K-KNUTH EQUIVALENCE FOR INCREASING TABLEAUX

CHRISTIAN GAETZ, MICHELLE MASTRIANNI, REBECCA PATRIAS, HAILEE PECK, COLLEEN ROBICHAUX, DAVID SCHWEIN, AND KA YU TAM

Abstract. A K-theoretic analogue of RSK insertion and Knuth equivalence relations was first introduced in 2006 [1]. The resulting K-Knuth equivalence relations on words and increasing tableaux on \([n]\) has prompted investigation into the equivalence classes of tableaux arising from these relations. Of particular interest are the tableaux that are unique in their class, which we refer to as unique rectification targets (URTs). In this paper we give several new families of URTs and a bound on the length of intermediate words connecting two K-Knuth equivalent words. In addition, we describe an algorithm to determine if two words are K-Knuth equivalent and to compute all K-Knuth equivalence classes of tableaux on \([n]\).

1. Introduction

In 2006, Buch et al. introduced a new combinatorial algorithm called Hecke insertion, used to insert a word into an increasing tableau [1]. The algorithm is a K-theoretic analogue of the well-known Schensted algorithm for the insertion of a word into a semistandard Young tableau.

If two words insert into the same tableau via Schensted’s insertion algorithm, they are said to be Knuth equivalent and can be connected via the Knuth equivalence relations. Knuth equivalence has a K-theoretic analogue referred to as K-Knuth equivalence, also introduced in [1]. An important difference between Knuth equivalence and K-Knuth equivalence is that, while insertion equivalence via the Schensted algorithm (resp. the Hecke algorithm) implies Knuth equivalence (resp. K-Knuth equivalence), the converse holds for the standard version but not for the K-theoretic version. Hence two words can be K-Knuth equivalent but insert into different tableaux via the Hecke insertion algorithm.

A K-Knuth equivalence class typically contains words from different insertion classes. There are some K-Knuth classes, however, for which all words in the class insert into the same tableau. A class with this property is called a unique rectification class, and its corresponding insertion tableau is a unique rectification target (URT). In both [1] and [6], Hecke insertion and K-Knuth equivalence were used to rederive a K-theoretic version of the Littlewood-Richardson rule for the cohomology rings of Grassmanians. In order to get a working version of this rule, non-URTs needed to be avoided. Hence Patrias and Pylyavskyy [6] posed the following natural question, an open problem.

Problem 1. Characterize all URTs or at least provide an efficient algorithm to determine if a given tableau is a URT.

Date: September 19, 2014.
This paper makes partial progress towards answering Problem 1. In more detail, we will extend previous results, many from a paper of Buch and Samuel [2], about URTs and K-Knuth equivalence. In Section 2 of the paper, we provide more background on Hecke insertion and K-Knuth equivalence and discuss the K-theoretic extension of the jeu de taquin algorithm of Thomas and Yong, which provides another way of determining whether two tableaux are in the same equivalence class. We also summarize Buch and Samuel’s results about URTs and give invariants for classes of K-Knuth equivalent tableaux. In Section 3, we give an algorithm to compute all K-Knuth classes of tableaux on a given alphabet \([n]\) and a run time for that algorithm. From this we derive a finite-time algorithm to determine if two given words are K-Knuth equivalent. Section 4 shows that every two K-Knuth equivalent tableaux on \([n]\) can be connected by intermediate words of length at most \(\frac{1}{2}n(n+1)(n+2) + 3\). The proof of this bound also includes a useful lemma stating that any words of length \(\ell\) in an insertion class can be connected to the row word of the corresponding insertion tableau by moving through intermediate words of length at most \(\ell\). In Sections 5 and 6, we give new families of URTs. We introduce a new technical invariant for K-Knuth classes called T-compatibility and use it to prove that all right-alignable tableaux are URTs. We also give a method for easily determining whether any given hook-shaped tableau is a URT. Finally, in Section 7, we discuss various findings on the number of K-Knuth equivalence classes of tableaux on an alphabet \([n]\) and the number of unique rectification classes among them. We then give additional conjectures and related results.

2. Background

The goal of this section is to acquaint the reader with the language of K-Knuth equivalence relations on increasing tableaux, which for the most part parallels the better-known Knuth equivalence relations [5].

2.1. Increasing Tableaux. In this section we will define in more detail increasing tableaux, the main subject of this paper, as well as related terminology, following the formalization of [2, Section 3.1]. Throughout this paper, \(\mathbb{N}\) will denote the set of positive integers.

Elements of the set \(\mathbb{N} \times \mathbb{N}\) are called boxes, and will form the building blocks of increasing tableaux. We will visualize \(\mathbb{N} \times \mathbb{N}\) as an infinite matrix comprised of boxes: the box \((i,j)\) appears in row \(i\) and column \(j\).

Suppose \(\alpha = (i_1,j_1)\) and \(\beta = (i_2,j_2)\) are boxes. We say that \(\alpha\) is strictly north of \(\beta\) if \(i_1 < i_2\) and weakly north of \(\beta\) if \(i_1 \leq i_2\); we say that \(\alpha\) is strictly northeast of \(\beta\) if \(i_1 < i_2\) and \(j_1 > j_2\), and we say that \(\alpha\) is weakly northeast of \(\beta\) if \(i_1 \leq i_2\) and \(j_1 \geq j_2\). The reader can formulate the analogous definitions for the remaining cardinal directions, which we omit. In addition, we say \(\alpha\) is above \(\beta\) to mean \(\alpha\) is north of \(\beta\), we say \(\alpha\) is directly above \(\beta\) to mean \(i_1 = i_2 - 1\) and \(j_1 = j_2\), and so on.

A shape \(\lambda\) is any finite subset of \(\mathbb{N} \times \mathbb{N}\). We say \(\lambda\) is a straight shape if whenever \(\lambda\) contains the box \(\alpha\) it contains all boxes weakly northwest of \(\alpha\). A skew shape \(\nu/\mu\) is the set difference of two straight shapes \(\nu \supseteq \mu\).
Example 2.1. Of the shapes below, the first is neither straight nor skew, the second is skew but not straight, and the third is straight.

We can identify a straight shape with a partition as follows. Given a straight shape \( \lambda \), let \( \lambda_i \) denote the number of boxes in row \( i \). If \( \lambda \) has \( \ell \) nonempty rows then \( \lambda \) is uniquely determined by the tuple \((\lambda_1, \lambda_2, \cdots, \lambda_\ell)\). By definition, \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_\ell \). The straight shape given in Example 2.1, for instance, corresponds to the partition \((4, 4)\).

A filling of a shape \( \lambda \) is any map \( T : \lambda \to \mathbb{N} \), which assigns an integer to each box of \( \lambda \). The image of a box \( \alpha \) under \( T \) is called the label of \( \alpha \). We say that the filling \( T \) is an increasing tableau (of shape \( \lambda \)) if the entries of \( T \) strictly increase down columns and from left to right along rows, that is, if \( T(\alpha) < T(\beta) \) whenever \( \alpha \) is weakly northwest of \( \beta \). In this paper all tableaux are increasing tableaux, and in particular we will not consider semistandard tableaux. A tableau \( T \) of shape \( \lambda \) is straight if \( \lambda \) is straight and skew if \( \lambda \) is skew. Unless otherwise mentioned, we will write “tableau” to mean “straight tableau.”

Example 2.2. Of the fillings below, only the third is an increasing tableau.

As with matrices, let \( \lambda^t \) denote the transpose of \( \lambda \), defined by \( \lambda^t = \{(j, i) : (i, j) \in \nu\} \).

Let \( T^t : \lambda^t \to \mathbb{N} \) denote the transpose of \( T \), defined by \( T^t(j, i) = T(i, j) \). The transpose of a tableau or shape is sometimes referred to as its conjugate.

Example 2.3. The tableau

\[
\begin{array}{cccc}
1 & 2 & 3 & 5 \\
3 & 4 & 5 & 6 \\
7 & 7 & & \\
\end{array}
\]

has transpose

\[
\begin{array}{ccc}
1 & 3 & 6 \\
2 & 4 & 7 \\
3 & 5 & \\
6 & 7 & \\
\end{array}
\]

Definition 2.4. A tableau \( T \) is initial if the set of labels of \( T \) is \([n]\) for some \( n \in \mathbb{N} \). A word \( w \) is initial if the set of letters appearing in \( w \) is \([n]\) for some \( n \in \mathbb{N} \).

Example 2.5. The word 124335 is initial but the word 14355 is not. Of the two tableaux below, the lefthand tableau is initial and the righthand tableau is not.
Initial tableaux are often easier to work with, and for this reason we will usually restrict our attention to initial tableaux. This restriction comes at no loss of generality, as the following definition will make clear.

**Definition 2.6.** Let \( w = w_1 w_2 \ldots w_k \) be a word and let \( a_1 < a_2 < \cdots < a_\ell \) be the ordered list of letters appearing in \( w \). The *standardization* of \( w \) is the word formed by replacing \( a_i \) with \( i \) in \( w \).

Similarly, let \( T \) be a tableau and let \( a_1 < a_2 < \cdots < a_\ell \) be the ordered list of letters appearing in \( T \). The standardization of \( T \) is the tableau formed from \( T \) by replacing every entry \( a_i \) with \( i \).

**Example 2.7.** The standardization of the word 35822 is 23411. The standardization of the tableau
\[
\begin{array}{ccc}
6 & 5 & 1 \\
3 & 2 & 4 \\
7 & & \\
\end{array}
\]
is
\[
\begin{array}{ccc}
1 & 2 & 3 \\
2 & 3 & 4 \\
6 & 7 & \\
\end{array}
\]

2.2. **Hecke Insertion.** Hecke insertion is an algorithm for inserting a positive integer into an increasing tableau, resulting in another increasing tableau, which may or may not be the same as the original. The elementary step of the Hecke insertion algorithm is the insertion of a positive integer into a row of the tableau. After the row is modified, either a new positive integer is inserted into the next row or the algorithm terminates.

The rules for Hecke inserting a positive integer \( x \) into row \( R \) of a tableau \( T \) are as follows. Suppose first that \( x \geq y \) for all \( y \in R \).

1. If adjoining a box containing \( x \) to the end of \( R \) results in a valid increasing tableau \( T' \), then \( T' \) is the result of the insertion and the algorithm terminates.
2. If adjoining a box containing \( x \) to the end of \( R \) does not result in a valid increasing tableau, then \( R \) is unchanged and the algorithm terminates.
3. Otherwise, let \( y \) be the smallest integer in \( R \) that is strictly larger than \( x \).
4. If replacing \( y \) with \( x \) results in an increasing tableau, then replace \( y \) with \( x \) and insert \( y \) into the next row.
5. If replacing \( y \) with \( x \) does not result in an increasing tableau, then insert \( y \) into the next row and do not change \( R \).

We write \( T \leftarrow x \) to denote the final tableau resulting from the Hecke insertion of \( x \) into \( T \).

It will occasionally be convenient to consider the column insertion of \( x \) into \( T \), which is computed by performing Hecke insertion with columns playing the role of rows. Formally, the column insertion of \( x \) into \( T \) is given by \((T^t \leftarrow x)^t\). From now on, “insertion” will always refer to Hecke insertion.

**Example 2.8.** [6, Example 2.3]
\[
\begin{array}{ccc}
1 & 2 & 3 \\
2 & 3 & 4 \\
6 \\
\end{array} \leftarrow 3 = \begin{array}{ccc}
1 & 2 & 3 \\
2 & 3 & 4 \\
6 & 7 \\
\end{array}
\]

In this example, inserting 3 into the first row invokes rule (4), so we insert 5 into the second row. This invokes (4) again, so we insert 6 into the third row. By (2), we get the tableau shown.
Example 2.9. [6, Example 2.4]

\[
\begin{array}{ccc}
2 & 4 & 6 \\
3 & 6 & 8 \\
\hline
7 \\
\end{array} \leftarrow 5 = \begin{array}{ccc}
2 & 4 & 5 \\
3 & 6 & 8 \\
7 & 8 \end{array}
\]

We first insert 5 into the first row, which by (3) replaces the 6 in the rightmost box, bumping the 6 into the second row. By rule (4), the second row is unchanged, and we insert an 8 into the third row. Rule (1) gives the resulting tableau.

Let \( w = w_1 \cdots w_n \) be a word. The insertion tableau of \( w \), written \( P(w) \), is formed by recursively Hecke inserting the letters of \( w \) from left to right:

\[
P(w) = (\cdots (\varnothing \leftarrow w_1) \leftarrow w_2) \cdots \leftarrow w_n).
\]

We define the insertion class of \( w \) to be \( \{ w' : P(w') = P(w) \} \).

Proposition 2.10. [2, Theorem 6.2] If \( w \) and \( w' \) are in the same insertion class, then \( w \equiv w' \).

Hence Hecke insertion equivalence implies \( K \)-Knuth equivalence. The converse, however, is false – \( K \)-Knuth equivalent words may not be insertion equivalent. So unlike Knuth equivalence classes, \( K \)-Knuth equivalence classes may contain more than one tableau.

Example 2.11. Let \( w = 1342 \) and \( w' = 13422 \). Clearly \( w \) and \( w' \) are \( K \)-Knuth equivalent. Notice, however, that

\[
P(w) = \begin{array}{c}
1 \\
2 \\
4 \\
3 \\
\end{array}
\]

and

\[
P(w') = \begin{array}{c}
1 \\
2 \\
4 \\
3 \\
\end{array}
\]

Patrias and Pylyavskyy [6] define a recording tableau \( Q(w) \) for the Hecke insertion of a word \( w \), in analogy to the recording tableau for Schensted insertion, which allows one to uniquely recover an inserted word \( w \) from the pair \( (P(w), Q(w)) \) via reverse Hecke insertion. We will not use this notion, but note it for completeness.

2.3. \( K \)-Knuth equivalence. Just as Hecke insertion is a \( K \)-theoretic analogue of the standard RSK insertion, \( K \)-Knuth equivalence is the corresponding analogue for Knuth equivalence. Recall that the Knuth equivalence relations are as follows:

\[
xzy \sim zyx, \quad (x < y < z)
\]
\[
yxz \sim yzx, \quad (x < y < z)
\]
\[
xyx \sim yxx, \quad (x < y)
\]
\[
xyy \sim yyx, \quad (x < y).
\]

Two words are said to be Knuth equivalent if one can be obtained from the other via a finite series of applications of the above Knuth relations.

In the \( K \)-theoretic case, the first two rules are precisely the same. However, this case has two additional rules with important consequences. The \( K \)-Knuth relations are as follows:
\[ xzy \equiv zxy, \quad (x < y < z) \]
\[ yxz \equiv yzx, \quad (x < y < z) \]
\[ p \equiv pp, \]
\[ pqp \equiv qpq. \]

Again, two words are said to be \textit{K-Knuth equivalent} if one can be obtained from the other via a series of applications of the above \textit{K}-Knuth relations.

The third and fourth relations have some important implications. The third rule implies that each \textit{K}-Knuth equivalence class of words has infinitely many words of arbitrarily large length. Because Hecke insertion results in an increasing tableau, there are only finitely many tableaux into which words on an alphabet \([n]\) (words containing at least one of each letter from \(\{1, 2, ..., n\}\)) can be inserted. Hence there are infinitely many equivalence classes on any alphabet \([n]\) with infinitely many words in each class. This is in contrast to the standard version, in which there are only finitely many words in each class, but infinitely many classes on any given alphabet.

The fourth rule implies that two words can be equivalent, but each letter could appear a different number of times in the two words. For example, \(121 \equiv 212\), but 1 appears twice in the first word and once in the second.

\section*{2.4. Reading Words.}

The Hecke insertion algorithm associates to each word an increasing tableau. In this section, we describe a way to associate to each increasing tableau a certain set of words, called \textit{reading words} for the tableau. A tableau can have many reading words, but they will all be \textit{K}-Knuth equivalent.

Let \(T\) be an increasing tableau. The most commonly used reading word for \(T\) is the \textit{row word} for \(T\), written \(\text{row}(T)\), which is obtained by reading the entries of \(T\) from left to right along each row, starting from the bottom row and moving upward. Similarly, the \textit{column word} for \(T\), written \(\text{col}(T)\), is obtained by reading the entries of \(T\) from bottom to top along each column, starting from the first column and moving rightward.

More generally, let \(\sigma\) be a sequence of boxes of \(T\). If for any two boxes \(\alpha\) and \(\beta\), we have that \(\alpha\) appears before \(\beta\) in \(\sigma\) whenever \(\alpha\) is southwest of \(\beta\) in \(T\), then \(\sigma\) is said to be a \textit{reading order} for \(T\). If \(\sigma\) is a reading order for \(T\), we write \(\text{row}_\sigma(T)\) to denote the word produced by reading the entries of \(T\) in the order given by \(\sigma\).

\begin{example}
If \(T = \begin{array}{cccc}
1 & 3 & 4 & 5 \\
2 & 4 & 5 \\
4 
\end{array}\)
then \(\text{row}(T) = 42451345\) and \(\text{col}(T) = 42143545\). A third valid reading word for \(T\) is \(42145345\).
\end{example}

\begin{proposition} \cite[Lemma 5.4]{2} \end{proposition}

\begin{proof}
If \(\sigma\) and \(\pi\) are two reading orders of an increasing tableau \(T\), then \(\text{row}_\sigma(T) \equiv \text{row}_\pi(T)\).
\end{proof}

Let \(T\) and \(T'\) be two increasing tableaux. If \(\text{row}(T) \equiv \text{row}(T')\), \(\text{row}(T) \equiv \text{row}(T')\), we say that \(T\) is \textit{K-Knuth equivalent} to \(T'\) and write \(T \equiv T'\), giving a \textit{K}-Knuth equivalence relation on the set of increasing tableaux.
2.5. **K-Jeu de Taquin.** The classical jeu de taquin (jdt) algorithm defines an equivalence relation on standard skew tableaux. Recall a tableau $T$ is standard if it has entries exactly $1, 2, \ldots, n$ for some $n$.

**Example 2.14.** Of the fillings below, only the second is standard.

\[
\begin{array}{ccc}
1 & 2 & 3 \\
3 & 4 & 5 \\
7 & 8
\end{array}
\quad
\begin{array}{ccc}
1 & 2 & 3 \\
4 & 6 & 7 \\
8 & 10
\end{array}
\]

In this section we give a $K$-theoretic extension of jdt to increasing tableaux, $K$-jdt, closely following [2]. The $K$-jdt algorithm gives an alternative method for testing tableaux equivalence.

**Definition 2.15.** We say two boxes $\alpha, \beta \in \mathbb{N} \times \mathbb{N}$ are neighbors if $\alpha$ is directly above, below, left, or right of $\beta$. Given a tableau $T$ and two entries $s$ and $s'$ of $T$, define a new tableau swap$_{s,s'}(T)$ of the same shape by

\[
\text{swap}_{s,s'}(T) : \alpha \mapsto \begin{cases} 
\begin{array}{c}
s' \\
T(\alpha)
\end{array} & \text{if } T(\alpha) = s \text{ and } T(\beta) = s' \text{ for some neighbor } \beta \text{ of } \alpha; \\
\begin{array}{c}
s \\
T(\alpha)
\end{array} & \text{if } T(\alpha) = s' \text{ and } T(\beta) = s \text{ for some neighbor } \beta \text{ of } \alpha; \\
T(\alpha) & \text{otherwise}
\end{cases}
\]

**Example 2.16.** Two examples of swaps may be found below.

\[
\begin{array}{ccc}
1 & 2 & \bullet \\
\bullet & 3 \\
3
\end{array}
\quad
\begin{array}{ccc}
1 & 2 & \bullet \\
\bullet & 3 \\
3
\end{array}
\quad
\begin{array}{ccc}
1 & 2 & \circ \\
\circ & 3 & 4 \\
0
\end{array}
\quad
\begin{array}{ccc}
1 & 2 & \circ \\
\circ & 3 & 4 \\
0
\end{array}
\]

To define $K$-jdt, we will extend the labels of a tableau to the unordered set $\mathbb{N} \cup \{\bullet, \circ\}$. The symbols $\bullet$ and $\circ$ are undefined and will move across the tableau as swaps are performed during $K$-jdt. A complete $K$-jdt move is composed of a series of these swaps.

**Definition 2.17** ([2]). For a tableau $T$ of shape $\lambda$ on letters in the interval $[a, b]$ and a subset $C \subset \lambda$ of the maximally southeast boxes in $\lambda$, we define the forward slide of $T$ starting from $C$ as

\[
\text{jdt}_C(T) = (\text{swap}_a, \text{swap}_{a+1}, \ldots \text{swap}_{b-1}, \text{swap}_b, (C \rightarrow \bullet) \cup T),
\]

where $[C \rightarrow \bullet]$ denotes the constant tableau of shape $C$ that puts $\bullet$ in each box. Similarly, if $\hat{C} \subset \mathbb{N} \times \mathbb{N} \setminus \nu$ is a subset of the maximally northwest boxes of $\mathbb{N} \times \mathbb{N} \setminus \nu$, then the reverse slide of $T$ starting from $\hat{C}$ is

\[
\text{jdt}_{\hat{C}}(T) = (\text{swap}_b, \text{swap}_{b-1}, \ldots \text{swap}_{a+1}, \text{swap}_a, (T \cup [\hat{C} \rightarrow \circ])),
\]

where $[\hat{C} \rightarrow \circ]$ denotes the constant tableau of shape $\hat{C}$ that puts $\circ$ in each box.

It is apparent from the definitions of swap that tableaux resulting from forward and reverse $K$-jdt slides remain increasing along rows and columns. It is also apparent that, by design, forward and reverse moves are inverses. Furthermore, one can use forward moves to transform a skew shape into a straight shape, and reverse moves to do the opposite.
Example 2.18. The examples below give one complete forward jdt slide and one complete reverse jdt slide, showing the sequence of swaps performed during the slide.

\[
\begin{array}{cccccc}
1 & 3 & \rightarrow & 2 & 4 & \rightarrow \ 2 & \bullet & 4 & \rightarrow \ 2 & 3 & 4 & \rightarrow \ 2 & 3 & 4 & \Rightarrow \ 2 & 3 & 4
\end{array}
\]

In the same way that the K-Knuth “moves” give an equivalence relation on words, K-jdt slides give an equivalence relation on tableaux.

Definition 2.19 ([2]). We say that two increasing tableaux \( S \) and \( T \) are K-\( \text{jeu de taquin equivalent} \) if \( S \) can be obtained by applying a sequence of forward and reverse K-\( \text{jeu de taquin} \) slides to \( T \).

The importance of K-jdt equivalence lies in the following theorem, proved in [2].

Theorem 2.20. [2, Theorem 6.2] For tableaux \( T \) and \( T' \), row \( (T) \equiv \text{row}(T') \) if and only if \( T \) and \( T' \) are K-jdt equivalent.

Therefore, K-jdt equivalence is the same as K-Knuth equivalence.

2.6. Unique Rectification Targets. After applying enough forward K-jdt slides to a skew tableau \( T \), a straight tableau called a K-rectification of \( T \) will eventually result. The rectification order is the choice of the placements of \( \bullet \)'s (resp. \( \circ \)'s) for each forward (resp. reverse) K-jdt slide. In contrast to the classical theory of jdt, different rectification orders may result in different K-rectifications; in other words, varying the placements of \( \bullet \)'s during K-jdt slides may result in different straight tableaux.

Example 2.21. Here is an example how different rectification orders may produce different K-rectifications. The tableau

\[
\begin{array}{cccc}
\bullet & 2 & \\
2 & 1 & 3 & 4
\end{array}
\]

has the rectifications

\[
\begin{array}{cccccc}
\bullet & 2 & \rightarrow & 2 & \bullet & 4 & \rightarrow \ 2 & \circ & 4 & \rightarrow \ 2 & 1 & 3 & 4 & \rightarrow \ 1 & 2 & 4 & \rightarrow \ 1 & 2 & 4 & \Rightarrow \ 1 & 2 & 4
\end{array}
\]

and

\[
\begin{array}{cccccc}
\bullet & 2 & \rightarrow & 2 & \bullet & 4 & \rightarrow \ 2 & \circ & 4 & \rightarrow \ 2 & 1 & 3 & 4 & \rightarrow \ 1 & 2 & 4 & \rightarrow \ 1 & 2 & 4 & \Rightarrow \ 1 & 2 & 4
\end{array}
\]

resulting in different tableaux.

In some instances the K-rectification may be unique, motivating the following definition.
**Definition 2.22.** [2, Definition 3.5] An increasing tableau $U$ is a *unique rectification target* (URT) if, for every increasing tableau $T$ that has $U$ as a rectification, $U$ is the only rectification of $T$.

Equivalently, an increasing tableau is a URT if it is the only tableau in its $K$-Knuth equivalence class. The literature gives several classes of URTs, which we summarize below.

**Definition 2.23.** A *minimal* tableau is a tableau in which each box is filled with the smallest positive integer that will make the filling a valid increasing tableau.

**Example 2.24.** The following tableau is minimal of shape $(4, 3, 3, 1)$:

```
  1 2 3 4
  2 3 4
  3 4 5
  4
```

**Proposition 2.25.** [2, Corollary 4.7] Every minimal tableau is a URT.

**Definition 2.26.** A *superstandard* tableau is a standard tableau that fills the first row with $1, 2, \ldots, \lambda_1$, the second row with $\lambda_1 + 1, \lambda_1 + 2, \ldots, \lambda_1 + \lambda_2$, etc., where $\lambda_i$ is the length of the $i^{th}$ row of the tableau.

**Example 2.27.** The following tableau is superstandard of shape $(4, 3, 3, 1)$:

```
  1 2 3 4
  5 6 7
  8 9 10
  11
```

**Proposition 2.28.** Every superstandard tableau is a URT.

**Definition 2.29.** A *fat hook* is a partition of the form $(a, b, c, d)$, where $a, b, c, d$ are nonnegative integers with $a \geq c$.

**Example 2.30.** The tableau below is a fat hook of shape $(4^2, 2^3)$.

```
   |   |   |
   |   |   |
   |   |   |
   |   |   |
```

**Definition 2.31.** Let $M_{\lambda}$ be the minimal increasing tableau corresponding to a fat hook $\lambda = (a^b, c^d)$, and let $U$ be an increasing tableau. We say $U$ fits in the corner of $M_{\lambda}$ if $U$ has at most $d$ rows, at most $a - c$ columns, and all integers contained in $U$ are strictly larger than all integers contained in $M_{\lambda}$.

In other words, $U$ fits in the corner of $M_{\lambda}$ if the entries of $U$ are strictly greater than the entries of $M_{\lambda}$ and if positioning $U$ in the corner of the hook results in an increasing tableau.
Theorem 2.32. [2, Theorem 6.9] Let \( \lambda \) be a fat hook, and let \( U \) be any unique rectification target that fits in the corner of \( M_\lambda \). Then \( M_\lambda \cup U \) is a unique rectification target.

Example 2.33. We may conclude that the tableau

\[
T = \begin{array}{cccc}
1 & 2 & 3 & 4 \\
2 & 3 & 4 & 5 \\
3 & 6 & 7 & \\
4 & 7 & \\
5 & 
\end{array}
\]

is a URT because \( T = M_{(4^2,1^3)} \cup U \), where

\[
M_{(4^2,1^3)} = \begin{array}{cccc}
1 & 2 & 3 & 4 \\
2 & 3 & 4 & 5 \\
3 & 4 & 5 & \\
4 & & & \\
5 & & & 
\end{array}
\quad \text{and} \quad U = \begin{array}{cc}
6 & 7 \\
7 & 
\end{array}
\]

2.7. \( K \)-Knuth invariants. Now that we have the notion of an equivalence class of tableaux, we will provide several invariants under the \( K \)-Knuth equivalence relation. These will aid in proving results concerning the relations between tableaux in equivalence classes. A comprehensive list will be provided at the end of the section for ease of reading.

Definition 2.34. For a word \( w \), let \( \lis(w) \) (resp. \( \lds(w) \)) denote the length of the longest strictly increasing (resp. decreasing) subsequence of \( w \).

If \( w \) is the row word of a tableau \( T \) then \( \lis(w) \) is the length of the first row of \( T \) and \( \lds(w) \) is the length of the first column of \( T \).

Example 2.35. We will use the reading word of the tableau \( T \) from Example 2.33 to illustrate this concept. We see that if \( w = \text{row}(T) = 54736723451234 \), then \( \lis(w) = 4 \), and \( \lds(w) = 2 \).

Proposition 2.36. [6, Lemma 2.17] If \( w_1 \equiv w_2 \) then \( \lis(w_1) = \lis(w_2) \) and \( \lds(w_1) = \lds(w_2) \).

The above equalities follow easily from the \( K \)-Knuth equivalence relations.

Theorem 2.37. [8, Theorem 1.3] For any word \( w \), the size of the first row and first column of \( P(w) \) are given by \( \lis(w) \) and \( \lds(w) \), respectively.

See Definition 5.9 for a detailed description of the northeast-hook-closed shapes in part (b) of the following definition and proposition.

Definition 2.38. For a word \( w \), let \( w|_{[a,b]} \) denote the word obtained from \( w \) by deleting all integers not contained in the interval \([a,b]\). Likewise, let \( T \) be an increasing tableau, not necessarily straight, and let \( T|_{[a,b]} \) denote the tableau obtained from \( T \) by removing all boxes not in \([a,b]\). 

Proposition 2.39. [2, Lemma 5.5]

Let \([a,b]\) be an integer interval.

(a) Let \( w_1 \) and \( w_2 \) be \( K \)-Knuth equivalent words. Then \( w_1|_{[a,b]} \) and \( w_2|_{[a,b]} \) are \( K \)-Knuth equivalent words.
(b) Let $T_1$ and $T_2$ be increasing tableaux of northeast-hook-closed shapes with $T_1 \equiv T_2$. Then $T_1|_{[a,b]}$ and $T_2|_{[a,b]}$ are K-Knuth equivalent tableaux.

Example 2.40. We have that $T_1 \equiv T_2$ for

$T_1 = \begin{array}{ccccc}
1 & 2 & 3 & 4 & 5 \\
2 & 3 & & & \\
3 & & & & \\
5 & & & & \\
\end{array}$

$T_2 = \begin{array}{ccccc}
1 & 2 & 3 & 4 & 5 \\
2 & 3 & 5 & & \\
3 & 5 & & & \\
5 & & & & \\
\end{array}$

Therefore $T_1|_{[3,5]} \equiv T_2|_{[3,5]}$, so that

$\begin{array}{ccc}
3 & 4 & 5 \\
3 & & 5 \\
5 & & & \\
\end{array}$

$\equiv$

$\begin{array}{ccc}
3 & 4 & 5 \\
3 & & 5 \\
5 & & & \\
\end{array}$

Part (b) of Proposition 2.39 gives the second invariant for equivalent tableaux, their restriction to an interval subalphabet. We can now form equivalent sub-tableaux by deleting the boxes of equivalent tableaux that are not included in the interval subalphabet.

Definition 2.41. Let $T$ be a straight tableau. The outer hook of $T$ is the sub-tableaux of $T$ consisting of the first row and the first column.

Example 2.42. The outer hook of the tableau below is shaded gray.

$\begin{array}{ccccc}
1 & 2 & 4 & 5 \\
3 & 4 & 8 & \\
6 & 7 & & & \\
\end{array}$

Our third invariant for K-Knuth classes is the outer hook.

Proposition 2.43. Let $T$ and $T'$ be initial tableau such that $T \equiv T'$. Then $T$ and $T'$ have the same outer hook.

Proof. Suppose $T$ and $T'$ have letters in the alphabet $[n]$. Consider the two tableau sequences

$T|_{[1]}, T|_{[2]}, \ldots, T|_{[n]}$

$T'|_{[1]}, T'|_{[2]}, \ldots, T'|_{[n]}$.

We have already seen that $T|_{[i]} \equiv T'|_{[i]}$ because $T \equiv T'$. Now proceed by induction on $n$. The tableaux $T|_{[1]}$ and $T'|_{[1]}$ are equivalent and contain one box, meaning $T|_{[1]} = T'|_{[1]}$. At the $i$th step of the sequence, if $T|_{[i]}$ and $T'|_{[i+1]}$ have the same outer hook then $T|_{[i+1]}$ and $T'|_{[i+1]}$ must have the same outer hook as well because the outer hook of $T|_{[i]}$ (resp. $T'|_{[i]}$) differs from the outer hook of $T|_{[i+1]}$ (resp. $T'|_{[i+1]}$) by the addition of at most two boxes, which must be labeled with $i+1$.

Our fourth invariant is simple and involves the transpose of a tableau: if $T_1 \equiv T_2$, then $T'_1 \equiv T'_2$. Invariance under the transpose follows from the fact that if $T_1$ and $T_2$ are K-Knuth equivalent then they are K-jdt equivalent, and any sequence of K-jdt moves connecting the two may be applied to their transposes.

Example 2.44. We have that

$\begin{array}{ccc}
1 & 2 & 4 \\
3 & & \\
\end{array}$

$\equiv$

$\begin{array}{ccc}
1 & 2 & 4 \\
3 & & \\
\end{array}$
Another invariant is the Hecke permutation, defined in [1] to provide a coarser equivalence than $K$-Knuth equivalence. Define $\Sigma$ to be the group of bijective maps 

$$\{ w : \mathbb{N} \to \mathbb{N} \mid w(x) = x \text{ for all but finitely many } x \in \mathbb{N} \}.$$ 

The adjacent transpositions $s_i = (i, i+1) \in \Sigma$ generate $\Sigma$ and give the group a natural presentation as a Coxeter group. We will use this Coxeter group structure to define a new product on $\Sigma$ that makes $\Sigma$ into a monoid.

Given a permutation $u$, let $\ell(u)$ denote the shortest length of a factorization $u = s_{i_1} \cdots s_{i_k}$ of $u$ as a product of the $s_i$ transpositions. Equivalently, $\ell(u)$ equals the number of inversions of $u$, that is, the number of tuples $i < j$ such that $u(i) > u(j)$ (see [4, Sec. 1.6 Exercise 2]).

**Definition 2.45.** Let $u \in \Sigma$ be a permutation. The Hecke product of $u$ and a transposition $s_i$ is defined by

$$u \cdot s_i = \begin{cases} u s_i & \text{if } \ell(us_i) > \ell(u); \\ u & \text{otherwise.} \end{cases}$$

Given a second permutation $v = s_{i_1} \cdot s_{i_2} \cdots s_{i_l} \in \Sigma$, the Hecke product of $u$ and $v$ is defined by

$$u \cdot v = u \cdot s_{i_1} \cdot s_{i_2} \cdots s_{i_l},$$

multiplying in left to right order.

The Hecke product is associative and gives a monoidal structure on $\Sigma$, allowing us to introduce the following concept.

**Definition 2.46.** The Hecke permutation of an increasing tableau $T$ is $w(T) = w(a) = s_{a_1} \cdot s_{a_2} \cdots s_{a_k}$, where $a = a_1 \cdots a_k$ is a reading word of $T$.

**Proposition 2.47.** The Hecke permutation $w(T)$ of an increasing tableau is invariant under $K$-Knuth moves.

Proposition 2.47 is equivalent to Corollary 6.5 in [2], using the fact that $K$-jdt equivalence implies $K$-Knuth equivalence.

Having the same Hecke permutation is a necessary but not sufficient condition for two tableaux to appear in the same $K$-Knuth class, meaning that the number of $K$-Knuth equivalence classes on $[n]$ is at least as large as $S_{n+1}$, the symmetric group on $n + 1$ elements. Hence there are a minimum of $(n + 1)!$ $K$-Knuth classes of tableaux on $[n]$ letters. Proposition 7.2 will make use of this fact.

**Example 2.48.** The Hecke permutation for the word 21231 is given by

$$1 \mapsto 3, \quad 2 \mapsto 2, \quad 3 \mapsto 4, \quad 4 \mapsto 1.$$ 

In summary, the following are invariant under the $K$-Knuth equivalence relations:

1. the longest strictly increasing (or decreasing) subword of a word,
2. the restriction of a word (or a tableau) to an interval subalphabet, up to $K$-Knuth equivalence,
3. the outer hook of a tableau,
(4) the transpose of a tableau, up to $K$-Knuth equivalence, and
(5) the Hecke permutation.

3. Algorithms

This section deals with computational aspects of the $K$-Knuth equivalence relation. In particular, we will describe algorithms to answer the following two problems:

(1) Determine if two words are $K$-Knuth equivalent.
(2) Compute all $K$-Knuth classes of tableaux on $[n]$.

Let $T_n$ be the set of (not necessarily initial) increasing tableaux on $[n]$, and let $N_n$ be its cardinality. We will prove the following theorem.

**Theorem 3.1.** If $w_1$ and $w_2$ are words of length at most $\ell$ on $[n]$, then we can determine whether they are $K$-Knuth equivalent in $O(\ell + n^3 N_n \alpha(n))$ time, where $\alpha(n)$ is the inverse Ackermann function.

Recall that the inverse Ackermann function is very slowly-growing and satisfies $\alpha(n) = o(\log^* n)$, where the iterated logarithm $\log^* n$ is, roughly speaking, the number of times the logarithm function is applied iteratively to $n$ until we arrive at a number less than or equal to 1. More precisely, $\log^*$ is defined recursively by

$$
\log^* n = \begin{cases} 
0 & \text{if } n \leq 1 \\
1 + \log^*(\log n) & \text{if } n > 1
\end{cases}
$$

Surprisingly enough, the way to achieve the efficiency is to determine all $K$-Knuth equivalence classes. More precisely, we will prove the following.

**Theorem 3.2.** We can divide all increasing tableaux on $[n]$ into $K$-Knuth equivalence classes in $O(n^3 N_n \alpha(n))$ time.

We first remark that it is fairly easy to construct the set $T_n$ of all (not necessarily initial) tableaux on $[n]$. Indeed, one can construct this set recursively. Given $T_{n-1}$, one can check for each $T \in T_{n-1}$ where one can add boxes with entry $n$ to $T$ to get $T' \in T_n$. This takes only $O(N_n)$ time. Therefore, we will assume that we have the set $T_n$ at our disposal at our disposal in the algorithms for Theorem 3.1 and Theorem 3.2.

We also note the following simple lemma. Consider the following directed graph $G$ with labelled edges: the set of vertices of $G$ is $T_n$, and for $1 \leq y \leq n$, there is an edge from $T$ to $T'$ with label $y$ if and only if $T' = T \leftarrow y$.

**Lemma 3.3.** We can construct the directed graph $G$ in $O(n^2 N_n \log n)$ time.

**Proof.** The graph $G$ contains $O(nN_n)$ edges. Given a row of length $L$ and a letter $y$, we can use binary search to find the smallest entry in the row greater than $y$ and then check whether replacing the entry with $y$ violates the increasing tableau condition in constant time. Therefore, insertion of a letter into a row of length $L$ takes $O(\log L)$ time, and each Hecke insertion takes $O(n \log n)$ time. Thus the construction of $G$ takes $O(n^3 N_n \log n)$ time.

Using Lemma 3.3, we will assume that we have the graph $G$ at our disposal in the algorithms for Theorem 3.1 and Theorem 3.2. We now show that Theorem 3.1 follows from Theorem 3.2.
Proof of Theorem 3.1. Note that \( w_1 \equiv w_2 \) if and only if \( P(w_1) \equiv P(w_2) \). Hence we can compute \( P(w_1) \) and \( P(w_2) \) and then check whether \( P(w_1) \equiv P(w_2) \) by computing all K-Knuth equivalence classes on \([n]\). Hecke insertion can be done by going through a path with corresponding labels in the directed graph \( G \), so computing \( P(w_1) \) and \( P(w_2) \) takes \( O(\ell) \) time. □

Before describing an algorithm to prove Theorem 3.2, we first recall two results about algorithms.

**Proposition 3.4.** [3, Section 21.3] Let \( X \) be a set of cardinality \( k \). The data structure disjoint-set forest can store a set partition \( P \) of \( X \) and perform the following operations:

1. Given \( x, y \in X \), we can check whether \( x \) and \( y \) are in the same set of \( P \);
2. If \( x, y \in X \) are in different sets in \( P \), we can merge the sets containing \( x \) and \( y \);
3. Output all sets in \( P \).

Moreover, if we perform \( m \) operations of type (1) and (2) and one operation of type (3), the total runtime is \( O((m+k)\alpha(k)) \).

**Proposition 3.5.** [3, Section 10.1] The data structure queue can perform the following operations:

1. Insert an element into the queue;
2. Remove some element from the queue and output the removed element.

Moreover, each insertion and removal takes \( O(1) \) time.

Throughout the algorithm we will maintain a set partition \( P \) of \( T_n \) and a queue \( Q \) that stores pairs \((T_1, T_2)\) of tableaux in \( T_n \).

We say that a pair \((a,b)\) of words is a primitive pair if either

1. \( a = p, b = pp \) for some letter \( p \);
2. \( a = pqp, b = qpq \) for some letters \( p \neq q \);
3. \( a = xzy, b = zxy \) for some letters \( x < y < z \); or
4. \( a = yxz, b = yzx \) for some letters \( x < y < z \).

**Algorithm 1** Algorithm for Computing all K-Knuth Classes

1. Initialization: \( P := \{\{T\} : T \in T_n\}; Q \) empty.
2. for all \( T \in T_n \) do
3. for all primitive pair \((a,b)\) do
4. Compute \( T_1 := T \leftarrow a \) and \( T_2 := T \leftarrow b \) using \( G \);
5. if \( T_1 \) and \( T_2 \) are in different sets of \( P \) then
6. Merge the sets in \( P \) containing \( T_1 \) and \( T_2 \) respectively;
7. Insert the pair \((T_1,T_2)\) into \( Q \).
8. end if
9. end for
10. end for
11. while \( Q \) is non-empty do
12. Remove a pair \((U_1,U_2)\) from \( Q \).
13. for all \( 1 \leq y \leq n \) do
14. Compute \( T_1 := U_1 \leftarrow y \) and \( T_2 := U_2 \leftarrow y \) using \( G \);
We claim that the set partition $\mathcal{P}$ gives the $K$-Knuth equivalence classes in $T_n$ at the termination of the algorithm.

**Theorem 3.6.** If $T_1$ and $T_2$ are $K$-Knuth equivalent, then they lie in the same set $S \in \mathcal{P}$ at the end of the algorithm.

To prove Theorem 3.6 we require two preliminary lemmas.

**Lemma 3.7.** Fix $1 \leq y \leq n$. Let $T_1$ and $T_2$ be two tableaux and let $T'_1 = T_1 \leftarrow y$. If $T_1$ and $T_2$ lie in the same set of $\mathcal{P}$ at the end, then $T'_1$ and $T'_2$ lie in the same set of $\mathcal{P}$ at the end.

**Proof.** We first prove the assertion for the special case where, at some point of the execution, $(T_1, T_2) \in Q$. In this case, $(T_1, T_2)$ is eventually removed from $Q$ in line 12. If $T'_1$ and $T'_2$ are in the same $S \in \mathcal{P}$ at this point, then the assertion holds. Otherwise, we will merge the sets containing $T'_1$ and $T'_2$.

Now we consider the general case. We need a simple claim to aid us in the proof.

**Claim 3.8.** There exists a sequence $T_1 = U_0, U_1, \ldots, U_r = T_2$ of tableaux such that, for each $i$, either $(U_i, U_{i+1})$ or $(U_{i+1}, U_i)$ is in $Q$ at some point.

We shall see how this claim proves the general case. For each $i$, let $U'_i = U_i \leftarrow y$. Claim 3.8 shows that, at the end of the program, $U'_i$ and $U'_{i+1}$ are in the same set of $\mathcal{P}$ for each $i$, i.e., all $U'_i$ are in the same $S \in \mathcal{P}$. In particular, $T'_1$ and $T'_2$ are in the same $S \in \mathcal{P}$.

It remains to prove the claim. Assume that $T_1$ and $T_2$ are in the same set of $\mathcal{P}$ after the $k$th merge but not before. We will proceed by induction on $k$. The case $k = 0$ is trivial. Suppose $k > 0$. Let $S_1$ and $S_2$ be the sets containing $T_1$ and $T_2$, respectively, before the $k$th merge. By assumption, the sets $S_1$ and $S_2$ merge at the $k$th merge. After that merge, we insert into $Q$ the pair $(U_1, U_2)$, for some $U_1 \in S_1$ and $U_2 \in S_2$. We know from the induction hypothesis that there is a chain of pairs in $Q$ connecting $T_1$ and $U_1$ as well as $T_2$ and $U_2$. The result then follows. □

**Lemma 3.9.** Let $T \in T_n$ and let $(a, b)$ be a primitive pair. Then $T_1 := T \leftarrow a$ and $T_2 := T \leftarrow b$ are in the same $S \in \mathcal{P}$ at the end of the algorithm.

**Proof.** At some point we check whether $T_1$ and $T_2$ are in the same set $S \in \mathcal{P}$. If they are in the same set, then we are fine. Otherwise, we will merge the sets containing them on the next line. □

**Proof of Theorem 3.6.** We start with a special case. Suppose there exist words $w_1$ and $w_2$ that differ by one $K$-Knuth move and such that $P(w_1) = T_1$ and $P(w_2) = T_2$. Write $w_1 = uav$ and $w_2 = ubv$ where $a$ and $b$ form a primitive pair, so that by Lemma 3.9 the tableaux $P(ua)$ and $P(ub)$ are in the same set of $\mathcal{P}$.

```plaintext
15: if $T_1$ and $T_2$ are in different sets of $\mathcal{P}$ then
16:   Merge the sets in $\mathcal{P}$ containing $T_1$ and $T_2$ respectively;
17:   Insert the pair $(T_1, T_2)$ into $Q$.
18: end if
19: end for
20: end while
21: Output all sets $S \in \mathcal{P}$.
```
Applying Lemma 3.7 multiple times, we conclude that \( T_1 \) and \( T_2 \) are in the same set of \( \mathcal{P} \).

For the general case, there is a sequence \( T_1 = U_0, U_1, \ldots, U_r = T_2 \) of tableaux such that for each \( i \) there exist two words differing by one \( K \)-Knuth move and inserting into \( U_i \) and \( U_{i+1} \) respectively. By the previous case \( U_i \) and \( U_{i+1} \) are in the same set of \( \mathcal{P} \), so at the end all \( U_i \)'s are in the same \( S \in \mathcal{P} \). In particular, \( T_1 \) and \( T_2 \) are in the same set of \( \mathcal{P} \). □

Having shown the correctness of the algorithm, we will now analyze the runtime.

**Theorem 3.10.** Algorithm 1 runs in \( O(n^3 N_n \alpha(n)) \) time, where \( N_n \) is the number of increasing tableaux on \([n] \).

**Proof.** There are \( O(n^3) \) primitive pairs, so lines 4 – 7 will be executed \( O(n^3 N_n) \) times. There are at most \( N_n - 1 \) merges, so lines 14 – 17 will be executed \( O(nN_n) \) times.

By Proposition 3.4 and Proposition 3.5 the runtime for the whole algorithm is \( O(n^3 N_n \alpha(N_n)) \). Since every tableau on \([n] \) has at most \( n^3 \) boxes and each box is either empty or contains an entry in \([n] \), we have \( N_n \leq (n+1)^n \). This implies that \( \alpha(N_n) = \Theta(\alpha(n)) \), giving the desired asymptotic bound. □

### 4. Length of Intermediate Words

If \( w \equiv w' \), then there exists a sequence \( w = w_0, w_1, \ldots, w_r = w' \) of words such that \( w_i \) and \( w_{i+1} \) differ by one \( K \)-Knuth move. It is natural to ask whether it is always possible to find such a sequence so that the intermediate words \( w_i \) always have length at most that of the longer one. Surprisingly, the answer is no, as one can check by computer that \( 4235124 \equiv 4523124 \) cannot be connected by words of length at most 7. However, it is possible to give an upper bound in terms of the size of the alphabet.

To put the result on a more formal footing, we will make the following definition.

**Definition 4.1.** Let \( w \) and \( w' \) be words and let \( k \) be a positive integer. We say that \( w \equiv^k w' \) if there exists a sequence \( w = w_0, w_1, \ldots, w_r = w' \) of words such that \( w_i \) and \( w_{i+1} \) differ by one \( K \)-Knuth move, and each word \( w_i \) has length at most \( k \).

We will prove the following result.

**Theorem 4.2.** Suppose \( T_1 \equiv T_2 \) are tableaux on \([n] \). Let \( N = \frac{1}{3} n(n+1)(n+2) + 3 \). Then \( \text{row}(T_1) \equiv^N \text{row}(T_2) \).

Computer evidence suggests that the bound in Theorem 4.2 can be tightened to the largest size of a tableau in the \( K \)-Knuth equivalence class, where the size of a tableau \( T \) of shape \( \lambda \) is the number of boxes contained in \( \lambda \).

**Conjecture 4.3.** Let \( T \) and \( T' \) be two tableaux with \( T \equiv T' \), and let \( k \) be the largest size of a tableau \( K \)-Knuth equivalent to \( T \) or \( T' \). Then \( \text{row}(T) \equiv^k \text{row}(T') \).

Conjecture 4.3 has been verified for tableaux on \([n] \) with \( n \leq 5 \).
4.1. Proof of Theorem 4.2. We will use the following lemma, which concerns $K$-Knuth equivalence within an insertion class, to prove Theorem 4.2. Let $|w|$ denote the number of letters in a word $w$.

Lemma 4.4. If $w$ is a word and $P(w) = T$, then $w |w| \equiv \text{row}(T)$.

We defer the proof of Lemma 4.4 to Section 4.2 because it is fairly technical. We first require a simple technical result.

Lemma 4.5. If $w$ is a word on $[n]$, then there are at most $\frac{1}{6}n(n+1)(n+2)$ letters $t$ such that, when we write $w = utv$, we have $P(ut) \neq P(u)$.

We may now conclude the proof of Theorem 4.2.

Proof of Lemma 4.5. Fix $i$ and $j$ with $i + j = k + 1$. If $P(ut)$ and $P(u)$ have different $(i,j)$th entries, then the insertion of $t$ either creates a new entry at the $(i,j)$th position or decreases the $(i,j)$th entry. At the end the $(i,j)$th entry must be at least $k$, so there are at most $n - k + 1$ letters in $w$ that change the $(i,j)$th entry. There are exactly $k$ pairs of $(i,j)$ with $i + j = k + 1$, so the result follows from
\[\sum_{k=1}^{n} k(n - k + 1) = \frac{1}{6}n(n+1)(n+2).\]

4.2. Proof of Lemma 4.4. The proof of Lemma 4.4 will consist of a careful analysis of the Hecke insertion algorithm. In essence, computing an insertion tableau $P(w)$ is the same as making a sequence of $K$-Knuth moves to the word $w$, and none of these moves lengthens the word.

We first require a simple technical result.

Lemma 4.6. If $a_1 < a_2 < \cdots < a_i < b$, then $a_1 a_2 \cdots a_{i-1} b a_i \equiv b a_1 a_2 \cdots a_i$. If $b < a_1 < a_2 < \cdots < a_i$, then $a_1 b a_2 \cdots a_i \equiv a_1 a_2 \cdots a_i b$. 

Proof. For the first (resp. second) statement, apply the K-Knuth moves \( zzy \to xzy \) (resp. \( yzx \to yxz \)) for \( x < y < z \) as many times as needed to move the \( b \).

The first step of the proof of Lemma 4.4 is the reduction to the special case where \( w = \text{row}(T')y \) for some tableau \( T' \) and some letter \( y \). Assuming this case has been proved, write \( w = w_1w_2 \cdots w_k \). Let \( T_i = P(w_1w_2 \cdots w_i) \), \( t_i = \text{row}(T_i) \), and \( u_i = t_iw_{i+1} \cdots w_k \). The assertion in the special case implies that \( t_{i+1} \equiv t_iw_{i+1} \), so that \( u_{i+1} \equiv u_i \). The general case then follows.

We further reduce the special case \( w = \text{row}(T')y \) to proving the following assertion.

**Claim 4.7.** Suppose \( R = r_1r_2 \cdots r_k \) is the row word of a row of \( T' \). If \( R \) is the first row, let \( S \) be the empty word; otherwise, let \( S = s_1s_2 \cdots s_\ell \) (with \( \ell \geq k \)) be the row word of the previous row. Suppose the insertion of \( y \) into \( R \) changes \( R \) to \( R' \).

Let \( m = k + \ell + 1 \).

1. If \( y \geq r_k \), then we have \( RyS \equiv R'S \).
2. If \( r_i \leq y < r_{i+1} \) for some \( i \), then we have \( RyS \equiv r_{i+1}R'S \).

We first examine how this claim implies the special case \( w = \text{row}(T')y \) of Lemma 4.4. Let \( R_i' \) (resp. \( R_i \)) \( (1 \leq i \leq n) \) be the row word of the \( i \)th row of \( T' \) (resp. \( T \)) (if the tableau has no such row, let the corresponding row word be the empty word). Suppose we only insert letters into the first through the \( m \)th row when we insert the letter \( y \) into the tableau \( T' \), and let \( y_i \) be the letter inserted into the \( i \)th row (so that \( y_1 = y \)). Note that \( R_i' = R_i \) for \( i > m \). It then follows from Claim 4.7 that

\[
\text{row}(T')y = R_nR_{n-1}' \cdots R_1'y_1y_2 \cdots y_m \equiv R_n'R_{n-1}' \cdots R_2' \equiv R_n'y_{n-1} \cdots R_2'y_2y_1 \equiv R_n'y_{n-1} \cdots R_2'y_2 \equiv \cdots \equiv R_n'y_{n-2}y_{n-1} \equiv R_n'y_{n-2} \equiv R_nR_{n-1} \cdots R_1 = R_nR_{n-1} \cdots R_1 = \text{row}(T).
\]

Proving Lemma 4.4 is therefore reduced to proving Claim 4.7. We will first prove the first part of the claim, the case where \( y \) does not replace any entry. The first part separates into the following cases.

(a) \( k = 0 \).
   (i) \( s_1 \neq y \).
   (ii) \( s_1 = y \).
(b) \( k > 0 \) and \( r_k = y \).
(c) \( k > 0 \), \( r_k < y \) and \( s_{k+1} \neq y \).
(d) \( k > 0 \), \( r_k < y \) and \( s_{k+1} = y \).
   (i) \( k = 1 \).
   (ii) \( k > 1 \).

Define \( \alpha = r_1r_2 \cdots r_{k-1} \).

Now we tackle individual cases.

(a) (i) We have \( RyS = yS \equiv R'S \) and the result is trivial.
   (ii) Let \( \gamma = s_2s_3s_4 \cdots s_\ell \). In this case,

\[
RyS = yy\gamma \equiv y\gamma = S = R'S.
\]
(b) We have $R = R' = \alpha y$. Thus $Ry = \alpha yy^{k+1} \equiv \alpha y = R'$. Hence $RyS^m \equiv R'S$.

(c) We have $RyS = \alpha r_k S = R'S$ and the result is trivial.

(d) Let

$$\beta = s_1 s_2 s_3 \cdots s_{k-2},$$

$$\gamma = s_{k+2} s_{k+3} \cdots s_t.$$

(i) We have $RyS = r_1 ys_1 y \gamma$ and $R'S = r_1 s_1 y \gamma$. Hence it suffices to prove that $r_1 ys_1 y \equiv r_1 s_1 y$. By standardization this is equivalent to $2313 \equiv 213$, which follows from the following computation:

$$2313 \equiv 2133 \equiv 213.$$

(ii) Applying Lemma 4.6 twice, we have

$$RyS = \alpha r_k \beta y s_{k-1} y \gamma \equiv \alpha \beta r_k s_{k-1} y s_k y \gamma.$$

Similarly, applying Lemma 4.6 gives us

$$R'S = \alpha r_k \beta y s_{k-1} y \gamma \equiv \alpha \beta s_{k-1} r_k s_k y \gamma.$$

Hence it suffices to prove that $r_k s_{k-1} y s_k y \equiv s_{k-1} r_k s_k y$. By standardization this is equivalent to $31424 \equiv 3124$, which follows from the following computation:

$$31424 \equiv 31242 \equiv 13242 \equiv 13422 \equiv 1342 \equiv 3124 \equiv 3124.$$

Next, we prove the second part of Claim 4.7, where $y$ replaces a letter. Again, there are several cases to consider.

(e) $i = 0$.

(i) $s_1 \neq y$.

(ii) $s_1 = y$.

(f) $i > 0$ and $s_i = y$.

(g) $i > 0$, $s_i < y$ and $s_{i+1} \neq y$.

(h) $i > 0$, $s_i < y$ and $s_{i+1} = y$.

(i) $i = 1$.

(ii) $i > 1$.

Define

$$\alpha = r_1 r_2 r_3 \cdots r_{i-1},$$

$$\beta = r_i + 2 r_i + 3 r_i + 4 \cdots r_k.$$ 

By Lemma 4.6, we have

$$r_{i+1} \beta y \equiv r_{i+1} y \beta.$$

Now we tackle individual cases.

(e) By (1), $RyS^m \equiv r_1 y \beta S$.

(i) We have $R \equiv r_1 y \beta S$ and the result follows.

(ii) Let $\chi = s_2 s_3 s_4 \cdots s_t$. In this case,

$$RyS = r_1 \beta y y \chi \equiv r_1 \beta y \chi = r_1 r_1 \beta y \chi = r_1 R' S.$$
(f) By equation (1),
\[ RyS \equiv \alpha r_{i+1} y \beta S \equiv \alpha r_{i+1} y r_{i+1} \beta S. \]

By Lemma 4.6, we have
\[ r_{i+1} R' S = r_{i+1} \alpha r_{i+1} y \beta S \equiv \alpha r_{i+1} y r_{i+1} \beta S. \]

Hence \( RyS \equiv r_{i+1} R' S. \)

(g) By Lemma 4.6 and (1), we have
\[ r_{i+1} R' S = r_{i+1} \alpha r_{i+1} y \beta S \equiv \alpha r_{i+1} y r_{i+1} \beta S \equiv \alpha r_{i+1} y S = RyS. \]

(h) Let
\[
\begin{align*}
\gamma &= s_1 s_2 \cdots s_{i-2}, \\
\delta &= s_{i+2} s_{i+3} \cdots s_k, \\
\epsilon &= r_{i+2 s_{i+2} s_{i+3} \cdots s_k}, \\
\chi &= s_{k+1} s_{k+2} s_{k+3} \cdots s_{\ell}.
\end{align*}
\]

Applying Lemma 4.6 multiple times, we have
\[ r_{i+1} R' S = r_{i+1} r_{i+1} \beta \gamma s_{i-1} \delta \chi \equiv r_{i+1} r_{i+1} \gamma s_{i-1} \delta \chi. \]

(i) By (1) and (2),
\[ RyS = r_1 r_2 \beta y s_1 y \delta \chi \equiv r_1 r_2 y s_1 y \delta \chi \equiv r_1 r_2 y s_1 y \epsilon. \]

Similarly, by (2),
\[ r_2 R' S = r_2 r_1 r_2 \beta s_1 y \delta \chi \equiv r_2 r_1 r_2 y s_1 y \epsilon. \]

Hence it suffices to prove that \( r_1 r_2 y s_1 y \equiv 5 \). By standardization this is equivalent to 24313 \( \equiv 42413 \), which follows from the following computation:
\[ 24313 \equiv 24313 \equiv 4213 \equiv 4213 \equiv 2431 \equiv 2431 \]
\[ \equiv 22431 \equiv 24231 \equiv 24231 \equiv 24231. \]

(ii) By (1) and (2),
\[ RyS = \alpha r_{i+1} \beta y \gamma s_{i-1} \delta \chi \equiv \alpha r_{i+1} y \beta \gamma s_{i-1} \delta \chi \equiv \alpha r_{i+1} y \gamma s_{i-1} \delta \chi \]
\[ \equiv \alpha r_{i+1} y \gamma s_{i-1} \epsilon. \]

Applying Lemma 4.6 then gives us
\[ RyS \equiv \alpha \gamma s_{i-1} r_{i+1} y s_i y \epsilon \chi. \]

Similarly, by Lemma 4.6 and (2),
\[ r_{i+1} R' S = r_{i+1} \alpha r_{i+1} \beta \gamma s_{i-1} \delta \chi \equiv r_{i+1} r_{i+1} \beta \gamma s_{i-1} \delta \chi \]
\[ \equiv \alpha r_{i+1} r_{i+1} \gamma s_{i-1} y \delta \chi \equiv \alpha \gamma s_{i-1} r_{i+1} y s_i y \epsilon \chi. \]
Hence it suffices to prove that \( r_i s_{i-1} r_{i+1} y s_i y \approx s_{i-1} r_{i+1} r_i s_r r_{i+1} y \). By standardization this is equivalent to \( 315424 \approx 153254 \), which follows from the following computation:

\[
315424 \equiv 315242 \equiv 312542 \equiv 132542 \equiv 135242 \equiv 135424 \\
\equiv 133542 \equiv 135342 \equiv 135324 \equiv 153524 \equiv 153254.
\]

This completes the proof of Lemma 4.4.

5. Right-Alignable Tableaux

In this section we give a new family of URTs, the right-alignable tableaux. Since superstandard and rectangular tableaux are right-alignable, we will in particular show that all tableaux in those two classes are URTs. Although we have assumed so far that tableaux are straight unless otherwise mentioned, in this section we relax that requirement. It will be useful to consider tableaux of more general shape.

**Definition 5.1.** Let \( \lambda = (\lambda_1, \ldots, \lambda_\ell) \) be a straight shape and let \( T : \lambda \to \mathbb{N} \) be a straight tableau of shape \( \lambda \). The right-justification of \( \lambda \) is the shape \( \{ (i, j + (\lambda_1 - \lambda_i)) : (i, j) \in \lambda \} \). The right-justification of \( T \) is the filling \( T_R : \lambda_R \to \mathbb{N} \) defined by \( T_R(i, j + (\lambda_1 - \lambda_i)) = T(i, j) \).

A tableau \( T \) is right-alignable if the right-alignment \( T_R \) strictly increases down columns and left-to-right across rows.

**Example 5.2.** The tableau

\[
T = \begin{array}{ccc}
1 & 2 & 3 \\
3 & 5 & \text{4}
\end{array}
\]

has right justification \( T_R = \begin{array}{ccc}
1 & 2 & 3 \\
3 & 5 & 4
\end{array} \).

Hence \( T \) is not right-alignable.

We will prove the following theorem in the next two sections.

**Theorem 5.3.** Every right-alignable tableau is a URT.

The following two corollaries follow easily from Theorem 5.3.

**Corollary 5.4.** Every superstandard tableau is a URT.

**Corollary 5.5.** Every rectangular tableau is a URT.

Theorem 5.3 is by no means sharp: there are URTs that are not right-alignable. For example, a minimal tableaux is right-alignable if and only if it is rectangular.

5.1. T-Compatibility. In this section we define a technical invariant for K-Knuth classes, T-compatibility, which will be used to show that right-alignable tableaux are URTs.

**Definition 5.6.** Given a word \( w \), let \( w^* \) denote the set of tuples \( (x, i) \), where \( x \) is the letter in \( w \) appearing at position \( i \). We say that \( (x, i) \) appears before (resp. after) \( (y, j) \) if \( i < j \) (resp. \( i > j \)).
For example, in 7556∗ the tuple (5, 2) appears before (5, 3) and after (7, 1). In general the set w∗ is identical to the word w, except that the same letters of w∗ are different if they occupy different positions. For clarity, we will abuse notation and identify a tuple (x, i) ∈ w∗ with its first component x. There is no risk of confusion unless the letter x appears twice in w∗, in which case we will distinguish the two by subscripts: x1 and x2, say.

**Definition 5.7.** Let λ be any shape and let T be an increasing tableau of shape λ. We say that a word w is T-compatible if there exists a function f : w∗ → λ such that

1. f is surjective,
2. the box f(x) has label x in T,
3. if f(x) is above f(y) then x appears after y in w∗, and
4. if f(x) is to the right of f(y) then x appears after y in w∗.

A straight tableau U is T-compatible if the row word row(U) is T-compatible.

The tableau T in Definition 5.7 can have any shape, and in particular could equal either of the following tableaux:

\[
\begin{array}{ccc}
1 & 3 & 4 \\
4 & 6 & 5 \\
\end{array}
\quad \quad \quad
\begin{array}{ccc}
1 & 4 & 7 \\
3 & 5 & \\
\end{array}
\]

At the same time, we require the tableau U in Definition 5.7 to have straight shape.

The requirements for the function f in Definition 5.7 could be reformulated as follows. If the letters of w∗ are ordered under the relation "y ≤ x if y appears before x in w∗", and if the boxes of T are ordered under the relation "α ≤ β if α is weakly southwest of β," then f satisfies the conditions of Definition 5.7 if and only if f is a surjective, monotonic map of posets.

**Example 5.8.** Let T = \([\begin{array}{ccc}
1 & 3 & 4 \\
4 & 6 & 5 \\
\end{array}]\). The word 35124 is T-compatible since the map f : 35124∗ → \([\begin{array}{ccc}
1 & 3 & 4 \\
4 & 6 & 5 \\
\end{array}]\) given by

\[3 \mapsto (2, 1), 5 \mapsto (2, 2), 1 \mapsto (1, 1), 2 \mapsto (1, 2), 4 \mapsto (1, 3)\]

is surjective, sends each letter to a box labeled with that letter, and respects the order of the letters in 35124∗. In this case 35124 = row(T), and it is true in general that any reading word for a tableau T is T-compatible: indeed, T-compatibility can be construed to generalize the reading word. Hence the word 31524 = col(T) is also T-compatible since the map f : 31524∗ → λ given by

\[3 \mapsto (2, 1), 1 \mapsto (1, 1), 5 \mapsto (2, 2), 2 \mapsto (1, 2), 4 \mapsto (1, 3)\]

satisfies the necessary properties. However, the word 31254 is not T-compatible. There is only one map 31254∗ → λ sending each letter to a box labeled with that letter: under such a map, 5 → (2, 2) and 2 → (1, 2). But (1, 2) is above (2, 2) while 5 comes before 2 in 31254∗.

The reader may have noticed that the three words we gave in example 5.8 were K-Knuth equivalent, but only the first and second were T-compatible: applying the K-Knuth move 31524 → 31254 fails to preserve T-compatibility. This failure reflects the fact that the shape (2, 2, 1) is missing a box in the bottom right corner.
More specifically, in order for $T$-compatibility to be an invariant for $K$-Knuth classes the tableau $T$ must contain enough boxes to be closed under forming certain hooks.

Recall that the $(i, j)$th entry of a tableau lies in the $i$th row and $j$th column.

**Definition 5.9.** A shape $\lambda$ is *northwest-hook-closed* if it is closed under forming northwest hooks: whenever $x = (i_1, j_1)$ and $y = (i_2, j_2)$ are boxes of $\lambda$ such that $i_1 \geq i_2$ and $j_1 \leq j_2$ then $\lambda$ contains the boxes $(r, j_1)$ for $i_1 \geq r \geq i_2$ and $(i_2, c)$ for $j_1 \leq c \leq j_2$.

A shape $\lambda$ is *northeast-hook-closed* if it is closed under forming northeast hooks: whenever $x = (i_1, j_1)$ and $y = (i_2, j_2)$ are boxes of $\lambda$ such that $i_1 \leq i_2$ and $j_1 \leq j_2$ then $\lambda$ contains the boxes $(r, j_2)$ for $i_1 \leq r \leq i_2$ and $(i_1, c)$ for $j_1 \leq c \leq j_2$.

A shape $\lambda$ is *southeast-hook-closed* if it is closed under forming southeast hooks: whenever $x = (i_1, j_1)$ and $y = (i_2, j_2)$ are boxes of $\lambda$ such that $i_1 \geq i_2$ and $j_1 \leq j_2$ then $\lambda$ contains the boxes $(r, j_2)$ for $i_1 \geq r \geq i_2$ and $(i_1, c)$ for $j_1 \leq c \leq j_2$.

Definition 5.9 generalizes hook closure as defined in [2, section 5]: Buch and Samuel’s hook closure is the same as our northeast-hook closure.

**Example 5.10.** Of the shapes below, only the first is northwest-hook-closed, only the second is northeast-hook-closed, and only the third is southeast-hook-closed. The fourth satisfies none of the three hook-closure properties we defined.

Lemma 5.11 will concern shapes that are northwest- and southeast-hook closed; such shapes are the reflections of skew shapes across a vertical axis, like the examples shown below.

Lemma 5.12 will concern shapes that are northeast-, northwest-, and southeast-hook closed; such shapes are the reflections of straight shapes across a vertical axis.
Lemma 5.11. Let $\lambda$ be a northwest- and southeast-hook-closed shape and let $T$ be an increasing tableau of shape $\lambda$. If $w \equiv w'$ and $w$ is $T$-compatible then $w'$ is $T$-compatible.

Proof. It suffices to assume that $w'$ differs from $w$ by one $K$-Knuth move. Let $f : w^* \to \lambda$ be the surjection showing that $w$ is $T$-compatible. For each $K$-Knuth move we will construct a function $f' : (w')^* \to \lambda$ showing that $w'$ is $T$-compatible and differing from $f$ only on the modified letters. Throughout the proof, we will write indices on letters that are the same but that appear in different positions in $w^*$.

If $w'$ is obtained from $w$ by replacing $p_1$ with $p_2p_3$, then define $f'(p_2) = f'(p_3) = f(p_1)$.

If $w'$ is obtained from $w$ by replacing $p_1p_2$ by $p_3$ then it suffices to show that $f(p_1) = f(p_2)$, in which case we may define $f'(p_3) = f(p_1)$. For the sake of contradiction, assume that $f(p_1) = (i_1, j_1) \neq f(p_2) = (i_2, j_2)$. Since $A$ increases along rows and columns, either $f(p_1)$ is strictly northeast of $f(p_2)$ or $f(p_2)$ is strictly northeast of $f(p_1)$. In either case, using the fact that $T$ is northwest- and southeast-hook closed, let $u$ be a letter of $w^*$ with $f(u) = (i_1, j_2)$ and let $v$ be a letter of $w^*$ with $f(v) = (i_2, j_1)$.

The letters $u$ and $v$ must lie between $p_1$ and $p_2$ in $w^*$, contradicting the assumption that the letters $p_1$ and $p_2$ are consecutive.

If $w'$ is obtained from $w$ by replacing $p_1q_1p_2$ by $q_2p_3q_3$, then it suffices to show that $f(q_1) = f(q_2)$, in which case we may define $f'(p_3) = f(q_1)$. Arguing as in the previous case, we assume that $f(p_1) \neq f(p_2)$ and deduce that there are two letters between $p_1$ and $p_2$, a contradiction.

If $w'$ is obtained from $w$ by replacing $xyz$ with $zxy$, provided $x < y < z$, then it suffices to show that $f(z)$ is strictly southeast of $f(x)$, in which case we may define $f'(x) = f(x)$ and $f'(z) = f(z)$. It is impossible for $f(z)$ to appear weakly southwest of $f(x)$ because $z$ comes after $x$ in $w^*$, and it is impossible for $f(z)$ to appear weakly northwest of $f(x)$ because $x < z$. Assume, then, for the sake of contradiction, that $f(z) = (i_2, j_2)$ is weakly northeast of $f(x) = (i_1, j_1)$. Since there are no letters between $x$ and $z$ in $w^*$, $f(z)$ must be directly right of $f(x)$, so that $i_2 = i_1$ and $j_2 = j_1 + 1$.

The box $f(y) = (i_3, j_3)$ cannot be weakly northwest of $f(x)$ or weakly southeast of $f(z)$ because $x < y < z$. And since $y$ appears after $x$ and $z$ in $w^*$, the box $f(y)$ must be directly above $f(z)$, so that $i_3 = i_2 - 1$ and $j_3 = j_2$.

Let $u$ be a letter of $w^*$ such that $f(u) = (i_3, j_1)$, which exists because $T$ is northwest-hook closed.

Then $u < z$ and $u$ lies between $x$ and $y$ in $w^*$, a contradiction.

Analogous arguments apply to the other three $K$-Knuth moves. □
5.2. Proof of Theorem 5.3. We have not yet specified why $T$-compatibility is useful: if, in addition to the hypotheses of Lemma 5.11, $T$ is northeast-hook closed, then there is at most one tableau which is $T$-compatible. By Lemma 5.11 that tableau must be a URT.

Lemma 5.12. Let $\lambda$ be a northeast-, northwest-, and southeast-hook-closed shape. If $T$ is a tableau of shape $\lambda$, then there is at most one straight tableau $U$ that is $T$-compatible.

Proof. Let $f : \text{row}(U) \to T$ denote the surjective map certifying that $U$ is $T$-compatible. Given a shape $\lambda$ let $\text{row}_i(\lambda)$ denote the $i$th row of $\lambda$; given a tableau $T$ let $\text{row}_i(T)$ denote the subword of $\text{row}(T)$ consisting of the letters in the $i$th row of $T$.

The proof consists of showing that $f(\text{row}_i(U)^*) = \text{row}_i(T)$, meaning that the letters in the $i$th row of $U$ are exactly the letters in the $i$th row of $T$. Hence $U$ is unique among straight tableaux that are $T$-compatible. We first prove an easy claim.

Claim 5.13. If $w$ is $T$-compatible and $x$ is the first letter of a strictly decreasing subword of $w^*$ of length $k$, then $f(x)$ appears in or below the $k$th row of $\lambda$.

Proof. Write the decreasing subword as $x = x_1 > x_2 > \cdots > x_k$. The box $f(x_{i+1})$ cannot be weakly southeast of the box $f(x_i)$ because $x_{i+1} < x_i$, and $f(x_{i+1})$ cannot be weakly southwest of $f(x_i)$ because $x_{i+1}$ appears after $x_i$ in $w^*$. Hence $f(x_{i+1})$ is strictly north of $f(x_i)$.

To finish the proof of lemma 5.12, we’ll show by backward induction on $i$ that $f(\text{row}_i(U)^*) = \text{row}_i(T)$. Claim 5.13 implies that $T$ has at least as many rows as $U$. And the rightmost column of $T$ gives a strictly decreasing subword of $U$ with maximal length, meaning that $T$ and $U$ have the same number of rows. Call that number $m$. For any box $\alpha \in \text{row}_m(U)$ with $f(x) = \alpha$, the letter $x$ appears as the first letter of a decreasing subword of $\text{row}(U)^*$ of length $m$, and hence lies in the $m$th row of $U$. Then $f(\text{row}_m(U)^*) = \text{row}_m(T)$.

Now assume that $f(\text{row}_i(U)^*) = \text{row}_i(T)$. Suppose $\alpha \in \text{row}_{i-1}(U)$, and let $x$ be any letter of $w^*$ for which $f(x) = \alpha$. If $\alpha$ does not lie above a box of $\text{row}_i(T)$, then $x$ is less than every entry of $\text{row}_j(U)^*$ for $i \leq j \leq m$. If $\alpha$ lies directly above a box of $\text{row}_i(T)$, then for every $j$ with $i \leq j \leq m$ the letter $x$ comes after a letter $y \in \text{row}_j(T)^*$ that is greater than $x$. In either case we conclude that $x$ cannot lie below the $(i-1)$th row of $U$, so that $x \in \text{row}_{i-1}(U)^*$.

Under what circumstances does the tableau $U$ mentioned in Lemma 5.12 actually exist? The proof of Lemma 5.12 shows that if a tableau $T$ is closed under forming northwest, northeast, and southeast hooks and if $U$ is $T$-compatible then $T$ and $U$ have the same entries in each row. Hence $T$ and $U$ have the same number of entries in each row and differ only by the relative positions of each row. The rows of $U$ must be aligned along their left edges because $U$ is straight, and the rows of $T$ must be aligned along their right edges because, as Example 2.1 mentioned, $T$ is the reflection of a straight shape about a vertical axis. In other words, we may obtain $T$ from $U$ by right-aligning the rows of $U$. In this way we complete the proof of Theorem 5.3.

Proof of Theorem 5.3. If $T_R$ is an increasing tableau then we may speak of $T_R$-compatible tableaux. In particular $\text{row}(T)$ is $T_R$-compatible, as we may show by
defining the map \( f : \text{row}(T)^* \to \lambda_R \) by \( f(T(i,j)) = (j + (\lambda_1 - \lambda_i)) \), where \( T \) has shape \( \lambda = (\lambda_1, \ldots, \lambda_\ell) \). If \( T \) is \( K \)-Knuth equivalent to some other tableau \( T' \) then \( T' \) is \( T_R \)-compatible by Lemma 5.11 and so \( T' = T \) by Lemma 5.12. Hence \( T \) is a URT.

5.3. Shapes of Non-URTs. Theorem 5.3 is not sharp: there are many URTs, most notably non-rectangular minimal tableaux, that do not satisfy the theorem’s hypotheses. However, the corollary that every rectangular tableau is a URT is almost sharp, as we will show in this section: every straight shape \( \lambda \) that is not a rectangle, except for \( \square \), has an increasing filling \( T \) that is not a URT.

Proposition 5.14. If \( \lambda \) is a straight shape that is not a rectangle or \( \square \) then there is an increasing tableau of shape \( \lambda \) that is not a URT.

Proof. Write \( \lambda = (\lambda_1, \ldots, \lambda_\ell) \). The proof will consist of expanding the prototypical non-URTs \( \square \equiv \square \) from Example 2.11.

Suppose \( \lambda \) has two parts, \( \lambda_2 = 1 \), and \( \lambda_1 \geq 3 \). Row inserting 2 into the tableau

\[
\begin{array}{cccccc}
1 & 3 & 4 & \cdots & \lambda_1 + 1 \\
3 & & & & \\
\end{array}
\]

one or two times yields the following respective tableaux, which are equivalent because \( 2 \equiv 22 \):

\[
\begin{array}{cccccc}
1 & 2 & 4 & \cdots & \lambda_1 + 1 \\
3 & & & & \\
\end{array}
\equiv
\begin{array}{cccccc}
1 & 2 & 4 & \cdots & \lambda_1 + 1 \\
3 & 4 & & & \\
\end{array}
\]

For the remaining cases we require a family of tableaux parameterized by two natural numbers, \( j \) and \( k \). Consider the tableau

\[
T = \begin{array}{cccccc}
1 & \cdots & k-1 & k & k+2 & k+3 & \cdots & j+1 \\
2 & \cdots & k & & & & \\
\end{array}
\]

\[ (j \geq 4, k \geq 2) \]

Row inserting \( k+1 \) into \( T \) yields the tableau

\[
T_{j,k} = \begin{array}{cccccc}
1 & \cdots & k-1 & k & k+1 & k+3 & k+4 & \cdots & j+1 \\
2 & \cdots & k & & & k+2 & & \\
\end{array}
\]

of shape \((j,k)\), and row inserting \( k+1 \) into \( T_{j,k} \) yields the tableau

\[
T'_{j,k} = \begin{array}{cccccc}
1 & \cdots & k-1 & k & k+1 & k+3 & k+4 & \cdots & j+1 \\
2 & \cdots & k & & k+2 & & k+3 \\
\end{array}
\]

of shape \((j,k+1)\).

Continuing with our case analysis, suppose \( \lambda \) has two parts, \( \lambda_1 > \lambda_2 \), and \( \lambda_2 \geq 2 \). If \( \lambda_1 > \lambda_2 + 1 \) then the tableau \( T_{\lambda_1,\lambda_2} \) has shape \( \lambda \) and is not a URT. If \( \lambda_1 = \lambda_2 + 1 \) then \( T'_{\lambda_1,\lambda_2} \) has shape \( \lambda \) and is not a URT.

We will handle the remaining cases, which concern tableaux having three or more parts, using the following observation. Given a tableau \( T \) of shape \( \lambda = (\lambda_1, \ldots, \lambda_\ell) \), let \( T^{(r,s)} \) denote the restriction of \( T \) to the rows \( r, r+1, \ldots, s \) of \( \lambda \).
Claim 5.15. If $T$ and $T'$ are two tableaux that differ only in the $i$th and $(i+1)$th rows, and if $T^{(i,i+1)} \equiv (T')^{(i,i+1)}$, then $T \equiv T'$.

Proof. The row words $\row(T)$ and $\row(T')$ may be connected by K-Knuth moves that only use letters from the $i$th and $(i+1)$th rows.

To continue with the proof of Proposition 5.14, suppose $\lambda$ has more than two parts and there is an index $i \geq 1$ such that $\lambda_i \geq \lambda_{i+1}+2$. Given a tableau $T$, let $T[n]$ denote the tableau formed by increasing the entries of $T$ by $n$: formally, $T[n](i,j) = T(i,j) + n$. There exists a tableau $T$ of shape $\lambda$ such that $T^{(i,i+1)} = T_{\lambda_i,\lambda_{i+1}}[n]$ for some $n$. Let $T'$ denote the tableau of shape $(\ldots, \lambda_i, \lambda_{i+1} + 1, \lambda_{i+2}, \ldots)$ that has the same labels as $T$ in every row except the $i$th and $(i+1)$th, and such that $(T')^{(i,i+1)} = T'_{\lambda_i,\lambda_{i+1}}[n]$. Then $T \equiv T'$ but $T \neq T'$, as desired.

By constructing an analogous argument using the transposes $T'_{j,k}$ and $(T')_{j,k}'$, we can show that the proposition holds if either $\lambda$ or $\lambda'$ has more than one entry in the second row and has two consecutive rows of different lengths. This exhausts all cases.

6. Hook-Shaped Tableaux

In this section, we examine a class of tableaux known as the hook-shaped tableaux and characterize which hook-shaped tableaux are URTs.

Definition 6.1. A straight shape $\lambda$ is hook-shaped if $\lambda = (m,1^n)$ for some $m \geq 1$ and $n \geq 0$. An increasing tableau $T$ of shape $\lambda$ is hook-shaped if $\lambda$ is hook-shaped.

Of the tableaux below, the tableau on the left is hook-shaped and the two tableaux on the right are not.

![Hook-Shaped Tableaux](https://via.placeholder.com/150)

Theorem 6.2. Suppose $T$ is a hook-shaped, increasing tableau such that

1. the first row of $T$ is labeled $1, 2, \ldots, n$, and
2. the first column is labeled $1, \ell_2, \ell_3, \ldots, \ell_m$, where $\ell_{i+1} = \ell_i + 1$ for $2 \leq i \leq m - 1$ and $\ell_2 \geq 2$.

Then $T$ is a URT.

Proof. We proceed by induction on $n$. Given a tableau $T$ satisfying the hypotheses of the claim, call $n$ the rightmost entry of $T$. If $n = 1$, the tableau $T$ is a URT by virtue of being rectangular. For the induction step, assume that every tableau with rightmost entry $n$ is a URT. We’ll deduce a contradiction from the existence of a tableau $T$ with rightmost entry $n + 1$ that is not a URT. Since minimal tableaux are URTs, we may assume that $\ell_2 \geq 3$. Let $T'$ be a tableau distinct from $T$ and lying in the same $K$-Knuth equivalence class. Then the skew tableaux $T' |_{[2,n]}$ and $T'' |_{[2,n]}$ are $K$-Knuth equivalent by Proposition 2.39. Perform $K$-jdt on $T |_{[2,n]}$ and $T'' |_{[2,n]}$ at position $(1,1)$, that is, at the position vacated by 1 upon restriction to the subalphabet $[2,n]$. Because the first rows of $T |_{[2,n]}$ and $T'' |_{[2,n]}$ are labeled consecutively and $\ell_2 \geq 3$, performing $K$-jdt has the effect of translating the first row of each tableaux one box to the left.

Let $S$ and $S'$ denote the tableaux resulting from $T |_{[2,n]}$ and $T'' |_{[2,n]}$, respectively. Since $S$ and $S'$ are distinct and $K$-Knuth equivalent we conclude that $S$ is not
a URT. But the tableau formed from $S$ by decreasing each of its entries by 1 has rightmost entry $n$ and satisfies the hypotheses of the claim, contradicting the induction hypothesis.

Example 6.3. These hook-shaped tableaux satisfy the hypotheses of Theorem 6.2 and are therefore URTs:

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
3 & 4 & 5 \\
\end{array}
\quad
\begin{array}{cccc}
1 & 2 & 3 & 4 & 5 \\
3 \\
5 \\
\end{array}
\quad
\begin{array}{cccc}
1 & 2 & 3 & 4 & 5 \\
5 \\
6 \\
\end{array}
\]

These hook-shaped tableaux do not satisfy the hypotheses of Theorem 6.2:

\[
\begin{array}{cccc}
1 & 2 & 3 & 5 \\
4 \\
\end{array}
\quad
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
2 & 4 \\
3 \\
5 \\
\end{array}
\quad
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
2 \\
3 \\
5 \\
\end{array}
\]

Next, we will show that Theorem 6.2 is sharp.

Claim 6.4. Let $T$ be a hook-shaped, increasing initial tableau with the following properties:

1. the first row of $T$ is labeled $1, \ell_2, \ell_3, ..., \ell_s$,
2. the first column of $T$ is labeled $1, m_2, m_3, ..., m_t$, and
3. at least one of $\ell_2, ..., \ell_s$ and $m_2, ..., m_t$ is a non-consecutive sequence.

Then $T$ is not a URT.

Proof. We consider two cases. For simplicity, we assume without loss of generality that every tableau $T$ in consideration is an initial tableau. Also for simplicity, we make a distinction between the first row and the first column of the tableaux in consideration, but this distinction is arbitrary due to the fact that transposes of non-URTs are non-URTs.

For simplicity of notation, $[r(1, \ell_2, \ell_3, ..., \ell_s), c(1, m_2, m_3, ..., m_t)]$ will denote the hook-shaped tableau whose first row has entries $1, \ell_2, \ell_3, ..., \ell_s$ and whose first column has entries $1, m_2, m_3, ..., m_t$:

\[
\begin{array}{cccc}
1 & \ell_2 & \ell_3 & \ell_s \\
m_2 \\
m_3 \\
\vdots \\
m_t \\
\end{array}
\]

We will often show that such a hook-shaped tableau is equivalent to a tableau with the same outer hook and one extra entry in position $(2,2)$. Tableaux of that shape will be denoted $[r(1, \ell_2, \ell_3, ..., \ell_s), c(1, m_2, m_3, ..., m_t), n]$, where $n$ is the new entry in $(2,2)$:

\[
\begin{array}{cccc}
1 & \ell_2 & \ell_3 & \ell_s \\
m_1 & k \\
m_2 \\
\vdots \\
m_t \\
\end{array}
\]
Case 1. \( T \) has the following properties:

1. The first row of \( T \) is of the form \( 1, \ell_2, \ldots, \ell_s \), where \( \ell_2, \ldots, \ell_s \) is any strictly increasing sequence, and
2. The first column of \( T \) is of the form \( 1, 2, m_3, \ldots, m_t \), where \( m_3, \ldots, m_t \) is a non-consecutive sequence (such that \( T \) is initial).

It will suffice to consider the tableau \([r(1, k), c(1, 2, \ldots, k - 1, k + p)]\), where \( k = \ell_1 \geq 3 \) and \( p \geq 1 \). This tableau is K-equivalent to \([r(1, k), c(1, 2, \ldots, k - 1, k + p)], k + p\]. The equivalence can be seen easily and directly by performing \( K \)-Knuth moves on the row words of the two tableaux.

The above simplified example of a hook-shaped non-URT generalizes to all initial tableaux with the properties specified in Case 1 via row and column insertion, explained below.

If we row-insert some sequence of letters greater than \( k \) to the first row of each tableau, the equivalence of the two tableaux does not change. Hence we have
\[
[r(1, k, k + 1, \ldots, k + (p - 1), \ell_{i+p}, \ldots, \ell_s), c(1, 2, \ldots, k - 1, k + p)]
\]
\[
\equiv [r(1, k, k + 1, \ldots, k + (p - 1), \ell_{i+p}, \ldots, \ell_s), c(1, 2, \ldots, k - 1, k + p), k + p].
\]

Similarly, we can column-insert integers less than \( k + 1 \) into the first column of each tableau and maintain equivalence. Thus we can obtain any sequence of integers between 1 and \( k \) in the first row. For example, let the sequence we want to obtain between 1 and \( k \) be \( \ell_1, \ldots, \ell_{i-1} \). We can first column-insert \( \ell_{i-1} - 1 \) into the first column. This has the effect of shifting everything to the right of 1 in the first row one box to the right, and inserting \( \ell_{i-1} \) into the position \((1, 2)\). We can repeat this process, inserting \( \ell_{i-2} - 1 \) into the first column, and so on, until we have the following equivalence:
\[
[r(1, \ell_1, \ldots, \ell_{i-1}, k, k + 1, \ldots, k + (p - 1), \ell_{i+p}, \ldots, \ell_s), c(1, 2, \ldots, k - 1, k + p)]
\]
\[
\equiv [r(1, \ell_1, \ldots, \ell_{i-1}, k, k + 1, \ldots, k + (p - 1), \ell_{i+p}, \ldots, \ell_s), c(1, 2, \ldots, k - 1, k + p), k + p].
\]

Finally, we can column-insert any sequence of integers larger than \( k + 1 \) in the first column. Hence we have the equivalence
\[
[r(1, \ell_1, \ldots, \ell_{i-1}, k, k + 1, \ldots, k + (p - 1), \ell_{i+p}, \ldots, \ell_s), c(1, 2, \ldots, k - 1, k + p, m_j, \ldots, m_t)]
\]
\[
\equiv [r(1, \ell_1, \ldots, \ell_{i-1}, k, k + 1, \ldots, k + (p - 1), \ell_{i+p}, \ldots, \ell_s), c(1, 2, \ldots, k - 1, k + p, m_j, \ldots, m_t), k + p].
\]

This completes the proof because the hook-shaped tableau above (assumed to be initial based on our construction) represents the general version of the tableau with properties specified in Case 1. Hence any tableau of that form is not a URT.

Case 2. \( T \) has the following properties:

1. The first row of \( T \) is of the form \( 1, 2, \ldots, n \).
2. The first column of \( T \) is of the form \( 1, m_1, \ldots, m_t \), where the sequence \( m_1, \ldots, m_t \) is non-consecutive.

Note that the case where \( m_1 = 2 \) was already covered by Case 1.

It will suffice to consider the tableau \([r(1, 2, \ldots, k), c(1, k - 1, k + p)]\). We show by induction that this is not a URT for all \( k \geq 3 \). When \( k = 3 \) the tableau is not
a URT due to the below equivalence, which can be easily checked by performing
K-Knuth equivalence moves on the row words of two tableaux:
$$[r(1, 2, 3), c(1, 2, 4)] \equiv [r(1, 2, 3), c(1, 2, 4), 4].$$

Assume that for some $k$, the following tableaux are equivalent:
$$[r(1, 2, \ldots, k), c(1, k - 1, k + p)] \equiv [r(1, 2, \ldots, k), c(1, k - 1, k + p), k + p].$$

We show that
$$[r(1, 2, \ldots, k), c(1, k - 1, k + (p + 1))] \equiv [r(1, 2, \ldots, k), c(1, k - 1, k + (p + 1)), k + (p + 1)]$$
by examining K-Knuth equivalence moves on their row words.

First, consider the row word of the tableau on the lefthand side of the above equivalence. We have, with commas separating distinct letters for ease of reading:

$$k + (p + 1), k, 1, 2, \ldots, k + 1 \equiv 1, k + (p + 1), k, 2, \ldots, k + 1.$$

In the word above, consider everything to the right of 1. Standardize it to obtain the word

$$k + 1, k - 1, \ldots, k.$$

By assumption, this word is K-Knuth equivalent to

$$k + 1, k - 1, k + 1, \ldots, k.$$

Thus

$$1, k + (p + 1), k, 2, \ldots, k + 1 \equiv 1, k + (p + 1), k, 2, \ldots, k + 1.$$

Since these two words respectively insert into the two tableaux depicted above, the tableaux are equivalent.

In order to see how this simplified example of a hook-shaped non-URT generalizes to all tableaux with the properties specified in Case 1, we again utilize row and column insertion. We have the equivalence

$$[r(1, 2, \ldots, k), c(1, k - 1, k + p)] \equiv [r(1, 2, \ldots, k), c(1, k - 1, k + p), k + p].$$

As in the first case, row- and column-inserting letters into each of the tableaux will not change their equivalence. Thus the following equivalence holds:

$$[r(1, 2, \ldots, k, k + 1, \ldots, k + (p - 1), k + p, k + (p + 1), \ldots, n),$$

$$c(1, m_1, \ldots, m_t, k - 1, k + p, m_{t+3}, \ldots, m_n)]$$

$$\equiv [r(1, 2, \ldots, k, k + 1, \ldots, k + (p - 1), k + p, k + (p + 1), \ldots, n),$$

$$c(1, m_1, \ldots, m_t, k - 1, k + p, m_{t+3}, \ldots, m_n), k + p].$$

Clearly, the hook-shaped tableau above is a general tableau (again assumed to be initial) such that the first row has entries $1, 2, \ldots, n$, and the first column has non-consecutive entries. Hence, every tableau of this form is not a URT. □

An important takeaway from the above proof is that a hook-shaped initial tableau

$$\begin{array}{ccc}
1 & a_2 & \ldots & a_n \\
b_2 & & & \\
& \vdots & & \\
b_n & & & \\
\end{array}$$
is equivalent to

\[ \begin{array}{c c c c c}
1 & a_2 & \ldots & a_n \\
b_2 & a_1 \\
\vdots \\
b_n \\
\end{array} \]

whenever \( a_i - a_{i-1} > 1 \).

7. Conjectures and Related Results

7.1. Sizes of Tableaux Classes. In the course of studying the \( K \)-Knuth equivalence relation on tableaux, we computed all equivalence classes of tableaux on \([n]\) for \( 0 \leq n \leq 7 \). We were unable to obtain asymptotic bounds on the size of \( K \)-Knuth equivalence classes, but they seem to grow at least as quickly as \( n! \).

Table 1. Sets of Initial Tableaux

<table>
<thead>
<tr>
<th>Alphabet Size</th>
<th>Initial Increasing Tableaux</th>
<th>( K )-Knuth Classes of Initial Tableaux</th>
<th>URTs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>87</td>
<td>79</td>
<td>71</td>
</tr>
<tr>
<td>5</td>
<td>849</td>
<td>620</td>
<td>459</td>
</tr>
<tr>
<td>6</td>
<td>11915</td>
<td>6036</td>
<td>3313</td>
</tr>
<tr>
<td>7</td>
<td>238405</td>
<td>70963</td>
<td>25904</td>
</tr>
</tbody>
</table>

Table 1 shows that the ratio of unique rectification classes of tableaux on \([n]\) to all \( K \)-Knuth classes of tableaux on \([n]\) decreases monotonically, and we expect the ratio to asymptotically tend to zero.

**Conjecture 7.1.** Let \( I_n \) denote the number of \( K \)-Knuth equivalence classes of initial tableaux on \([n]\), and let \( U_n \) denote the number of URTs on \([n]\). Then \( \lim_{n \to \infty} U_n/I_n = 0 \).

7.2. Composition of \( K \)-Knuth Classes of Tableaux.

**Proposition 7.2.** For every \( n \geq 2 \), there is an equivalence class of tableaux on \([2n]\) containing at least \( n! \) distinct tableaux.

**Proof.** The comments following the proof of Claim 6.4 have as a corollary that for every \( k = 2, 3, \ldots, n \) there is an equivalence
Let $T$ denote the tableau on the left, let $T'$ denote the tableau on the right, and let $w = \text{row}(T)$. A simple computation shows that $P(\text{row}(T)) = T$, and row inserting $2k - 2$ into $T$ shows that $T' = P(w(2k - 2))$. Hence $w = \text{row}(T) \equiv \text{row}(T') \equiv w(2k - 2)$.

It follows that if we Hecke insert any of the positive integers $2, 4, 6, \ldots, 2(n-1)$ in any order into the first row of $T$, the resulting tableau lies in the same equivalence class as $T$. Hence any tableau $T'$ with the following two properties is $K$-Knuth equivalent to $T$.

1. The first row and the first column of $T'$ agree with the first row and the first column of $T$, respectively.
2. Let $U$ denote the restriction of $T'$ to the $(2n-1) \times n$ rectangle consisting of the boxes $(i,j)$ satisfying $2 \leq i \leq 2n$ and $2 \leq j \leq n + 1$. Then the tableau $U$ uses the letters $4, 6, \ldots, 2n$.

There are at least $n!$ possibilities for the tableau $U$, as we saw in Section 2.7. Thus the class of tableaux $K$-Knuth equivalent to $T$ contains at least $n!$ tableaux. □

The process for generating tableaux described in the proof of Proposition 7.2 can produce many tableaux in the equivalence class of a hook-shaped tableaux. It suggests an important relationship between $K$-Knuth equivalence and row insertion, a relationship we have yet to fully understand. For completeness we include the following proposition.

**Proposition 7.3.** Let $T$ be an initial, hook-shaped tableaux whose first row is labeled $1, a_1, a_2, \ldots$ and whose second row is labeled $1, b_1, b_2, \ldots$. Let

$$A = \{a_i : i \geq 1, a_{i + 1} - a_i \geq 2\}$$

$$B = \{b_i : i \geq 1, b_{i + 1} - b_i \geq 2\}.$$ 

Then the set of straight tableaux that are $K$-Knuth equivalent to $T$ includes all the tableaux that may be obtained by making the following insertions into $T$, in any order: (1) row inserting elements of $A$ into the first row of $T$ and (2) column inserting elements of $B$ into the first column of $T$.

**Example 7.4.** The following tableaux are $K$-Knuth equivalent. They may all be obtained by row inserting $2$ and column inserting $3$.

\begin{align*}
&\begin{array}{cccc}
1 & 2 & 4 & 5 \\
2 & & & \\
3 & & & \\
5 & & & \\
6 & & & \\
\end{array} & \begin{array}{cccc}
1 & 2 & 4 & 5 \\
2 & 4 & & \\
3 & & & \\
5 & & & \\
6 & & & \\
\end{array} & \begin{array}{cccc}
1 & 2 & 4 & 5 \\
2 & 5 & & \\
3 & & & \\
5 & & & \\
6 & & & \\
\end{array} & \begin{array}{cccc}
1 & 2 & 4 & 5 \\
2 & 4 & 5 & \\
3 & & & \\
5 & & & \\
6 & & & \\
\end{array} \end{align*}

Proposition 7.3 does not give all tableaux in a class, however: the tableaux above are also equivalent to the tableau

\begin{align*}
&\begin{array}{cccc}
1 & 2 & 4 & 5 \\
2 & 4 & 5 & \\
3 & & & \\
5 & & & \\
6 & & & \\
\end{array},
\end{align*}

which cannot be obtained by making the described row and column insertions.
Consequently, the size of a $K$-Knuth equivalence class of tableaux on $[n]$ has no constant upper bound as $n$ increases. In fact, the number of standard tableaux in a $K$-Knuth equivalence class is also unbounded.

**Proposition 7.5.** For every $n > 0$, there exists a $K$-Knuth class containing at least $2^n$ standard tableaux.

**Proof.** We first consider the tableaux

$$T = \begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 7 \\ 6 \end{bmatrix} \quad \text{and} \quad T' = \begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 7 \end{bmatrix}.$$  

Note that $36731452 \equiv 36734152$ and the two words insert into $T$ and $T'$ respectively, so that $T \equiv T'$.

We will build, using $3 \times 3$ blocks, a standard tableau $U$ whose equivalence class contains at least $2^n$ standard tableaux. For each $i, j \geq 1$ with $i + j \leq n + 1$, we construct an interval $I_{i,j}$ of positive integers with the following properties:

1. The intervals are pairwise disjoint.
2. The union of all the $I_{i,j}$ is an interval of the form $[1, N]$ for some $N > 0$.
3. For every $i$ and $j$, the interval $I_{i,j}$ is after $I_{i+1,j}$ and before $I_{i,j+1}$.
4. If $i + j \leq n$, then $I_{i,j}$ has length 9; if $i + j = n + 1$, then $I_{i,j}$ has length 7.

Now we construct the tableaux $T_{i,j}$ with three rows and three columns as follows. If $i + j \leq n$, let $T_{i,j}$ be an arbitrary increasing tableau on $I_{i,j}$ with shape $(3, 3, 3)$ and no repeated entries (so $T_{i,j}$ is a $3 \times 3$ square). If $i + j = n + 1$, let $T_{i,j}$ (resp. $T_{i,j}'$) be the unique increasing tableau on $I_{i,j}$ such that $\text{std}(T_{i,j}) = T$ (resp. $\text{std}(T_{i,j}') = T'$).

Finally, we build a tableau $U$ with $3n$ rows and $3n$ columns such that the $3 \times 3$ block $T_{i,j}$ occupies the $(3i - 2)$th through $3i$th rows and the $(3j - 2)$th through $3j$th columns. The above construction guarantees that $U$ is a standard tableau.

For each $i$, define

$$R_i = \text{row}(T_{i,1})\text{row}(T_{i,2})\text{row}(T_{i,3}) \cdots \text{row}(T_{i,n-i})\text{row}(T_{i,n-i+1}),$$

$$R'_i = \text{row}(T'_{i,1})\text{row}(T'_{i,2})\text{row}(T'_{i,3}) \cdots \text{row}(T'_{i,n-i})\text{row}(T'_{i,n-i+1}).$$

The fact $T \equiv T'$ implies $\text{row}(T_{i,n-i+1}) \equiv \text{row}(T'_{i,n-i+1})$. This means that $R_i \equiv R'_i$. Observe that $w = R_n R_{n-1} \cdots R_1$ is a reading word of $U$. Let $S$ be a subset of $[n]$.

If $U'$ is the tableau obtained by replacing $T_{i,n-i+1}$ with $T'_{i,n-i+1}$ for all $i \in S$, then the word $w'$ obtained by replacing $R_i$ with $R'_i$ for all $i \in S$ is a reading word for $U'$. Hence it follows that $U \equiv U'$ for every choice of $S \subseteq [n]$. Since $[n]$ has $2^n$ subsets, the $K$-Knuth equivalence class contains at least $2^n$ standard tableaux. \qed

For instance, if $n = 3$, one possible construction of $U$ is

$$\begin{bmatrix} 1 & 2 & 3 & 10 & 11 & 12 & 19 & 20 & 23 \\ 4 & 5 & 6 & 13 & 14 & 15 & 21 & 22 & 25 \\ 7 & 8 & 9 & 16 & 17 & 18 & 24 \\ 26 & 27 & 28 & 35 & 36 & 39 \\ 29 & 30 & 31 & 37 & 38 & 41 \\ 32 & 33 & 34 & 40 \\ 42 & 43 & 46 \\ 44 & 45 & 48 \\ 47 \end{bmatrix}$$
7.3. Shapes of Tableaux. Which shapes appear in a $K$-Knuth class of tableaux? We initially suspected that each tableau class contains a minimum and maximum shape, ordering the shapes under inclusion, but the following class disproves our conjecture:

\[
\begin{array}{cccc}
1 & 2 & 5 \\
2 & 3 & 6 \\
3 & 4 & 5 \\
4 & 5 & \end{array}
\quad
\begin{array}{cccc}
1 & 2 & 5 \\
2 & 3 & 6 \\
3 & 4 & 5 \\
4 & 5 & \end{array}
\quad
\begin{array}{cccc}
1 & 2 & 5 \\
2 & 3 & 6 \\
3 & 4 & 5 \\
4 & 5 & \end{array}
\quad
\begin{array}{cccc}
1 & 2 & 5 \\
2 & 3 & 6 \\
3 & 4 & 5 \\
4 & 5 & \end{array}
\]

However, it seems – and we have not been able to find a counterexample – that if two shapes $\lambda_1 \subseteq \lambda_2$ appear among the shapes in an equivalence class, then every shape in the interval $[\lambda_1, \lambda_2]$ of Young’s lattice appears among the shapes in that class.

Conjecture 7.6. Let $T_1, T_2, \ldots, T_k$ be a $K$-Knuth class of straight tableaux and let $T_i$ have shape $\lambda_i$. Then the set $\Sigma = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ has the following property: if $\lambda_i, \lambda_j \in \Sigma$ then $[\lambda_i, \lambda_j] \subseteq \Sigma$.

This conjecture has been verified for $K$-Knuth classes on $[n]$ for $n \leq 7$.

7.4. Changes in Tableau Shape. Initially, we had conjectured that if two words $w$ and $w'$ differ by just one $K$-Knuth move, then the shapes of $P(w)$ and $P(w')$ differ by just one box. However, we found a counterexample: $5451342154 \equiv 54513422154$ because $2 \equiv 2$, but the two words insert into the following tableaux, respectively:

\[
\begin{array}{cccc}
1 & 2 & 4 & 5 \\
2 & 5 & \end{array}
\quad
\begin{array}{cccc}
1 & 2 & 4 & 5 \\
2 & 4 & 5 & \end{array}
\quad
\begin{array}{cccc}
1 & 2 & 4 & 5 \\
2 & 4 & 5 & \end{array}
\quad
\begin{array}{cccc}
1 & 2 & 4 & 5 \\
2 & 4 & 5 & \end{array}
\]

Acknowledgments

This research was conducted at the 2014 REU program at the University of Minnesota, Twin Cities, and was supported by NSF grant DMS-1148634. For their mentorship and support, we would like to thank Vic Reiner, Joel Lewis, and Gregg Musiker. We are especially grateful to Pavlo Pylyavskyy for introducing us to this problem and guiding us throughout the project.

References
