A Timed Version of the Plactic Monoid

Amritanshu Prasad

Abstract. Timed words are words where letters of the alphabet come with time stamps. We extend the definitions of semistandard tableaux, insertion, Knuth equivalence, and the plactic monoid to the setting of timed words. Using this, Greene's theorem is formulated and proved for timed words, and algorithms for the RSK correspondence are extended to real matrices.

1. Introduction

Lascoux and Schützenberger [**11**] introduced the plactic monoid to give a proof of the Littlewood-Richardson rule based on a strategy outlined by Robinson [**18**], and ideas of Schensted [**19**] and Knuth [**8**]. This theory revolves around bijections involving words and tableaux.

These bijections are restrictions to lattice points of certain volume-preserving piecewise linear bijections between convex polyhedra [**7, 6, 3, 14**]. The importance of this viewpoint is borne out in the work of Knutson and Tao, who proved the saturation of the monoid of triples (λ, μ, ν) of integer partitions such that the representation V_λ occurs in the tensor product $V_\mu \otimes V_\nu$ of representations of $GL_n(\mathbf{C})$. This led to the resolution of *Horn's conjecture* on the possible sets of eigenvalues of a sum of Hermitian matrices [**9, 10**].

Here we develop the monoid-theoretic foundations for piecewise linear correspondences interpolating bijections involving tableaux. This is done by generalizing the plactic monoid from the framework of the free monoid of words to the setting of *timed words*. Timed words were introduced by Alur and Dill [**1**] in their approach to the formal verification of real-time systems using timed automata. While words represent a sequence of events, timed words represent a sequence of events where the time of occurrence of each event is also recorded. We use a finite version of their definition of timed words. While each letter occurs discretely (an integer number of times) in a classical word, it appears for a positive duration (which is a real number) in a timed word.

2010 *Mathematics Subject Classification.* 05A05,68R15,05E10.

Key words and phrases. timed words, Greene invariants, Knuth relations, plactic monoid.

Supported by a Swarna Jayanti Fellowship of the Department of Science & Technology, Government of India. These results were announced at the 83rd annual conference of the Indian Mathematical Society, held at Sri Venkateshwara University, Tirupati where the author delivered the Srinivasa Ramanujan Memorial award lecture. An expository article on the classical version of Greene's theorem, and its extension to the timed case (based on the lecture) will appear in *The Mathematics Student* [**17**].

In Section 2.1 we introduce the timed versions of semistandard Young tableaux (called *timed tableaux*). Schensted's insertion algorithm is generalized to timed tableaux in Section 2.2. Greene invariants of timed words are introduced in Section 3.1. Timed versions of Knuth relations are introduced in Section 3.2.1. These relations are conceptually very similar to the relations introduced by Knuth in [**8**]. However, it was a delicate task to arrive at relations that are at once simple enough so that one can show that they preserve Greene invariants (Lemma 3.3.1) but also powerful enough to transform any timed word to its insertion tableau (Lemma 3.2.4). With this groundwork, the extension of Greene's theorem to timed words becomes routine (Theorem 3.4.1).

Standard properties of Knuth equivalence, such as the existence of a unique tableau in each Knuth class, and the characterization of Knuth equivalence in terms of Greene invariants are extended to timed words in Sections 4.1 and 4.2.

The RSK algorithm [**8**] is extended from integer matrices in Sections 5.1 and 5.2. Viennot's light-and-shadows version of the Robinson-Schensted correspondence, which was extended to integer matrices in [**15**], is now extended to real matrices. The piecewise linear nature of these algorithms can be easily seen from the timed version of Greene's theorem.

All the algorithms described here are straightforward to implement. An implementation in python is available at `http://www.imsc.res.in/~amri/timed_plactic/timed_tableau.py`. A jupyter worksheet with demos of many of the theorems and proofs in this paper is available at `http://www.imsc.res.in/~amri/timed_plactic/timed_tableau.ipynb` and in html format at `http://www.imsc.res.in/~amri/timed_plactic/timed_tableau.html`.

2. Insertion in Timed Tableaux

2.1. Timed Tableaux. Let $A_n = \{1, \dots, n\}$, to be thought of as a linearly ordered alphabet.

Definition 2.1.1 (Timed Word). A *timed word* of length r in the alphabet A_n is a piecewise-constant right-continuous function $w : [0, r) \to A_n$ with finitely many discontinuities. We write $l(w) = r$. In other words, for some finite sequence $0 = r_0 < r_1 < \dots < r_k = r$ of transition points, and letters $c_1, \dots, c_k$ in A_n, $w(x) = c_i$ if $r_{i-1} \leq x < r_i$. Given such a function, we write

$$w = c_1^{t_1} c_2^{t_2} \cdots c_k^{t_k}, \tag{1}$$

where $t_i = r_i - r_{i-1}$. We call this an *exponential string* for w. The *weight* of w is the vector:

$$\mathrm{wt}(w) = (m_1, \dots, m_n),$$

where m_i is the Lebesgue measure of the pre-image of i under w, in other words,

$$m_i = \mathrm{meas}(w^{-1}(i)) = \sum_{j \in \{1,\dots,k\},\ c_j = i} t_j.$$

The exponential string, as defined above, is not unique; if two successive letters c_i and c_{i+1} are equal, then we can merge them, replacing $c_i^{t_i} c_{i+1}^{t_{i+1}} = c_i^{t_i + t_{i+1}}$.

The above definition is a finite variant of Definition 3.1 of Alur and Dill [**1**], where $r = \infty$, and there is an infinite increasing sequence of transition points.

Given timed words w_1 and w_2, their *concatenation* is defined in the most obvious manner—their exponential strings are concatenated (and if necessary, successive equal values merged). The monoid formed by all timed words in an alphabet A_n, with product defined by concatenation, is denoted by $A_n^\dagger$. The submonoid of $A_n^\dagger$ consisting of timed words where the exponents $t_1, t_2, \ldots, t_k$ in exponential string (1) are integers is the free monoid A_n^* of words in the alphabet A_n (see e.g., [**12**, Chapter 1]).

Definition 2.1.2 (Timed Subword). Given a timed word $w : [0, r) \to A_n$, and $S \subset [0, r)$ a finite disjoint union of intervals of the form $[a, b) \subset [0, r)$, the *timed subword w_S of w with respect to S* is defined by:

$$w_S(t) = w(\inf\{u \in [0, r) \mid \mathrm{meas}([0, u) \cap S) \geq t\}) \text{ for } 0 \leq t < \mathrm{meas}(S).$$

Intuitively, w_S is obtained from w by cutting out the segments that are outside S. Given timed words v and w, v is said to be a *timed subword* of w if there exists $S \subset [0, r)$ as above such that $v = w_S$. Timed subwords $v_1, \ldots, v_k$ of w are said to be *pairwise disjoint* if there exist pairwise disjoint subsets $S_1, \ldots, S_k$ as above such that $v_i = w_{S_i}$ for $i = 1, \ldots, k$.

Definition 2.1.3 (Timed Row). A *timed row* in the alphabet A_n is a weakly increasing timed word in $A_n^\dagger$. In exponential notation every timed row is of the form $1^{t_1} \cdots n^{t_n}$ where $t_i \geq 0$ for $i = 1, \ldots, n$. The set of all timed rows in $A_n^\dagger$ is denoted $\mathrm{R}_n^\dagger$. The set of all timed rows of length l is denoted $\mathrm{R}_n^\dagger(l)$.

Definition 2.1.4 (Row Decomposition). Every timed word w has a unique decomposition:

$$w = u_l u_{l-1} \cdots u_1,$$

such that u_i is a timed row for each $i = 1, \ldots, l$, and $u_i u_{i-1}$ is not a row for any $i = 2, \ldots, l$. We shall refer to such a decomposition as the *row decomposition* of w.

Given two timed rows u and v, say that u is *dominated* by v (denoted $u \lhd v$) if

(1) $l(u) \geq l(v)$, and
(2) $u(t) < v(t)$ for all $0 \leq t < l(v)$.

Definition 2.1.5 (Real Partition). A *real partition* is a weakly decreasing finite sequence $\lambda = (\lambda_1, \ldots, \lambda_l)$ of non-negative real numbers. Two real partitions are regarded as equal if one may be obtained from the other by the removal of trailing zeroes.

Definition 2.1.6 (Timed Tableau). A *timed tableau* in A_n is a timed word w in A_n with row decomposition $w = u_l u_{l-1} \cdots u_1$ such that $u_1 \lhd \cdots \lhd u_l$. The *shape* of w is the real partition $(l(u_1), l(u_2), \ldots, l(u_l))$, and the *weight* of w is the weight of w as a timed word (see Definition 2.1.1). The set of all timed tableaux in A_n is denoted $\mathrm{Tab}_n^\dagger$. The set of all timed tableaux of shape λ is denoted $\mathrm{Tab}_n^\dagger(\lambda)$. The set of all timed tableaux of shape λ and weight μ is denoted $\mathrm{Tab}_n^\dagger(\lambda, \mu)$.

The above is a direct generalization of the notion of the reading word of a tableau in the sense of [**13**].

Example 2.1.7. $w = 3^{0.8} 4^{1.1} 1^{1.4} 2^{1.6} 3^{0.7}$ is a timed tableau in A_5 of shape $(3.7, 1.9)$ and weight $(1.4, 1.6, 1.5, 1.1, 0)$.

2.2. Timed Insertion. Given a timed word w and $0 \leq a < b \leq l(w)$, according to Definition 2.1.2, $w_{[a,b)}$ is the timed word of length $b-a$ such that:

$$w_{[a,b)}(t) = w(a+t) \text{ for } 0 \leq t < b-a.$$

Definition 2.2.1 (Timed row insertion). Given a timed row u, a letter $c \in A_n$, and a real number $t_c \geq 0$, the insertion $\mathrm{RINS}(u, c^{t_c})$ of c^{t_c} into u is defined as follows: if $u(t) \leq c$ for all $0 \leq t < l(u)$, then

$$\mathrm{RINS}(u, c^{t_c}) = (\varnothing, uc^{t_c}),$$

where $\varnothing$ denotes the *empty word* of length zero. Otherwise, there exists $0 \leq t < l(u)$ such that $u(t) > c$. Let

$$t_0 = \min\{0 \leq t < l(u) \mid u(t) > c\}.$$

Define

$$\mathrm{RINS}(u, c^{t_c}) = \begin{cases} (u_{[t_0, t_0+t_c)}, u_{[0,t_0)} c^{t_c} u_{[t_0+t_c, l(u))}) & \text{if } l(u) - t_0 > t_c, \\ (u_{[t_0, l(u))}, u_{[0,t_0)} c^{t_c}) & \text{if } l(u) - t_0 \leq t_c. \end{cases}$$

When $t_c = 1$ and u lies in the image of A_n^* in $A_n^\dagger$, this coincides with the first step of the algorithm INSERT from Knuth [**8**] where c is inserted into a row: if $\mathrm{RINS}(u, c^1) = (v', u')$, then u' is the new row obtained after insertion, v' is the letter bumped out by c.

If v is a row of the form $c_1^{t_1} \cdots c_k^{t_k}$, define $\mathrm{RINS}(u, v)$ by induction on k as follows: Having defined $(v', u') = \mathrm{RINS}(u, c_1^{t_1} \cdots c_{k-1}^{t_{k-1}})$, let $(v'', u'') = \mathrm{RINS}(u', c_k^{t_k})$. Then define

$$\mathrm{RINS}(u, v) = (v'v'', u'').$$

Remark 2.2.2. The insertion of the timed row $v = c_1^{t_1} \cdots c_k^{t_k}$ into u is achieved by successively inserting $c_i^{t_i}$ as i runs from 1 to k. The segments are ejected are taken from u from left to right, with no overlaps. It follows that, if $(v', u') = \mathrm{RINS}(u, v)$, then v' is again a timed row.

Remark 2.2.3. Definition 2.2.1 is in terms of exponential strings, which are not uniquely associated to timed words. Therefore, in order that row insertion be well-defined for timed words, it is necessary to check that the result of row insertion of $c^{t_1+t_2}$ is the same as the result of row insertion of c^{t_1} followed by row insertion of c^{t_2}. This is straightforward.

Example 2.2.4. $\mathrm{RINS}(1^{1.4}2^{1.6}3^{0.7}, 1^{0.7}2^{0.2}) = (2^{0.7}3^{0.2}, 1^{2.1}2^{1.1}3^{0.5})$.

Definition 2.2.5 (Timed Tableau Insertion). Let w be a timed tableau with row decomposition $u_l \ldots u_1$, and let v be a timed row. Then $\mathrm{INSERT}(w, v)$, the insertion of v into w, is defined as follows: first v is inserted into u_1. If $\mathrm{RINS}(u_1, v) = (v_1', u_1')$, then v_1' is inserted into u_2; if $\mathrm{RINS}(u_2, v_1') = (v_2', u_2')$, then v_2' is inserted in u_3, and so on. This process continues, generating $v_1', \ldots, v_l'$ and $u_1', \ldots, u_l'$. $\mathrm{INSERT}(w, v)$ is defined to be $v_l' u_l' \cdots u_1'$. It is quite possible that $v_l' = \varnothing$.

Example 2.2.6. If w is the timed tableau from Example 2.1.7, then

$$\mathrm{INSERT}(w, 1^{0.7}2^{0.2}) = 3^{0.7}4^{0.2}2^{0.7}3^{0.3}4^{0.9}1^{2.1}2^{1.1}3^{0.5}.$$

When the timed tableau w lies in the image of A_n^*, then $\mathrm{INSERT}(w, c^1)$ is the same as the result of applying the algorithm $\mathrm{INSERT}(c)$ from Knuth [**8**] to the semistandard Young tableau w.

Definition 2.2.7. Given real partitions $\lambda = (\lambda_1, \dots, \lambda_l)$ and $\mu = (\mu_1, \dots, \mu_{l-1})$, we say that μ *interleaves* λ if the inequalities

$$\lambda_1 \geq \mu_1 \geq \lambda_2 \geq \mu_2 \geq \cdots \geq \lambda_{l-1} \geq \mu_{l-1} \geq \lambda_l$$

hold. In other words, the successive parts of μ lie in-between the successive parts of λ.

Theorem 2.2.8. *For any timed tableau w in A_n and any timed row v in A_n,* $\mathrm{INSERT}(w, v)$ *is again a timed tableau in A_n. We have*

$$\mathrm{wt}(\mathrm{INSERT}(w, v)) = \mathrm{wt}(w) + \mathrm{wt}(v),$$

and $\mathrm{shape}(w)$ *interleaves* $\mathrm{shape}(\mathrm{INSERT}(w, v))$*.*

Proof. The main step in this proof is the following lemma:

Lemma 2.2.9. *Suppose x and y are timed rows in A_n such that $x \lhd y$. For any timed row v in A_n, suppose $(v', x') = \mathrm{RINS}(x, v)$, and $(v'', y') = \mathrm{RINS}(y, v')$.*

(1) $x' \lhd y'$
(2) $\mathrm{wt}(v'') + \mathrm{wt}(x') + \mathrm{wt}(y') = \mathrm{wt}(x) + \mathrm{wt}(y) + \mathrm{wt}(v)$
(3) $l(y') \leq l(x)$.

Proof of the lemma. By inserting in stages, assume that $v = c^t$ for some $c \in A_n$ and some $t > 0$.

If x never exceeds c, then $x' = xc^t$, and $y' = y$, and so $x' \lhd y'$. Otherwise, when c^t is inserted into x, v' is a segment of x corresponding to an interval $[t_0, t_0 + \delta)$ such that $x(t_0) > c$. This segment in x is replaced by a segment c^δ to obtain x'. Let $c_1^{t_1} \cdots c_k^{t_k}$, with $c < c_1 < \cdots < c_k$, be the exponential string of v'.

Proceed by induction on k. If $k = 1$, $v' = c_1^{t_1}$. Now $y(t_0) > x(t_0) = c_1$, so $c_1^{t_1}$ will displace a segment of y, one that begins to the left of t_0, with $c_1^{t_1}$, and so $x' \lhd y'$.

For $k > 1$, perform the insertion of c^t into x in two steps, first inserting c^{t_1}, and then inserting c^{t-t_1}. If $(v_1', x_1') = \mathrm{RINS}(x, c^{t_1})$, then $v_1' = c_1^{t_1}$. Let $(v_1'', y_1') = \mathrm{RINS}(y, v_1')$. By the $k = 1$ case, $x_1' \lhd y_1'$.

Now $(c_2^{t_2} \cdots c_k^{t_k}, y') = \mathrm{RINS}(y_1', c^{t-t_1})$, so y' is obtained by inserting $c_2^{t_2} \cdots c_k^{t_k}$ into y_1'. Therefore, by induction hypothesis, $x' \lhd y'$, proving (1).

The assertion (2) about weights is straightforward. For (3), observe that v' is a concatenation of segments from x. Write $v' = wz$, where w consists of segments that come from $u_{[0, l(y))}$ and z consists of segments that come from $u_{[l(y), l(x))}$. Then, from the arguments in the proof of the first part of the lemma, the segments in w will all replace segments of y, so if $(v'', y'') = \mathrm{RINS}(y, w)$, then $l(y'') = l(y)$. Now $(v', y') = \mathrm{RINS}(y'', z)$, whence $l(y') \leq l(y'') + l(z) \leq l(y) + [l(x) - l(y)] = l(x)$. □

We now prove that $\mathrm{INSERT}(w, v)$ is a timed tableau. Suppose w has row decomposition $u_l u_{l-1} \cdots u_1$. Using the notation of Definition 2.2.5, and writing v_0 for v, we have $(v_i', u_i') = \mathrm{RINS}(u_i, v_{i-1}')$ and $(v_{i+1}', u_{i+1}') = \mathrm{RINS}(u_{i+1}, v_i')$. The first assertion of Lemma 2.2.9, with $v = v_i'$, $x = u_i$, and $y = u_{i+1}$, shows that $u_i' \lhd u_{i+1}'$ for $i = 1, \dots, l-1$. Taking $x = u_l$ and $y = \varnothing$ gives $u_l' \lhd v_l'$. Therefore $\mathrm{INSERT}(w, v)$ is a timed tableau. The third assertion of Lemma 2.2.9, with the same settings, gives $l(u_{i+1}') \leq l(u_i)$, showing that $\mathrm{shape}(w)$ interleaves $\mathrm{shape}(\mathrm{INSERT}(w, v)$. The assertion about weights in the theorem follows easily from the second assertion of Lemma 2.2.9. □

Definition 2.2.10 (Insertion Tableau of a Timed Word). Let w be a timed word with row decomposition $u_1 \cdots u_l$. The insertion tableau of w is defined as:

$$P(w) = \mathrm{INSERT}(\cdots \mathrm{INSERT}(\mathrm{INSERT}(u_1, u_2), u_3), \dots, u_l).$$

Example 2.2.11. If $w = 3^{0.8}1^{0.5}4^{1.1}1^{0.9}2^{1.6}3^{0.7}1^{0.7}2^{0.2}$ has four rows in its row decomposition. $P(w)$ is calculated via the following steps:

w	$P(w)$
$3^{0.8}$	$3^{0.8}$
$3^{0.8}1^{0.5}4^{1.1}$	$3^{0.5}1^{0.5}3^{0.3}4^{1.1}$
$3^{0.8}1^{0.5}4^{1.1}1^{0.9}2^{1.6}3^{0.7}$	$3^{0.8}4^{1.1}1^{1.4}2^{1.6}3^{0.7}$
$3^{0.8}1^{0.5}4^{1.1}1^{0.9}2^{1.6}3^{0.7}1^{0.7}2^{0.2}$	$3^{0.7}4^{0.2}2^{0.7}3^{0.3}4^{0.9}1^{2.1}2^{1.1}3^{0.5}$

Definition 2.2.12 (Schützenberger Involution on Timed Words). Given $w = c_1^{t_1} \cdots c_k^{t_k} \in A_n^\dagger$, define

$$w^\sharp = (n - c_k + 1)^{t_k} \cdots (n - c_1 + 1)^{t_1}, \tag{2}$$

in effect, reversing both the order on the alphabet, and the positional order of letters in the timed word.

Lemma 2.2.13. *Let u and v be timed rows. Suppose* $\mathrm{RINS}(u, v) = (v', u')$, *and* $l(v') = l(v)$. *Then* $\mathrm{RINS}(u'^\sharp, v'^\sharp) = (v^\sharp, u^\sharp)$.

Proof. It suffices to consider the case where $v = c^t$. The hypothesis $l(v') = l(v)$ implies that $t_0 = \inf\{t \mid u(t) > c\}$ satisfies $0 \le t_0 \le l(u) - c$, and

$$u' = u_{[0,t_0)} c^t u_{[t_0+t, l(u))}, \text{ and } v' = u_{[t_0, t_0+t)}.$$

Using induction as in the proof of Theorem 2.2.8, we may assume that v' is constant, so $v' = d^t$ for some $d > c$.

Now

$$u'^\sharp = u^\sharp_{[t_0+t, l(u))}(n - c + 1)^t u^\sharp_{[0,t_0)} \text{ and } v'^\sharp = (n - d + 1)^t.$$

Since all the values of $u_{[t_0+t,l(u))}$ are greater than or equal to d, all the values of $u^\sharp_{[t_0+t,l(u))}$ are less than or equal to $n - d + 1$. Moreover, $n - c + 1 > n - d + 1$. It follows immediately from Definition 2.2.1 that $\mathrm{RINS}(u'^\sharp, v'^\sharp) = (v^\sharp, u^\sharp)$. □

Corollary 2.2.14. *The timed row insertion algorithm gives rise to a bijection:*

$$\mathrm{RINS} : \mathrm{R}_n^\dagger(r) \times \mathrm{R}_n^\dagger(s) \tilde{\to} \{(v', u') \in \mathrm{R}_n^\dagger(r + s - r') \times \mathrm{R}_n^\dagger(r') \mid r' \ge \max(r, s),\ u' \lhd v'\}.$$

Proof. Suppose $(u, v) \in \mathrm{R}_n^\dagger(r) \times \mathrm{R}_n^\dagger(s)$, and $(v', u') = \mathrm{RINS}(u, v)$. Then (u, v) can be recovered from (v', u') (given the prior knowledge of r and s) as follows: let $(v_1^\sharp, u_1^\sharp) = \mathrm{RINS}({u'_{[0,r)}}^\sharp, v'^\sharp)$. Then using Lemma 2.2.13, u and v can be recovered as $u = u_1$, and $v = v_1 u'_{[r,r')}$. □

Theorem 2.2.15 (Timed Pieri Rule). *The timed insertion algorithm gives rise to a bijection:*

$$\mathrm{INSERT} : \mathrm{Tab}_n^\dagger(\lambda) \times \mathrm{R}_n^\dagger(r) \tilde{\to} \coprod_{\substack{\lambda \text{ interleaves } \mu \\ l(\lambda)+r=l(\mu)}} \mathrm{Tab}_n^\dagger(\mu)$$

PROOF. Let $\lambda = (\lambda_1, \dots, \lambda_l)$. Let $w \in \mathrm{Tab}_n^\dagger(\lambda)$ have row decomposition $u_l \cdots u_1$, and $x \in \mathrm{R}_n^\dagger(r)$. Suppose that $w' = \mathrm{INSERT}(w, x)$ has row decomposition $u'_{l+1} \cdots u'_1$ (with the possibility that $u'_{l+1} = \varnothing$). We already know that shape(w) interleaves shape(w) (Theorem 2.2.8). Given timed rows u' and v' such that $u' \triangleleft v'$, and non-negative real numbers r and s such that $r \leq l(u')$, let $\mathrm{RINS}_r^{-1}(v', u')$ denote the unique pair of rows (u, v) such that $l(u) = r$, $l(v) = s$, and $(v', u') = \mathrm{RINS}(u, v)$ (see Corollary 2.2.14). Then the rows of w can be recovered from w' as follows:

$$\begin{aligned}
(x_l, u_l) &= \mathrm{RINS}_{\lambda_l}^{-1}(u'_{l+1}, u'_l),\\
(x_{l-1}, u_{l-1}) &= \mathrm{RINS}_{\lambda_{l-1}}^{-1}(x_l, u'_{l-1}),\\
(x_{l-2}, u_{l-2}) &= \mathrm{RINS}_{\lambda_{l-2}}^{-1}(x_{l-1}, u'_{l-2}),\\
&\vdots\\
(x_1, u_1) &= \mathrm{RINS}_{\lambda_1}^{-1}(x_2, u'_1),
\end{aligned}$$

and finally x can be recovered as $x = x_1$. □

DEFINITION 2.2.16 (Deletion). Let $w' \in \mathrm{Tab}_n^\dagger(\mu)$ and let λ be a real partition that interleaves μ. Then we write

$$\mathrm{DELETE}_\lambda(w') = (v, w) \text{ if and only if } w \in \mathrm{Tab}_n^\dagger(\lambda) \text{ and } \mathrm{INSERT}(w, v) = w'.$$

The pair (v, w) is computed from w' and λ by the algorithm described in the proof of Theorem 2.2.15.

3. Greene's Theorem

3.1. Greene Invariants for Timed Words.

DEFINITION 3.1.1 (Greene Invariants for Timed Words). Given $w \in A_n^\dagger$, its *kth Greene invariant* $a_k(w)$ is defined as the maximum possible sum of lengths of a set of k pairwise disjoint subwords of w (see Definition 2.1.2) that are all timed rows:

$$a_k(w) = \sup\{l(u_1) + \cdots + l(u_k) \mid u_1, \dots, u_k \text{ are pairwise disjoint subwords, and each } u_i \text{ is a timed row}\}$$

LEMMA 3.1.2. *If w is a timed tableau of shape $\lambda = (\lambda_1, \dots, \lambda_l)$, then for each $1 \leq k \leq l$,*

$$a_k(w) = \lambda_1 + \cdots + \lambda_k.$$

PROOF. This proof is very similar to the proof of the corresponding result for ordinary tableaux [**5, 13**]. Suppose w has row decomposition $u_l u_{l-1} \cdots u_1$. Then $u_1, \dots, u_k$ are pairwise disjoint subwords that are rows, so

$$a_k(w) \geq \lambda_1 + \cdots + \lambda_k.$$

Conversely, any row subword of w cannot have overlapping segments from two different rows u_i and u_j of w, because if $i > j$, then $u_i(t) > u_j(t)$, but in the row decomposition of w, u_i occurs before u_j. Therefore, k disjoint subwords can have length at most the sum of lengths of the largest k rows of w, which is $\lambda_1 + \ldots + \lambda_k$. □

3.2. Timed Knuth Equivalence and the Timed Plactic Monoid.

Definition 3.2.1 (Timed Knuth Relations). Assume that x, y and z are timed rows such that xyz is also a timed row. The timed Knuth relations are given by:

$$(\kappa_1) \qquad xzy \equiv zxy \text{ if } l(z) = l(y) \text{ and } \lim_{t\to l(y)^-} y(t) < z(0),$$

$$(\kappa_2) \qquad yxz \equiv yzx \text{ if } l(x) = l(y) \text{ and } \lim_{t\to l(x)^-} x(t) < y(0).$$

Definition 3.2.2 (Timed Plactic Monoid). The *timed plactic monoid* $\mathrm{pl}^\dagger(A_n)$ is the quotient $A_n^\dagger/\equiv$, where $\equiv$ is the congruence generated by the timed Knuth relations (κ_1) and (κ_2).

In other words, two elements of $A_n^\dagger$ are said to *differ by a Knuth relation* if they are of the form uv_1w and uv_2w, where v_1 and v_2 are terms on opposite sides of one of the timed Knuth relations (κ_1) and (κ_2). Knuth equivalence $\equiv$ is the equivalence relation generated by Knuth relations. Since this equivalence is stable under left and right multiplication in $A_n^\dagger$, the concatenation product on $A_n^\dagger$ descends to a product on the set $\mathrm{pl}^\dagger(A_n)$ of Knuth equivalence classes, giving it the structure of a monoid.

Lemma 3.2.3. *Two timed words v and w differ by a Knuth relation (κ_1) if and only if $v^\sharp$ and $w^\sharp$ (see Definition 2.2.12) differ by a Knuth relation (κ_2).*

Proof. When the involution $w \mapsto w^\sharp$ is applied to the Knuth relation (κ_1), the Knuth relation (κ_2) is obtained. □

Lemma 3.2.4. *Every timed word is Knuth equivalent to its timed insertion tableau.*

Proof. First we show that, for timed rows u and v, if $(v', u') = \mathrm{RINS}(v, u)$, then $v'u' \equiv uv$. By insertion in stages, we assume that $v = c^t$. If $u(t) \leq c$ for all $0 \leq t < l(u)$, there is nothing to show. Otherwise, a segment y of u, beginning at t_0, and of length $t_1 = \min(l(u) - t_0, t)$ is displaced by the segment c^{t_1} of c^t. Write $u = x'yx''$. It suffices to show $x'yx''c^{t_1} \equiv yx'c^{t_1}x''$. But this can be done in two steps as follows (the segment to which the Knuth relation is applied is underlined):

$$x'\underline{yx''c^{t_1}} \equiv_{\kappa_2} x'\underline{yc^{t_1}x''} = \underline{x'yc^{t_1}}x'' \equiv_{\kappa_1} \underline{yx'c^{t_1}}x''.$$

From Definition 2.2.10, it suffices to show that $\mathrm{INSERT}(w, v) \equiv wv$ for every timed tableau w and every timed row v. Suppose w has row decomposition $u_l u_{l-1} \cdots u_1$ then, with the notations of Definition 2.2.5,

$$\begin{aligned} wv &= u_l \cdots u_2 u_1 v \\ &\equiv u_l \cdots u_2 v_1' u_1' \\ &\equiv u_l \cdots v_2' u_2' u_1' \\ &\vdots \\ &\equiv v_l' u_l' \cdots u_2' u_1' = \mathrm{INSERT}((w, v), \end{aligned}$$

by repeated application of the assertion at the beginning of this proof. □

3.3. Knuth Equivalence and Greene Invariants.

LEMMA 3.3.1. *If two timed words are Knuth equivalent, then they have the same Greene invariants.*

PROOF. It suffices to prove that if two words differ by a Knuth relation they have the same Greene invariants. For the Knuth relation (κ_1), suppose that xyz is a timed row with $l(z) = l(y)$, and the last letter of y is strictly less than the first letter of z. For any timed words w and u, we wish to show that Greene invariants coincide for $wxzyu$ and $wzxyu$. Since every timed row subword of $wzxyu$ is also a timed row subword of $wxzyu$, $a_k(wzxyu) \leq a_k(wxzyu)$ for all k.

To prove the reverse inequality, for any set of pairwise disjoint row subwords $v_1, \dots, v_k$ of $wxzyu$, it suffices to construct pairwise disjoint row subwords $v_1', \dots, v_k'$ of $wzxyu$ such that $\sum_{i=1}^k l(v_k') \geq \sum_{i=1}^k l(v_k)$. Write $v_i = w_i x_i z_i y_i u_i$ for each i, where w_i, x_i, z_i, y_i and u_i are (possibly empty) row subwords of w, x, z, y and u respectively.

Since the last letter of y is strictly smaller than the first letter of z, it cannot be that $y_i \neq \varnothing$ and $z_i \neq \varnothing$ simultaneously for the same i. If, for all i, $x_i = \varnothing$, or $z_i = \varnothing$, then each v_i remains a row subword of $wzxyu$, so we may take $v_i' = v_i$ for all i.

Otherwise, there exists i such that $v_i = w_i x_i z_i u_i$, with $x_i \neq \varnothing$, and $z_i \neq \varnothing$. Without loss of generality, assume that this is the case for $i = 1, \dots, r$, and not for $i = r+1, \dots, k$ for some $1 \leq r \leq k$. If $y_i = \varnothing$ for all i, then set

$$v_i' = \begin{cases} w_1 x_1 y u_1 & \text{for } i = 1, \\ w_i x_i u_i & \text{for } 1 < i \leq r, \\ v_i & \text{for } r < i \leq k. \end{cases}$$

Since $l(y) = l(z) \geq \sum_{i=1}^r l(z_i)$, $\sum_{i=1}^k l(v_i') \geq \sum_{i=1}^k l(v_i)$. By construction the words $v_1', \dots, v_k'$ are pairwise disjoint timed row subwords of w.

Finally, suppose there exists at least one index i such that $y_i \neq \varnothing$, say $y_{r+1} \neq \varnothing$. Also, assume that $\max z_1$ (the largest letter of z_1) is at least as large as $\max z_i$ for $i = 2, \dots, r$. Let z_0 be the timed row obtained by concatenating all the segments of z_i for $1 \leq i \leq r$ in the order in which they occur in z. It follows that $\max z_0 = \max z_1$, and $l(z_0) = l(z_1) + \cdots + l(z_r)$. Let x_0 be the timed row obtained by concatenating all the segments of x_1 and x_{r+1} in the order in which they occur in x. If $\min x_1 \leq \min x_{r+1}$, define

$$v_i' = \begin{cases} w_1 x_0 y_{r+1} u_{r+1} & \text{for } i = 1, \\ w_i x_i u_i & \text{for } 1 < i \leq r, \\ w_{r+1} z_0 u_1 & \text{for } i = r+1, \\ v_i & \text{for } r+1 < i \leq k. \end{cases}$$

If $\min x_1 > \min x_{r+1}$, then interchange w_1 and w_{r+1} in the above definition. In both cases the words $v_1', \dots, v_k'$ are pairwise disjoint row subwords of $wzxyu$ whose lengths add up to l.

For the Knuth relation (κ_2), a similar argument can be given. However, a more elegant method is to use Lemma 3.2.3, noting that $a_k(w) = a_k(w^\sharp)$ for all $k \geq 1$ and all $w \in A_n^\dagger$, thereby reducing it to (κ_1). □

3.4. The timed version of Greene’ theorem.

Theorem 3.4.1 (Timed version of Greene’s theorem). *For every $w \in A_n^\dagger$, if the timed tableau $P(w)$ has shape $\lambda = (\lambda_1, \dots, \lambda_l)$, then*

$$a_k(w) = \lambda_1 + \dots + \lambda_k \text{ for } k = 1, \dots, l.$$

Proof. Greene’s theorem holds when w is a timed tableau (Lemma 3.1.2). By Lemma 3.3.1, Greene invariants remain unchanged under the timed versions of Knuth relations. By Lemma 3.2.4, every timed word is Knuth equivalent to its timed insertion tableau. Therefore, the Greene invariants of a timed word are given by the shape of its insertion tableau as stated in the theorem. □

Remark 3.4.2. The proof of Lemma 3.1.2 shows that the supremum in the definition of Greene invariants (Definition 3.1.1) is attained for timed tableaux. From the proof of Lemma 3.3.1, it follows that this supremum is attained for every timed word in the Knuth equivalence class of a timed tableau, and therefore for every $w \in A_n^\dagger$.

4. Knuth Equivalence Classes

4.1. Tableaux in Knuth Equivalence Classes. Given $w \in A_n^\dagger$, let $\bar{w}$ denote the word in $A_{n-1}^\dagger$ whose exponential string is obtained by removing all terms of the form n^t with $t > 0$ from the exponential string of w. The word $\bar{w}$ is called the restriction of w to A_{n-1}.

Lemma 4.1.1. *For every timed tableau $w \in A_n^\dagger$, $\bar{w}$ is also a timed tableau. Moreover,* shape$(\bar{w})$ *interleaves* shape(w)*.*

Proof. Suppose w has row decomposition $u_l u_{l-1} \cdots u_1$. Since n is the largest element of A_n, we may write $u_i = u_i' n^{t_i}$ for some $t_i \geq 0$. Clearly $l(u_i) \geq l(u_i')$. Since w is semistandard, $l(u_i') \geq l(u_{i+1})$ for $i = 1, \dots, l-1$. It follows that the shape of w', which is $(l(u_1'), \dots, l(u_l'))$ interleaves the shape of w, which is $(l(u_1), \dots, l(u_l))$. Since $u_i \lhd u_{i+1}$, it follows that $u_i' \lhd u_{i+1}$ for $i = 1, \dots, l-1$. □

Lemma 4.1.2. *If $v, w \in A_n^\dagger$ are Knuth equivalent, then their restrictions to A_{n-1}, $\bar{v}$ and $\bar{w}$ are Knuth equivalent in $A_{n-1}^\dagger$.*

Proof. Applying the restriction to A_{n-1} map $w \mapsto \bar{w}$ to both sides of the Knuth relation (κ_1) gives: $x\bar{z}y$ and $\bar{z}xy$. Write $y = y'y''$, where $l(y') = l(\bar{z})$, we have

$$x\bar{z}y'y'' \equiv \bar{z}xy'y'',$$

a Knuth relation in in $A_{n-1}^\dagger$. A similar argument works for the Knuth relation (κ_2). □

Theorem 4.1.3. *Every Knuth equivalence class in $A_n^\dagger$ contains a unique timed tableau.*

Proof. The existence of a timed tableau in each Knuth equivalence class is ensured by Lemma 3.2.4. The proof of uniqueness is by induction on n. The base case, where $n = 1$ is trivially true. Now suppose v and w are Knuth equivalent timed tableaux in $A_n^\dagger$. By Lemmas 4.1.1 and 4.1.2 $\bar{v}$ and $\bar{w}$ are Knuth equivalent timed tableaux in $A_{n-1}^\dagger$. By the induction hypothesis, $\bar{v} = \bar{w}$. Let $\lambda = (\lambda_1, \dots, \lambda_l)$ be the shape of this timed tableau. By Lemma 3.3, v and w have the same Greene invariants, and therefore the same shape $\mu = (\mu_1, \dots, \mu_{l+1})$. It follows that both v and w are obtained from $\bar{v} = \bar{w}$ by appending $n^{\mu_i - \lambda_i}$ to the ith row of $\bar{v} = \bar{w}$ for each i, hence $v = w$. □

4.2. Characterization of Knuth Equivalence Classes.

Classical Knuth equivalence can be characterized in terms of Greene invariants (see [**11**, Theorem 2.15]). The same characterization works for timed Knuth equivalence.

Theorem 4.2.1. *Let w and w' be timed words in $A_n^\dagger$. Then w and w' are Knuth equivalent if and only if, for all timed words u and v in $A_n^\dagger$, $a_k(uwv) = a_k(uw'v)$ for all $k \geq 1$.*

Proof. If w and w' are Knuth equivalent, then so are uwv and $uw'v$. By the timed version of Greene's theorem (Theorem 3.4.1) $a_k(uwv) = a_k(uw'v)$ for all $u, v \in A_n^\dagger$.

For the converse, suppose that w and w' are not Knuth equivalent. Then $P(w) \neq P(w')$. If $P(w)$ and $P(w')$ do not have the same shape, then by Theorem 3.4.1, they do not have the same Greene invariants, so taking $u = v = \varnothing$ proves the result.

Now suppose that w and w' are rows of the same length. If $w \neq w'$, there exist decompositions $w = xc^t y$ and $w' = xc'^t y'$, where $c \neq c'$, and $t > 0$. If $c < c'$, then for $T > t + l(y)$,

$$a_1(xc^t yc^T) = l(x) + t + T, \text{ while } a_1(xc'^t yc^T) = l(x) + T,$$

thereby proving the result.

In the general case, suppose $w = u_l u_{l-1} \cdots u_1$ and $w' = u'_l u'_{l-1} \cdots u'_1$ are row decompositions. Let i be the least integer such that $u_i \neq u'_i$. By the proof for rows, there exists $c \in A_n$, and $T > 0$ such that when $(v, x) = \mathrm{RINS}(u_i, c^T)$ and $(v', x') = \mathrm{RINS}(u'_i, c^T)$, then $l(x) \neq l(x')$. Also, note that c is at least i, the least possible value of the ith row of a tableau.

Now assume that $T > l(u_j)$ for $j = 1, \dots, i-1$. Take $u = 1^T 2^T \cdots (i-1)^T$. Then we have

$$P(uw) = u_l \cdots u_{i+1} i^T u_i (i-1)^T u_{i-1} \cdots 1^T u_1,$$
$$P(uw') = u'_l \cdots u'_{i+1} i^T u'_i (i-1)^T u_{i-1} \cdots 1^T u_1.$$

Now $z = c^T (i-1)^T u_{i-1} \cdots 1^T u_1$ is a timed tableau. Let

$$(v, \bar{z}) = \mathrm{DELETE}_{(T+\lambda_1, \dots, T+\lambda_{i-1})}(z).$$

Then when $P(uwv)$ and $P(uw'v)$ are computed, the calculations are the same for the first $i-1$ rows. But then c^T is inserted into u_i and u'_i to obtain the ith rows of $P(uwv)$ and $P(uw'v)$, which, by our earlier argument, will have different lengths. □

5. The Real RSK Correspondence

5.1. Definition using Timed Insertion Tableaux.

Let $M_{m \times n}(\mathbf{R}^+)$ denote the set of all $m \times n$ matrices with non-negative real entries. Given $A = (a_{ij}) \in$

$M_{m\times n}(\mathbf{R}^+)$, define its *timed column word* u_A, and *timed row word* v_A as follows:

$$u_A = 1^{a_{11}}2^{a_{12}}\cdots n^{a_{1n}}\,1^{a_{21}}2^{a_{22}}\cdots n^{a_{2n}}\cdots 1^{a_{m1}}2^{a_{m2}}\cdots n^{a_{mn}}.$$

$$v_A = 1^{a_{11}}2^{a_{21}}\cdots m^{a_{m1}}\,1^{a_{12}}2^{a_{22}}\cdots m^{a_{m2}}\cdots 1^{a_{1n}}2^{a_{2n}}\cdots m^{a_{mn}}.$$

The timed word u_A is obtained by reading column numbers of A along its rows, timed by its entries. The timed word v_A is obtained by reading the row numbers of A along its columns, timed by its entries. Define:

$$\mathrm{RSK}(A) = (P(u_A), P(v_A)). \tag{3}$$

This is a direct generalization of the definition of the RSK correspondence given in [**16**, Section 18].

Example 5.1.1. Let

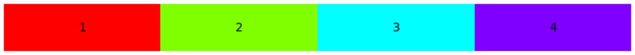
$$A = \begin{pmatrix} 0.16 & 0.29 & 0.68 & 0.44 \\ 0.29 & 0.70 & 0.38 & 0.45 \\ 0.32 & 0.29 & 0.43 & 0.70 \end{pmatrix}.$$

Then $(P,Q) = \mathrm{RSK}(A)$ are given by:

$$P = 3^{0.32}4^{0.29}2^{0.60}3^{0.65}4^{0.55}1^{0.77}2^{0.67}3^{0.52}4^{0.75},$$

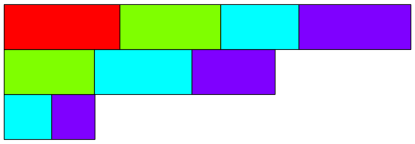
$$Q = 3^{0.61}2^{1.38}3^{0.43}1^{1.57}2^{0.45}3^{0.70},$$

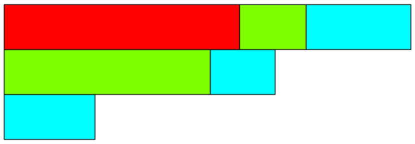
which have common shape $(2.71, 1.81, 0.61)$. Visually, four different colours can be used to depict the letters of our alphabet:



,

then the tableaux P and Q can be represented by the images:





From this visual representation, it is evident that P and Q are timed tableaux of the same shape.

Theorem 5.1.2. *The function* RSK *defines a bijection:*

$$\mathrm{RSK} : M_{m\times n}(\mathbf{R}^+) \xrightarrow{\sim} \coprod_{\lambda} \mathrm{Tab}_n^{\dagger}(\lambda) \times \mathrm{Tab}_m^{\dagger}(\lambda),$$

where λ *runs over all real partitions with at most* $\min(m,n)$ *parts.*

Remark 5.1.3. Let μ_i denote the sum of the ith row of A, and ν_j the sum of the jth column. Let $\mu = (\mu_1, \dots, \mu_m)$, and $\nu = (\nu_1, \dots, \nu_n)$. Then, if $\mathrm{RSK}(A) = (P,Q)$ then $\mathrm{wt}(P) = \nu$, and $\mathrm{wt}(Q) = \mu$.

Remark 5.1.4 (Relation to Knuth's definition). Knuth [**8**] defined $\mathrm{RSK}(A) = (P,Q)$ for integer matrices in a slightly different manner. His definition of $P = P(u_A)$ is exactly the same as the definition here. However Q is defined as a *recording tableau* which has the same shape as P by its very construction. With Knuth's construction, each step (insertion followed by recording) is reversible, and it is clear that a bijection is obtained. The symmetry property of the RSK correspondence, that $\mathrm{RSK}(A^T) = (Q,P)$ if $\mathrm{RSK}(A) = (P,Q)$ is then stated as a non-trivial theorem.

The definition (3) is the extension to real matrices of the definition in [**16**, Section 18] for integer matrices. With this definition it is immediate that if $\mathrm{RSK}(A) = (P, Q)$, then $\mathrm{RSK}(A^T) = (Q, P)$ since $u_{A^T} = v_A$. However, it is not immediately clear that P and Q have the same shape, and that the correspondence is invertible. These are proved using Greene's theorem in [**16**]. For real matrices, the timed version of Greene's theorem allows the proof of [**16**] to be carried out for real matrices. For the sake of completeness, this argument is given in full detail below.

LEMMA 5.1.5. *For every $A \in M_{m\times n}(\mathbf{R}^+)$, the tableaux $P(u_A)$ and $P(v_A)$ have the same shape.*

PROOF. Any timed subword w of u_A is of the form

$$w = 1^{b_{11}}2^{b_{12}}\cdots n^{b_{1n}}\,1^{b_{21}}2^{b_{22}}\cdots n^{b_{2n}}\cdots 1^{b_{m1}}2^{b_{m2}}\cdots n^{b_{mn}},$$

where $0 \leq b_{ij} \leq a_{ij}$ for all (i, j). If w is a row, then the indices $(i_1, j_1), \dots, (i_k, j_k)$ for which $b_{ij} > 0$, when taken in the order in which they appear in w, must satisfy $i_1 \leq \dots \leq i_k$, and $j_1 \leq \dots \leq j_k$. Define a partial order on the set

$$P_{mn} = \{(i, j) \mid 1 \leq i \leq m, 1 \leq j \leq n\}$$

by $(i, j) \leq (i', j')$ if and only if $i \leq i'$ and $j \leq j'$. Then it follows that the kth timed Greene invariant of u_A (Definition 3.1.1) is given by:

$$a_k(u_A) = \max_C \sum_{(i,j)\in C} a_{ij},$$

where the maximum is taken over the set of all subsets $C \subset P_{mn}$ which can be written as a union of k chains. Since the order relation on P_{mn} corresponds to the order relation on P_{nm} under $(i, j) \leftrightarrow (j, i)$, it follows that $a_k(v_A) = a_k(u_A)$ for all k. Thus, the timed version of Greene's theorem (Theorem 3.4) implies that P and Q have the same shape. □

5.2. Insertion-Recording Algorithm for $\mathrm{RSK}(A)$**.** Given real partitions λ and μ such that λ interleaves μ, and $w \in \mathrm{Tab}^{\dagger}_{m-1}(\lambda)$, define the *inflation of w to shape μ by m* to be the unique tableau $\mathrm{INFL}_\mu(w, m)$ of shape μ whose restriction to $m - 1$ is equal to w. In the notation of Section 4.1, $\overline{\mathrm{INFL}_\mu(w, m)} = w$.

Given $A \in M_n(\mathbf{R}^+)$, let $r_{i,A} = 1^{a_{i1}}2^{a_{i2}}\cdots n^{a_{in}}$. Then $u_A = r_{1,A}r_{2,A}\cdots r_{m,A}$.

Insertion-Recording Algorithm

```
• $P \leftarrow \varnothing$, $Q \leftarrow \varnothing$.
• For $i = 1, \dots, m$, repeat the following steps:
    – $P \leftarrow \mathrm{INSERT}(P, r_{i,A})$.
    – $\lambda \leftarrow \mathrm{shape}(P)$.
    – $Q \leftarrow \mathrm{INFL}_\lambda(Q, i)$
• Return $(P, Q)$.
```

LEMMA 5.2.1. *For every $A \in M_{m\times n}(\mathbf{R}^+)$, the output of the insertion-recording algorithm is* $\mathrm{RSK}(A)$ *as defined in (3).*

PROOF. The proof is by induction on the number m of rows in A. The base case of $m = 1$ is trivial.

Now suppose A' denotes the submatrix consisting of the first $m - 1$ rows of A. Then $u_A = u_{A'}r_{m,A}$, so that $P(u_A) = \mathrm{INSERT}(P(u_{A'}), r_{m,A})$, Also, the restriction $\bar{v}_A$ of v_A to $m - 1$ is $v_{A'}$.

Since $v_{A'}$ is the restriction of v_A to A_{m-1}, by Lemma 4.1.2, $P(v_{A'})$ is Knuth equivalent to the restriction of $P(v_A)$ to A_{m-1}. Theorem 4.1.3, implies that $P(v_{A'})$ is equal to the restriction of $P(v_A)$ to A_{m-1}. Therefore $P(v_A) = \mathrm{INFL}_\lambda(P(v_{A'}), m)$, which is the output of the insertion-recording algorithm. □

Proof of Theorem 5.1.2. The proof uses the fact that the insertion-recording algorithm is invertible. Following the notation of the proof of Lemma 5.2.1, it suffices to recover $r_{m,A}$, $P(u_{A'})$ and $P(v_{A'})$ from $P(u_A)$ and $P(v_A)$ to reverse the insertion-recording algorithm. For this, observe that $P(v_{A'})$ is just the restriction of $P(v_A)$ to A_{m-1}. If μ is the shape of $P(v_{A'})$, then $(r_{m,A}, P(u_{A'})) = \mathrm{DELETE}_\mu(P(u_A))$ (see Definition 2.2.16). □

5.3. Light-and-Shadows Real RSK. Viennot described a visual version of the Robinson-Schensted-Correspondence for permutations, using the *light and shadows* method [**20**]. This algorithm was extended to the RSK correspondence on integer matrices by Fulton using the matrix-ball method [**4**]. Another such extension, called the VRSK algorithm, was given in [**15**, Chapter 3]. In VRSK one can work directly with the matrices themselves, without having to draw them as configurations of points in the plane, which get unwieldy when matrices have large entries. Another unforeseen advantage of the VRSK algorithm is that a minor variant works for real matrices, giving the correspondence of (3). This new algorithm, which we call the *light-and-shadows real RSK* is introduced in this section. The piecewise linear nature of the RSK correspondence becomes clear from this algorithm.

Definition 5.3.1 (Sequence of Leading Points). For a matrix $A \in M_{m\times n}(\mathbf{R}^+)$, consider the set

$$\mathrm{supp}(A) = \{(i,j) \in P_{mn} \mid a_{ij} > 0\}.$$

Then the sequence $L(A)$ of leading points of A is the set $\max(\mathrm{supp}(A))$ (with respect to the poset structure on P_{mn}) arranged in a sequence

$$L(A) = (i_1, j_1), \dots, (i_r, j_r)$$

such that $j_1 < \cdots < j_r$ and (since this set is an antichain in P_{mn}) $i_1 > \cdots > i_r$.

Light-and-Shadows Real RSK

```
• $P \leftarrow \varnothing$, $Q \leftarrow \varnothing$.
• While $A$ is non-zero repeat the following steps:
    – Set $S \leftarrow 0_{m\times n}$ ($m \times n$ zero matrix)
    – Set $p = \varnothing$, $q = \varnothing$.
    – While $A$ is non-zero repeat the following steps:
        ∗ Compute $L(A) = (i_1, j_1), \dots, (i_r, j_r)$ of $A$
        ∗ Let $m(A) = \min\{a_{i_1 j_1}, \dots, a_{i_r, j_r}\}$
        ∗ Set $a_{i_s, j_s} \leftarrow a_{i_s, j_s} - m(A)$ for $s = 1, \dots r$
        ∗ Set $s_{i_{s+1}, j_s} \to s_{i_{s+1}, j_s} + m$ for $s = 1, \dots, r-1$
        ∗ Set $p \leftarrow pj_1^m$, $q \leftarrow qi_r^m$
    – $P \leftarrow pP$, $Q \leftarrow qQ$
    – $A \leftarrow S$
• Return $(P, Q)$
```

Theorem 5.3.2. *When the light-and-shadows real RSK algorithm is applied to* $A \in M_{m\times n}(\mathbf{R}^+)$*, it return* $\mathrm{RSK}(A)$.

Proof. For the proof, we introduce an algorithm that is midway between the insertion-recording algorithm of Section 5.2 and the light-and-shadows real RSK.

Row-wise RSK Algorithm

```
• $P \leftarrow \varnothing$, $Q \leftarrow \varnothing$
• While $A$ is non-zero repeat the following steps:
    – Set $p \leftarrow \varnothing$, $q \leftarrow \varnothing$
    – For $i = 1, \dots, m$, repeat the following steps:
        * Set $S \leftarrow 0_{m\times n}$.
        * Set $(v, u) = \mathrm{RINS}(p, 1^{a_{i1}} \cdots n^{a_{in}})$
        * If $v = 1^{s_1} \cdots n^{s_n}$, set $s_{ij} \leftarrow s_j$ for $j = 1, \dots, n$.
    – Set $A \leftarrow S$.
    – $P \leftarrow pP$, $Q \leftarrow \mathrm{INFL}_{\mathrm{shape}(P)}(Q, i)$.
• return $(P, Q)$
```

The main loop of this algorithm starts with a matrix A, and replaces it with the matrix $S = s_{ij}$ computed using timed row insertion. It also computes the first row of the tableau P and Q as p, q. We claim that the function $A \mapsto (S, p, q)$ of the main loop of the row-wise RSK algorithm is the same as the function $A \mapsto (S, p, q)$ of the main loop of the light-and-shadows real RSK. We call S the *shadow matrix* of A.

Write A as a block matrix $\binom{A'}{A''}$, where A' is an $(m-1)\times n$ matrix and A'' is a $1 \times n$ matrix. It suffices to show that if the light-and-shadows real RSK algorithm return (P', Q') on A' and (P, Q) on A, the $P = \mathrm{INSERT}(P', u_{A''})$. Here $u_{A''}$ is just the row:

$$1^{a_{m1}} \cdots n^{a_{mn}}.$$

The inner loop of the light-and-shadows real RSK algorithm produces a sequence $A' = A'_1, A'_2, \dots, A'_h$ of matrices as it runs on input A'. Let $L'_k = L(A'_k)$ and $m'_k = m(A'_k)$, for $k = 1, \dots, h$. When the inner loop finishes running, we have $p' = {j'_1}^{m'_1} \cdots {j'_h}^{m'_h}$, and $q' = {i'_1}^{m'_1} \cdots {i'_h}^{m'_h}$, where j'_k is the least non-zero column, and i'_k is the least non-zero row of A'_k.

Now A is obtained from A' by adding a new row $\begin{pmatrix} a_{m1} & \cdots & a_{mn} \end{pmatrix}$. To begin with, assume that this row has only one non-zero entry, a_{mj_0}. Let $L_1, L_2, \dots$, and $m_1, m_2, \dots$ be the corresponding sequences of leading points, and their corresponding smallest entries respectively. If $j_0 \geq j_i$ for all i, then $L_k = L'_k$ for all $k = 1, \dots, h$. In addition, A has a singleton sequence of leading points $\{(m, j_0)\}$. As a result, the output of the main loop is $p = p' u_{A''}$, and $q = q' m^{a_{mj_0}}$, and the shadow matrix of A is the same as the shadow matrix of A'. The same outcome is obtained from the main loop of the row-wise RSK algorithm. The hypothesis that $j_0 \geq j_i$ for all i is equivalent to saying that A'' has its non-zero entries to the left of any non-zero entries of A'. Therefore $P(u_A) = P(u_{A'}) u_{A''}$, and the shadow matrix of A' generated by row-wise RSK is the same as the shadow matrix of A generated by row-wise RSK.

Now suppose that $j_0 < j_l$ for some l, and take the least such value l. Then the sequences of leading points $L_1, \dots, L_{l-1}$ of A are the same as the sequences $L'_1, \dots, L'_{l-1}$. If $a_{mj_0} \geq m'_{j'_l} + \dots m'_{j'_h}$, then $L_k = \{(m, j_0)\} \cup L'_k$ for $k = l, \dots, h$.

Therefore $p = {j'_1}^{m'_1} \cdots {j'_{l-1}}^{m'_{l-1}} j_0^{a_{mj_0}}$. From the definition of timed row insertion (Definition 2.2.1), $\mathrm{RINS}(p', j_0^{a_{mj_0}}) = (j_l^{m_l} \cdots j_h^{m_h}, p)$. Also, $q = q' m^{a_{mj_0}}$. Finally, S is obtained from S' by adding m'_k to the (m, j_k)th entry of S' for each $k = l, \dots, h$.

Otherwise, $m_{j_l} + \cdots + m_{j_{q-1}} < a_{mj_0} \leq m_{j_1} + \cdots + m_{j_q}$ for some $l \leq q < h$. In this case the sequences of leading points for A are given by:

$$L_k = \begin{cases} L'_k & \text{for } 1 \leq k < l-1, \\ (m, j_0) \cup L'_k & \text{for } l \leq k \leq q, \\ L'_{k-1} & \text{for } q \leq l \leq h, \end{cases}$$

$p = {j'_1}^{m'_1} \cdots j_{l-1}^{m'_{l-1}} j_0^{a_{m0}} j_q^{m'_{j_1} + \cdots + m'_{j_q} - a_{mj_0}} j_{q+1}^{m'_{q+1}} \cdots j_h^{m'_h}$. Again from the definition of timed row insertion, $\mathrm{RINS}(p', j_0^{a_{mj_0}}) = (j_l^{m'_l} \cdots j_{q-1}^{m'_{q-1}} j_q^{a_{mj_0} - (m'_1 + \cdots + m'_{l-1})}, p)$. Also, $q = q' m^{a_{mj_0}}$. Finally, the value m'_k is added to the (m, j_k)th entry of S' for $k = l, \dots, q-1$, and $a_{mj_0} - (m'_1 + \cdots + m'_{l-1})$ is added to the (m, j_q)th entry of S' to obtain S.

Thus we have seen that when A'' has a single non-zero entry, the effect of this entry modifies the outputs of both the row-wise real RSK algorithm and the light-and-shadows real RSK algorithm in exactly the same manner.

If the last row of A has more than one non-zero entry, they may be dealt with sequentially (from left to right) to get the same outcome. □

5.4. Piecewise Linear RSK.

Definition 5.4.1 (Gelfand-Tsetlin Pattern). A *Gelfand-Tsetlin pattern* of size n is a triangle $T = (\lambda_i^{(k)} \mid 1 \leq k \leq n, 1 \leq i \leq k)$ of non-negative real numbers:

$$\begin{array}{ccccccccc} \lambda_1^{(n)} & & \lambda_2^{(n)} & & \cdots & & \cdots & & \lambda_n^{(n)} \\ & \lambda_1^{(n-1)} & & \lambda_2^{(n-1)} & & \cdots & & \lambda_{n-1}^{(n-1)} & \\ & & \ddots & & \ddots & & \iddots & & \\ & & & \ddots & & \iddots & & & \\ & & & & \lambda_1^{(1)} & & & & \end{array}$$

such that $\lambda_i^{(k)} \geq \lambda_i^{(k-1)} \geq \lambda_{i+1}^{(k)}$ for $k = 2, \dots, n$ and $i = 1, \dots, k-1$. The shape of a Gelfand-Tsetlin pattern of size n is its *top row*, $\lambda^{(n)}$.

Whenever $n \geq k$, define $r_k^n : A_n^\dagger \to A_k^\dagger$ by taking $r_k^n(w)$ to be the timed word whose exponential string is obtained from the exponential string of w by deleting all terms of the form c^t where $c > k$. It is easy to see that if $w \in A_n^\dagger$ is a timed tableau, then so is $r_k^n(w)$ for all $k = 1, \dots, n$.

In this section, given a timed tableau w in A_n, its shape will always be written as a real partition with n components, $\lambda = (\lambda_1, \dots, \lambda_n)$, where $\lambda_1 \geq \cdots \lambda_n \geq 0$.

Given a timed tableau w in A_n, define partitions $\lambda^{(k)} = (\lambda_1^{(k)}, \dots, \lambda_k^{(k)})$ by $\lambda^{(k)} = \mathrm{shape}(r_k^n(w))$. By Lemma 4.1.1, $\lambda_{(k-1)}$ interleaves $\lambda^{(k)}$ for all $k = 2, \dots, n$. Therefore the numbers $\lambda_i^{(k)}$ form a Gelfand-Tsetlin pattern, which we denote by $GT(w)$. The shape of λ is also the shape of $GT(w)$. Conversely, given a Gelfand-Tsetlin pattern T, it is easy to reconstruct the unique timed tableau w such that $T = GT(w)$.

The space of all Gelfand-Tsetlin patterns of size n, being defined by a finite collection of homogeneous inequalities in $\binom{n+1}{2}$ variables, forms a polyhedral cone in $R^{\binom{n+1}{2}}$. In terms of the notation introduced in Section 5.1, we have the following lemma, which is well-known for the RSK correspondence on integer matrices (see e.g., [**6**, Prop. 2.26]).

Lemma 5.4.2. *Given* $A \in M_{m\times n}(\mathbf{R}^+)$ *with* $\mathrm{RSK}(A) = (P, Q)$*, let* $(\lambda_i^{(k)}) = GT(P)$ *and* $(\mu_i^{(k)}) = GT(Q)$*, Gelfand-Tsetlin patterns of size* n *and* m *respectively. Then*

$$\lambda_1^{(j)} + \cdots \lambda_k^{(j)} = \max_{C \subset P_{mj} \textit{ union of at most } k \textit{ chains}} \sum_{(i,j)\in C} a_{ij}, \tag{4}$$

$$\mu_1^{(i)} + \cdots + \mu_k^{(i)} = \max_{C \subset P_{in} \textit{ union of at most } k \textit{ chains}} \sum_{(i,j)\in C} a_{ij}, \tag{5}$$

for all $1 \leq j \leq n$*, and* $1 \leq i \leq m$*.*

Proof. Let A_j is the submatrix of A consisting of its first j columns, then $u_{A_j} = r_j^n(u_A)$. By a repeated application of Lemma 4.1.2, $P(u_{A_j}) = r_j^n(P(u_A))$. But the j row of $GT(P)$ is, by definition, the shape of $r_j^n(P(u_A))$. But the shape of $P(u_{A_j})$ is given by (4), as explained in the proof of Lemma 5.1.5. The identity (5) has a similar proof. □

Corollary 5.4.3. *The RSK correspondence defines a continuous piecewise linear bijection from the cone* $M_{m\times n}(\mathbf{R}^+)$ *onto the cone of pairs of Gelfand-Tsetlin patterns* $((\lambda_i^{(k)}), (\mu_i^{(k)}))$ *of sizes* n *and* m *respectively, with* $\lambda^{(n)} = \mu^{(m)}$ *(after padding the shorter of the two with zeros).*

Lemma 5.4.2 clearly demonstrates the piecewise linear nature of the RSK correspondence: The algorithms of Sections 5.1–5.3 allow for fast computations of this piecewise linear map. While Eqs. (4) and (5) are used to *define* the RSK correspondence in [**6**], in this article, the RSK *algorithm* is extended to real matrices, and Lemma 5.4.2 is proved for the extended algorithm.

A more detailed analysis of the piecewise linear nature of the real RSK correspondence will be carried out in [**2**].

Acknowledgements. I thank R. Ramanujam for introducing me to the work of Alur and Dill on timed automata. I thank Xavier Viennot for teaching me about the wonderful world of the RSK correspondence, and encouraging me to work on this project. I thank Maria Cueto, Sachin Gautam, Sridhar P. Narayanan, Digjoy Paul, K. N. Raghavan, Ramanathan Thinniyam, and S. Viswanath for many interesting discussions, which helped me develop the ideas in this article. I thank Darij Grinberg for carefully reading, and sending me corrections to an earlier version of this article.

References

[1] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126, 1994.

[2] M. Cueto, S. Gautam, and A. Prasad. The real RSK correspondence, 2018. In preparation.

[3] J. A. De Loera and T. B. McAllister. Vertices of Gelfand–Tsetlin polytopes. *Discrete & Computational Geometry*, 32(4):459–470, Nov 2004.

[4] W. Fulton. *Young Tableaux: With Applications to Representation Theory and Geometry.* Cambridge University Press, 1997.

[5] C. Greene. An extension of Schensted's theorem. *Advances in Mathematics*, 14:254–265, 1974.

[6] A. N. Kirillov. Introduction to tropical combinatorics. In *Physics and Combinatorics*, pages 82–150. World Scientific, 2011.

[7] A. N. Kirillov and A. D. Berenstein. Groups generated by involutions, Gelfand-Tsetlin patterns, and combinatorics of Young tableaux. *Algebra i Analiz*, 7:92–152, 1995.

[8] D. E. Knuth. Permutations, matrices, and generalized Young tableaux. *Pacific Journal of Mathematics*, 34:709–727, 1970.

[9] A. Knutson and T. Tao. The honeycomb model of $gl_n(\mathbf{c})$ and tensor products i: proof of the saturation conjecture. *Journal of the AMS*, 12(4):1055–1090, 1999.

[10] A. Knutson and T. Tao. Honeycombs and sums of Hermitian matrices. *Notices of the AMS*, 48(2):175–186, 2001.

[11] A. Lascoux and M.-P. Schützenberger. Le monoïde plaxique. In *Noncommutative structures in algebra and geometric combinatorics (Naples)*. Consiglio Nazionale delle Ricerche, Roma, 1978.

[12] M. Lothaire. *Combinatorics on Words*. Cambridge University Press, 1983.

[13] M. Lothaire. *Algebraic Combinatorics on Words*. Cambridge University Press, 2002. Chapter 5: The Plactic Monoid, by A. Lascoux, B. Leclerc and Y. Thibon; also available from `http://www-igm.univ-mlv.fr/~berstel/Lothaire/AlgCWContents.html`.

[14] I. Pak. Four questions on Birkhoff polytope. *Annals of Combinatorics*, 4:83–90, 2000.

[15] A. Prasad. *Representaion Theory: A Combinatorial Viewpoint*. Cambridge University Press, 2015.

[16] A. Prasad. An introduction to Schur polynomials. https://arxiv.org/abs/1802.06073, 2018. Notes from a course at the NCM school on Schubert Varieties, held at The Institute of Mathematical Sciences, Chennai, in November 2017.

[17] A. Prasad. Knuth's moves on timed words. *The Mathematics Student*, 87(3–4), July–December 2018.

[18] G. d. B. Robinson. On the representations of the symmetric groups. *American Journal of Mathematics*, 60:745–760, 1938.

[19] C. Schensted. Longest increasing and decreasing subsequences. *Order*, 7:179–191, 1961.

[20] G. Viennot. Une forme géométrique de la correspondance de robinson-schensted. In *Combinatoire et représentation du groupe symétrique*, pages 29–58. Springer, 1977.

The Institute of Mathematical Sciences, Chennai

Homi Bhabha National Institute, Mumbai