{q}_i}}{(1 - \mathbf{t}^{\mathbf{b}_{j1}}) \cdots (1 - \mathbf{t}^{\mathbf{b}_{jk}})}$$

in polynomial time for each i, j. Since the Hadamard product is bilinear, it follows that we can compute $f(A; \mathbf{t}) \star f(B; \mathbf{t})$ in polynomial time as well. □

Theorem 3.6. *Any rational strategy for a lattice game produces algorithms for*

- *determining whether a position is a P-position or an N-position, and*
- *computing a legal move to a P-position, given any N-position.*

These algorithms run in polynomial time if the rational strategy is short.

Proof. Suppose we wish to determine whether $\mathbf{p} \in \mathcal{B}$ is a P-position or an N-position. Let $f(\mathcal{P}; \mathbf{t})$ be a rational strategy for the lattice game. By definition, $\mathcal{P}$ and $\mathbf{p}$ both lie in the cone C. It follows from Lemma 3.5 that we can compute $f(\mathcal{P} \cap \mathbf{p}; \mathbf{t}) =$

$f(\mathcal{P};\mathbf{t}) \star \mathbf{t}^{\mathbf{p}}$ in $O(\iota^c)$ time, where ι is the complexity of $f(\mathcal{P};\mathbf{t})$ and $\mathbf{t}^{\mathbf{p}}$, and c is some positive integer. We get

$$f(\mathcal{P} \cap \mathbf{p};\mathbf{t}) = \begin{cases} \mathbf{t}^{\mathbf{p}} & \text{if } \mathbf{p} \in \mathcal{P} \\ 0 & \text{if } \mathbf{p} \in \mathcal{N}. \end{cases}$$

Given an N-position $\mathbf{q}$, simply apply this algorithm to all positions $\mathbf{q} - \boldsymbol{\gamma}$ for each legal move $\boldsymbol{\gamma} \in \Gamma$. Since $\mathbf{q} \in \mathcal{N}$, at least one $\mathbf{q} - \boldsymbol{\gamma}$ lies in $\mathcal{P}$, hence this procedure will end in $O(\iota^c|\Gamma|)$ time. □

Remark 3.7. The eventual goal of this project is to solve Dawson's Chess. That is, given any position in Dawson's Chess, we desire efficient algorithms to determine whether the next player to move has a winning strategy, and if so, to find one. This is equivalent to determining whether a given position $\mathbf{p}$ is a P-position or an N-position. If $\mathbf{p} \in \mathcal{N}$, then the next player to move indeed has a winning strategy by moving the game to a P-position. This is the problem of determining those $\boldsymbol{\gamma} \in \Gamma$ for which $\mathbf{p} - \boldsymbol{\gamma}$ lies in $\mathcal{P}$. By Theorem 3.6, we can do all of this if we have a Dawson's Chess rational strategy for heaps of sufficient size. Alas, it is not known whether rational strategies exist for general squarefree games, or even for Dawson's Chess.

Definition 3.8. A lattice game with ruleset Γ and game board $\mathbb{N}^d \smallsetminus \mathcal{D}$ is *squarefree* if for each $\boldsymbol{\gamma} \in \Gamma$ and $\mathbf{p}, \mathbf{q} \in \mathbb{N}^d$, if $\mathbf{p} + \mathbf{q} - \boldsymbol{\gamma} \in \mathbb{N}^d$ then $\mathbf{p} - \boldsymbol{\gamma} \in \mathbb{N}^d$ or $\mathbf{q} - \boldsymbol{\gamma} \in \mathbb{N}^d$.

The notion of squarefree lattice games captures the notion of disjunctive sum for combinatorial games: the position $\mathbf{p} + \mathbf{q}$ is supposed to be the sum of two summands $\mathbf{p}$ and $\mathbf{q}$, and if we can make a move on $\mathbf{p} + \mathbf{q}$, we should be able to make a move on one of its summands. See [GM09, Erratum] and [Fin11] for further details.

Conjecture 3.9. *Every squarefree lattice game possesses a rational strategy.*

It is known that arbitrary lattice games need not possess rational strategies [Fin11]. The smallest known counterexample is on $\mathbb{N}^3$; its rule set has size 28.

Remark 3.10. The question, then, is how to find a rational strategy for Dawson's Chess. Observe that a fixed lattice game structure only suffices to encode a heap game for heaps of bounded size. Let G_n denote the lattice game corresponding to Dawson's Chess with heaps of size at most n. If we can find the rational strategy for any given n, then this is good enough, although we must be careful about the complexity of finding such rational strategies as a function of n. In the next sections, we shall see that affine stratifications serve as data structures from which to extract the rational strategy in polynomial time. Thus the problem will be reduced to finding affine stratifications for G_n for all n, and there is hope that some regularity might arise, as n grows, to allow the possibility of computing them in time polynomial in n.

4. Affine stratifications as data structures

Definition 4.1 ([GM09, Definition 8.6])**.** An *affine stratification* of a subset $\mathcal{W} \subseteq \mathbb{Z}^d$ is a partition

$$\mathcal{W} = \biguplus_{i=1}^{r} W_i$$

of $\mathcal{W}$ into a disjoint union of sets W_i, each of which is a finitely generated module for an affine semigroup $A_i \subset \mathbb{Z}^d$; that is, $W_i = F_i + A_i$, where $F_i \subset \mathbb{Z}^d$ is a finite set. An *affine stratification of a lattice game* is an affine stratification of its set of P-positions.

The choice to require an affine stratification of $\mathcal{P}$, as opposed to $\mathcal{N}$, may seem arbitrary, but in the end these are equivalent, due to the following result.

Theorem 4.2. *If A and $B \subset A$ both possess affine stratifications, then $A \smallsetminus B$ possesses an affine stratification.*

The plan for Theorem 4.2 is to show that removing a translated normal affine semigroup (an affine semigroup is *normal* if it is the intersection of a real cone with a lattice; see [MiSt05, Chapter 7]) from a normal affine semigroup yields an affinely stratified set, and intersecting affinely stratified sets results in an affinely stratified set.

Lemma 4.3. *Suppose B is the intersection of a rational convex polyhedron and a subgroup of $\mathbb{Z}^d$. If A is a normal affine semigroup and $\mathbf{b} + A \subset B$ for some $\mathbf{b} \in B$, then $B \smallsetminus (\mathbf{b} + A)$ has an affine stratification.*

Proof. First we assume that $\mathbf{b} = \mathbf{0}$ and that B is a normal affine semigroup and $\mathbb{R}_{\geq 0}A = \mathbb{R}_{\geq 0}B$. Since $A \subset B$, that means $\mathbb{Z}A$ is a sublattice of $\mathbb{Z}B$ in $\mathbb{Z}^d$, hence B can be written as a finite disjoint union of cosets of A.

Now, suppose B is an arbitrary intersection of a rational convex polyhedron Π_B and a lattice L, and $\mathbf{b} \in B$ is arbitrary. We will reduce to the previous case by "carving" away pieces of B that do not lie in $\mathbb{R}_{\geq 0}A$. Suppose $\mathbb{R}_{\geq 0}A$ has a facet (a $(d-1)$-dimensional face) which is not contained in a facet of Π_B. Let H be the bounding hyperplane of this facet and H_- the corresponding negative halfspace (the half that is outside of $\mathbb{R}_{\geq 0}A$). Then $H_- \cap \Pi_B$ is a rational convex polyhedron. to reduce the number of facets of $\mathbb{R}_{\geq 0}A$ which do not lie in a facet of Π_B. Thus we have "carved out" a piece $H_- \cap \Pi_B$ of Π_B. By [Mil10, Lemma 2.4], $H_- \cap \Pi_B \cap L$ is a finitely generated module over an affine semigroup. Now replace Π_B with $\Pi_B \smallsetminus H_-$ and repeat. Each time we repeat the argument, we carve out a piece of the original Π_B which has an affine stratification, and furthermore we reduce the number of facets of $\mathbb{R}_{\geq 0}A$ that do not lie in the current Π_B. Eventually we reduce to the case where each facet of $\mathbb{R}_{\geq 0}A$ lie in some facet of Π_B, which is actually the first case above where Π_B is a cone and $\mathbf{b} = \mathbf{0}$. By [Mil10, Corollary 2.8], the union of these pieces possesses an affine stratification.

There is a degenerate case when A is not d-dimensional, but then we may reduce to a lower dimension by carving away $\mathbb{Z}^d \smallsetminus A$. □

Lemma 4.4. *If $\mathcal{W}$ and $\mathcal{W}'$ have affine stratifications, then $\mathcal{W} \cap \mathcal{W}'$ has an affine stratification.*

Proof. By [Mil10, Theorem 2.6], we may write

$$\mathcal{W} = \biguplus_{i=1}^{r} W_i \quad \text{and} \quad \mathcal{W}' = \biguplus_{j=1}^{s} W_i'$$

where each W_i and W_j' is a translate of a normal affine semigroup. Therefore, it suffices to show that the intersection of a translate of a normal affine semigroup with a translate of another normal affine semigroup has an affine stratification, for the union of all of these intersections would then have an affine stratification, by [Mil10, Corollary 2.8].

Suppose our two translates are $\mathbf{a}_1 + A_1$ and $\mathbf{a}_2 + A_2$. If their intersection is empty, then trivially it has an affine stratification, so we may assume that there is some $\mathbf{a} \in (\mathbf{a}_1 + A_1) \cap (\mathbf{a}_2 + A_2)$. Then $\mathbf{a}_1 - \mathbf{a} + \mathbb{Z}A_1 = \mathbb{Z}A_1$ and $\mathbf{a}_2 - \mathbf{a} + \mathbb{Z}A_2 = \mathbb{Z}A_2$. Therefore

$$\begin{aligned}(\mathbf{a}_1 + \mathbb{Z}A_1) \cap (\mathbf{a}_2 + \mathbb{Z}A_2) &= \mathbf{a} + (\mathbf{a}_1 - \mathbf{a} + \mathbb{Z}A_1) \cap (\mathbf{a}_2 - \mathbf{a} + \mathbb{Z}A_2) \\ &= \mathbf{a} + (\mathbb{Z}A_1 \cap \mathbb{Z}A_2),\end{aligned}$$

i.e., the intersection of the cosets is itself a coset of a lattice. Moreover, the intersection $(\mathbf{a}_1 + \mathbb{R}_{\geq 0}A_1) \cap (\mathbf{a}_2 + \mathbb{R}_{\geq 0}A_2)$ is a polyhedron. By [Mil10, Lemma 2.4], since A_1 and A_2 are normal, we have

$$\begin{aligned}(\mathbf{a}_1 + A_1) \cap (\mathbf{a}_2 + A_2) &= ((\mathbf{a}_1 + \mathbb{R}_{\geq 0}A_1) \cap (\mathbf{a}_1 + \mathbb{Z}A_1)) \cap ((\mathbf{a}_2 + \mathbb{R}_{\geq 0}A_2) \cap (\mathbf{a}_2 + \mathbb{Z}A_2)) \\ &= ((\mathbf{a}_1 + \mathbb{R}_{\geq 0}A_1) \cap (\mathbf{a}_2 + \mathbb{R}_{\geq 0}A_2)) \cap ((\mathbf{a}_1 + \mathbb{Z}A_1) \cap (\mathbf{a}_2 + \mathbb{Z}A_2))\end{aligned}$$

is an intersection of a polyhedron with a coset of a lattice and hence is a finitely generated module over an affine semigroup. In particular, the intersection has an affine stratification. □

Proof of Theorem 4.2. First, assume A is a normal affine semigroup. Suppose

$$B = \biguplus_{i=1}^{r} B_i$$

where each B_i is a translate of a normal affine semigroup. By Lemma 4.3, each $A \smallsetminus B_i$ has an affine stratification. Therefore, by Lemma 4.4, $A \smallsetminus B = A \smallsetminus (\biguplus_{i=1}^{r} B_i) = \bigcap_{i=1}^{r}(A \smallsetminus B_i)$ has an affine stratification. For the general case where A has an affine stratification, each A_i reduces to the previous case, and then we obtain the result by taking the union. □

Conjecture 4.5. *Every squarefree lattice game possesses an affine stratification.*

Example 4.6. Consider again the game of NIM with heaps of size at most 2. An affine stratification for this game is $\mathcal{P} = 2\mathbb{N}^2$; that is, $\mathcal{P}$ consists of all nonnegative integer points with both coordinates even.

Example 4.7. The misère lattice game on $\mathbb{N}^5$ whose rule set forms the columns of

$$\Gamma = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

was one of the motivations for the definitions in [GM09] because the illustration of the winning positions in this lattice game provided by Plambeck and Siegel [PlSi07, Figure 12] possesses an interesting description as an affine stratification. An explicit description can be found in [GM09, Example 8.13].

In what follows, we define the complexity of an affine stratification to be the complexity of the generators and the affine semigroups involved. Roughly speaking, the complexity of an integer k is its binary length (more precisely, $1 + \lceil \log_2 k \rceil$), so the complexity is roughly the sum of the binary lengths of the integer entries of the generators in the finite sets F_i and the coefficients of the vectors generating the affine semigroups A_i; see [Bar06, Section 2] for additional details. To say that an algorithm is *polynomial time when the dimension d is fixed* means that the running time is bounded by $\iota^{\phi(d)}$ for some fixed function ϕ, where ι is the complexity.

Definition 4.8 (**Complexity of an affine semigroup**)**.** Fix an affine semigroup $A = \mathbb{N}\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$ in $\mathbb{Z}^d$. Let $\mathbf{a}_i = (a_{1i}, \ldots, a_{di})$. Then A may be represented by a $d \times n$ matrix with entries a_{ij}. The *complexity* of A is the number of bits needed to represent these dn numbers, which is equal to

$$dn + \sum_{i=1}^{n} \sum_{j=1}^{d} \log_2 |a_{ij}|.$$

Definition 4.9 (**Complexity of an affine stratification**)**.** Let

$$\mathcal{P} = \biguplus_{i=1}^{r} W_i$$

be an affine stratification, where $W_i = F_i + A_i$ for some affine semigroup $A_i \subset \mathbb{Z}^d$ and finite set $F_i \subset \mathbb{Z}^d$. Let $m_i = |F_i|$ and $F_i = \{\mathbf{b}_{i1}, \ldots, \mathbf{b}_{im_i}\}$ where $\mathbf{b}_{ij} = (b_{ij1}, \ldots, b_{ijd})$, and let $A_i = \mathbb{N}\{\mathbf{a}_{i1}, \ldots, \mathbf{a}_{in}\}$, where $\mathbf{a}_{ij} = (a_{ij1}, \ldots, a_{ijd})$. The *complexity* of the affine stratification is the number

$$d\Big(nr + \sum_{i=1}^{r} m_i\Big) + \sum_{i=1}^{r} \sum_{s=1}^{d} \Big(\sum_{j=1}^{n} \log_2 |a_{ijs}| + \sum_{j=1}^{m_i} \log_2 |b_{ijs}| \Big)$$

of bits needed to represent each F_i and A_i.

Remark 4.10. The existence of affine stratifications as in [GM09, Conjecture 8.9] is equivalent to the same statement with the extra hypothesis that the rule set generates a saturated (also known as "normal") affine semigroup. There are also a number of ways to characterize the existence of affine stratifications, in general [Mil10, Theorem 2.6], using various combinations of hypotheses such as normality of the affine semigroups involved, or disjointness of the relevant unions. However, some of these freedoms increase complexity in untamed ways, and are therefore unsuitable for efficient algorithmic purposes. Definition 4.1 characterizes the notion of affine stratification in the most efficient terms, where algorithmic computation of rational strategies is concerned; allowing the unions to overlap would make it easier to find affine stratifications, but harder to compute rational strategies from them.

5. Computing rational strategies from affine stratifications

In this section, we prove the following.

Theorem 5.1. *A rational strategy can be algorithmically computed from any affine stratification, in time polynomial in the input complexity of the affine stratification when the dimension d is fixed and the numbers of module generators over the semigroups A_i are uniformly bounded above.*

The proof of the theorem requires a few intermediate results, the point being simply to keep careful track of the complexities of the constituent elements of affine stratifications.

Lemma 5.2. *Fix $k, d \in \mathbb{N}$. Let $A, B \subseteq \mathbb{Z}^d$ lie in the same pointed rational cone C. If $f(A; \mathbf{t})$ and $f(B; \mathbf{t})$ are short rational generating functions with $\leq k$ binomials in their denominators, then for some $c \in \mathbb{N}$ there is an $O(\iota^c)$ time algorithm for computing the rational function $f(A \cup B; \mathbf{t})$, where ι is an upper bound on the complexity of $f(A; \mathbf{t})$ and $f(B; \mathbf{t})$. If A and B are disjoint, then the complexity of $f(A \cup B; \mathbf{t})$ is bounded by 2ι, and $f(A \cup B; \mathbf{t})$ can be computed in $O(\iota)$ time.*

Proof. This follows from the fact that

$$f(A \cup B; \mathbf{t}) = f(A; \mathbf{t}) + f(B; \mathbf{t}) - f(A \cap B; \mathbf{t})$$

and that $f(A \cap B; \mathbf{t}) = f(A; \mathbf{t}) \star f(B; \mathbf{t})$ can be computed in polynomial time, by Lemma 3.5. □

Corollary 5.3. *Fix $k, d \in \mathbb{N}$. Let $A_1, \ldots, A_m \subseteq \mathbb{Z}^d$ lie in the same pointed rational cone C. If $f(A_1; \mathbf{t}), \ldots, f(A_r; \mathbf{t})$ are rational generating functions with $\leq k$ binomials in their denominators, and $A = A_1 \cup \cdots \cup A_m$, then for some $c \in \mathbb{N}$ there is an $O(2^m \iota^c)$ time algorithm for computing $f(A; \mathbf{t})$ as a rational generating function, where ι is an upper bound on the complexity of $f(A_1; \mathbf{t}), \ldots, f(A_r; \mathbf{t})$. If the A_i are pairwise disjoint, then the complexity bound is $O(m\iota)$.*

Proof. This follows from Lemma 5.2 and the fact that the number of binomials in the denominators in the rational generating functions may increase by a factor of up to 2 after computing each union. If the A_i are pairwise disjoint, then

$$f(A;\mathbf{t}) = \sum_{i=1}^{m} f(A_i;\mathbf{t})$$

and no intersections need to be computed. □

Lemma 5.4. *Fix n and d. If $A \subseteq \mathbb{Z}^d$ is a pointed affine semigroup generated by n integer vectors and has complexity ι, then for some positive integer c there is an $O(\iota^c)$ time algorithm for computing $f(A;\mathbf{t})$.*

Proof. Let $A = \mathbb{N}\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$. It is algorithmically easy to embed A into $\mathbb{N}^d$: if A has dimension d, then find d linearly independent facets and take their integer inner normal vectors as the columns of the embedding ν; if A has dimension $d' < d$, then find d' linearly independent facets and any $d - d'$ linear integer functions that vanish on A. Use the discussion of [BaWo03, Section 7.3] to compute $f(\nu(A);\mathbf{t})$. Then apply ν^{-1} to the exponents in $f(\nu(A);\mathbf{t})$ to get $f(A;\mathbf{t})$. □

Lemma 5.5. *Fix d. Let $W = F + A$, where $A \subseteq \mathbb{Z}^d$ is a pointed affine semigroup with complexity ι and $F \subseteq \mathbb{Z}^d$ is a finite set with $|F| = m$. For some $c \in \mathbb{N}$ there is an $O(2^m \iota^c)$ time algorithm for computing $f(W;\mathbf{t})$ as a rational function.*

Proof. Let $F = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$. Since F is finite, any linear function that is positive on $A \smallsetminus \{0\}$ is bounded below on W. Therefore, there exists a pointed rational cone that contains each $\mathbf{b}_j + A$. For each j, $f(\mathbf{b}_j + A;\mathbf{t}) = \mathbf{t}^{\mathbf{b}_j} f(A;\mathbf{t})$, each of which has complexity $O(\iota)$ and can be computed in $O(\iota^{c'})$ time, for some $c' > 0$, by Lemma 5.4. Since W is the union of the $\mathbf{b}_j + A$, it follows from Corollary 5.3 that $f(W;\mathbf{t})$ can be computed in $O(2^m \iota^c)$. □

We now return to proving our main theorem.

Proof of Theorem 5.1. Write

$$\mathcal{P} = \biguplus_{i=1}^{r} W_i$$

where $W_i = F_i + A_i$ for affine semigroups $A_i \subseteq \mathbb{Z}^d$ and finite sets $F_i \subseteq \mathbb{Z}^d$. Let ι be an upper bound on the complexity of each of the A_i. Since the sizes of the F_i are fixed, by Lemma 5.5 we can compute each $f(W_i;\mathbf{t})$ in $O(\iota^c)$ time, for some positive integer c. Since the W_i are pairwise disjoint, by Corollary 5.3 we can compute $f(\mathcal{P};\mathbf{t})$ in $O(r\iota^c)$ time. □

There is little hope that the complexity of calculating affine stratifications—or even merely rational strategies—should be polynomial in the input complexity when certain parameters are not fixed. Indeed, the complexity of the generating function for an affine semigroup fails to be polynomial in the number of its generators. Thus it makes sense to restrict complexity estimates to lattice games with rule sets of fixed complexity. On the other hand, there is hope that the complexity of an affine stratification should be bounded by the complexity of the rule set. Therefore, once the complexity of the rule set has been fixed, the algorithms dealing with affine stratifications could be polynomial.

6. Determining misère congruence

This section provides an algorithm to determine whether two given positions are misère congruent. The notion of misère congruence is simply the translation of "indistinguishability" [Pla05, PlSi07] into the language of lattice games.

Definition 6.1. Two positions $\mathbf{p}$ and $\mathbf{q}$ are *(misère) congruent* if

$$(\mathbf{p}+C)\cap\mathcal{P}-\mathbf{p}=(\mathbf{q}+C)\cap\mathcal{P}-\mathbf{q}.$$

Theorem 6.2. *Given a rational strategy $f(\mathcal{P};\mathbf{t})$ and $\mathbf{p},\mathbf{q}\in\mathcal{B}$, there is a polynomial time algorithm for determining whether $\mathbf{p}$ and $\mathbf{q}$ are misère congruent.*

Proof. Let $S_\mathbf{p}=(\mathbf{p}+C)\cap\mathcal{P}-\mathbf{p}$ and $S_\mathbf{q}=(\mathbf{q}+C)\cap\mathcal{P}-\mathbf{q}$. Since $\mathbf{p}\in C$, we have $\mathbf{p}+C\subseteq C$, and $\mathcal{P}\subseteq C$ by definition, so we may apply Lemma 3.5 to compute $f((\mathbf{p}+C)\cap\mathcal{P};\mathbf{t})$ in polynomial time. Then we can compute $f(S_\mathbf{p};\mathbf{t})$ in polynomial time since $f(S_\mathbf{p};\mathbf{t})=\mathbf{t}^{-\mathbf{p}}f((\mathbf{p}+C)\cap\mathcal{P};\mathbf{t})$. Similarly, we compute $f(S_\mathbf{q};\mathbf{t})$ in polynomial time. Then $\mathbf{p}$ and $\mathbf{q}$ are congruent if and only if $f(S_\mathbf{p};\mathbf{t})-f(S_\mathbf{q};\mathbf{t})=0$. □

Corollary 6.3. *Given an affine stratification of a lattice game, there is a polynomial time algorithm that decides on the misère congruence of any pair of positions.*

Proof. In polynomial time, Theorem 5.1 produces a rational strategy for the lattice game and then Theorem 6.2 decides on the congruence. □

References

[Bar06] Alexander Barvinok, *The complexity of generating functions for integer points in polyhedra and beyond*, International Congress of Mathematicians, Vol. III, 763–787, Eur. Math. Soc., Zürich, 2006.

[BaWo03] Alexander Barvinok and Kevin Woods, *Short rational generating functions for lattice point problems*, J. Amer. Math. Soc. **16** (2003), no. 4, 957–979. (electronic)

[Bou02] Charles L. Bouton, *Nim, a game with a complete mathematical theory*, Ann. of Math. (2) **3** (1901/02), no. 1–4, 35–39.

[Daw34] Thomas Dawson, *Fairy Chess Supplement*, The Problemist: British Chess Problem Society **2** (1934), no. 9, p. 94, Problem No. 1603.

[Fin11] Alex Fink, *Lattice games without rational strategies*, preprint (2011), 9 pages.

[GM09] Alan Guo and Ezra Miller, *Lattice point methods for combinatorial games*, Adv. in Appl. Math., **46** (2010), 363–378. doi:10.1016/j.aam.2010.10.004 arXiv:math.CO/0908.3473; Erratum: preprint, 2011.

[GMW09] Alan Guo, Ezra Miller, and Michael Weimerskirch, *Potential applications of commutative algebra to combinatorial game theory*, in *Kommutative Algebra*, abstracts from the April 19–25, 2009 workshop, organized by W. Bruns, H. Flenner, and C. Huneke, Oberwolfach rep. **22** (2009), pp. 23–26.

[GuSm56] Richard K. Guy and Cedric A. B. Smith, *The G-values of various games*, Proc. Cambridge Philos. Soc. **52** (1956), 514–526.

[Mil10] Ezra Miller, *Affine stratifications from finite misère quotients*, preprint, 2010. arXiv:math.CO/1009.2199v1

[MiSt05] Ezra Miller and Bernd Sturmfels, *Combinatorial Commutative Algebra*, Graduate Texts in Mathematics, vol. 227, Springer–Verlag, New York, 2005.

[Pla05] Thane E. Plambeck, *Taming the wild in impartial combinatorial games*, Integers **5** (2005), no. 1, G5, 36 pp. (electronic)

[PlSi07] Thane E. Plambeck and Aaron N. Siegel, *Misère quotients for impartial games*, J. Combin. Theory Ser. A **115** (2008), no. 4, 593–622. arXiv:math.CO/0609825v5

[Wei09] Michael Weimerskirch, *An algorithm for computing indistinguishability quotients in misère impartial combinatorial games*, preprint, 2009.

Department of Mathematics, Duke University, Durham, NC 27708

Current address, Guo: MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139

E-mail address: aguo@mit.edu, ezra@math.duke.edu