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 \((\lambda, \mu, \nu)\) of integer partitions such that the representation \(V_\lambda\) occurs in the tensor product \(V_\mu \otimes V_\nu\) of representations of \(GL_n(\mathbb{C})\). 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.
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_tableau.py](http://www.imsc.res.in/~amri/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](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](http://www.imsc.res.in/~amri/timed_plactic/timed_tableau.html).

2. Insertion in Timed Tableaux

2.1. Timed Tableaux. Let $A_n = \{1, \ldots, n\}$, 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 $w : [0, r) \to A_n$. We write $l(w) = r$. In other words, for some finite sequence $0 = r_0 < r_1 < \ldots < r_k = r$ of transition points, and letters $c_1, \ldots, c_k$ in $A$, $w(x) = c_i$ if $r_{i-1} \leq x < r_i$. Given such a function, we write

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

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

$$\text{wt}(w) = (m_1, \ldots, m_n),$$

where $m_i$ is the Lebesgue measure of the pre-image of $i$ under $w$, i.e., $m_i = \text{meas}(w^{-1}(i))$.

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, will be denoted by $A_1^n$. The submonoid of $A_1^n$ consisting of timed words where the exponents $t_1, t_2, \ldots, t_k$ in exponential string $[1]^{t_i}$ are integers is the free monoid $A_1^n$ of words in the alphabet $A_1^n$ (see e.g., [12], Chapter 1).

**Definition 2.1.2 (Timed Subword).** Given a timed word $w: [0, r) \rightarrow A_1^n$, and $S \subset [0, r)$ a finite disjoint union of intervals of the form $[a, b) \subset [0, r)$, the timed subword of $w$ with respect to $S$ is defined as the timed word:

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

Intuitively, $w_S$ is obtained from $w$ by cutting out the segments that are outside $S$. Given two words $v$ and $w$, $v$ is said to be a subword of $w$ if there exists $S \subset [0, r)$ as above such that $v = w_S$. 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 $A_1^n$ is a weakly increasing timed word in $A_1^n$. In exponential notation every timed row is of the form $1_{t_1}^n \cdots 1_{t_n}^m$ where $t_i \geq 0$ for $i = 1, \ldots, n$. The set of all timed rows in $A_1^n$ is denoted $R_1^n$. The set of all timed rows of length $l$ is denoted $R_1^n(l)$.

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

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

such that $u_i$ is a timed row for each $i = 1, \ldots, l$, and $u_{i-1} u_i$ 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 $u \lessdot v$) if

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

**Definition 2.1.5 (Timed Tableau).** A timed tableau in $A_1^n$ is a timed word $w$ in $A_1^n$ with row decomposition $w = u_1 u_{l-1} \cdots u_1$ such that $u_1 \lessdot \cdots \lessdot u_l$. The shape of $w$ is the weakly decreasing tuple $(l(u_1), l(u_2), \ldots, l(u_l))$ of positive real numbers (henceforth called a real partition), and the weight of $w$ is the weight of $w$ as a timed word (see Definition 2.1.1). The set of all timed tableau in $A_1^n$ is denoted $\text{Tab}_n^1$. The set of all timed tableau of shape $\lambda$ is denoted $\text{Tab}_n^1(\lambda)$. The set of all tableau of shape $\lambda$ and weight $\mu$ is denoted $\text{Tab}_n^1(\lambda, \mu)$.

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

**Example 2.1.6.** $w = 3^{0.8} 4^{1.1} 1^{1.4} 2^{1.6} 3^{0.7}$ 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

**Definition 2.1.2 (Timed Subword).** 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:

$$w_{[a,b)}(t) = w(a + t) \text{ for } 0 \leq t < b - a.$$
Otherwise, there exists $0 \leq t < l(u)$ such that $w(t) > c$. Let
\[ t_0 = \min\{0 \leq t < l(u) \mid u(t) > c\}. \]

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

When $t = 1$ and $u$ lies in the image of $A^*$ in $A^\dagger$, this coincides with the first step of the algorithm $\text{INSERT}$ from Knuth [8] where $c$ is inserted into a row: if $RINS(u, c) = (v', u')$, 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 $c_1^{t_1} \ldots c_k^{t_k}$. Define $RINS(u, v)$ by induction on $k$ as follows: Having defined $(v', u') = RINS(u, c_1^{t_1} \ldots c_{k-1}^{t_{k-1}})$, let $(v'', u'') = RINS(u', c_k^{t_k})$. Then define
\[
RINS(w, u) = (v''v', u'').
\]

**Example 2.2.2.** \(RINS(1^{1.4}2^{1.6}3^{1.9}0^{0.7}, 1^{0.7}0^{0.2}) = (2^{0.7}3^{0.2}, 1^{2.1}2^{1.1}3^{0.5}).\)

**Definition 2.2.3 (Timed Tableau Insertion).** Let $w$ be a timed tableau with row decomposition $u_1 \ldots u_k$, and let $v$ be a timed row. Then $\text{INSERT}(w, v)$, the insertion of $v$ into $w$ is defined as follows: first $v$ is inserted into $u_1$. If $RINS(u_1, v) = (v'_1, u'_1)$, then $v'_1$ is inserted into $u_2$; if $RINS(u_2, v'_1) = (v'_2, u'_2)$, then $v'_2$ is inserted in $u_3$, and so on. This process continues, generating $v'_1, \ldots, v'_k$ and $u'_1, \ldots, u'_k$. $\text{INSERT}(t, v)$ is defined to be $v'_1v'_2 \ldots v'_k$. It is quite possible that $v'_k = \emptyset$.

**Example 2.2.4.** If $w$ is the timed tableau from Example 2.1.6 then
\[
\text{INSERT}(w, 1^{0.7}2^{0.2}) = 3^{0.7}4^{0.2}2^{0.7}3^{0.3}4^{0.9}1^{2.1}2^{1.1}3^{0.5}.
\]

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

**Definition 2.2.5.** Given real partitions $\lambda = (\lambda_1, \ldots, \lambda_l)$ and $\mu = (\mu_1, \ldots, \mu_{l-1})$, we say that $\mu$ interleave $\lambda$ if the inequalities
\[
\lambda_1 \geq \mu_1 \geq \lambda_2 \geq \mu_2 \geq \ldots \geq \lambda_{l-1} \geq \mu_{l-1} \geq \mu_l.
\]
In other words, the successive parts of $\mu$ lie in-between the successive parts of $\lambda$.

**Theorem 2.2.6.** For any timed tableau $w$ in $A_n$ and any timed row $v$ in $A_n$, $\text{INSERT}(w, v)$ is again a timed tableau in $A_n$. We have
\[
\text{wt}(\text{INSERT}(w, v)) = \text{wt}(w) + \text{wt}(v),
\]
and $\text{shape}(w)$ interleave $\text{shape}(\text{INSERT}(w, v))$.

**Proof.** We first prove that $\text{INSERT}(w, v)$ is a timed tableau. For this purpose, by inserting in stages, one may assume that $v = c_i$ for some $c_i \in A_n$ and some $t > 0$. Let $w$ have row decomposition $u_0u_1 \ldots u_k$. If $u_1(t) \leq c$ for all $0 \leq t < l(u_1)$, then $u'_1 = u_1c_i'$, and $u'_2 = u_2$, and so $u'_1 \subseteq u'_2$. Otherwise, when $c_i'$ is inserted into $u_1$, $v_1'$ is a segment of $u_1$ corresponding to an interval $[t_0, t_0 + \delta)$ such that $u_1(t_0) > c$. This segment in $u_1$ is replaced by a segment $c_i$ to obtain $u'_1$. Let $v_1' = c_1^{t_1} \ldots c_k^{t_k}$ (so $c < c_1 < \ldots < c_k$) be the exponential notation for $v_1'$.

Proceed by induction on $k$. If $k = 1$, $v_1' = c_1^{t_1}$. Now $u_2(t_0) > u_1(t_0) = c_1$, so $c_1^{t_1}$ will displace a segment of $u_2$ that begins to the left of $t_0$ with $c_1^{t_1}$, and so $u'_1 \subseteq u'_2$.  

For $k > 1$, perform the insertion of $c^i$ into $w$ in two steps, first inserting $c^{i_1}$, and then inserting $c^{i_1-t}$. If $(v''_1, v''_2) = \text{RINS}(u_1, c^{i_1})$, then $v''_1 = c^{i_1}_1$. Let $(v''_2, v''_2') = \text{RINS}(u_2, v''_2)$. By the $k = 1$ case, $u''_1 \lhd u''_2$.

Now $(c^{i_2} \cdots c^{i_k}, u'_1) = \text{RINS}(u_1, c^{i_1-t})$, and $u'_2$ is obtained by inserting $c^{i_2} \cdots c^{i_k}$ into $u''_2$. Therefore, by induction hypothesis, $u'_1 \lhd u'_2$. Repeating this argument with the remaining rows shows that $u'_1 \lhd u'_2 \lhd \cdots \lhd u'_f$, as required.

To see that shape$(w)$ interleaves shape$(\text{INSERT}(w, v))$, observe that $v'_1$ is a concatenation of segments of $u_1$. Write $v'_1 = xy$, where $x$ consists of segments that come from $u_{1,0}^{l(u_2)}$ and $y$ consists of segments that come from $u_{1,1,0}^{l(u_2)}$. Then, from the arguments of the first part of this proof, the segments in $x$ will all replace segments of $u_2$, so if $(v, u_2) = \text{RINS}(u_2, x)$, then $l(u_2) = l(u_2)$. Since $y$ is inserted into $\tilde{u}_2$ to obtain $u'_2$. Now $(v'_2, u'_2') = \text{RINS}(\tilde{u}_2, y)$, whence

$$l(u'_2) \leq l(\tilde{u}_2) + l(y) \leq l(u_2) + [l(u_1) - l(u_2)] = l(u_1).$$

The same argument shows that $l(u'_i) \leq l(u_i)$ for $i > 2$ as well, so that shape$(w)$ interleaves shape$(\text{INSERT}(w, v))$.

The assertion about weights is obvious.

**Definition 2.2.7 (Insertion Tableau of a Timed Word).** Let $w$ be a timed word with row decomposition $u_1 \cdots u_t$. The insertion tableau of $w$ is defined as:

$$P(w) = \text{INSERT}(\cdots \text{INSERT}(\text{INSERT}(u_1, u_2), u_3), \ldots, u_t).$$

**Example 2.2.8.** 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:

| $w$                          | $P(w)$                          |
|-----------------------------|---------------------------------|
| $3^{0.8}$                   | $3^{0.8}$                       |
| $3^{0.8}1^{0.5}4^{1.1}$     | $3^{0.8}1^{0.5}4^{1.1}1^{0.9}2^{1.6}3^{0.7}$ |
| $3^{0.8}1^{0.5}4^{1.1}1^{0.9}2^{1.6}3^{0.7}$ | $3^{0.8}1^{0.5}4^{1.1}1^{0.9}2^{1.6}3^{0.7}$ |
| $3^{0.8}1^{0.5}4^{1.1}1^{0.9}2^{1.6}3^{0.7}1^{0.7}2^{0.2}$ | $3^{0.8}1^{0.5}4^{1.1}1^{0.9}2^{1.6}3^{0.7}1^{0.7}2^{0.2}$ |

**Definition 2.2.9 (Schützenberger Involution on Timed Words).** Given $w = c^{i_1} \cdots c^{i_k} \in A^+_n$, define

$$w^d = (n - c_k + 1)^{i_k} \cdots (n - c_1 + 1)^{i_1},$$

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

**Lemma 2.2.10.** Let $u$ and $v$ be timed rows. Suppose $\text{RINS}(u, v) = (v', u')$, and $l(v') = l(v)$. Then $\text{RINS}(u^d, v^d) = (v^d, u^d)$.

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

$$u' = u_{t_0}^{l(u)}c^i u_{l(u) + t_0}^{l(u')},$$

and $v' = u_{t_0}^{l(u)}c^i u_{l(u) + t_0 + t_0}'$. Using induction as in the proof of Theorem 2.2.6, we may assume that $v'$ is constant, so $v' = d^i$ for some $d > c$. 

\[\square\]
Now
\[ u'^t = u[0,t_{o+\tau_{l(u)}}](n-c+1)^tu[0,t_{o}) \quad \text{and} \quad v'^t = (n-d+1)^t. \]

Since all the values of \( u[t_{o+\tau_{l(u)}}] \) are greater than or equal to the last value of \( d \), all the values of \( u[0,t_{o+\tau_{l(u)}}] \) 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 \( \text{RINS}(u'^t, v'^t) = (v^t, u^t) \). \[\Box\]

**Corollary 2.2.11.** The timed row insertion algorithm gives rise to a bijection:

\[ \text{RINS}: \mathbb{R}^t_n(r) \times \mathbb{R}^t_n(s) \rightarrow \{(v', u') \in \mathbb{R}^t_n(r + s - r') \times \mathbb{R}^t_n(r') \mid r' \geq \max(r, s), \ u' \prec v'\}. \]

**Proof.** Suppose \((u, v) \in \mathbb{R}^t_n(r) \times \mathbb{R}^t_n(s)\), and \((v', u') = \text{RINS}(u, v)\). Then \((u, v)\) can be recovered from \((v', u')\) (given the prior knowledge of \( r \) and \( s \)) as follows: let \((v_1', u_1') = \text{RINS}(u[c_{0,r}], v[c_{0,r}])\). Then using Lemma 2.2.10, \( u \) and \( v \) can be recovered as \( u = u_1 \), and \( v = v_1 u'[v,r,s] \). \[\Box\]

**Theorem 2.2.12 (Timed Pieri Rule).** The timed insertion algorithm gives rise to a bijection:

\[ \text{INSERT}: \mathbb{Tab}^t_n(\lambda) \times \mathbb{R}^t_n(r) \rightarrow \bigcup_{\lambda \text{ interolves } \mu \atop \mu \in \mathbb{Ring}_n} \mathbb{Tab}^t_n(\mu). \]

**Proof.** Let \( \lambda = (\lambda_1, \ldots, \lambda_l) \). Let \( w \in \mathbb{Tab}^t_n(\lambda) \) have row decomposition \( u_1 \cdots u_1 \), and \( x \in \mathbb{R}^t_n(r) \). Suppose that \( w' = \text{INSERT}(w, x) \) has row decomposition \( u_1' \cdots u_1' \) (with the possibility that \( u_{l+1} = \emptyset \)). We already know that \( \text{shape}(w) \) interolves \( \text{shape}(w) \) (Theorem 2.2.6). Given timed rows \( u' \) and \( v' \) such that \( u' \prec v' \), and non-negative real numbers \( r \) and \( s \) such that \( r \leq l(u') \), let \( \text{RINS}^{-1}_n(u', v') \) denote the unique pair of rows \((u, v)\) such that \( l(u) = r \), \( l(v) = s \), and \( (v', u') = \text{RINS}(u, v) \) (see Corollary 2.2.11). Then the rows of \( w \) can be recovered from \( w' \) as follows:

\[
(x_l, u_l) = \text{RINS}^{-1}_n(u_{l+1}', u_l'),
(x_{l-1}, u_{l-1}) = \text{RINS}^{-1}_n(x_l, u_{l-1}'),
(x_{l-2}, u_{l-2}) = \text{RINS}^{-1}_n(x_{l-1}, u_{l-2}'),
\]

and finally \( x \) can be recovered as \( x = x_1 \). \[\Box\]

**Definition 2.2.13 (Deletion).** Let \( w' \in \mathbb{Tab}^t_n(\mu) \) and let \( \lambda \) be a real partition that interolves \( \mu \). Then we write

\[ \text{DELETE}_n(w') = (v, w) \quad \text{if and only if} \quad w \in \mathbb{Tab}^t_n(\lambda) \quad \text{and} \quad \text{INSERT}(w, v) = w'. \]

The pair \((v, w)\) is computed from \( w' \) and \( \lambda \) by the algorithm described in the proof of Theorem 2.2.12.
3. Greene’s Theorem

3.1. Greene’s Invariants for Timed Words.

Definition 3.1.1 (Greene’s Invariants for Timed Words). Given \( w \in A_n^\dagger \), its \( k \)th Greene’s 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:

\[
a_k(w) = \max \{ l(u_1) + \cdots + l(u_k) \mid u_1, \ldots, u_k \text{ are pairwise disjoint subwords, and each } u_i \text{ is a timed row} \}
\]

Lemma 3.1.2. If \( w \) is a timed tableau of shape \( \lambda = (\lambda_1, \ldots, \lambda_l) \), 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]. Indeed, \( u_1, \ldots, u_k \) are pairwise disjoint subwords that are rows, so

\[
a_k(w) \geq \lambda_1 + \cdots + \lambda_l.
\]

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 \( \lambda_1 + \cdots + \lambda_k \). □

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:

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

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

Definition 3.2.2 (Timed Plactic Monoid). The timed plactic monoid \( \text{pl}(A_n) \) is the quotient \( A_n^\dagger \equiv \), where \( \equiv \) is the congruence generated by the timed Knuth relations \( \{\kappa_1\} \) and \( \{\kappa_2\} \).

In other words, two elements of \( A_n^\dagger \) are said to differ by a Knuth move if they are of the form \( v_1 w \) and \( v_2 w \), where \( v_1 \) and \( v_2 \) are terms on opposite sides of one of the timed Knuth relations \( \{\kappa_1\} \) and \( \{\kappa_2\} \). Knuth equivalence \( \equiv \) is the equivalence relation generated by Knuth moves. 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 \( \text{pl}(A) \) of Knuth equivalence classes, giving it the structure of a monoid.

Lemma 3.2.3. Then \( v \) and \( w \) differ by a Knuth move \( \{\kappa_1\} \) if and only if \( v^\dagger \) and \( w^\dagger \) (see Definition 2.2.9) differ by a Knuth move \( \{\kappa_2\} \).

Proof. When the involution \( w \mapsto w^\dagger \) is applied to the Knuth relation \( \{\kappa_1\} \), the Knuth relation \( \{\kappa_2\} \) is obtained. □

Lemma 3.2.4. Every timed word is Knuth equivalent to its timed insertion tableau.
PROOF. It suffices to show that, for every timed row \( u \), and every term \( c' \), if \( (v, u') = \text{RINS}(u, c') \), then \( uc' \equiv vu' \). If \( u(t) \leq c \) for all \( 0 \leq t < l(u) \), there is nothing to show. Otherwise, a segment \( v \) of \( u \), beginning at \( t_0 \), and of length \( t_1 = \min(l(l(u) - t_0), t) \) is displaced by the segment \( c_{t_1} \) of \( c' \). Write \( u = u'vu'' \). It suffices to show \( u'vuc_{t_1}u'' \equiv vu'c_{t_1}u'' \). But this can be done in two Knuth moves as follows (the segment to which the Knuth move is applied is underlined):

\[
\frac{u'vuc_{t_1}u''}{\equiv \kappa_2} \frac{u'vuc_{t_1}u''}{=} \frac{u'vuc_{t_1}u''}{=} \frac{u'vuc_{t_1}u''}{\equiv \kappa}, \frac{vu'c_{t_1}u''}{},
\]

completing the proof of the lemma. \( \square \)

### 3.3. Knuth Equivalence and Greene’s 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 move then they have the same Greene invariants. For the Knuth move \( \{c_1\} \), 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’s invariants coincide for \( wzxyu \) and \( wxzyu \). Now suppose that \( v_1, \ldots, v_k \) are pairwise disjoint row subwords of \( wzxyu \) whose lengths add up to \( a_k(wzxyu) \). We may 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 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 \( y_i \neq \emptyset \) and \( z_i \neq \emptyset \) simultaneously for the same \( i \). Renumber \( v_1, \ldots, v_k \) in such a way that \( v_i = w_i x_i z_i u_i \) for \( i = 1, \ldots, r \) and \( v_i = w_i x_i y_i u_i \) for \( i = r + 1, \ldots, k \). If \( x_i = \emptyset \) for \( i = 1, \ldots, r \), then all the \( v_i \)’s remain row subwords of \( wzxyu \), and so \( a_k(wzxyu) \geq a_k(wxzyu) \). If \( y_i = \emptyset \) for \( i = r + 1, \ldots, k \), then set

\[
\begin{align*}
v'_1 &= w_1 x_1 y_1 u_1, \\
v'_i &= w_i x_i u_i & \text{for } i = 2, \ldots, k.
\end{align*}
\]

Then \( v'_i \) are pairwise disjoint row subwords of \( wzzyu \), and \( l(v'_1) + \cdots + l(v'_k) \geq l(v_1) + \cdots + l(v_k) \), since \( l(y) = l(z) \geq l(z_1) + \cdots + l(z_r). \) It follows that \( a_k(wzzyu) \geq a_k(wxzyu) \).

If at least one \( x_i \neq \emptyset \) and one \( y_i \neq \emptyset \), assume that \( x_1 \) has the least first letter among \( x_1, \ldots, x_r \), and that \( y_k \) has the largest last letter among \( y_{r+1}, \ldots, y_k \). Let \( x_0 \) be the row subword of \( x \) obtained by concatenating the segments of \( x_1, \ldots, x_r \) in the order in which they appear in \( x \) (so \( x_0 \) is a row of length \( l(x_1) + \cdots + l(x_r) \)). Let \( y_0 \) be the row subword of \( y \) obtained by concatenating the segments of \( y_{r+1}, \ldots, y_k \) in the order in which they appear in \( y \) (so \( y_0 \) is a row of length \( l(y_{r+1}) + \cdots + l(y_k) \)). Define

\[
\begin{align*}
v'_1 &= w_1 x_0 y_0 u_k, \\
v'_i &= w_i z_i u_i & \text{for } i = 2, \ldots, r, \\
v'_i &= w_i x_i u_i & \text{for } i = r + 1, \ldots, k, \\
v'_k &= w_k z_1 u_1.
\end{align*}
\]

Then \( v'_1, \ldots, v'_k \) are pairwise disjoint row subwords of \( wzzyu \) with total length \( l(v'_1) + \cdots + l(v'_k) \), so \( a_k(wzzyu) \geq a_k(wxzyu) \). The reverse inequality \( a_k(wzzyu) \geq a_k(wxzyu) \) is obvious, since every row subword of \( wzzyu \) is also a row subword of \( wzzyu \). It follows that \( a_k(wzzyu) = a_l(wxzyu) \) for all \( k \).
For the Knuth move \[[k_2]\] a similar argument can be given, however, a more elegant method is to use Lemma 3.2.3 noting that \(a_k(w) = a_k(w^t)\) for all \(k \geq 1\) and all \(w \in A_n^\dagger\).

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 \(\lambda = (\lambda_1, \ldots, \lambda_l)\), then
\[
a_k(w) = \lambda_1 + \cdots + \lambda_k \text{ for } k = 1, \ldots, l.
\]

**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.

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 \(A_{n-1}^\dagger\).

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

**Proof.** Suppose \(w\) has row decomposition \(u_1u_2\ldots u_n\). Since \(n\) is the largest element of \(A_n^\dagger\), we may write \(u_i = u_i^t n^{t_i}\) for some \(t_i \geq 0\). Clearly \(l(u_i) \geq l(u_i^t)\). Since \(w\) is semistandard, \(l(u_i^t) \geq l(u_{i+1})\) for \(i = 1, \ldots, l-1\). It follows that the shape of \(w^t\), which is \((l(u_1^t), \ldots, l(u_l^t))\) interleaves the shape of \(w\), which is \((l(u_1), \ldots, l(u_l))\). Since \(u_i \preceq u_{i+1}\), it follows that \(u_i^t \preceq u_{i+1}\) for \(i = 1, \ldots, l-1\).

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

**Proof.** Applying the restriction to \(A_{n-1}^\dagger\) map \(w \rightarrow \bar{w}\) to both sides of the Knuth relation \([k_2]\) gives: \(x\bar{z}y\) and \(\bar{x}\bar{z}y\). Write \(y = y'y'\), where \(l(y') = l(\bar{z})\), we have
\[
x\bar{z}y'y'' \equiv \bar{x}\bar{z}y'y''
\]
a Knuth relation in in \(A_{n-1}^\dagger\). A similar argument works for the Knuth relation \([k_2]\).

**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 \(A_1^\dagger\). 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 \(\lambda = (\lambda_1, \ldots, \lambda_l)\) be the shape of this timed tableau. By Lemma 3.3 \(v\) and \(w\) have the same Greene invariants, and therefore the same shape \(\mu = (\mu_1, \ldots, \mu_{l+1})\). It follows that both \(v\) and \(w\) are obtained from \(\bar{v} = \bar{w}\) by appending \(n^{\mu_i-\lambda_i}\) to the \(i\)th 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 Thm. 2.15]). The same characterization works for timed Knuth equivalence.

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

**Proof.** If \( w \) and \( w' \) are Knuth equivalent, then so are \( uvw \) and \( uww'v \). By the timed version of Greene’s theorem (Theorem 3.4.1, they do not have the same shape, then by Theorem 3.4.1 they do not have the same Greene invariants, so taking \( u = v = \emptyset \) proves the result.

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

\[
a_1(x^t y c^T) = l(x) + t + T, \quad \text{while} \quad a_1(x^{c'} y c^T) = l(x) + T,
\]

thereby proving the result.

In the general case, suppose \( w = u_1 u_{i-1} \cdots u_1 u_1 \) and \( w' = u_1' u_{i-1} \cdots u_1' u_1' \) 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 \( (v, x) = \text{RINS}(u_i, c^T) \) and \((v', x') = \text{RINS}(u'_i, c^T)\), then \( l(x) \neq l(x') \). Also, note that \( c \) is at least \( i \), the least possible value of the \( i \)th row of a tableau.

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

\[
P(uvw) = u_{i-1} \cdots u_{i+1} t^T u'_i (i-1)^T u_{i-1} \cdots 1^T u_1, \quad P(uww') = u_{i-1} \cdots u_{i+1} t^T u'_i (i-1)^T u_{i-1} \cdots 1^T u_1.
\]

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

\[
(v, z) = \text{DELETE}_{(T+\lambda_1, \ldots, T+\lambda_{i-1})}(z).
\]

Then when \( P(uvw) \) and \( P(uww') \) 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 \( i \)th rows of \( P(uvw) \) and \( P(uww') \), which, by our earlier argument, will have different lengths. \( \square \)

5. The Real RSK Correspondence

5.1. Definition using Timed Insertion Tableaux. Let \( M_{m \times n}(\mathbb{R}^+) \) denote the set of all \( m \times n \) matrices with non-negative real entries. Given \( A = (a_{ij}) \in M_{m \times n}(\mathbb{R}^+) \), define its timed column word \( u_A \), and timed row word \( v_A \) as follows:

\[
u_A = 1^{a_{11}} 2^{a_{12}} \cdots n^{a_{1n}} 1^{a_{21}} 2^{a_{22}} \cdots n^{a_{2n}} \cdots 1^{a_{m1}} 2^{a_{m2}} \cdots n^{a_{mn}}.
\]
\[
v_A = 1^{a_{11}} 2^{a_{21}} \cdots m^{a_{1m}} 1^{a_{12}} 2^{a_{22}} \cdots m^{a_{2m}} \cdots 1^{a_{1n}} 2^{a_{2n}} \cdots m^{a_{nm}}.
\]
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:

$$\text{RSK}(A) = (P(u_A), P(v_A)).$$

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

$$P = 3^{0.32} 4^{0.29} 5^{0.68} 4^{0.55} 1^{0.77} 2^{0.67} 3^{0.52} 4^{0.75},$$
$$Q = 3^{0.61} 2^{1.38} 3^{0.43} 1^{1.57} 2^{0.45} 3^{0.70},$$

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 $\text{RSK}$ defines a bijection:

$$\text{RSK} : M_{m \times n}(\mathbb{R}^+) \rightarrow \bigsqcup_\lambda \text{Tab}_n^f(\lambda) \times \text{Tab}_m^f(\lambda),$$

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

**Remark 5.1.3.** Let $\mu_i$ denote the sum of the $i$th row of $A$, and $\nu_j$ the sum of the $j$th column. Let $\mu = (\mu_1, \ldots, \mu_m)$, and $\nu = (\nu_1, \ldots, \nu_n)$. Then, if $\text{RSK}(A) = (P, Q)$ then $\text{wt}(P) = \nu$, and $\text{wt}(Q) = \mu$.

**Remark 5.1.4 (Relation to Knuth’s definition).** Knuth [8] defined $\text{RSK}(A) = (P, Q)$ 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 $\text{RSK}(A^T) = (Q, P)$ if $\text{RSK}(A) = (P, Q)$ 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 $\text{RSK}(A) = (P, Q)$, then $\text{RSK}(A^T) = (Q, P)$ 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 \( A \in M_{m \times n}(\mathbb{R}^+) \), 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
\[
w = b_{11} b_{12} \ldots b_{kn},
\]
where \( 0 \leq b_{ij} \leq a_{ij} \) for all \( (i,j) \). If \( w \) is a row, then the indices \( (i_1,j_1), \ldots, (i_k,j_k) \) for which \( b_{ij} > 0 \), when taken in the order in which they appear in \( w \), must satisfy \( i_1 \leq \cdots \leq i_k \), and \( j_1 \leq \cdots \leq j_k \). 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 \( k \)th timed Greene’s 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 \( C \subseteq P_{mn} \) which can be written as a union of \( k \) chains. Since the order relation on \( P_{mn} \) corresponds to the order relation on \( P_{mn} \) 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. \( \square \)

### 5.2. Insertion-Recording Algorithm for \( \text{RSK}(A) \)

Given real partitions \( \lambda \) and \( \mu \) such that \( \lambda \) interlaces \( m \), and \( w \in \text{Tab}^i_{m-1}(\lambda) \), define the inflation of \( w \) to shape \( \mu \) by \( m \) to be the unique tableau \( \text{INFL}_\mu(w, m) \) of shape \( \mu \) whose restriction to \( m-1 \) is equal to \( w \). In the notation of Section 4.1, \( \text{INFL}_\mu(w, m) = w \).

Given \( A \in M_m(\mathbb{R}^+) \), let \( r_{i,A} = 1^{a_{i1}} 2^{a_{i2}} \ldots n^{a_{in}} \). Then \( u_A = r_{1,A} r_{2,A} \ldots r_{m,A} \).

**Insertion-Recording Algorithm**

- \( P \leftarrow \emptyset, Q \leftarrow \emptyset \).
- For \( i = 1, \ldots, m \), repeat the following steps:
  - \( P \leftarrow \text{INSERT}(P, r_{i,A}) \).
  - \( \lambda \leftarrow \text{shape}(P) \).
  - \( Q \leftarrow \text{INFL}_\lambda(Q, i) \).
- Return \( (P, Q) \).

**Lemma 5.2.1.** For every \( A \in M_{m \times n}(\mathbb{R}^+) \), the output of the insertion-recording algorithm is \( \text{RSK}(A) \) 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 \( P(u_A) = \text{INSERT}(P(u_{A'}), r_{m,A}) \). Also, the restriction \( v_A \) 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 \( P(v_A) = \text{INFL}_\lambda(P(v_{A'}), m) \), which is the output of the insertion-recording algorithm. \( \square \)
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 \( \mu \) is the shape of \( P(v_A) \), then \( (r_{m,A}, P(u_A)) = \text{DELETE}_\mu(P(u_A)) \) (see Definition 2.2.13).

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 \( A \in M_{m \times n}(\mathbb{R}^+) \), consider the set

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

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

\[
L(A) = (i_1, j_1), \ldots, (i_r, j_r)
\]

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

- \( P \leftarrow \emptyset, Q \leftarrow \emptyset \).
- While \( A \) is non-zero repeat the following steps:
  - Set \( S \leftarrow 0_{m \times n} \) (\( m \times n \) zero matrix)
  - Set \( p = \emptyset, q = \emptyset \).
  - While \( A \) is non-zero repeat the following steps:
    - Compute \( L(A) = (i_1, j_1), \ldots, (i_r, j_r) \) of \( A \)
    - Let \( m(A) = \min\{a_{i_1,j_1}, \ldots, a_{i_r,j_r}\} \)
    - Set \( a_{i_s,j_s} \leftarrow a_{i_s,j_s} - m(A) \) for \( s = 1, \ldots, r \)
    - Set \( s_{i_{s+1},j_s} \leftarrow s_{i_{s+1},j_s} + m \) for \( s = 1, \ldots, r - 1 \)
    - Set \( p = p_j^{m}, q = q_i^{m} \)
  - \( P \leftarrow pP, Q \leftarrow qQ \)
  - \( A \leftarrow S \)
- Return \((P, Q)\)

Theorem 5.3.2. When the light-and-shadows real RSK algorithm is applied to \( A \in M_{m \times n}(\mathbb{R}^+) \), it return 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
\begin{itemize}
  \item $P \leftarrow \emptyset$, $Q \leftarrow \emptyset$
  \item While $A$ is non-zero repeat the following steps:
    \begin{itemize}
      \item Set $p \leftarrow \emptyset$, $q \leftarrow \emptyset$
      \item For $i = 1, \ldots, m$, repeat the following steps:
        \begin{itemize}
          \item Set $S \leftarrow 0_{m \times n}$.
          \item Set $(v, u) = \text{RINS}(p, 1^{a_1}, \ldots, n^{a_n})$.
          \item If $v = 1^{a_1}, \ldots, n^{a_n}$, set $s_{ij} \leftarrow s_j$ for $j = 1, \ldots, n$.
        \end{itemize}
      \item Set $A \leftarrow S$.
      \item $P \leftarrow pP$, $Q \leftarrow qQ$.
    \end{itemize}
  \item return $(P, Q)$
\end{itemize}

The main loop of this algorithm starts with a matrix $A$, and replaces 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 $(A'_0, A'_1, \ldots, A'_h)$, 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 $P = \text{INSERT}(P', u_{A''})$. Here $u_{A''}$ is just the row:

$$1^{a_1}, \ldots, n^{a_n}.$$

The inner loop of the light-and-shadows real RSK algorithm produces a sequence $A' = A'_1, A'_2, \ldots, A'_h$ of matrices as it runs on input $A'$. Let $L'_k = L(A'_k)$ and $m'_k = m(A'_k)$, for $k = 1, \ldots, h$. When the inner loop finishes running, we have $p' = j'_1 m'_1, j'_2 m'_2, \ldots, j'_h m'_h$, and $q' = j'_1 m'_1, j'_2 m'_2, \ldots, j'_h m'_h$, 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 $(a_{m_1}, \ldots, a_{m_n})$. To begin with, assume that this row has only one non-zero entry, $a_{m_0}$. Let $L_1, L_2, \ldots$, and $m_1, m_2, \ldots$ be the corresponding sequences of leading If $j_0 \geq j_i$ for all $i$, then $L_k = L'_k$ for all $k = 1, \ldots, h$. In addition, $A$ has a singleton sequence of leading points $(j_0, m_0)$. As a result, the output of the main loop is $p = p'u_{A''}$, and $q = q'm_{a_{m_0}}$, 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 $L_1, \ldots, L_{l-1}$ of $A$ are the same as the sequences $L'_1, \ldots, L'_{l-1}$. If $a_{m_0} \geq m'_l + \ldots + m'_{j_l}$, then $L_k = \{(m, j_0)\} \cup L'_k$ for $k = 1, \ldots, h$. Therefore $p = j'_1 m'_1, \ldots, j'_1 m'_1, j'_1 m'_1, j'_0, \ldots$. From the definition of timed row insertion (Definition 2.2.1), RINS($p', j'_0, m'_1, \ldots, n^{a_n}$) = $(j'_1 m'_1, \ldots, j'_1 m'_1, p)$. Also, $q = q'm_{a_{m_0}}$. Finally, $S$ is obtained from $S'$ by adding $m'_k$ to the $(m, j_k)$th entry of $S'$ for each $k = l, \ldots, h$. 


Otherwise, \( m_j + \cdots + m_{\lambda_1-1} < a_{\lambda_1} \leq m_j + \cdots + m_q \) for some \( l \leq q < h \). In this case the sequences of leading points for \( A \) are given by:

\[
L_k = \begin{cases} 
L_k' & \text{for } 1 \leq k < l - 1, \\
(m, j_0) \cup L_k' & \text{for } l \leq k \leq q, \\
L_{k-1} & \text{for } q < l \leq h,
\end{cases}
\]

\[
p = j_1^{m_1'} \cdots j_{l-1}^{m_l-1} j_0^{a_{m_0} - a_{\lambda_1} - m_1' - \cdots - m_q'}.\]

Again from the definition of timed row insertion, \( \text{RINS}(p', a_{m_0}) = (j_1^{m_1'} \cdots j_{l-1}^{m_l-1} j_0^{a_{m_0} - (m_1' + \cdots + m_q')}, p) \). Also, \( q = q' a_{m_0} \).

Finally, the value \( m_k' \) is added to the \((m, j_k)\)th entry of \( S' \) for \( k = l, \ldots, q - 1 \), and \( a_{m_0} - (m_1' + \cdots + m_{q-1}') \) 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.

\[\square\]

5.4. Piecewise Linear RSK.

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

\[
\begin{array}{cccc}
\lambda^{(1)}_1 & \lambda^{(2)}_1 & \cdots & \lambda^{(n)}_1 \\
\lambda^{(1)}_2 & \lambda^{(2)}_2 & \cdots & \lambda^{(n)}_2 \\
\vdots & \vdots & \ddots & \vdots \\
\lambda^{(1)}_n & \lambda^{(2)}_n & \cdots & \lambda^{(n)}_n
\end{array}
\]

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

Whenever \( n \geq k \), define \( r^n_k : A^n_k \to A^\dagger_k \) by taking \( r^n_k(w) \) to be the timed word whose exponential string is obtained from the exponential string of \( w \) by deleting all terms of the form \( c^x \) where \( c > k \). It is easy to see that if \( w \in A^n_k \) is a timed tableau, then so is \( r^n_k(w) \) for all \( k = 1, \ldots, n \).

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

Given a timed tableau \( w \) in \( A_n \), define partitions \( \lambda^{(k)} = (\lambda^{(k)}_1, \ldots, \lambda^{(k)}_k) \) by \( \lambda^{(k)} = \text{shape}(r^n_k(w)) \). By Lemma 4.1.1, \( \lambda^{(k-1)} \) interleaves \( \lambda^{(k)} \) for all \( k = 2, \ldots, n \). Therefore the numbers \( \lambda_i^{(k)} \) form a Gelfand-Tsetlin pattern, which we denote by \( \text{GT}(w) \). The shape of \( \lambda \) is also the shape of \( \text{GT}(w) \). Conversely, given a Gelfand-Tsetlin pattern \( T \), it is easy to reconstruct the unique timed tableau \( w \) such that \( T = \text{GT}(w) \).

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 fairly lemma, which is well-know for the RSK correspondence on integer matrices (see e.g., [6 Prop. 2.26]).
Lemma 5.4.2. Given $A \in M_{m \times n}(\mathbb{R}^+)$ with RSK$(A) = (P, Q)$, let $(\lambda^{(k)})$ = GT$(P)$ and $(\mu^{(k)})$ = GT$(Q)$, Gelfand-Tsetlin patterns of size $n$ and $m$ respectively. Then

\begin{align}
\lambda_1^{(i)} + \cdots + \lambda_k^{(i)} &= \max_{C \subseteq P_{m \times j} \text{ union of at most } k \text{ chains}} \sum_{(i,j) \in C} a_{ij}, \\
\mu_1^{(i)} + \cdots + \mu_k^{(i)} &= \max_{C \subseteq P_{m \times n} \text{ union of at most } k \text{ chains}} \sum_{(i,j) \in C} a_{ij},
\end{align}

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 $u_{A_j} = r_j^P(u_A)$. By a repeated application of Lemma 4.1.2, $P(u_{A_j}) = r_j^P(P(u_A))$. But the $j$ row of GT$(P)$ is, by definition, the shape of $r_j^P(P(u_A))$. 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 $M_{m \times n}(\mathbb{R}^+)$ onto the cone of pairs of Gelfand-Tsetlin patterns $((\lambda^{(k)}), (\mu^{(k)}))$ 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.

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 $gl_n(\mathbb{C})$ 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. *Representation 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