EVACUATION OF RECTANGULAR STANDARD YOUNG TABLEAUX AND CHIP-FIRING

ROHIT AGRAWAL, VLADIMIR SOTIROV, AND FAN WEI

ABSTRACT. We elaborate upon a bijection discovered by Cools et al. in [CDPR10] between rectangular standard Young tableaux and representatives, known as G-parking functions, of chip configurations on certain graphs modulo the chip-firing equivalence relation. We present an explicit formula for computing the G-parking function associated to a given tableau, and prove that evacuation on tableaux corresponds (under the bijection) to a reflection on the graph.

1. Introduction

In [CDPR10], Cools et al. investigate the divisors of the family of graphs Γ_g consisting of a chain of g loops with edge lengths $(\ell_1, \ldots, \ell_g, m_1, \ldots, m_g)$ that are generic in the sense that ℓ_i/m_i (as a reduced fraction) is not the ratio of two integers with sum less than $2g-2$. These graphs arise as the dual graphs of the special fiber of a strongly connected semistable regular model of smooth curves over a discretely valued field, and the analysis of their divisors is key to Cools et al.'s tropical proof of the Brill-Noether Theorem.

In the course of their proof, Cools et al. find a bijection between linear equivalence classes of rank r degree d divisors such that $g = (g-d+r)(r+1)$, and $(g-d+r) \times (r+1)$ rectangular standard Young tableaux. This bijection raises the natural question of what the standard operations on rectangular standard Young tableaux such as conjugation, evacuation, and promotion translate to for divisors on generic graphs Γ_g. In this paper we will prove our advisor Gregg Musiker's conjecture that evacuation corresponds to a reflection of the generic graph Γ_g. In the course of our proof, we will provide and use an explicit formula for computing the reduced divisor that represents the linear equivalence class of divisors associated to a given rectangular standard Young tableau.

The organization of our paper is as follows. In Section 2 we review the prerequisite theory of divisors on finite graphs. We do so using the language of the chip-firing game on finite graphs introduced by Baker and Norine in their seminal paper [BN07]. In Section 3 we review in detail the construction of the *lingering lattice paths* due to Cools et al. associated to (reduced) divisors of generic Γ_g, and through which Cools et al. derive the bijection to standard Young tableaux. Finally, in Section 4 we state and prove our results on explicit formulae for the bijection, as well as our theorem that evacuation on tableaux corresponds, via the bijection, to reflection of generic graphs Γ_g.

The reader who feels comfortable with the material in Sections 2 and 3 may proceed straight to Section 4, but is advised to familiarize themselves with Lemma 3.6,

Date: March 1, 2012.

which is crucial to our argument in Section 4. Of additional interest in Section 3 are Lemma 3.13 and Theorem 3.14 which constitute a slightly clearer proof of why lingering lattice paths determine the rank of a G-parking function on a graph Γ_g than the one present in Cools et al.'s original exposition.

2. The Chip-Firing Game and Divisors on Graphs

By a *graph* G we will mean a pair $G = (V, E)$ where V is a finite set of vertices, and where $E\colon V \times V \to \mathbb{Z}_{\geq 0}$ is an edge function assigning a number of edges between any two vertices. We will require throughout that the edges are undirected, i.e. that $E(u, v) = E(v, u)$ for all $u, v \in V$, and we will also disallow loops, i.e. $E(v, v) = 0$ for all $v \in V$, since we will have no use for edges that merely connect a vertex to itself.

We will say that a sequence of vertices $(v_1, \ldots, v_n)$ in a graph G is a *path* if $E(v_i, v_{i+1}) > 0$ for $1 \leq i < n$. We will say that a graph G is *connected* if for any two vertices v_1 and v_n there exists a sequence of vertices $(v_2, \ldots, v_{n-1})$ such that $(v_1, \ldots, v_n)$ is a path.

By a *full subgraph* H of $G = (V, E)$ determined by a subset $S \subset V$ of vertices of G, we will mean the graph $H = (S, E|_S)$ where E_S is the restriction of the edge function to that subset, i.e. the vertices of H are those specified by the subset $S \subset V$ and its edges are precisely those edges in G that join only vertices in S. Furthermore, for any full subgraph $H = (S, E_S)$ of $G = (V, E)$ we define $G \setminus H$ to be the full subgraph $(V \setminus S, E_{V \setminus S})$.

We may occasionally abuse notation by referring to a subset $S \subset V$ as a subgraph S of $G = (V, E)$ – whenever we do this, we mean the full subgraph of G generated by S.

2.1. The chip-firing game.

The *chip-firing game* has had a somewhat convoluted development. It was introduced in general by Dhar in [Dha90], where it is called the abelian sandpile model, and where the several key general results were proved. Independently, it was introduced as a game on graphs and named the "chip-firing game" by Bjorner, Lovasz, and Shor in [BLS91]. Finally, it has been reinterpreted by Baker and Norine in [BN07] as a convenient language for navigating certain facts regarding divisors on finite graphs and their linear equivalence classes. The term *divisor* has been borrowed from the theory of Riemann surfaces, as finite graphs can be seen as discrete analogues of Riemann surfaces. Since our work is purely combinatorial, we will only mention in passing how the theory of the chip-firing game on a finite graph translates to the theory of divisors; the interested reader may find the details of the connection in [BN07].

Definition 2.1. A *chip configuration* c on a graph G is an element of $\mathbb{Z}^V$, i.e. it is an assignment of an integer to every vertex of the graph. We interpret positive integers as numbers of "chips" and negative integers as "debts"; for any $c \in \mathbb{Z}^V$ we denote by $c(v)$ the number of chips at vertex v, with negative numbers implying debt.

Note that chip configurations on a graph are precisely the *divisors* on that graph.

Definition 2.2. We define the *Laplacian matrix* $L(G)$ *of a graph* G to be the $|V| \times |V|$ matrix $L(G) = (a_{vw})$ where $a_{vw} = \begin{cases} \deg(v) & v = w \\ -E(v, w) & v \neq w \end{cases}$.

Definition 2.3. *Firing a vertex* v is an operation on chip configurations which acts as follows: for every edge connecting v to a vertex w, a chip is removed from v and added to the vertex w. More formally, the configuration $c \in \mathbb{Z}^V$ is sent to the configuration $c' = c - P_v$ where P_v is the column corresponding to v in the Laplacian matrix $L(G)$.

Note that this definition immediately implies that firing a sequence of vertices is independent of the sequence's order since the columns of $L(G)$ certainly generate an abelian group. Therefore, it makes sense to talk about *firing a multiset S of vertices*, which will naturally be the transformation obtained by firing each $v \in S$ as many times as it occurs in S in whatever order. For the sake of simplicity, we will refer to firing a subset of vertices $S \subset V$ as *firing the subgraph H in G* where $H = (S, E_S)$ is a subgraph of G. The behavior of subgraph-firing will prove key to our development of the theory of the chip-firing game, so we proceed to define the notions of interior and boundary vertices of a full subgraph, which will allow us to explicitly describe subgraph-firing in Lemma 2.5.

Definition 2.4. If H is a full subgraph of G, define the *interior of H*, $\operatorname{Int} H$, to be the set of vertices in H such that no edge of G joins them to a vertex in $G \setminus H$. Define also the *boundary of H*, δH, to be the set of vertices of H for which there exists an edge of G joining them to $G \setminus H$.

Lemma 2.5 (Subgraph-firing)**.** *Let $G = (V, E)$ be a graph and let H be a subset of vertices. Then firing H inside of G has the following effects:*

(1) *the number of chips on the interior vertices* $\operatorname{Int} H$ *does not change;*
(2) *the number of chips on δH changes in exactly the same way it would by firing the set δH inside the subgraph of $G \setminus \operatorname{Int} H$ obtained by removing all edges joining vertices of δH to vertices of δH. In particular, the number of chips on each vertex of δH necessarily decreases.*

Proof. Suppose that firing H takes the initial configuration c to c', and consider any vertex $v \in V$. Then v both sends and receives a chip along each edge joining it to a vertex in H, while it only loses chips along edges joining it to vertices in $G \setminus H$. □

Corollary 2.6. *Firing all vertices of a graph G does not change the configuration c.*

Proof. Taking $H = G$ in the Subgraph-firing Lemma is enough since then the number of chips on $\operatorname{Int} G = G$ must remain unchanged. □

Corollary 2.7. *If firing v_0 in a graph $G = (V, E)$ sends a configuration c to c', then firing $V \setminus \{v_0\}$ is the inverse operation sending c' to c. More generally, the inverse of firing a multiset S of vertices in a graph G, with k being the largest number of occurrences of any particular vertex v in S, is firing the complement multiset $kV \setminus S$ where kV is the multiset consisting of k copies of each vertex of V.*

Proof. Let $H = G \setminus \{v_0\}$. Then the boundary δH consists of precisely those vertices in G that have an edge joining them to v_0, i.e. the neighbors of v_0. Note that $G \setminus \operatorname{Int} H = \delta H \cup \{v_0\}$.

Then the Subgraph-firing Lemma says that firing $G \setminus \{v_0\} = H$ in G is equivalent to firing δH in the subgraph of $G \setminus \operatorname{Int} H = \delta H \cup \{v_0\}$ obtained by removing all edges joining vertices of δH to vertices of δH. Since δH consists of the neighbors of v_0, it follows that the total effect will be sending a chip to v_0 along each edge incident to v_0, which is precisely the inverse of chip-firing v_0. □

This important corollary shows in particular that obtaining one chip configuration from another by a sequence of chip-firing operation is necessarily a symmetric relation between chip configurations. Since this relation is trivially reflexive and transitive, the corollary shows that it is an equivalence relation. In fact, this equivalence relation is the correct analogue of linear equivalence of divisors, as described in [BN07].

Definition 2.8. We say that two chip configurations c and c' are *chip-firing equivalent*, denoted $c \sim c'$, if one can be obtained from the other by a sequence of chip-firing operations.

The following lemma, easily proved by the Subgraph-firing Lemma, will be crucial to the investigation of chip configurations on the graphs Γ_g in the latter sections of the paper.

Lemma 2.9 (Conservation of momentum)**.** *Suppose that* $(v_{-1}, \dots, v_j, \dots, v_{j+k})$ *is a path in a graph* G *(the* v_i *do not have to be distinct), and suppose that there exist subgraphs* H_i *of* G *for* $0 \le i < j+k$ *such that:*

(1) $\delta H_i = \{v_i\}$;
(2) *the degree of each* v_i *in* $G \setminus \operatorname{Int} H_i$ *is* 2*, and the two edges in* $G \setminus \operatorname{Int} H_i$ *involving* v_i *join* v_i *to* v_{i-1} *and* v_{i+1}.

Then any chip configuration c *on* G *is chip-firing equivalent to the chip configuration* c' *on* G *obtained by transferring* k *chips from* v_0 *to* v_{-1} *and one chip from* v_j *to* v_{j+k}.

Proof. The equivalence is given by firing the sequences of graphs $(H_0, H_1, \dots, H_j)$, $(H_0, H_1, \dots, H_{j+1}), \dots, (H_0, H_1, \dots, H_{j+k-1})$. To see this, notice that the Subgraph-firing Lemma guarantees that firing H_i is the same as firing $\delta H_i = \{v_i\}$ in the subgraph $G \setminus \operatorname{Int} H_i$. By our hypothesis there are only two edges involving v_i in $G \setminus \operatorname{Int} H_i$, and they join v_i to v_{i-1} and v_{i+1}. Hence firing H_i in G for $0 \le i < j+k$ transfers one chip from v_i to v_{i-1} and one chip from v_i to v_{i+1}.

For $1 \le i < k+j$, firing $(H_0, \dots, H_i)$ is the same as first firing $(H_0, \dots, H_{i-1})$ and then firing H_i. Inductively, we may assume that firing $(H_0, \dots, H_{i-1})$ transfers one chip from v_0 to v_{-1} and one chip from v_{i-1} to v_i. Since firing H_i certainly transfers one chip from v_i to v_{i-1} and one chip from v_i to v_{i+1}, we have that firing $(H_0, \dots, H_{i-1})$ and then H_i overall transfers one chip from v_0 to v_{-1} and one chip from v_i to v_{i+1}. Similarly, a second induction shows that firing the sequences of graphs $(H_0, H_1, \dots, H_j)$, $(H_0, H_1, \dots, H_{j+1}))$, ..., $(H_0, H_1, \dots, H_{j+k-1})$ will overall transfer k chips from v_0 to v_{-1} and one chip from v_j to v_{j+k}, as desired. □

2.2. **Linear equivalence classes of divisors and their ranks.** The equivalence classes of chip configurations on a graph G under chip-firing are nothing more than the linear equivalence classes of divisors on the graph. The study of these equivalence classes for connected graphs G is facilitated by the existence of nice systems of representatives for each vertex v known as G-parking functions. These were first introduced in the context of graphs by Postnikov and Shapiro in [PS04] and were subsequently reinterpreted by Baker and Norine in [BN07] as v-reduced divisors.

Definition 2.10. We say that a chip configuration $c \in \mathbb{Z}^V$ for a connected graph $G = (V, E)$ is *G-parking relative to v_0* if $c(v) \geq 0$ for all $v \neq v_0$ and if firing any subset of vertices of $V \setminus \{v_0\}$ produces a new configuration c' such that $c'(w) < 0$ for some $w \neq v_0$.

In [Dha90], where Dhar introduced the general chip-firing game, called the abelian sandpile model at the time, he also presented a simple algorithm which in the case of (connected) graphs specializes to transforming any chip configuration c into its chip-firing equivalent G-parking function, thus demonstrating existence of G-parking functions in every equivalence class.

Theorem 2.11 (Dhar's Burning Algorithm)**.** *Fix a vertex v_0 of a connected graph G.*

Then every chip configuration on G is chip-firing equivalent to a G-parking function relative to v_0, and furthermore such a G-parking function is determined by the following algorithm.

(1) *Transform the configuration c into a configuration c' such that $c'(w) \geq 0$ for all $w \neq v_0$ by firing vertex v_0 a sufficient number of times and slowly forging a path to the vertices in c that are in debt.*
(2) *Start a "fire" from the vertex v_0 and let it travel along the edges of the graph;*
(3) *If a vertex has fewer chips than the number of edges along which the "fire" reaches it, declare the vertex "burnt" and continue the fire along all edges coming out of that vertex;*
(4) *If a vertex has more chips than the number of edges along which the "fire" reaches it, declare the vertex "unburnt" and do not perpetuate the "fire" along any more edges coming out of that vertex;*
(5) *The "unburnt" vertices determine a subgraph H of G: fire those vertices and then start a "fire" again;*
(6) *If the whole graph burns, then the configuration is a G-parking function relative to v_0.*

Proof. First, we make rigorous the first step of the algorithm, which claims that every chip configuration is equivalent to one that is non-negative on $V \setminus \{v_0\}$. Consider a path $P = (v_0, v_1, \ldots, v_n)$ that joins v_0 to a vertex v_n such that $c(v_n) < 0$ and induct on the length of P. If $n = 1$, then simply firing v_0 $-c(v_1)$ times will increase $c(v_1)$ to $c'(v_1) = 0$ and decrease only the number of chips on v_0. For a path of length $n+1$, do the above process until $c'(v_n) \geq -c(v_{n+1}) \deg(v_n)$ then fire v_n $-c(v_{n+1})$ times. Then $c''(v_n) \geq 0$ still and $c''(v_{n+1}) \geq 0$ as desired.

Next, we will show that the fire of the burning algorithm does in fact determine a subset of vertices that can be fired without any of them going into debt. The fire determines recursively subsets $I_i \subset V$ of "burnt" vertices as follows: set $I_0 = \{v_0\}$, and for every i declare the edges incident to I_i to be "burnt", and let I_{i+1} be the union of I_i and the vertices v in $V \setminus I_i$ such that $c(v) < \#\{\text{edges that join } v \text{ to "burnt" vertices}\}$. We have that $\{v_0\} = I_0 \subset I_1 \subset \cdots$ is an ascending chain of subsets of V and since V is finite it must stabilize, i.e. there exists an n such that $I_n = I_{n+1} = \cdots$.

It follows that $S = V \setminus I_n$ is the set of vertices of V that ultimately remain "unburnt". The Subgraph-firing Lemma then says that firing S is the same as firing δS in the subgraph of $G \setminus \operatorname{Int} S$ obtained by removing all edges joining vertices of

δS to vertices of δS. It is plain to see, however, that δS consists of those vertices $v \in V$ with edges joining them to $I_n = V \setminus S$, and that, because $I_{n+1} = I_n$, for every $v \in \delta S$ we have that $c(v) \geq \#\{$edges that join v to "burnt" vertices, that is, to $I_n = V \setminus S\}$. It follows then by the Subgraph-firing Lemma that firing S inside G does not bring any of the vertices in δS into debt, and also does not change the number of chips on any vertex in $\operatorname{Int} S$. Hence, the chip configuration so obtained is, like the one it was obtained from, non-negative on $V \setminus \{v_0\}$ and furthermore the total number of chips on $V \setminus \{v_0\}$ has not increased.

Finally, we note that there are only finitely many chip configurations that are non-negative on $V \setminus \{v_0\}$ with at most a certain number of chips on $V \setminus \{v_0\}$, and make the claim that the action of repeatedly running a "fire" from v_0 and then firing the "unburnt" subgraph will never produce a chip configuration that has already occurred. Since there are only finitely many chip configurations possible, it will follow that eventually firing the "unburnt" subgraph of G does not change the chip configuration, which happens only if the graph is empty, that is, if running a fire from v_0 will "burn" the whole graph G and leave no "unburnt" subgraph to fire.

Thus, to finish the proof of correctness of the algorithm, we have only to prove our final claim that the action of repeatedly running a "fire" from v_0 and then firing the "unburnt" graph will never return to a chip configuration that has already occurred. To see this, notice that chip-firing a multiset of $V \setminus \{v_0\}$ inside G cannot ever be be reversed by chip-firing another multiset of $V \setminus \{v_0\}$ since the chip-firing inverse of a multiset inside G is unique and necessarily involves chip-firing a multiset including v_0, as per Corollary 2.7. Now certainly firing the set of "unburnt" vertices $S \subset G \setminus \{v_0\}$ does not leave the configuration c unchanged (as the number of chips on δS necessarily decreases), and hence the subsequent firings of "unburnt" subgraphs of $G \setminus \{v_0\}$ cannot invert the firing of S and return us to the configuration c. Hence, our claim is proved, and with it – the correctness of the algorithm. □

With existence settled, we now prove uniqueness of G-parking functions, once again with the help of the Subgraph-firing Lemma.

Proposition 2.12. *No two distinct G-parking functions relative to a vertex v_0 are chip-firing equivalent.*

Proof. Let c and c' be two G-parking functions relative to a vertex v_0, and let S be the multiset whose firing takes c to c'. Let k be the largest number of occurrences of any particular vertex $v \in S$. We may assume without loss of generality that the vertex v_0 does not occur with multiplicity k since if it does, then the multiset that represents the inverse transformation from c' to c does not by Corollary 2.7.

Then let M_k be the set of vertices in S that occur k times, and note that we can decompose S into the union of the sets M_k, $M_k \cup M_{k-1}$, $M_k \cup M_{k-1} \cup M_{k-2}$, $\ldots$, $M_k \cup M_{k-1} \cup \cdots \cup M_1$. Note that each of these contains M_k and hence by the Subgraph-firing Lemma, firing any of these sets does not increase the number of chips on M_k. Yet firing M_k the first time must surely give us a debt on one of the boundary vertices since M_k does not contain v_0 and c is a G-parking function relative to v_0, and therefore firing all of S will at best preserve and at worst exasperate that debt. This would mean, however, that $c'(v) < 0$ for that boundary vertex v of M_k, contradicting the fact that c' is a G-parking function. Therefore, the multiset S is empty and $c = c'$. □

Finally, we define the crucial notions of the genus of a (connected) graph, and the rank and degree of a chip configuration introduced by Baker and Norine in [BN07], and state without proof the general results necessary for the analysis in Section 3 of the G-parking functions on the graphs Γ_g.

Definition 2.13. The *rank* $r(c)$ of a configuration c on a connected graph G is -1 if c is not chip-firing equivalent to any non-negative chip configuration, and otherwise it is the largest number of chips that can be removed from any set of vertices and give us a configuration c' equivalent to some non-negative chip configuration.

Note that a simple application of Dhar's burning algorithm to a non-negative configuration c' relative to any vertex v will necessarily produce a non-negative G-parking function c'. This implies that equivalence to a non-negative configuration c' of some configuration c can conclusively be checked by computing the G-parking function of c relative to any vertex v_0, and checking whether the number of chips on v_0 is non-negative. Also note that rank is invariant under chip-firing since subtraction of chips is the same as subtraction in the abelian group of chip configurations $\mathbb{Z}^{|V|}$, and hence commutes with subtraction of columns of the Laplacian $L(G)$.

Definition 2.14. For any chip configuration c on a graph G define the *degree* $\deg(c)$ of c to be $\sum_{v\in V} c(v)$, i.e. the sum total of all chips and debts on G.

Definition 2.15. For any connected graph G, define the genus g of G by $g = |E| - |V| + 1$ where $|E|$ is the total number of edges and $|V|$ is the total number of vertices of the graph.

In the course of their tropical proof of the Brill-Noether theorem in [CDPR10], Cools et al. make use of the Tropical Riemann-Roch theorem, first proved by Baker and Norine in [BN07] and Luo's theorem, first proved in [Luo11]. For the sake of our review of Cools et al.'s construction of lingering lattice paths in the following section, we include the statements of these theorems here.

Theorem 2.16 (Tropical Riemann-Roch Theorem)**.** *If K is the canonical chip configuration on a graph G given by $K(v) = \deg(v) - 2$, then for any other chip configuration we have $r(c) - r(K - c) = \deg(c) + 1 - g$.*

Theorem 2.17 (Luo's Theorem)**.** *Let A be a finite subset of a connected graph Γ such that the closure in Γ of each connected component of $\Gamma \setminus A$ is contractible. Then the rank $r(c)$ of a non-negative chip configuration c is the largest number of chips that can be removed from any multiset of vertices contained in A and then give us a configuration c' equivalent to a non-negative chip configuration.*

3. Lingering Lattice Paths of chip configurations on Γ_g and standard Young tableaux

In this section we reinterpret the crucial construction of lingering lattice paths due to Cools et. al. in [CDPR10] as a procedure for determining the rank of G-parking functions on the graphs of interest Γ_g. The following exposition is mostly a slight rearrangement of what is already present in their paper, emphasizing the precise roles played by both the genericity of Γ_g and the condition $(r+1)(g-d+r) = g$ behind the bijection between rank r degree d chip configurations on Γ_g and $(g - d + r) \times (r + 1)$ standard Young tableaux. The only notable difference is our proof of Theorem 3.14 (Theorem 3.6 in [CDPR10]), in which we attempt to give explicit motivation behind the concrete computations in the proof that Cools et al. give.

3.1. **G-parking functions and genericity of the graphs Γ_g.**

Definition 3.1. Let Γ_g be a graph consisting of g consecutively concatenated loops γ_1, γ_2,...,γ_g such that $v_1, \dots v_{g-1}$ are their intersection points, while v_0 and v_g are points on the first and last loop respectively. Define also the sequence of pairs $(m_i, l_i)_{1\leq i\leq g}$ where l_i is the clockwise distance from v_i to v_{i-1} and m_i is the counter-clockwise distance (so that $m_i + l_i$ is the number of vertices on γ_i). For an illustration, see Figure 4.5.

Our first order of business is to characterize at least some set of G-parking functions on Γ_g.

Proposition 3.2. *A configuration c on Γ_g is G-parking relative to v_i if and only if c restricted to every cut loop $\gamma_j \setminus \{v_j\}$ for $j \leq i$ and $\gamma_j \setminus \{v_{j-1}\}$ for $j > i$ is of degree at most* 1*.*

Proof. We make use of Dhar's burning algorithm to determine the necessary conditions for a configuration c to be G-parking relative to v_i.

First, we suppose that the vertex v_i is burnt and consider how the fire spreads to the loop γ_i (to the left of v_i). There will be two fires starting from v_i – the one going counter-clockwise from v_i and the one going clockwise from v_i. If there is no chip on $\gamma_i \setminus \{v_i\}$, then both fires will reach v_{i-1} and the whole cut loop will have burnt. If there is a single chip on γ_i, then both the clockwise and the counter-clockwise fire will reach the vertex with that chip, and burn it. If there are at least two vertices on $\gamma_i \setminus \{v_{i-1}\}$ with at least one chip each, then one is closest to v_i along one of the two directions (clockwise and counter-clockwise from v_i) and hence will stop one of the fires, while another vertex with a chip will be closest along the other direction and so will stop the other one. Consequently, no part of the arc of γ_i containing those two vertices will burn. Finally, a vertex with at least two chips will not burn as there are only two fires within the loop.

Evidently, the same reasoning holds for the right loop γ_{i+1}, and inductively the proposition follows for the whole graph. □

Given the above proposition, we obtain a concise way of recording G-parking functions relative to v_0, which will prove useful for the construction of the lingering lattice paths.

Definition 3.3. For any G-parking function c relative to v_0 we associate a sequence $(d_0; x_1, \dots, x_g)$ where d_0 is the number of chips on v_0 and x_i is the counter-clockwise distance of the single chip on i^{th} cut loop $\gamma_i \setminus \{v_{i-1}\}$ from v_{i-1}, with $x_i = 0$ if no such chip exists.

Next, we use Luo's theorem to simplify our computations of the rank of chip configurations on c.

Proposition 3.4. *A configuration c on Γ_g has rank at least r if and only if removing any r chips from $\{v_0, \dots, v_g\}$ transforms c into a chip configuration c' equivalent to a non-negative configuration.*

Proof. Following [CDPR10], the connected components of $\Gamma_g \setminus \{v_0, \dots, v_g\}$ are all contractible, and hence an application of Luo's Theorem shows the proposition. □

For the sake of simplicity, we will collect the r chips that are to be subtracted from a chip configuration c at the vertices $\{v_0, v_1 \ldots, v_g\}$ in a configuration e that is non-negative on $\{v_0, \ldots, v_g\}$ and zero everywhere else on G. For want of a better term and for the sake of emphasis, whenever we wish to use a configuration e in order to determine the rank of c, we will refer to any non-negative configuration e as an *effective divisor*.

Remark 3.5. Supposing that c is a G-parking function relative to v_0, we would like to know whether $c-e$ is chip-firing equivalent to a non-negative chip configuration. This may be checked by computing the G-parking function equivalent to $c-e$ relative to any vertex. Cools et al. adopt, at least implicitly, a more sophisticated strategy. First decompose e as a sum $e = e_0 + e_1 + \cdots + e_g$, where e_i is a chip configuration with some non-negative number of chips on v_i and zero everywhere else. Then let $c_0 = c$ and define c_i to be the G-parking function relative to v_i of $c_{i-1} - e_{i-1}$. As long as each of c_i has more chips on v_i than e_i, $c - e$ will be chip-firing equivalent to a non-negative chip configuration. In particular, it will be equivalent to the G-parking function $c_g - e_g$ relative to v_g.

The lingering lattice paths Cools et al. construct will then be a tool for keeping track of the possible values of c_i presuming each of the $c_j - e_j$ for $j < i$ had a non-negative number of chips. Before we construct the lingering lattice paths, however, we need to know how to transform a G-parking function relative to v_{i-1} to its equivalent G-parking function relative to v_i.

Lemma 3.6. *Let c be a G-parking function on Γ_g relative to v_{i-1}. Suppose that c has k chips on v_{i-1}. Then c is chip-firing equivalent to the G-parking function c' relative to v_i which agrees with c everywhere outside the ith closed loop γ_i, and on γ_i restricts according to the following cases:*

(1) *If there is no chip on $\gamma_i \setminus \{v_{i-1}\}$, then c' has $k-1$ chips on v_i and one chip that is $(k-1)m_i$ away clockwise from v_{i-1};*
(2) *if there is a chip on $\gamma_i \setminus \{v_{i-1}\}$ and $x_i \not\equiv (k+1)m_i \bmod (l_i + m_i)$ then c' has k chips on v_i plus a chip that is $km_i - x_i$ away clockwise from v_{i-1};*
(3) *if there is a chip on $\gamma_i \setminus \{v_{i-1}\}$ and $x_i \equiv (k+1)m_i \bmod (l_i + m_i)$, then c' has $k+1$ chips on v_i.*

Proof. This is an immediate consequence of repeatedly applying conservation of momentum. To apply the lemma, define first for every vertex $v \in \gamma_g$ a graph H_v as follows: for $v \neq v_{i-1}, v_i$, set $H_v = v$, while for $v = v_{i-1}, v_i$ set $H_{v_{i-1}} = \{v_0\} \bigcup \left(\bigcup_{j=1}^{i-1} \gamma_j\right)$ and $H_{v_i} = \left(\bigcup_{j=i+1}^{g} \gamma_j\right) \bigcup \{v_g\}$.

Second, consider an arbitrarily large path obtained by starting at v_{i-1} and running counter-clockwise loops around γ_i. It is plain to see that for any subpath of that arbitrarily long path, the graphs H_v for v in the subpath satisfy the conditions of Conservation of momentum. Hence, we can move chips via m_i applications of Conservation of momentum as follows:

In the first case we move $k-1$ chips from v_{i-1} a distance of m_i in the counter-clockwise direction, and one chip from v_{i-1} a distance of $(k-1)m_i$ in the clockwise direction.

In the second case and third case, we move the k chips a distance of m_i in the counter-clockwise direction, and the already present single chip a distance of km_i in the clockwise direction. □

Note that the first case of the above lemma may simply move all the chips from v_{i-1} to v_i if it so happens that $(k-1)m_i \equiv \ell_i \bmod \ell_i + m_i$, which is equivalent to $\ell_i + m_i$ dividing km_i. In order to define the lingering lattice paths, we need to eliminate this possibility. Fortunately, we can do so by limiting the size of k using the Tropical Riemann-Roch theorem, and restricting our attention to the graphs Γ_g which Cools et al. call generic.

Proposition 3.7. *If* $\deg(c) > 2g-2$ *for a configuration* c *on* Γ_g*, then* $r(c) = \deg(c) - g$.

Proof. The Tropical Riemann-Roch theorem states that $r(c) - r(K-c) = \deg(c) + 1 - g$ where g is the genus of the graph and K is the canonical divisor given by $K(v) = \deg(v) - 2$.

On the one hand, the genus of Γ_g is exactly g since each of the cut loops $\gamma_i \setminus \{v_i\}$ has just one more edge than it has vertices, giving us that $g = |E| - |V| + 1 = g - 1 + 1 = g$ where the -1 comes from the vertex v_0 that is not a part of any cut loop.

On the other hand, every vertex of Γ_g that is not in $\{v_1, \dots, v_{g-1}\}$ is of degree 2, while $\deg(v_i) = 4$ for $1 \le i \le g-1$. Hence, the canonical divisor K which satisfies $K(v) = \deg(v) - 2$ has 2 chips on each of $v_1, v_2, \dots, v_{g-1}$ and no chips anywhere else, which means that its degree is $2g-2$.

Computing $r(K-c)$ when $\deg(c) > 2g-2$ is then easy since $\deg(K-c)$ will have to be negative, which, since chip-firing does not change the degree of a configuration, implies that $K-c$ cannot be equivalent to a non-negative configuration and hence that $r(K-c) = -1$. As a result, the Tropical Riemann-Roch theorem gives us that $r(c) + 1 = \deg(c) + 1 - g$ and so $r(c) = \deg(c) - g$. ☐

The above proposition allows us to assume that $\deg(c) \le 2g-2$ for the chip configurations c whose rank we wish to compute using the lingering lattice paths. Hence, in the above lemma we may take $k \le 2g-2$, and the first case will always move all but one chip from v_{i-1} to v_i if and only if $\ell_i + m_i$ does not divide km_i for any $k \le 2g-2$. In order to ensure this latter condition, we restrict our attention to the following graphs.

Definition 3.8. A graph Γ_g is called *generic* if when we take the ratios ℓ_i/m_i and reduce them to lowest terms a_i/b_i, we have that $a_i + b_i > 2g-2$.

Proposition 3.9. *The condition that* Γ_g *be generic is equivalent to requiring that the first case of Lemma* 3.6 *always moves all but one chip from* v_{i-1} *to* v_i *for* $1 \le k \le 2g-2$.

Proof. We need a condition that implies that $km_i \ne 0 \bmod \ell_i + m_i$ for $1 \le k \le 2g-2$. Let $m_i = b_i \gcd(\ell_i, m_i)$ and $\ell_i = a_i \gcd(\ell_i, m_i)$, i.e. let a_i/b_i be the expression for ℓ_i/m_i reduced to lowest terms. Then $km_i \ne 0 \bmod \ell_i + m_i$ holds if and only if $kb_i \ne 0 \bmod a_i + b_i$. We need it to hold for $1 \le k \le 2g-2$ and since b_i is relatively prime to $a_i + b_i$, this happens if and only if $a_i + b_i > 2g-2$, as desired. ☐

3.2. **Lingering lattice paths.**

Definition 3.10. For any non-negative integer r we define the *open Weyl chamber* $\mathcal{C} \subset \mathbb{Z}^r$ by $\mathcal{C} = \{y \in \mathbb{Z}^r : y(0) > y(1) > \cdots > y(r-1) > 0\}$.

Definition 3.11. For any generic graph Γ_g and any G-parking function c relative to v_0 given by sequence $(d_0; x_1, \ldots, x_g)$ and of total degree at most $2g-2$, we associate for any positive integer $r \leq d_0$ the *lingering lattice path* $P \subset \mathbb{Z}^r$ consisting of points $p_0, \ldots, p_g$ in $\mathbb{Z}^r$ with $p_i = (p_i(0), p_i(1), \ldots, p_i(r-1))$ such that $p_0 = (d_0, d_0-1, \ldots, d_0-(r-1))$ and:

$$p_i - p_{i-1} = \begin{cases} (-1, \ldots, -1) & \text{if } x_i = 0 \\ e_j & \begin{array}{l}\text{if } x_i \equiv (p_{i-1}(j)+1)m_i \bmod \ell_i + m_i \\ \text{and both } p_{i-1}, p_{i-1}+e_j \in \mathcal{C}\end{array} \\ 0 & \text{otherwise} \end{cases}$$

where $e_0, e_1, \ldots, e_{r-1}$ are the standard basis vectors for $\mathbb{Z}^r$.

The above definition contains the implicit claim that, in using this procedure to construct P, there are never two distinct j such that $x_i \equiv (p_{i-1}(j)+1)m_i \bmod \ell_i + m_i$. This claim demands proof, which we give below.

Proposition 3.12. *Suppose that c is a G-parking function relative to v_0 on generic Γ_g and given by the sequence $(d_0; x_1, \ldots, x_g)$ with $\deg(c) \leq 2g-2$ and that $p_0 = (d_0, d_0-1, d_0-2, \ldots, d_0-(r-1))$, $p_1, \ldots, p_{i-1} \in \mathbb{Z}^r$ for $i < g$ are points in the open Weyl chamber whose consecutive differences satisfy the rule of Definition* 3.11. *Then the following are true:*

1. *for any $1 \leq j \leq r-1$ and $k < i$, $p_k(j-1) > p_k(j)$;*
2. *$p_k(0) = c_k(v_k)$ where c_k is the G-parking function relative to v_k that is chip-firing equivalent to c;*
3. *$(p_{i-1}(j)+1)m_i \equiv (p_{i-1}(j')+1)m_i \bmod \ell_i + m_i$ implies $j = j'$, allowing us to define p_i.*

Proof. For the first property, notice that the rule of the definition does one of three things at any step from p_{k-1} to p_k: it either decreases all entries by 1, keeps all entries constant, or increases a single entry by 1 as long as the resulting p_k remains in the open Weyl chamber. All of these preserve the strictly decreasing property, and since $p_0 = (d_0, d_0-1, \ldots, d_0-(r-1))$ is strictly decreasing, so is each p_k for $k < i$.

For the second property, we obviously have that $c = c_0$ and $p_0(0) = d_0 = c_0(v_0)$. By Lemma 3.6, we see that the value of $c_k(v_k)$ is related to the value of $c_{k-1}(v_{k-1})$ in precisely the same way that $p_k(0)$ is related to $p_{k-1}(0)$.

Finally, to show the third property, we use the other two. Evidently we have that $p_{i-1}(j), p_{i-1}(j') \leq p_{i-1}(0) = c_{i-1}(v_{i-1}) \leq \deg(c_{i-1}) = \deg(c) \leq 2g-2$ (the number of chips on a special vertex v_{i-1} of a G-parking function c_{i-1} relative to v_{i-1} is necessarily less than or equal to $\deg(c_{i-1})$ since all other vertices have non-negative number of chips on them by definition). Then the difference $p_{i-1}(j) - p_{i-1}(j')$ is at most $2g-2$ and the genericity condition ensures that $(p_{i-1}(j) - p_{i-1}(j'))m_i \equiv \bmod \ell_i + m_i$ if and only if $p_{i-1}(j) - p_{i-1}(j')$ is 0. □

Having defined the lingering lattice paths, and keeping in mind Remark 3.5, it is easy to see that the lingering lattice path associated to a G-parking function relative to c contains the kind of information we would need in order to determine the rank. In particular, the definition makes it plain to see that if c' is a G-parking function relative to v_{i-1} with $c'(v_{i-1}) = p_{i-1}(j)$ and $p_{i-1}(j-1) - p_{i-1}(j) > 1$, then certainly $p_i(j) = c''(v_i)$ where c'' is the G-parking function relative to v_i that is equivalent to

c'. As Remark 3.5 suggests, keeping track of how many chips there are on vertex v_i for the G-parking function relative to v_i is key to Cools et al.'s implicit procedure for computing the rank. It is at this point that we diverge from [CDPR10] and offer our own, slightly more elegant proof of the fact that a G-parking functions is of rank r if and only if the associated lingering lattice path in $\mathbb{Z}^r$ is contained in the open Weyl chamber. Specifically, we replace the combined argument of their Proposition 3.10 and Theorem 3.6 with the following:

Lemma 3.13. *Suppose that c is a G-parking function relative to v_0 on generic Γ_g and given by the sequence $(d_0; x_1, \ldots, x_g)$ with $\deg(c) \leq 2g-2$. Suppose that $P \subset \mathbb{Z}^r$ is the associated lingering lattice path.*

Suppose also that c' is any G-parking function relative to v_{i-1} and c'' is a G-parking function relative to v_i that is equivalent to c', then the lingering lattice path P satisfies the following properties:

1. *$c'(v_{i-1}) \geq p_{i-1}(j)$ if and only if $c''(v_i) \geq p_i(j)$;*
2. *if $j = 0$ or $p_i(j-1) - p_i(j) > 1$, then $c'(v_{i-1}) = p_{i-1}(j)$ if and only if $c''(v_i) = p_i(j)$.*

Proof. Both properties are more or less obvious if we compare the Definition 3.11 and the statement of Lemma 3.6.

To verify the first property, we consider separately the case in which the cut loop $\gamma_i \setminus \{v_{i-1}\}$ does not contain a chip, and the case in which it does. First, in the case in which there is no chip on the cut loop, $p_{i-1}(j)$ decreases by 1 to $p_i(j)$ by definition, and $c'(v_{i-1})$ decreases by 1 to $c''(v_i)$ by Lemma 3.6. Hence in this case the inequalities $c'(v_{i-1}) \geq p_{i-1}(j)$ and $c'(v_i) \geq p_i(j)$ are logically equivalent.

Second, in the case in which there is a chip on the cut loop, then by definition we have two subcases: either $p_{i-1}(j) = p_i(j)$, or $p_{i-1}(j)$ increases by 1 to $p_i(j)$. If $p_{i-1}(j) = p_i(j)$, then the presence of a chip on the cut loop means that $c'(v_{i-1})$ does not decrease to $c''(v_i)$, hence $c''(v_i) \geq c'(v_{i-1}) \geq p_i(j) = p_{i-1}(j)$ as desired.

If $p_{i-1}(j)$ increases by 1 to $p_i(j)$, then it does so because $x_i \equiv (p_{i-1}(j)+1)m_i \mod \ell_i + m_i$. If $c'(v_{i-1}) = p_{i-1}(j)$ or $c''(v_i) = p_i(j)$, then Lemma 3.6 implies that $c''(v_i) = c'(v_{i-1}) + 1$ and hence that $c''(v_i) = c'(v_{i-1}) + 1 = p_{i-1}(j) + 1 = p_i(j)$. Alternatively, if $c'(v_{i-1}) > p_{i-1}(j)$, then Lemma 3.6 implies that $c''(v_i) = c'(v_{i-1}) \geq p_{i-1}(j) + 1 = p_i(j)$. Thus, our casework is complete and we have verified the first property.

To verify the second property we only need to note that either of the conditions $j = 0$ and $p_{i-1}(j-1) - p_{i-1}(j) > 1$ precludes the possibility in the definition of the lingering lattice path that $x_i \equiv (p_{i-1}(j)+1)m_i \mod \ell_i + m_i$ but $p_{i-1} + e_j \notin \mathcal{C}$. With that case precluded, the remaining cases of Definition 3.11 and the corresponding relationships between $p_{i-1}(j)$ and $p_i(j)$ match precisely with the cases of Lemma 3.6 and the corresponding relationships between $c'(v_{i-1})$ and $c''(v_i)$. Hence we certainly have that $c'(v_{i-1}) = p_{i-1}(j)$ if and only if $c''(v_i) = p_i(j)$. □

Theorem 3.14. *A G-parking function relative to v_0 on generic graph Γ_g and of degree at most $2g-2$ has rank at least r if and only if the associated lingering lattice path in $\mathbb{Z}^r$ is in the Weyl chamber $\mathcal{C}$.*

Proof. To see that if the lingering lattice path in $\mathbb{Z}^r$ is in $\mathcal{C}$, then c has rank at least r, first suppose that c' is a G-parking function relative to v_i such that $c'(v_i) \geq p_i(j)$ and that n is a positive integer such that $j + n < r$. Then the fact that $p_i \in \mathcal{C}$

implies that $p_i(j) > p_i(j+1) > \cdots > p_i(j+n)$. Hence $p_i(j+n) \leq p_i(j) - n$ so that removing d chips from v_i results in a chip configuration c'' that is G-parking relative to v_i and such that $c''(v_i) = c'(v_i) - n = \geq p_i(j) - n \geq p_i(j+n)$.

Second, fix an effective divisor $e' = e_0 + e_1 + \cdots + e'_k$ of degree r, where e_i are effective divisors that necessarily have zero chips away from v_i, and e'_k is an effective divisor that has zero chips everywhere except at v_k. Let $e = e_0 + e_1 + \cdots + e_k$ be the effective divisor of degree $r-1$ obtained by removing one chip from e' at v_k, so that e_k is the effective divisor obtained by removing one chip from e'_k at v_k. Let $c_0 = c$ and let c_i be the G-parking function relative to v_i that is chip-firing equivalent to $c_{i-1} - e_{i-1}$. We know that $c_0(v_0) = p_0(0)$ by Proposition 3.12, and by the above we have that $(c_0 - e_0)(v_0) = c_0(v_0) - \deg(e_0) = p_0(0) - \deg(e_0) \geq p_0(\deg(e_0))$.

Hence, we may inductively suppose that $(c_i - e_i)(v_i) \geq p_i(\deg(e_0 + e_1 + \cdots + e_i))$ for some $0 \leq i < k$. Since c_{i+1} is the G-parking function relative to v_{i+1} that is chip-firing equivalent to $c_i - e_i$, by the first property of Lemma 3.13 the inequality of the inductive hypothesis implies the inequality $c_{i+1}(v_{i+1}) \geq p_{i+1}(\deg(e_0 + e_1 + \cdots + e_i)$. Hence $(c_{i+1} - e_{i+1})(v_{i+1}) = c_{i+1}(v_{i+1}) - \deg(e_{i+1}) = p_{i+1}(\deg(e_0 + e_1 + \cdots + e_i)) - \deg(e_{i+1}) \geq p_{i+1}(\deg(e_0 + e_1 + \cdots + e_{i+1})$.

Hence, by induction we have that the G-parking function c_k relative to v_k that is equivalent to $c - e$ satisfies $c_k(v_k) \geq p_k(r-1)$. Since p_k is in the open Weyl chamber $\mathcal{C}$, we have that $p_k(r-1) \geq 1$, which means that c_k with one chip removed from v_k has a non-negative number of chips on v_k. Since c_k with a chip removed from v_k is the G-parking function relative to v_k that is equivalent to $c - e'$, this shows that as long as the lingering lattice path in $\mathbb{Z}^r$ associated to c is in the open Weyl chamber $\mathcal{C}$, we can subtract any effective divisor of degree r from c and obtain a non-negative chip configuration, i.e. that the rank of c is at least r.

To see the converse, we will suppose that the lingering lattice path in $\mathbb{Z}^r$ associated to c is not in the open Weyl chamber, and we will construct an effective divisor e of degree r such that $c - e$ is not equivalent to a non-negative chip configuration.

First, however, for every k such that $0 \leq k \leq g$ we will construct an effective divisor e of degree $0 \leq j < r$ such that $p_{k+1}(j) = c_{k+1}(v_{k+1})$ where c_{k+1} is the G-parking function of $c - e$ relative to v_{k+1}, presuming that $p_0, p_1, \ldots, p_k$ are all in the open Weyl chamber.

Since $p_0 = (d_0, d_0 - 1, \ldots, d_0 - (r-1))$ and $c(v_0) = d_0$, it follows that e' consisting of $j < r$ chips on v_0 and zero chips everywhere else is such that $(c - e)(v_0) = c_0(v_0) = d_0 - j = p_0(j)$. Hence, we may inductively suppose that we can construct an effective divisor e' of degree $0 \leq j - n < r$ such that if c'_k is the G-parking function relative to v_k equivalent to $c - e'$, then $c'_k(v_k) = p_k(j)$.

Then let n be the largest number (possibly 0) such that $p_{k+1}(j-n) - p_{k+1}(j) = n$ if $j > 0$ and set $n = 0$ otherwise. Then certainly either $n - j = 0$, or $p_{k+1}(j - (n+1)) - p_{k+1}(j-n) > 1$ and $p_{k+1}(j-n) > p_{k+1}(j - (n-1)) > \ldots p_{k+1}(j)$ is a string of consecutive integers. Nevertheless, the second property of Lemma 3.13 applies. So let e' be an effective divisor such that $c - e'$ is equivalent to G-parking function c'_k relative to v_k such that $c'_k(v_k) = p_k(j-n)$. Then the second property of Lemma 3.13 states that the G-parking function c''_{k+1} relative to v_{k+1} and equivalent to c'_k satisfies $c''_{k+1}(v_{k+1}) = p_{k+1}(j-n)$. It follows then that subtracting n chips from v_{k+1} gives a G-parking function c_{k+1} relative to v_{k+1} such that $c_{k+1}(v_{k+1}) = c''_{k+1}(v_{k+1}) - n = p_{k+1}(j-n) - n = p_{k+1}(j)$. Hence, the effective divisor e such that $c - e$ is equivalent to c_{v_k} is given by adding n chips to e' at v_{k+1}.

To finish the proof that the lingering lattice path in $\mathbb{Z}^r$ associated to c fits in the Weyl chamber if c is of rank at least r, suppose that $P \subset \mathbb{Z}^r$ does not fit inside the open Weyl chamber, and let p_i be the first point of P not in $\mathcal{C}$. Then the fact that p_i is the first such point implies that $p_i(r-1) = 0$. Using the above, we know that we can construct an effective divisor e' of rank $r-1$ such that $c-e'$ is equivalent to a G-parking function c_i' relative to v_i with $c_i'(v_i) = 0$. Adding an additional chip to e' at v_i gives a rank r effective divisor e such that $c-e$ is equivalent to a G-parking function c_i relative to v_i such that $c_i(v_i) = -1$. Hence, $c-e$ is not equivalent to a non-negative chip configuration and the rank of c is less than r. □

The above theorem allows us to compute the rank of a G-parking function as follows. For convenience, imagine graphing P in $\mathbb{Z}^{d_0}$ as d_0 non-intersecting paths in the plane that go diagonally up or diagonally down starting at the heights $(d, d-1, \dots, 1)$ on the y-axis. If at some point $p_i \notin \mathcal{C}$, i.e. if the lowest path touches the x-axis, then remove this lowest path (i.e. project onto $\mathbb{Z}^{d_0-1}$). Then the new path P' will have p_i' in the open Weyl chamber, and we may continue constructing the path. Iterating this process will ultimately leave us with a path P'' in $\mathbb{Z}^r$ that fits in the open Weyl chamber, and such that the path in $\mathbb{Z}^{r+1}$ does not fit.

The observant reader will have noticed that the lingering lattice paths which disallow lingering, that is, which only increase at some level or decrease at all levels at every step are in bijection with standard Young tableaux. We formalize this fact and related results due to Cools et al. regarding the number of lingering steps below.

Definition 3.15. For any configuration c on graph Γ_g, let $\rho = g-(r+1)(g-d+r)$.

Theorem 3.16. *The following holds for G-parking functions relative to v_0 of degree $d > 2g-2$ on any graph Γ_g, and for G-parking functions relative to v_0 of degree $d \leq 2g-2$ on generic graphs.*

1. *If ρ is negative, then any c of degree d is of rank -1.*
2. *If ρ is non-negative, then the number of chips d_0 at v_0 is $\leq r+\rho$.*
3. *If ρ is non-negative, then the number of lingering steps where the path neither increases nor decreases is bounded above by* $\min\{\rho, g\}$.
4. *If ρ is zero, then $d_0 = r$, association between configurations and paths is bijective, and the paths are in one-to-one correspondence with rectangular standard Young tableaux of shape $(g-d+r) \times (r+1)$.*

Proof. All of these are trivial for $d > 2g-2$ since then the Tropical Riemann-Roch theorem tells us that $r = d-g$ which gives us $\rho = g$.

If $d \leq 2g-2$, then all of the properties will follow from counting the number of decreases and giving a lower bound for the number of increases. In particular, property 1) will follow by a contrapositive to showing that if the associated path P is in the open Weyl chamber (and hence the rank is non-negative), then ρ is a non-negative number such that the number of lingering steps in P is at most $\min\{\rho, g\}$ and $d_0 \leq r+\rho$.

So suppose that P is in the open Weyl chamber. Then $d-d_0$ is the number of single chips distributed among the cut loops $\gamma_i \setminus \{v_{i-1}\}$. Hence there are $d-d_0$ cut loops with single chips on them out of total of g cut loops, and hence $g-(d-d_0) = g-d+d_0$ cut loops without chips on them, which correspond to the decreases along the path P.

Since P ins the open Weyl chamber, we have that p_g must be in the open Weyl chamber, i.e. $r-j \leq p_g(j)$. We have, however, that $p_0 = (d_0, d_0-1, \ldots, d_0-(r-1))$ and so $r-j \leq p_g(j) = p_0(j) - (g-d+d_0) + \#\{\text{increases in the } j^{\text{th}} \text{ direction}\}$ where $g-d+d_0$ is the number of decreases (same in every direction). Since $p_0(j) = d_0 - j$, the above inequality will be satisfied id and only if the total number of increases in any particular direction j is at least $g-d+r$.

Therefore, the total number of increases in all directions combined is at least $r(g-d+r)$. Since number of increases plus number of decreases plus number of lingering steps is surely the total number of steps, we obtain that the number of lingering steps is at most $g-(g-d+d_0+r(g-d+r))$, which is surely non-negative. We know, however, that $r \leq d_0$, so that setting $\rho = g-(r+1)(g-d+r)$, the assertion that $g-(g-d+d_0+r(g-d+r)) \geq 0$ transforms into $d_0 \leq r+\rho$, while the statement that $g-(d+d_0+r(g-d+r)$ is an upper bound for the number of lingering steps implies that $\min\{\rho, g\}$ is such an upper bound.

Finally, if $\rho = 0$, then necessarily $d_0 = r$, and $g = (r+1)(g-d+r)$, which implies that each direction has exactly $g-d+r$ increases, that there are exactly $g-d+r$ decreases, and that at every step the number of decreases up to that point is at most the number of increases in each direction. These paths correspond to rectangular standard Young tableaux of shape $(g-d+r) \times (r+1)$ as follows. At the i^{th} step one add the number i to the tableaux, with increases in the e_j^{th} direction/j^{th} path corresponding to adding the number to the $j+1^{\text{st}}$ column (recall that Cools et al.'s indexing has $e_0, e_1, \ldots, e_{r-1}$ as the standard basis vectors for $\mathbb{Z}^r$), and decreases corresponding to adding the number to the $r+1^{\text{st}}$ column. □

4. Evacuation on generic Γ_g with $\rho = 0$

In this section we state and prove our original result, conjectured by our supervisor Gregg Musiker.

Definition 4.1. Let T be an $m \times n$ rectangular standard Young tableau with $p = mn$, that is, an $m \times n$ matrix whose entries are the integers from 1 to p such that $a_{i,j+1}, a_{i+1,j} > a_{i,j}$. Then we define the *evacuation*, $\text{ev}(T)$, of T to be the $m \times n$ rectangular standard Young tableau with $(i,j)^{\text{th}}$ entry $(p+1-a_{m+1-i,n+1-j})$.

For more details on evacuation of tableaux, see the wonderful survey in [Sta08] by Richard Stanley. Geometrically, $\text{ev}(T)$ can be pictured as rotating the rectangular standard Young tableaux 180° and flipping the entries according to the rule $i \to p+1-i$.

Definition 4.2. Given a chip configuration c on a graph Γ_g, we define the *reflection* of c, which we write as $\sigma(c)$, to be the chip conifugration c' on Γ'_g obtained by setting $v'_i = v_{g-i}$ and γ'_i to be the image of γ_{g-i} under the reflection that exchanges v_{g-i-1} and v_{g-i}.

Theorem 4.3. *Under the bijection of Theorem* 3.16 *between chip-firing equivalence classes of rank r degree d chip configurations c on generic Γ_g such that $(g-d+r)(r+1) = g$ and rectangular $(g-d+r) \times (r+1)$ standard Young tableaux, if such a G-parking function c relative to v_0 corresponds to a standard Young tableau T, then the G-parking function relative to v'_0 that is chip-firing equivalent to $\sigma(c)$ corresponds to* $\text{ev}(T)$*, the evacuation of T.*

For the sake of our readers, we separate our exposition of the proof in two parts. First, we offer a more detailed description of the bijection in question including an explicit formulae for computing the associated path from a tableau, and for computing the G-parking function c relative to v_0 both from the associated path and from the associated tableau. We then use these formulae to compute how evacuation of tableaux and reflections act on the chip configurations, and thus to see that they act in precisely the same way.

4.1. **The bijection in detail.** Recall that we can describe any G-parking function relative to v_0 on the graph Γ_g by a sequence $(d_0; x_1, \ldots, x_g)$ where x_i is the counter-clockwise distance of the single chip on the cut loop $\gamma_i \setminus \{v_{i-1}\}$ from v_{i-1}. Since such a sequence determines a G-parking function relative to v_0, and hence a unique representative of the chip-firing equivalence class of chip configurations, we will say that any rank r degree d chip configuration with $g = (g - d + r)(t + 1)$ determines such a sequence.

Recall also that to any rank r degree d G-parking function c relative to v_0 on a generic graph Γ_g described by a sequence $(d_0; x_1, \ldots, x_g)$ we associate a lingering lattice path $P \subset \mathbb{Z}^r = (p_0, p_1, \ldots, p_g)$ given by:

(1) $p_0 = (r, r-1, \ldots, 1)$;

(2) $$p_i - p_{i-1} = \begin{cases} (-1, \ldots, -1) & \text{if } x_i = 0 \\ e_j & \begin{array}{l}\text{if } x_i \equiv (p_{i-1}(j) + 1)m_i \bmod \ell_i + m_i \\ \text{and both } p_{i-1}, p_{i-1} + e_j \in \mathcal{C}\end{array} \\ 0 & \text{otherwise} \end{cases}$$

where $\mathcal{C} \subset \mathbb{Z}^r$ is the open Weyl chamber of points $y \in \mathcal{C}$ such that $y(1) > y(2) > \cdots > y(r)$ and $e_1, \ldots, e_r$ is the standard basis for $\mathbb{Z}^r$ (for the sake of convenience, in this section we depart from Cools et al.'s indexing and take $e_1, e_2, \ldots, e_r$ to be the standard basis vectors for $\mathbb{Z}^r$),

which satisfies the following properties:

(1) $p_i(1) > p_i(2) > \cdots > p_i(r) \geq 1$;
(2) when $g = (g - d + r)(r + 1)$ we have that:
   (a) $p_i - p_{i-1}$ is never 0,
   (b) the number of steps in the $j - 1^{\text{st}}$ direction up to a certain point (i.e. the number of times $p_i - p_{i-1} = e_{j-1}$ for $1 \leq i \leq k$) is always greater than or equal to the number of steps in the j^{th} direction, and always greater than or equal to the number of steps in $(-1, \ldots, -1)$ which we call the $r + 1^{\text{st}}$ direction;
   (c) $p_g = (r, r-1, \ldots, 1)$;
   (d) $r = d_0$, the number of chips on v_0.

Restricting our attention only to the cases when $g = (g - d + r)(r + 1)$, to any such P we can associate a $(g - d + r) \times (r + 1)$ rectangular standard Young tableau T where the equation $g = (g - d + r)(r + 1)$ determines d, by placing the number i for $1 \leq i \leq g$ in the j^{th} column if $p_i - p_{i-1} = e_j$, and in the $r + 1^{\text{st}}$ column if $p_i - p_{i-1} = (-1, -1, \ldots, -1)$.

This case in which the chip configurations c satisfy $(g - d + r)(r + 1) = g$ allows us to make a few simplifications to our notation. Note that the fact that $p_i - p_{i-1}$ is never 0 if and only if for every i with a non-zero x_i, there exists a j such that $x_i \equiv (p_{i-1}(j) + 1)m_i \bmod \ell_i + m_i$. Additionally, $x_i \neq m_i$, i.e. the single chip on $\gamma_i \setminus \{v_{i-1}\}$ is never on v_i, since that would imply that $p_{i-1}(j)m_i \equiv 0 \bmod \ell_i + m_i$

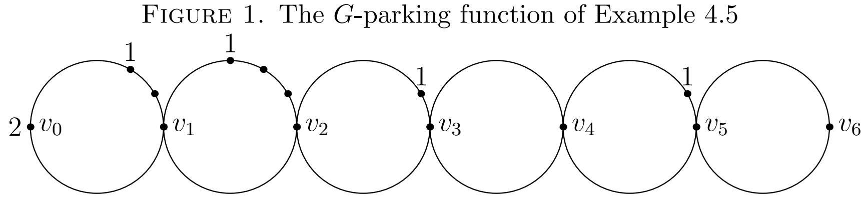
which is impossible by genericity of Γ_g and $1 \leq p_{i-1}(j) \leq 2g-2$. Thus, we make the following definition.

Definition 4.4. For any rank r degree d G-parking function c relative to v_0 on generic Γ_g with $(g-d+r)(r+1) = g$, we associate a sequence $(d_0; \underline{x}_1, \underline{x}_2, \ldots, \underline{x}_g)$ where $\underline{x}_i$ is 0 if there is no chip on $\gamma_i \setminus \{v_{i-1}\}$, and otherwise $\underline{x}_i$ is such that $\underline{x}_i m_i$ is the counter-clockwise distance of the chip from v_i. Equivalently, given that the G-parking function c determines $(d_0; x_1, \ldots, x_g)$, we have $(\underline{x}_i + 1)m_i = x_i$ if $x_i \neq 0$ and $\underline{x}_i = 0$ if $x_i = 0$.

This has the advantage of simplifying the rule for generating the lingering lattice path of a rank r degree d G-parking function c such that $(g-d+r)(r+1) = g$ to:

$$p_i - p_{i-1} = \begin{cases} (-1, \ldots, -1) & \text{if } \underline{x}_i = 0 \\ e_j & \begin{array}{l}\text{if } \underline{x}_i m_i \equiv p_{i-1}(j) m_i \bmod \ell_i + m_i \\ \text{and both } p_{i-1}, p_{i-1} + e_j \in \mathcal{C}\end{array} \end{cases} \tag{1}$$

Figure 1. The G-parking function of Example 4.5



Example 4.5. Consider G-parking function on Γ_g given by $(2; 2, 3, 1, 0, 1, 0)$ in which the $\underline{x}_i$ have values

$$\underline{x}_1 = 2, \underline{x}_2 = 3, \underline{x}_3 = 1, \underline{x}_4 = 0, \underline{x}_5 = 1, \underline{x}_6 = 0$$

and which is illustrated in Figure 1.

We decide to construct the lingering lattice path so we start with $p_0 = (2, 1)$ and the rule (1). Note that in general, rule (1) would encounter a situation not covered by its cases if and only if the rank r of c does not satisfy $(g-d+r)(r+1) = g$, in which case we would switch to the more general present in the original definition of the lingering lattice paths. We encounter no problem, however, as we obtain:

$$p_0 = \begin{pmatrix}2\\1\end{pmatrix}, p_1 = \begin{pmatrix}3\\1\end{pmatrix}, p_2 = \begin{pmatrix}4\\1\end{pmatrix}, p_3 = \begin{pmatrix}4\\2\end{pmatrix}, p_4 = \begin{pmatrix}3\\1\end{pmatrix}, p_5 = \begin{pmatrix}3\\2\end{pmatrix}, p_6 = \begin{pmatrix}2\\1\end{pmatrix}$$

The fact that rule (1) never failed implies that the rank of c certainly satisfies $(g-d+r)(r+1) = g$, and so in particular is certainly d_0. Hence the rank is 2, so $r+1 = 3$ and $g-d+r = 2 = g/(r+1) = 6/3 = 2$, so we can assoicate a 2×3 standard Young tableau, which we fill as follows:

Since p_0 increases to p_1 along the top (i.e. the e_1) direction, 1 must go in the first column; since p_2 increases to p_3 along the second from the top (i.e. e_2) direction, 3 must go in the second column; since p_3 decreases to p_4 (i.e. changes along $(-1, \ldots, -1)$, 4 must go in the last (third) column, and so forth. We obtain the tableau below:

1	3	4
2	5	6

Since we are interested in how evacuation of tableaux pulls back through the bijection, we better also see how to run the bijection in reverse. The following two propositions will tell us how to obtain the lingering lattice path from the tableau, and the numbers $\underline{x}_i$ from the lingering lattice path. First, however, we will need a certain statistics on standard Young tableaux.

Definition 4.6. Suppose that T is a rectangular $(g-d+r)\times(r+1)$ standard Young tableau. Fix a positive integer i such that $i \le g$.

We define l_j to be the number of cells in the j^{th} column of T whose entries are at most i, and l'_j to be the number of cells in the j^{th} column of T whose entries are at most $i-1$.

Proposition 4.7. *Suppose that T is rectangular $(g-d+r)\times(r+1)$ standard Young tableau associated to a lingering lattice path P.*

For a fixed i, and the l_j defined as above to be the number of cells in the j^{th} column whose entries are at most i, we have that the i^{th} step p_i of the lingering lattice path associated to T is given by:

$$p_i = p_0 + \begin{pmatrix} l_1 - l_{r+1} \\ l_2 - l_{r+1} \\ \cdots \\ l_r - l_{r+1} \end{pmatrix} = \begin{pmatrix} r + l_1 - l_{r+1} \\ r-1+l_2-l_{r+1} \\ \cdots \\ 1 + l_r - l_{r+1} \end{pmatrix} \tag{2}$$

Proof. Recall that in the bijection between the lingering lattice paths and the standard Young tableaux, a number $k \le i$ is placed in column $j < r+1$ when the $p_k - p_{k-1} = e_j$, i.e. when there has been an increase in the j^{th} direction. The number l_j hence counts the number of increases that have occurred in the j^{th} direction by step the p_i. On the other hand, a number $k \le i$ is placed in column $r+1$ when $p_k - p_{k-1} = (-1,-1,\ldots,-1)$, i.e. when there has been a decrease along all directions. Hence, l_{r+1} counts the number of decreases that have occurred by step p_i. Knowing that we start with $p_0 = (r, r-1, \ldots, 1)$, the proposition follows. □

Proposition 4.8. *Suppose that P is a lingering lattice path in $r = d_0$ directions and length g with no lingering steps, and suppose that the associated G-parking function on a generic graph Γ_g is given by the sequence $(d_0; \underline{x}_1, \ldots, \underline{x}_g)$.*

For a fixed i, we have that:

$$\underline{x}_i = \begin{cases} 0 & \text{if } p_i - p_{i-1} = (-1,-1,\ldots,-1) \\ p_{i-1}(j) & \text{if } p_i - p_{i-1} = e_j \end{cases}$$

Proof. This is the obvious reversal of the rule for constructing the lingering lattice paths (keeping in mind our convention). □

Combining the two propositions above, we obtain the following direct description of the bijection:

Proposition 4.9. *Supposing that T is the rectangular $(g-d+r)\times(r+1)$ standard Young tableau associated to a G-parking function on a generic Γ_g described by $(r; \underline{x}_1, \ldots, \underline{x}_g)$.*

For a fixed i, and the l'_s defined as above to be the number of cells in the s^{th} column of T whose entries are at most $i-1$, we have that:

$$\underline{x}_i = r + 1 - j + l'_j - l'_{r+1} \tag{3}$$

where l'_s is the number of cells in the s^{th} column of T whose entries are less than or equal to $i-1$.

Proof. Suppose that the number i is in the j^{th} column of T for $1 \leq j < r+1$. This means that $p_i - p_{i-1} = e_j$ and hence Proposition 4.8 gives us: $\underline{x}_i = p_{i-1}(j)$, while Proposition 4.7 gives us $p_{i-1} = r + 1 - j + l'_j - l'_{r+1}$.

If the number i is in the $r+1^{\text{st}}$ column of T, then $p_i - p_{i-1} = (-1, \dots, -1)$ and $\underline{x}_i = 0 = r + 1 - (r+1) + l'_{r+1} - l'_{r+1}$, so the formula holds in both cases. □

Example 4.10. Consider the standard Young tableau T with which Example 4.5 finished, and which is given below:

1	3	4
2	5	6

Then $\underline{x}_i = r + 1 - j + l'_j - l'_{r+1}$ where i is in column j and l'_s is the number of entries in column s that are at most $i-1$. Throughout, $r = 2$. We summarize the computations int the following table:

i	j	l'_j	l'_{2+1}	$r+1-j+l'_j-l'_{r+1}$	$\underline{x}_i$
1	1	0	0	$2+1-1+0-0=$	2
2	1	1	0	$2+1-1+1-0=$	3
3	2	0	0	$2+1-2+1-0=$	1
4	2	1	1	$2+1-3+0-0=$	0
5	2	1	1	$2+1-2+1-1=$	1
6	3	1	1	$2+1-3+1-1=$	0

As expected, we obtain back the chip configuration we started with, namely the one given by the sequence $(2; 2, 3, 1, 0, 1, 0)$.

4.2. **The actions of** $\text{ev}(T)$ **and** $\sigma(c)$**.** Given the propositions in the preceding section and the definition of evacuation on rectangular standard Young tableaux (rotate by 180° and flip the numbers according to the rule $i \to n+1-i$ where n is the number of cells in the tableau), we should be able to explicitly determine what evacuation does to the sequence $(d_0; \underline{x}_1, \dots, \underline{x}_g)$.

Proposition 4.11. *If the $(g-d+r)\times(r+1)$ standard Young tableaux T is associated to a G-parking function described by $(d_0; \underline{x}_1, \dots, \underline{x}_g)$ on generic Γ_g with $d_0 = r$ and $g = (g-d+r) \times (r+1)$, then $\text{ev}(T)$ is associated to such a function described by $(d_0; \underline{x}'_1, \dots, \underline{x}'_g)$ where*

$$\underline{x}'_{g+1-i} = j - 1 + l_j - l_1 \tag{4}$$

where l_s is the number of cells in the s^{th} column of T whose entries are at most i.

Proof. First, since the dimensions of the tableau do not change under evacuation, certainly the number of chips on the special vertex remains $d_0 = r$. Hence, all we need to find are the $\underline{x}'_i$ for $1 \leq i \leq g$. Explicitly, we will find the formula for $\underline{x}'_{g+1-i}$.

Evidently, evacuation takes column j to column $r+2-j$ so that if l_j is the number of cells in column j of T whose entries are at most i, then l_j is also the number of cells in column $r+2-j$ of $\text{ev}(T)$ whose entries are at least $g+1-i$ (since evacuation also flips the values of the entries).

In terms of the associated lingering lattice path $(p'_1, \dots, p'_g)$ to $\text{ev}(T)$, for $j \neq 1$ l_j counts the number of increases in the $r+2-j^{\text{th}}$ direction from p'_{g-i} to $p'_g =$

$(r, r-1, \dots, 1)$. Similarly, l_1 counts the number of decreases along all directions from p'_{g-i} to $p'_g = (r, r-1, \dots, 1)$. As a sanity check, note that when $i = 0$, all of the l_s are 0 as they should be since there are no steps from p'_{g-0} to p'_g. Furthermore, when $i = 1$, we have $l_1 = 1$ and $l_s = 0$ for $s \neq 1$ which is consistent with the fact that the step from p'_{g-1} to p'_g can only be a decrease, since it can neither linger nor increase as $p'_g = (r, r-1, \dots, 1)$ is the lowest the lingering lattice path can go.

Given that the l_j are counting steps in the $r+2-j^{\text{th}}$ direction from p'_{g-i} to p'_g, we obtain the following analogue of Proposition 4.7:

$$p'_{g-i} = p'_g - \begin{pmatrix} l_1 - l_{r+1} \\ l_1 - l_r \\ \dots \\ l_1 - l_2 \end{pmatrix} = \begin{pmatrix} r + l_{r+1} - l_1 \\ r - 1 + l_r - l_1 \\ \dots \\ 1 + l_2 - l_1 \end{pmatrix}$$

Next we obtain the analogue of Proposition 4.9 using exactly the same argument. Suppose that i is in the j^{th} column of T. Then we have that $g+1-i$ is in the $r+1-j^{\text{th}}$ column of $\text{ev}(T)$. If $j \geq 2$, then this means that $p'_{g+1-i} - p'_{g-i} = e_{r+2-j}$ and hence $x'_{g+1-i} = p'_{g-i}(r+2-j) = j - 1 + l_j - l_1$ where l_s is the number of cells in the s^{th} column of T whose entries are at most i.

Otherwise, if $j = 1$, we have that $g+1-i$ is in the $r+1^{\text{st}}$ column of $\text{ev}(T)$, which means that $p'_{g+1-i} - p'_{g-i} = (-1, \dots, -1)$ and $x'_{g+1-i} = 0 = 1 - 1 + l_1 - l_1$, so the formula holds in both cases. □

All that is left to do is determine that the sequence related to $\sigma(c)$ is the same as the one we obtained above for the G-parking function associated to $\text{ev}(T)$. Since the rank remains fixed under reflection, as does the degree, we know that we will still have $(g-d+r)(r+1) = g$ which guarantees that the number of chips on the special vertex will remain the same. Hence, we only need to figure out the $\underline{x}'_i$ of $\sigma(c)$. This is, however, easy with the following lemma, which relates $\underline{x}'_{g+1-i}$ to the $\underline{x}_i$, whose values we already have in terms of the combinatorics of the standard Young tableau.

Lemma 4.12. *Supposing that c is a rank r degree d G-parking function relative to v_0 on a generic graph Γ_g such that $g = (g+d-r)(r+1)$ and described by $(d_0; \underline{x}_1, \dots, \underline{x}_g)$ with $d_0 = r$, then $\sigma(c)$ on Γ'_g is chip-firing equivalent to a G-parking function relative to v'_0 described by $(d_0; \underline{x}'_1, \dots, \underline{x}'_g)$ where*

$$\underline{x}'_{g+1-i} = \max\{p_{i-1}(1) - \underline{x}_i - 1, 0\} \tag{5}$$

Proof. Note that the sequence $(d_0; \underline{x}'_1, \dots, \underline{x}'_g)$ of the G-parking function relative to v'_0 that is chip-firing equivalent to $\sigma(c)$ on Γ'_g can be interpreted as follows: if c' is the G-parking function relative to v_g that is chip-firing equivalent to c, then $\underline{x}'_{g+1-i}$ is the clockwise distance of the one chip on γ_i from v_{i-1} on Γ_g.

These clockwise distances, however, are determined by successively computing the equivalent G-parking functions relative to vertices v_1, then v_2, $\dots$, v_g. Lemma 3.6 applies and gives us the following.

Suppose that k is the number of chips on v_{i-1} of the G-parking function relative to v_{i-1} that is chip-firing equivalent to c. Then combining Lemma 3.6 with our notation for $\underline{x}_i$, we obtain:

(1) if $\underline{x}_i$ is 0, i.e. if there is no chip on $\gamma_i \setminus \{v_{i-1}\}$, then there will be one chip on $\gamma_i \setminus \{v_i\}$ that is a clockwise distance $(k-1)m_i$ away from v_i;

(2) if $\underline{x}_i m_i \not\equiv k m_i \bmod (l_i + m_i)$, then there is one chip on $\gamma_i \setminus \{v_i\}$ that is a clockwise distance $km_i - (\underline{x}_i + 1)m_i$ away from v_i;
(3) if $\underline{x}_i m_i \equiv k m_i \bmod (l_i + m_i)$, then there are no chips left on $\gamma_i \setminus \{v_i\}$.

Now, k is of course $p_{i-1}(1)$ since the top entries of the lingering lattice path keep track of exactly the number of chips at v_i of the G-parking functions relative to v_i chip-firing equivalent to c. Hence, the formula $\underline{x}'_{g+1-i} = \max\{p_{i-1}(1) - \underline{x}_i - 1, 0\}$ holds for all of the three cases above. □

Proof of Theorem 4.3. It is enough to show that the $\underline{x}'_{g+1-i}$ of the G-parking function relative to v'_0 equivalent to $\sigma(c)$ satisfy equation (4): $\underline{x}'_{g+1-i} = j - 1 + l_j - l_1$, where l_s is the number of cells of in the s^{th} column of the standard Young tableau associated to c whose entries are at most i , and j is the column in which i appears.

This is easy, however, since we have established the following equations:

(2) $p_{i-1}(1) = r + l'_1 - l'_{r+1}$;
(3) $x_i = r + 1 - j + l'_j - l'_{r+1}$;
(5) $x'_{g+1-i} = \max\{p_{i-1}(1) - x_i - 1, 0\}$.

where l'_s is the number of cells of in the s^{th} column of the standard Young tableau associated to c whose entries are at most $i - 1$, and j is the column in which i appears. Evidently:

$$x'_{g+1-i} = \max\{r + l'_1 - l'_{r+1} - (r + 1 - j + l'_j - l'_{r+1}) - 1, 0\} = \max\{j - 2 + l'_j - l'_1, 0\}$$

If we let l_s be the number of cells in the s^{th} column of T whose entries are at most i, then the fact that i is in the j^{th} column of T implies that $l_s = l'_s$ for $s \neq j$ and $l_j = l'_j + 1$. This gives us that if $j \neq 1$, then the formula in fact $\underline{x}'_{g+1-i} = \max\{j - 1 + l_j - l_1, 0\}$. Since the tableau is standard, we necessarily have $l_j \geq l_1$. Since also $j \geq 1$, we have that in fact $j - 1 + l_j - l_1 \geq 0$, so that the final formula for $j \neq 1$ is $\underline{x}'_{g+1-i} = j - 1 + l_j - l_1$.

If $j = 1$, then $\underline{x}'_{g+1-i} = \max\{1 - 2 + l'_1 - l'_1, 0\} = \max\{-1, 0\} = 0 = 1 - 1 + l_1 - l_1$, so the formula is still satisfied. □

5. Acknowledgments

This research was conducted at the 2011 summer REU (Research Experience for Undergraduates) program at the University of Minnesota, Twin Cities, and was partially funded by NSF grants DMS-1001933 and DMS-1067183. The authors would like to thank Profs. Gregg Musiker, Vic Reiner, and Pavlo Pylyavskyy, who directed the program, for their support. We would also like to express our particular gratitude to Prof. Musiker for introducing us to this problem, and for his invaluable guidance in both the research process and the writing of this report.

References

[BLS91] A. Björner, L. Lovász, and P. W. Shor, *Chip-firing games on graphs*, European Journal of Combinatorics **12** (1991), 283–291.

[BN07] M. Baker and S. Norine, *Riemann-Roch and Abel-Jacobi theory on a finite graph*, Advances in Mathematics **215** (2007), no. 2, 766–788.

[CDPR10] F. Cools, J. Draisma, S. Payne, and E. Robeva, *A tropical proof of the Brill-Noether theorem*, To appear in Advanced Mathematics; Arxiv preprint arXiv:1001.2774 (2010).

[Dha90] D. Dhar, *Self-organized critical state of sandpile automaton models*, Physical Review Letters **64** (1990), no. 14, 1613–1616.

[Luo11] Ye Luo, *Rank-determining sets of metric graphs*, Journal of Combinatorial Theory Series A **118** (2011), no. 6, 1775–1793.

[PS04] Alexander Postnikov and Boris Shapiro, *Trees, parking functions, syzygies, and deformations of monomial ideals*, Transactions of the American Mathematical Society **356** (2004), no. 8, 3109–3142.

[Sta08] Richard Stanley, *Promotion and evacuation*, The Electronic Journal of Combinatorics **16** (2008), no. 2.

E-mail address: `agraw025@umn.edu`

E-mail address: `sotirov@wisc.edu`

E-mail address: `fanwei@alum.mit.edu`