Journal of Algebraic Combinatorics, 19, 67–89, 2004

Box-Ball Systems and Robinson-Schensted-Knuth Correspondence

KAORI FUKUDA fukuda@math.kobe-u.ac.jp
Department of Mathematics, Kobe University, Rokko, Kobe 657-8501, Japan

Received June 6, 2001; Revised and Accepted March 12, 2003

Abstract. We study a box-ball system from the viewpoint of combinatorics of words and tableaux. Each state of the box-ball system can be transformed into a pair of tableaux (P, Q) by the Robinson-Schensted-Knuth correspondence. In the language of tableaux, the P-symbol gives rise to a conserved quantity of the box-ball system, and the Q-symbol evolves independently of the P-symbol. The time evolution of the Q-symbol is described explicitly in terms of the box-labels.

Keywords: box-ball system, Robinson-Schensted-Knuth correspondence, soliton cellular automaton, young tableau, Knuth equivalence

1. Introduction

The box-ball system (BBS), introduced in [6, 8], is a class of soliton cellular automata (ultra-discrete integrable systems). On this subject, remarkable progress has been made in connection with the discretization of nonlinear integrable systems [7, 9], and also with the crystal theory of representations of quantum algebras [1, 3]. In this paper, we study the box-ball system from the viewpoint of combinatorics of words and tableaux. Our discussion is based on the fact that each state of the BBS can be identified with a pair of tableaux (P, Q) by means of the Robinson-Schensted-Knuth (RSK) correspondence. The main points of this paper are as follows:

- The P-symbol provides a conserved quantity under the time evolution of BBS.
- The Q-symbol evolves independently of the P-symbol; the time evolution of the Q-symbol can be described combinatorially in terms of the box-labels.

The second statement implies that *equivalent* states (which have the same Q-symbol) evolve similarly, giving rise to equivalent states after any number of steps.

This paper is organized as follows. In Section 2, we review some necessary facts from combinatorics of words and tableaux; the bi-word defined in Section 2 plays a crucial role in this paper. In Section 3, we consider a *standard* version of the BBS, and formulate our main results in terms of the standard BBS. Section 4 is devoted to the proofs of the main results which will be introducecd in Section 3 for the standard BBS. In Section 5, we consider two generalizations of the standard BBS, and extend the results in Section 3 to those cases. The final section is devoted to a summary with examples.

We remark that there is another way due to Torii et al. [10] to construct conserved quantities for the BBS by the Robinson-Schensted correspondence. Their procedure and result, however, are essentially different from those we are going to discuss below.

2. Preliminaries

In this section we recall from the textbook of Fulton [2] (or Knuth [4]) some fundamental facts on combinatorics of words and tableaux, which we will freely use throughout this paper.

2.1. *Tableau word*

A *Young diagram* (a) is a finite collection of boxes, arranged in left-justified rows, with a weakly decreasing number of boxes in each row. We usually identify a partition, say $\lambda = (\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_l \geq 0)$, with the corresponding diagram. A *Young tableau* (b), or simply *tableau*, is a way of putting an integer in each box of a Young diagram that is weakly increasing across each row and strictly increasing down each column (*column-strict tableau* in the terminology of Macdonald [5]). We say that λ is the *shape* of the tableau. In the figure below, each shape is as follows; $(a) : \lambda = (4, 3, 1)$, $(b) : \lambda = (4, 3, 2)$, $(c) : \lambda = (3, 2, 1)$. A *standard tableau* (c) is a tableau in which the entries are numbers from 1 to n, each occurring once. See the figure below.

$$(a)\ \begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & \\ \cline{1-3} \\ \cline{1-1} \end{array} \qquad (b)\ \begin{array}{|c|c|c|c|}\hline 1 & 3 & 3 & 4 \\ \hline 2 & 4 & 5 \\ \cline{1-3} 3 & 5 \\ \cline{1-2} \end{array} \qquad (c)\ \begin{array}{|c|c|c|}\hline 1 & 2 & 4 \\ \hline 3 & 5 \\ \cline{1-2} 6 \\ \cline{1-1} \end{array}$$

We now recall the algorithm of *bumping* (*row-bumping*, or *row-insertion*), for constructing a new tableau from a tableau by inserting an integer.

The rule of bumping $\underline{T \leftarrow i}$ (for inserting an integer i in a tableau T): If there are no integers larger than i in the first row, add a new empty box at the right end, and put i in it. Otherwise, among the integers larger than i, find the leftmost one, say j, and put i in the box by bumping j out (i.e., replace j with i). Then, insert j, the bumped number, into the second row in the same way. Repeat this procedure until the bumped number can be put in a new box at the right end of the row.

Given a *word* (sequence of numbers) $w = w_1 w_2 \ldots w_n$, we define the *tableau* $\mathrm{Tab}(w)$ *of* w by bumping the entries of w from left to right, in the empty tableau $\emptyset$: $\mathrm{Tab}(w) = (\cdots((\emptyset \leftarrow w_1) \leftarrow w_2) \leftarrow \cdots) \leftarrow w_n$. Conversely, given a tableau T, we define the *word* $\mathrm{W}(T)$ *of* T by reading the entries of T *from left to right and bottom to top* (see figure 1).

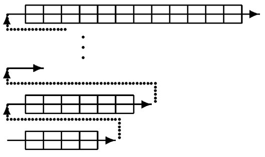


Figure 1. Reading route of a tableau word $\mathrm{W}(T)$.

Note that Tab(W(T)) $= T$. We say that a word w is a *tableau word* if it is the word of a tableau (see the example below).

$$w = 55137271314532$$

$$\text{Bumping}\ \Big\downarrow$$

$$T = \mathrm{Tab}(w) = \begin{array}{|c|c|c|c|c|}\hline 1&1&1&2&5\\ \hline 2&3&3&7\\ \cline{1-4} 3&4&7\\ \cline{1-3} 5&5\\ \cline{1-2}\end{array}$$

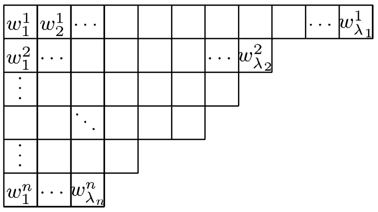
The word of tableau T : $\mathrm{W}(T) = 55347233711125$

Any tableau word w can be expressed in the form

$$w = w_1^n w_2^n \dots w_{\lambda_n}^n w_1^{n-1} \dots w_{\lambda_{n-1}}^{n-1} \dots w_1^1 \dots w_{\lambda_1}^1,$$

where $\lambda = (\lambda_1, \dots, \lambda_n)$ $(\lambda_1 \geq \dots \geq \lambda_n)$ is the *shape* of w and $w_j^i \leq w_{j+1}^i$, $w_j^i < w_j^{i+1}$ (see the figure below.) We remark that there is a bijection between the tableaux and the tableau words.

Tab(w) =



2.2. Knuth equivalence

We next describe the bumping algorithm in the language of *words*. The basic rule is given by

$$(u\, x'\, v)\, x \longrightarrow x'\, u\, x\, v \qquad (u \leq x < x' \leq v). \tag{1}$$

Here, x and x' are two numbers, and u and v are weakly increasing words; inequality $u \leq v$ means that every letter in u is smaller than or equal to every letter in v. In this expression, x stands for the number to be inserted into the row $(ux'v)$, and x' for the number to be bumped out from the row. This rule of bumping is decomposed into a sequence of rearrangements of three numbers of the following two types:

$$yzx \longrightarrow yxz \qquad (x < y \leq z), \tag{2}$$
$$xzy \longrightarrow zxy \qquad (x \leq y < z). \tag{3}$$

These two transformations, as well as their inverses, are called *elementary Knuth transformations*.

Definition 1 We call two words w and w' *Knuth equivalent* if they can be transformed into each other by a sequence of elementary Knuth transformations. We write $w \approx w'$ to denote that words w and w' are Knuth equivalent.

The following lemma will be used in the argument of Section 4.

Lemma 1 *If w and w' are Knuth equivalent words, and w_0 and w_0' are the results of removing the p largest numbers from each, for any p, then w_0 and w_0' are Knuth equivalent words.*

We refer the proof of this lemma to [2], for example.

Example 1

$$5152431245 \approx 5415213245.$$

A sequence of elementary Knuth transformations between these two Knuth equivalent words is given as follows:

$$\begin{aligned}
&5\mathbf{152}431245 \approx 5\mathbf{512}431245 && (1 \leq 2 < 5) \\
&= 551\mathbf{243}1245 \approx 551\mathbf{423}1245 && (2 \leq 3 < 4) && \cdots *_1 \\
&= 55\mathbf{142}31245 \approx 55\mathbf{412}31245 && (1 \leq 2 < 4) && \cdots *_2 \\
&= \mathbf{554}1231245 \approx \mathbf{545}1231245 && (4 < 5 \leq 5) \\
&= 5451\mathbf{231}245 \approx 5451\mathbf{213}245 && (1 < 2 \leq 3) && \cdots *_3 \\
&= 5\mathbf{451}213245 \approx 5\mathbf{415}213245 && (1 < 4 \leq 5)
\end{aligned}$$

Consider the two words 1243124 and 4121324, obtained by removing 5's from 5152431245 and 5415213245, respectively. These two words are again Knuth equivalent:

$$\begin{aligned}
&1243124 \approx 4121324. \\
&1\mathbf{243}124 \approx 1\mathbf{423}124 && (2 \leq 3 < 4) && \cdots *_1 \\
&= \mathbf{142}3124 \approx \mathbf{412}3124 && (1 \leq 2 < 4) && \cdots *_2 \\
&= 41\mathbf{231}24 \approx 41\mathbf{213}24 && (1 < 2 \leq 3) && \cdots *_3
\end{aligned}$$

2.3. *Biword*

We say that a two-rowed array

$$\boldsymbol{w} = \begin{pmatrix} i_1 & i_2 & \cdots & i_k & \cdots & i_n \\ j_1 & j_2 & \cdots & j_k & \cdots & j_n \end{pmatrix}$$

is a *biword* if the columns are arranged according to the lexicographic order:

$$\begin{cases} i_1 \leq i_2 \leq \cdots \leq i_n, \\ j_k \leq j_{k+1} \quad \text{if } i_k = i_{k+1} \quad (k = 1, \ldots, n-1). \end{cases}$$

Then we define the *dual biword* w^* of w as follows, first by interchanging the top and the bottom rows, and by rearranging the columns so that w^* should be in lexicographic order:

$$w^* = \begin{pmatrix} j_{\sigma(1)} & j_{\sigma(2)} & \cdots & j_{\sigma(k)} & \cdots & j_{\sigma(n)} \\ i_{\sigma(1)} & i_{\sigma(2)} & \cdots & i_{\sigma(k)} & \cdots & i_{\sigma(n)} \end{pmatrix},$$

where $\sigma \in S_n$ is a permutation of indices $1, 2, \ldots, n$ such $j_{\sigma(1)} \leq j_{\sigma(2)} \leq \cdots \leq j_{\sigma(n)}$ and that $i_{\sigma(k)} \leq i_{\sigma(k+1)}$ if $j_{\sigma(k)} = j_{\sigma(k+1)}$.

Example 2 The dual biword of

$$w = \begin{pmatrix} 1 & 2 & 2 & 4 & 5 & 7 \\ 3 & 1 & 5 & 2 & 2 & 1 \end{pmatrix}$$

is

$$w^* = \begin{pmatrix} 1 & 1 & 2 & 2 & 3 & 5 \\ 2 & 7 & 4 & 5 & 1 & 2 \end{pmatrix}.$$

2.4. RSK correspondence

There is a bijection between the biwords w and the pairs of tableaux (P, Q) of the same shape (RSK correspondence). The *P-symbol P* is the tableau obtained from the bottom row $(j_1, j_2, \ldots, j_n)$ by bumping. The *Q-symbol Q* is another tableau of the same shape which keeps the itinerary of the bumping procedure; it is obtained by filling the number i_k at each step in the box that has newly appeared when the number j_k is inserted.

Example 3 For the biword

$$w = \begin{pmatrix} 1 & 2 & 2 & 4 & 5 & 7 \\ 3 & 1 & 5 & 2 & 2 & 1 \end{pmatrix},$$

the corresponding pair of tableaux (P, Q) is obtained as in figure 2 on the next page.

Remark 1 The RSK correspondence can also be formulated as a bijection between the matrices with nonnegative integer entries and the pair of tableaux of the same shape. Note that the matrix $A = (a_{ij})$ corresponding to a biword w is defined by setting a_{ij} to be the number of columns of the form $\binom{i}{j}$ in w.

$$\begin{tikzcd}
P_0 = \emptyset \arrow[d] & Q_0 = \emptyset \arrow[d] \\
P_1 = \begin{array}{|c|}\hline 3\\\hline\end{array} \arrow[d] & Q_1 = \begin{array}{|c|}\hline 1\\\hline\end{array} \arrow[d] \\
P_2 = \begin{array}{|c|}\hline 1\\\hline 3\\\hline\end{array} \arrow[d] & Q_2 = \begin{array}{|c|}\hline 1\\\hline 2\\\hline\end{array} \arrow[d] \\
P_3 = \begin{array}{|c|c|}\hline 1&5\\\hline 3\\\cline{1-1}\end{array} \arrow[d] & Q_3 = \begin{array}{|c|c|}\hline 1&2\\\hline 2\\\cline{1-1}\end{array} \arrow[d] \\
P_4 = \begin{array}{|c|c|}\hline 1&2\\\hline 3&5\\\hline\end{array} \arrow[d] & Q_4 = \begin{array}{|c|c|}\hline 1&2\\\hline 2&4\\\hline\end{array} \arrow[d] \\
P_5 = \begin{array}{|c|c|c|}\hline 1&2&2\\\hline 3&5\\\cline{1-2}\end{array} \arrow[d] & Q_5 = \begin{array}{|c|c|c|}\hline 1&2&5\\\hline 2&4\\\cline{1-2}\end{array} \arrow[d] \\
P = P_6 = \begin{array}{|c|c|c|}\hline 1&1&2\\\hline 2&5\\\cline{1-2} 3\\\cline{1-1}\end{array} & Q = Q_6 = \begin{array}{|c|c|c|}\hline 1&2&5\\\hline 2&4\\\cline{1-2} 7\\\cline{1-1}\end{array}
\end{tikzcd}$$

Figure 2. Bumping procedure.

It is known that the P-symbol and the Q-symbol are interchanged if we switch the roles of the top and the bottom rows in the biword (see [2], for example).

Proposition 2.1 *If a biword* $\mathbf{w}$ *corresponds to the pair of tableaux* (P, Q), *then the dual biword* $\mathbf{w}^*$ *of* $\mathbf{w}$ *corresponds to the pair* (Q, P).

3. Box-ball system

In this section, we formulate the main results of this paper in terms of the *standard* version of the box-ball system (BBS), corresponding to the standard tableaux in the context of the RSK correspondence. A BBS is a system of finite number of balls of n colors evolving in the infinite array of boxes indexed by $\mathbf{Z}$. By a "standard" BBS, we mean a BBS in which n balls of n different colors are placed in the infinite array of boxes and all the boxes have capacity one. We use the numbers $1, 2, \ldots, n$ to denote the colors of balls, and the symbol $e = n + 1$ to indicate a vacant place.

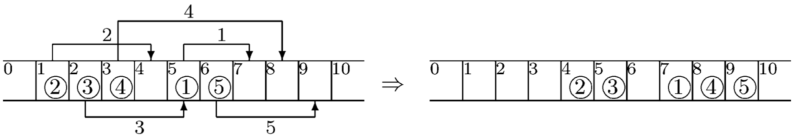
3.1. *Standard BBS: Original algorithm*

We first formulate the *standard* version of the BBS. A *state* of this system is a way to arrange n balls of different colors $1, 2, \ldots, n$ in the array of boxes indexed by $\mathbf{Z}$, under the condition that at most one ball can be placed in each box. One step of time evolution of the standard BBS, from time t to $t + 1$, is defined as follows:

1. Every ball should be moved only once within the interval between time t and $t + 1$.
2. Move the ball of color 1 to the nearest right empty box.
3. In the same way, move the balls of colors $2, 3, \ldots, n$, in this order.

We refer to this rule as the *original algorithm* of the standard BBS.

Example 4 The following figure shows an example with $n = 5$.



In the following figure, we show how Example 4 evolves as a BBS.

```
         _______234_____15_________________________________
         __________234____15_______________________________
         _____________234___15_____________________________
         ________________234__15___________________________
Time  t :___________________234_15_________________________
Time t+1:______________________23_145______________________
         ________________________23__145___________________
         __________________________23___145________________
         ____________________________23____145_____________
         ______________________________23_____145__________
```

Observe that there are groups of numbers behaving like solitons. For a study of the BBS from the viewpoint of solitons, we refer the reader to [1] and the references therein.

Remark 2 We remark that the BBS is a reversible system. In the original algorithm described above, exchange the roles of left and right, and move the balls according to the reversed order $n, n-1, \ldots$. Then we obtain the state at time $t-1$ (see the figure in Example 4 upside down).

Remark 3 A generalization of the standard BBS can be given by using more than one ball for some colors. One can also formulate a BBS such that more than one ball can be put in some boxes. A detailed description of such generalizations will be given in Section 5.

3.2. Biword formulation

We next attach a biword to each state of the standard BBS and formulate our main theorem.

Each state of the standard BBS can be represented by a doubly infinite sequence $\cdots a_{-1}\ a_0 a_1 \cdots$ of numbers $1, \dots, n$ and $e = n + 1$ such that $a_i = e$ except for a finite number of i's; if the box i is not empty, we define a_i to be the color of the ball contained in the box i, and set $a_i = e$ otherwise. Then we make a record of all pairs $\binom{i}{a_i}$ of *box-labels* i and *ball-colors* a_i (such that $a_i \neq e$), by scanning the sequence from left to right:

$$w = \begin{pmatrix} i_1 & i_2 & \cdots & i_k & \cdots & i_n \\ a_{i_1} & a_{i_2} & \cdots & a_{i_k} & \cdots & a_{i_n} \end{pmatrix}$$

We read $\binom{i_k}{a_{i_k}}$ in w as:

“ The box of label i_k contains a ball of color a_{i_k}.”

In this way, we obtain a bijection between the possible states of the standard BBS and the biwords

$$w = \begin{pmatrix} i_1 & i_2 & \cdots & i_n \\ j_1 & j_2 & \cdots & j_n \end{pmatrix}$$

such that $i_1 < i_2 < \cdots < i_n$ and that $\{j_1, j_2, \dots, j_n\} = \{1, 2, \dots, n\}$. When w is the biword attached to a state of the standard BBS, the dual biword w^* is of the form

$$w^* = \begin{pmatrix} 1 & 2 & \cdots & k & \cdots & n \\ b_1 & b_2 & \cdots & b_k & \cdots & b_n \end{pmatrix}$$

We remark that the bottom row

$$b = (b_1, b_2, \dots, b_k, \dots, b_n)$$

of the dual biword w^* represents the sequence of the box-labels of all nonempty boxes, arranged according to the ordering of colors. We refer to $b = (b_1, \dots, b_n)$ as the *box-label sequence* associated with the state $\cdots a_{-1} a_0 a_1 \cdots$.

Example 5 The two states of Example 4, at time t and at $t + 1$, are rewritten as follows in terms of the biwords, respectively:

$$w = \begin{pmatrix} 1 & 2 & 3 & 5 & 6 \\ 2 & 3 & 4 & 1 & 5 \end{pmatrix} \quad \Rightarrow \quad w' = \begin{pmatrix} 4 & 5 & 7 & 8 & 9 \\ 2 & 3 & 1 & 4 & 5 \end{pmatrix}.$$

The corresponding dual biwords are given by

$$\boldsymbol{w}^* = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 2 & 3 & 6 \end{pmatrix} \quad \Rightarrow \quad (\boldsymbol{w}')^* = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 7 & 4 & 5 & 8 & 9 \end{pmatrix}.$$

In terms of the box-label sequences, the same time evolution is expressed as

$$b = (5, 1, 2, 3, 6) \quad \Rightarrow \quad b' = (7, 4, 5, 8, 9).$$

Given a state $\cdots a_{-1} a_0 a_1 \cdots$ of the standard BBS, we denote by (P, Q) the pair of tableaux assigned to the biword $\boldsymbol{w}$ through the RSK correspondence. The P-symbol P (resp. Q-symbol Q) is by definition the tableau obtained by bumping from the bottom row of $\boldsymbol{w}$ (resp. from the bottom row of the dual biword $\boldsymbol{w}^*$ of $\boldsymbol{w}$). Note also that P is a standard tableau of n boxes, and that Q is a tableau of the same shape in which the entries are mutually distinct integers.

The time evolution of the standard BBS is then translated into the time evolution of the corresponding biword, and also, via the RSK correspondence, into the time evolution of the pair of tableaux (P, Q) of the same shape.

Theorem 3.1 *We regard the standard BBS as the time evolution of the pairs of tableaux (P, Q) through the RSK correspondence in the way explained above. Then,*
1. *The P-symbol is a conserved quantity under the time evolution of the BBS.*
2. *The Q-symbol evolves independently of the P-symbol.*

As we will see below, the time evolution of the standard BBS can be described locally by the so-called *carrier algorithm*; Theorem 3.1 will be proved in Section 4 by applying the carrier algorithm. We remark that the time evolution of the Q-symbol can also be described by using the carrier algorithm (see Proposition 4.1).

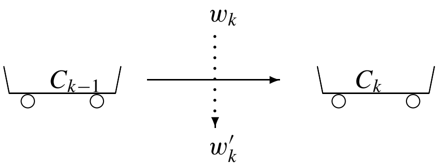
3.3. Carrier algorithm

The *carrier algorithm* is a way to transform a finite sequence $w = (w_1, w_2, \ldots w_n)$ of numbers into another sequence $w' = (w'_1, w'_2, \ldots, w'_n)$, by means of a weakly increasing sequence $C = (c_1, \ldots, c_m)$, called the *carrier*. In this transformation, the carrier moves along the word w from left to right; while the carrier passes each number w_k, the carrier loads w_k and unloads w'_k:

$$\left\{ \begin{array}{ccccccccccc} & w_1 & & w_2 & & w_3 & & \cdots & & w_n & \\ & \vdots & & \vdots & & \vdots & & \cdots & & \vdots & \\ C = C_0 & \longrightarrow & C_1 & \longrightarrow & C_2 & \longrightarrow & C_3 & \cdots & C_{n-1} & \longrightarrow & C_n = C' \\ & \vdots & & \vdots & & \vdots & & \cdots & & \vdots & \\ & w'_1 & & w'_2 & & w'_3 & & \cdots & & w'_n & \end{array} \right\}$$

The rule of loading and unloading is defined as follows:

The rule of loading/unloading: Let $C_{k-1} = (c_1^{(k-1)}, c_2^{(k-1)}, \ldots, c_m^{(k-1)})$ $(c_1^{(k-1)} \leq c_2^{(k-1)} \leq \cdots \leq c_m^{(k-1)})$ be the sequence of numbers which have already been loaded on the carrier. Let w_k be the number to be loaded. Compare w_k with the numbers in C_{k-1}. If there are some numbers larger than w_k in C_{k-1}, then one of the smallest among them is unloaded, and w_k is loaded instead. If there is no such number, a minimum in C_{k-1} is unloaded, and w_k is loaded instead (see the figure below).



$$w'_k = \begin{cases} \min\left\{c_i^{(k-1)} \in C_{k-1} \,\middle|\, c_i^{(k-1)} > w_k\right\} & \text{if } \left\{c_i^{(k-1)} \in C_{k-1} \,\middle|\, c_i^{(k-1)} > w_k\right\} \neq \emptyset, \\ c_1^{(k-1)} & \text{otherwise.} \end{cases}$$

$$C_k = \text{ the sequence of numbers obtained from } C_{k-1} \text{ by replacing a } w'_k \text{ by } w_k.$$

Given two finite sequences $C = (c_1, c_2, \ldots, c_m)$ $(c_1 \leq c_2 \leq \cdots \leq c_m)$ and $w = (w_1, w_2, \ldots, w_n)$, from $C_0 = C$, we obtain the new sequences $C' = C_n$ and w' by repeating the rule of loading/unloading above. We call this transformation $(C, w) \to (C', w')$ the *carrier algorithm*.

Remark 4 The carrier algorithm can be understood as a repetition of Knuth transformations. We apply the basic rule (1) of Section 2.2, to the k-th step of loading/unloading mentioned above. When C_{k-1} contains a number greater than w_k, we have

$$C_{k-1}w_k = (ux'v)x \quad \to \quad x'(uxv) = w'_k C_k \quad (u \leq x < x' \leq v),$$

where $x' = w_k$ and $x = w'_k$; otherwise,

$$C_{k-1}w_k = (x'v)x \quad \to \quad x'(vx) = w'_k C_k \quad (x' \leq v \leq x)$$

is the trivial transformation. Hence we have $C_{k-1}w_k \approx w'_k C_k$ for each $k = 1, \ldots, n$:

$$\begin{aligned} Cw = C_0 w_1 w_2 w_3 \ldots w_n &\approx w'_1 C_1 w_2 w_3 \ldots w_n \\ &\approx w'_1 w'_2 C_2 w_3 \ldots w_n \\ &\approx \qquad \vdots \\ &\approx w'_1 w'_2 w'_3 \ldots w'_n C_n = w'C' \end{aligned}$$

3.4. Time evolution with a carrier

In the following, we give two propositions that will be used in the proof of Theorem 3.1. The time evolution of the standard BBS from one state to the next can be described in two different ways; the original algorithm and the transformation of the box-label sequences. We describe these two algorithms by using the carrier as introduced above.

We take an interval $[p, q]$ of $\mathbf{Z}$ so that it contains all i with $a_i \neq e$, and all i with $a'_i \neq e$ as well. A choice of such an interval is given by $p = \min\{i \in \mathbf{Z} \mid a_i \neq e\}$, $q = \max\{i \in \mathbf{Z} \mid a_i \neq e\} + n$.

Proposition 3.2 *For a given state of the standard BBS, by ignoring the infinite sequences of e's on both sides, let* $A = (a_p, a_{p+1}, \ldots, a_{q-1}, a_q)$ *be the remaining sequence of numbers; with* p, q *defined as above. Then, the original algorithm* $A \to A'$ *from time* t *to* $t + 1$, *can be described by the carrier algorithm with a sequence* $C = (e, e, \ldots, e)$ *of* n *e's chosen as the initial state.*

We remark that, in this procedure, the final state of the carrier is identical to the initial state: $C' = (e, \ldots, e)$. The proof of this proposition will be given in Section 4.

Remark 5 This algorithm with a carrier was introduced for the first time in 1997 by Takahashi-Matsukidaira [7]. As for the *one-colored version* (in which each box has an arbitrary finite capacity, and all balls have the same color), they proved in [7] that the original algorithm and carrier algorithm provide the same time evolution of BBS.

Example 6 Take the same example as in Example 4. With $n = 5$, we take the interval $[p, q] = [1, 11]$, and set

$$C = (e, e, e, e, e), \qquad A = (2, 3, 4, e, 1, 5, e, e, e, e, e).$$

After eleven times of loading/unloading, we obtain

$$A' = (e, e, e, 2, 3, e, 1, 4, 5, e, e), \qquad C' = (e, e, e, e, e).$$

The following figure shows the intermediate steps of the procedure.

$$e,e,e,e,e \underset{e}{\overset{2}{+}} 2,e,e,e,e \underset{e}{\overset{3}{+}} 2,3,e,e,e \underset{e}{\overset{4}{+}} 2,3,4,e,e \underset{2}{\overset{e}{+}} 3,4,e,e,e \underset{3}{\overset{1}{+}} 1,4,e,e,e \underset{e}{\overset{5}{+}} \to$$

$$\to 1,4,5,e,e \underset{1}{\overset{e}{+}} 4,5,e,e,e \underset{4}{\overset{e}{+}} 5,e,e,e,e \underset{5}{\overset{e}{+}} e,e,e,e,e \underset{e}{\overset{e}{+}} e,e,e,e,e \underset{e}{\overset{e}{+}} e,e,e,e,e$$

Next, we discuss the transformation of the box-label sequences. Recall that the box-label sequence $b = (b_1, \ldots, b_k, \ldots, b_n)$ is defined as the bottom row of the dual biword $\boldsymbol{w}^*$

(see Section 3.2). Notice that $b_k \in [p, q]$ for all $k = 1, 2, \dots, n$, with p, q defined as before.

Proposition 3.3 *For a given state of the standard BBS, the transformation of the box-label sequence $b \to b'$ from time t to $t + 1$ can be described by the carrier algorithm with the initial state of the carrier $C = (l_1, l_2, \dots, l_m)$ defined as the increasing sequence consisting of the labels of all empty boxes in the interval $[p, q]$.*

We refer to the procedure of Proposition 3.3 as the box-label algorithm. The proof of this proposition will be given in Section 4.

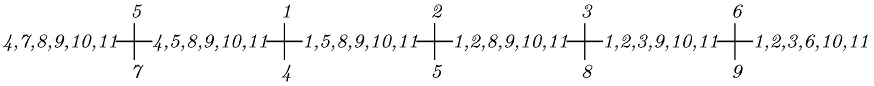
Example 7 Take the same example as in Example 4. With $n = 5$, we take the interval $[p, q] = [1, 11]$, and set

$$C = (4, 7, 8, 9, 10, 11), \quad b = (5, 1, 2, 3, 6).$$

After five times of loading/unloading, we obtain

$$b' = (7, 4, 5, 8, 9), \quad C' = (1, 2, 3, 6, 10, 11).$$

The following figure shows the intermediate steps of the procedure.



4. Proof of the main results

4.1. *Proof of Proposition 3.3*

Note that a state of the standard BBS, represented as an infinite sequence $\cdots a_{-1}a_0a_1 \dots$, is identified with a function $a : \mathbf{Z} \to \{1, \dots, n, n + 1 = e\}$ of *finite support*; the support of a is defined by $\operatorname{supp}(a) = \{i \in \mathbf{Z}; a_i \neq e\}$. The time evolution a' of a is determined by the injective mapping $f : \operatorname{supp}(a) \to \mathbf{Z}$ such that $a'_{f(i)} = a_i$ for $i \in \operatorname{supp}(a)$, and by $a'_j = e$ for $j \notin \operatorname{Im}(f)$. We first describe how the mapping f is defined by the original algorithm.

We now visualize the original algorithm of BBS by means of a 2-dimensional diagram as in figure 3. First, write the state a at time t at the top (a); write each a_i again, down in the same column at the row corresponding to the number itself (b)–(g); here we are using the datum of Example 4. Then, following the original algorithm of BBS, connect “1” to its partner, nearest e on the right as in figure 3(i). Then look at “2”, draw lines by the same method (j). In this example, “3” should be moved to the empty box which had originally

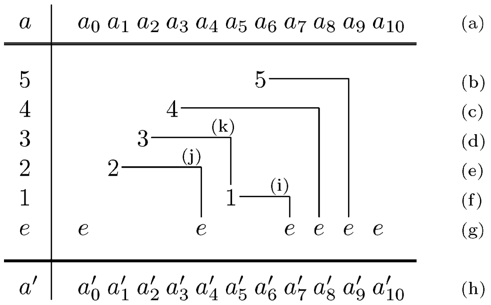

a
a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10}
(a)
5
4
3
2
1
e
(b)
(c)
(d)
(e)
(f)
(g)
(k)
(j)
(i)
e e e e e e
a'
a'_0 a'_1 a'_2 a'_3 a'_4 a'_5 a'_6 a'_7 a'_8 a'_9 a'_{10}
(h)

Figure 3. Two-dimensional illustration.

been occupied by the 1 on the right. Considering this 1 as the partner of the 3, connect the 3 to it (k). Do the same thing until all $a_i(\neq e)$ have been connected to their partners $a'_{f(i)}$. The general rule for drawing lines can be described as follows:

$(*)$ *Connect each number with the leftmost one among all the smaller numbers on the right that have not been connected from above.*

In the original algorithm, the values $f(i)$ are determined in increasing order of a_i. Therefore, assume that $f(j)$ is known for all j such that $a_j < a_i$; let $X_i = \{f(j) \mid a_j < a_i\}$ and $Y_i := \{k \mid k \in \mathbf{Z} \setminus X_i, a_k < a_i \text{ or } a_k = e\}$. Then $f(i)$ is determined as follows,

$$f(i) = \min\{k \mid k \in Y_i, k > i\}; \tag{4}$$

the minimum exists because $a_k = e$ for infinitely many $k > i$ while X_i is finite.

Here, we notice that a state of the standard BBS can be described as a dual biword

$$\boldsymbol{w}^* = \begin{pmatrix} a_{b_1} & a_{b_2} & \cdots & a_{b_k} & \cdots & a_{b_n} \\ b_1 & b_2 & \cdots & b_k & \cdots & b_n \end{pmatrix}.$$

in which $a_{b_k} = k$ for all $k = 1, 2, \dots, n$ (recall the latter half of Section 3.4). From the explanation above, we can use the carrier algorithm with the initial state $Y = Y_{b_1}$ for deriving the values $f(i)$'s (box-labels). We remark that Y_{b_k} represents the carrier for $k = 1, 2, \dots, n$; see the following chart.

$$\left\{ \begin{array}{ccccccccc} & b_1 & & b_2 & & b_3 & \cdots & b_n & \\ & \vdots & & \vdots & & \vdots & \cdots & \vdots & \\ Y = Y_{b_1} & \longrightarrow & Y_{b_2} & \longrightarrow & Y_{b_3} & \longrightarrow & Y_{b_4} \quad \cdots \quad Y_{b_n} & \longrightarrow & Y' \\ & \vdots & & \vdots & & \vdots & \cdots & \vdots & \\ & f(b_1) & & f(b_2) & & f(b_3) & \cdots & f(b_n) & \end{array} \right\}$$

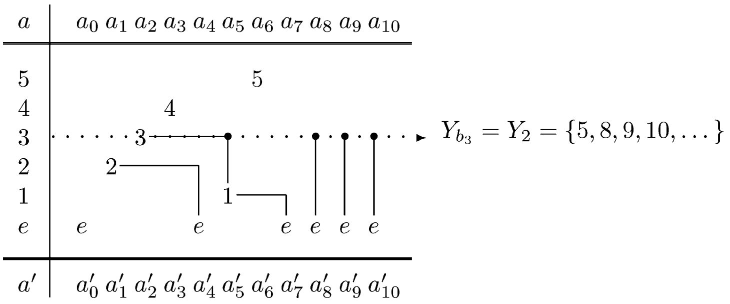

a
$a_0\ a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ a_{10}$
5
4
3
2
1
e
$Y_{b_3} = Y_2 = \{5, 8, 9, 10, \ldots\}$
a'
$a'_0\ a'_1\ a'_2\ a'_3\ a'_4\ a'_5\ a'_6\ a'_7\ a'_8\ a'_9\ a'_{10}$

Figure 4. Example for the proof of Proposition 3.3.

Notice that $Y' = Y_{f(b_1)}$ is the next initial state for the carrier algorithm from time $t+1$ to $t+2$. In this way, we can describe the box-label algorithm as the carrier algorithm, namely, we have the transformation of the box-label sequences: $(b_1, \ldots, b_n) \to (f(b_1), \ldots, f(b_n))$. We have thus proved Proposition 3.3 (see the example in figure 4).

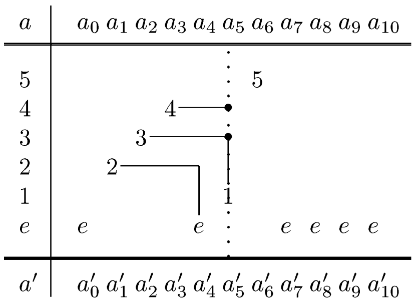
4.2. *Proof of Proposition 3.2*

Look at figure 3 again. The state a' at time $t+1$ can be determined by means of this diagram (h). In what follows, by a *chain*, we mean a decreasing sequence of numbers that are connected together by lines ($a_i \to a_{f(i)} \to a_{f(f(i))} \to \cdots$), and by a *perfect chain* a chain whose bottom is e: $a_{i_0} \to a_{i_1} \to \cdots \to a_{i_r}$ such that $i_0 \notin \mathrm{Im}\ (f)$, $f(i_k) = i_{k+1}$ for $k = 1, 2, \ldots, r-1$, and $a_{i_r} = e$. We analyze how a'_i is determined from a_i by looking locally at the i-th column.

(i) If an e is alone, it remains at the same position at time $t+1$. Note that each $a_i(\neq e)$ belongs to a unique perfect chain.
(ii) If $a_i(\neq e)$ is at the top of a perfect chain, namely $i \notin \mathrm{Im}(f)$, it is replaced with e (i.e., $a'_k = e$).
(iii) If $a_i(\neq e)$ belongs to a perfect chain and it is not at the top, it is replaced at time $t+1$ with $a'_i = a_k$ connected with it from above.

Notice that the perfect chains never intersect with each other. In view of this fact, we see that the same set of non-intersecting perfect chains can be obtained by observing the sequence of numbers at time *t from left to right*, rather than *from bottom to top* as in the rule ($*$).

In the algorithm with a carrier, the function f is determined instead by increasing values of $f(j)$. Let i be a value for which we search a k such that $f(k) = i$ and assume that the set $A_i = \{j \mid j \in \mathbf{Z}, a_j \neq e, f(j) < i\}$ is known; put $B_i = \{j \mid j \in \mathbf{Z}, j < i, j \notin A_i\}$. If $a_j = e$ or $a_j \leq a_i$ for all $j \in B_i$, then we have $i \notin \mathrm{Im}(f)$; otherwise with $k = \min\{a_j \mid j \in B_i, a_j > a_i$ and $a_j \neq e\}$, the index k is the unique one with $f(k) = i$. This defines the same function f as the original definition, as can be proved by



a
$a_0\ a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ a_{10}$
5
4
3
2
1
e
5
4
3
2
1
e e e e e e
a'
$a'_0\ a'_1\ a'_2\ a'_3\ a'_4\ a'_5\ a'_6\ a'_7\ a'_8\ a'_9\ a'_{10}$

$C_5 = \{3, 4, e, e, \dots\}$

Figure 5. Example for the proof of Proposition 3.2.

an induction on i. We remark that each set $C_i = \{a_j \mid j \in B_i\}$ corresponds to a carrier ($C = C_p$):

$$\left\{\begin{array}{ccccccccc}
 & a_p & & a_{p+1} & & a_{p+2} & \cdots & & a_q & \\
 & \vdots & & \vdots & & \vdots & \cdots & & \vdots & \\
C = C_p & \longrightarrow & C_{p+1} & \longrightarrow & C_{p+2} & \longrightarrow & \cdots & C_q & \longrightarrow & C' \\
 & \vdots & & \vdots & & \vdots & \cdots & & \vdots & \\
 & a_p{}' & & a_{p+1}{}' & & a_{p+2}{}' & \cdots & & a_q{}' &
\end{array}\right\}$$

We have thus completed Proposition 3.2 (see the example in figure 5).

4.3. *Proof of Theorem 3.1*

In the notation of Proposition 3.2, we get $CA \approx A'C$ from Remark 4.

$$\begin{aligned}
CA &= C_p(a_p, a_{p+1}, \dots, a_{q-1}, a_q) \\
&\approx a'_p C_{p+1}(a_{p+1}, a_{p+2}, \dots, a_{q-1}, a_q) \\
&\approx (a'_p, a'_{p+1}) C_{p+2}(a_{p+2}, a_{p+3}, \dots, a_{q-1}, a_q) \\
&\vdots \\
&\approx (a'_p, a'_{p+1}, \dots, a'_{q-1}, a'_q) C' = A'C
\end{aligned}$$

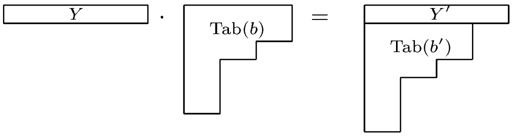
We know that Knuth equivalent words correspond to the same tableau. Since e is thought of as larger than any other number, by virtue of Lemma 1, we see that the results A_e and A'_e of removing e's from CA and $A'C$, respectively, are Knuth equivalent, i.e., $A_e \approx A'_e$. Hence the bumping of A_e and A'_e give the same tableau P; this P-symbol is conserved by the time evolution. We remark that the sequence A_e is nothing but the bottom row of the biword $\mathbf{w}$ we introduced before. We have completed the proof of the first statement of Theorem 3.1.

We next consider the box-label algorithm in order to prove the second statement of Theorem 3.1. We denote by T^* the time evolution of box-label sequences, so that $T^*(b) = b'$. With the notation in the proof of Proposition 3.3, we obtain the following sequence of

Knuth equivalent words:

$$Yb = Y_{b_1}b_1 \dots b_n \approx b'_1 Y_{b_2} b_2 \dots b_n \approx \cdots \approx b'_1 \dots b'_n Y' = b'Y'.$$

We now look at the tableau Tab(Yb). From the definition of the carrier algorithm, it is clear that, while inserting b into Y, the first row of the tableau is kept track of by the carrier. Hence we see that the first row of the resulting tableau Tab(Yb) is identical to Y', and that the tableau obtained from Tab(Yb) by removing the first row is identical to Tab(b'). This implies that both Y' and Tab(b') depend only on the Knuth equivalence class of Yb.



Supposing that a word a is Knuth equivalent to b (i.e., Tab(a) = Tab(b)), consider the time evolution $Ya \to a'Y''$ by the carrier algorithm with the same initial state of the carrier Y. Since $Ya \approx Yb$, from the consideration above, we conclude that $Y' = Y''$ and Tab(a') = Tab(b'), hence $a' \approx b'$.

Lemma 2 *If a and b are two Knuth equivalent words, then so are the resulting $T^*(a)$ and $T^*(b)$.*

Let $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$ be the biwords corresponding to the states at time t and $t+1$, respectively; let w_1, w_2 (resp. w_1^*, w_2^*) be the bottom rows of the biwords $\boldsymbol{w_1}$, $\boldsymbol{w_2}$ (resp. the dual biwords $\boldsymbol{w_1^*}$, $\boldsymbol{w_2^*}$). The tableaux Q_1 and Q_2 are the Q-symbols for time t and $t+1$, respectively. From Proposition 2.1, we see that $Q_i = Q(w_i) = P(w_i^*)$ for each $i = 1, 2$ (see figure 6).

By the theory of the RSK correspondence, among the words that correspond to tableaux of a given shape, the tableau words are precisely those for which the tableaux go thorough a specific sequence of shapes during insertion, and which therefore have a specific type

$$\begin{tikzcd}
\boldsymbol{w_1} \arrow[r, "RSK"] \arrow[d, "dual"'] \arrow[ddr] & (P_1, Q_1) \arrow[d] \arrow[ddr] & \\
\boldsymbol{w_1^*} \arrow[r] \arrow[ddr, "T^*"'] & (Q_1, P_1) \arrow[ddr] & \left\{ \begin{array}{lcl} P_2 & = & P_1 \\ Q_2 & = & T^*(Q_1) \end{array} \right. \\
 & \boldsymbol{w_2} \arrow[r] \arrow[d] & (P_2, Q_2) \arrow[d] \\
 & \boldsymbol{w_2^*} \arrow[r] & (Q_2, P_2)
\end{tikzcd}$$

Figure 6. Time evolution T^* in the dual version.

of Q-symbol. Then if w_1^* is a tableau word, namely $w_1^* = \mathrm{W}(P(w_1^*))$, this means that its Q-symbol $Q(w_1^*)$ is that specific Q-symbol, but since $Q(w_1^*) = P_1 = P_2 = Q(w_2^*)$, this shows that $w_2^* = T^*(w_1^*)$ is also a tableau word. Hence we obtain the following lemma:

Lemma 3 *If b is a tableau word, $T^*(b)$ is a tableau word of the same shape.*

See figure 6 again; put $b = \mathrm{W}(Q_1) = \mathrm{W}(Q(w_1))$, then b is Knuth equivalent to w_1^*, which is the box-label sequence of a possible state of the BBS, so that $T^*(b)$ is defined. Then by Lemma 2 $T^*(b)$ is Knuth equivalent to $T^*(w_1^*) = w_2^*$ while by Lemma 3 it is a tableau word; since there is only one tableau word Knuth equivalent to w_2^*, namely $\mathrm{W}(P(w_2^*)) = \mathrm{W}(Q_2)$, this shows that $T^*(b) = \mathrm{W}(Q_2) = \mathrm{W}(Q(w_2))$, i.e., that $\mathrm{W}(Q(w_2))$ is completely determined by b and hence by $\mathrm{W}(Q(w_1))$, thereby completing the proof of the second statement of Theorem 3.1.

Identifying tableau words with tableaux, we can define the time evolution of the Q-symbol Q by

$$T^*(Q) = \mathrm{Tab}(T^*(\mathrm{W}(Q))).$$

Summarizing, with the interval $[p, q] \subset \mathbf{Z}$ again, we have

Proposition 4.1 *In the standard BBS, the time evolution of the Q-symbol Q is described by the box-label algorithm with a carrier. The initial state of the carrier is given with $C = (l_1, \ldots, l_m)$ defined as the increasing sequence consisting of the labels of all empty boxes in the interval $[p, q]$. The carrier runs along the rows of the tableau Q from left to right, and bottom to top.*

Therefore, the evolution of the Q-symbol can be directly computed by the box-label algorithm at the level of tableau words read off from the tableau (recall figure 1 in Section 2), without the need to recompute a tableau from the resulting word.

5. Generalization of the BBS

In this section, we consider two generalizations of the standard BBS, which we call the advanced BBS and the generalized BBS. In both cases, we allow to use an arbitrary finite number of balls for each color. An *advanced* BBS is a BBS in which all the boxes have capacity one. A *generalized* BBS is a BBS in which the capacity of each box is specified individually. When we consider a generalized BBS, we denote by δ_j the capacity of the box labeled j and assume $\delta_j \geq 1$ for all $j \in \mathbf{Z}$. Then an advanced BBS is considered as the special case of a generalized BBS such that $\delta_j = 1$ for all $j \in \mathbf{Z}$.

We first discuss the original rule for the advanced BBS in which all the boxes have capacity one. In this case, we may use an arbitrary finite number of balls for each color. One step of time evolution of the advanced BBS, from time t to $t + 1$, is defined as follows:

$$\text{One step}\left\{\begin{array}{lcl}
\text{Time } t: & \cdots ee5e\mathbf{1}254\check{e}e3\mathbf{1}2\check{e}45eeeeeeeeee\cdots & \\
& \downarrow & \text{Move 1's.}\\
& \cdots ee5ee\mathbf{2}541\check{e}30\mathbf{2}145\check{e}eeeeeeee\cdots & \\
& \downarrow & \text{Move 2's.}\\
& \cdots ee5eee5412\mathbf{3}\check{e}e1452eeeeeeee\cdots & \\
& \downarrow & \text{Move 3.}\\
& \cdots ee5eee5\mathbf{4}12\check{e}3e1\mathbf{4}52\check{e}eeeeeeee\cdots & \\
& \downarrow & \text{Move 4's.}\\
& \cdots ee\mathbf{5}\check{e}ee\mathbf{5}\check{e}1243e1e\mathbf{5}24\check{e}eeeeeee\cdots & \\
& \downarrow & \text{Move 5's.}\\
\text{Time } t+1: & \cdots eee5eee51243e1ee245eeeeeee\cdots &
\end{array}\right.$$

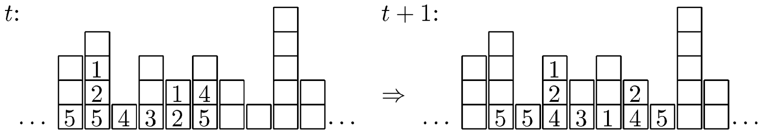
Figure 7. Original algorithm in the advanced BBS.

1. Every ball should be moved only once within the interval between time t and $t+1$.
2. Move the leftmost ball of color 1 to the nearest right empty box.
3. Among the remaining balls of color 1, if any, move the leftmost one to the nearest right empty box.
4. Repeat the same procedure until all the balls of color 1 have been moved (in figure 7 below, the balls to be moved are printed in the **boldface**, and the empty boxes to be filled in are denoted by $\check{e}$).
5. In the same way, move the balls of color 2, 3, ... n, in this order, until all the balls have been moved.

The following figure is an evolution of the example in figure 7:

```
          ____________155__2414__3__2__5________________________
          _______________155_2_1443__2__5_______________________
          __________________1525__41342__5______________________
Time  t :____________________5_1254__312_45____________________
Time t+1:_____________________5___51243_1__245_________________
          ______________________5___5__41213___245______________
          _______________________5___5__4_2_113___245___________
```

We next describe the generalized BBS where each box has an arbitrary finite capacity. (see the figure below).



In this case, we denote each box by a sequence of numbers limited by two walls "|". We fill in the box with e's so that the number of indices should represent the capacity of the box. In particular, the expression $|e\cdots e|$ (m-tuple of e) stands for an empty box of capacity m. Figure 8 is an example of the case where the boxes have capacity ..., 3, 4, 1, 3, 2, 3, 2, 1, 5, 2, ...

Then we can also apply the same rule of time evolution as before to this generalized version; the only difference is that, inside a box, the balls can be rearranged arbitrarily (for

$$
\begin{array}{lcl}
t: & \cdots|ee5|e\mathbf{1}25|4|\breve{e}e3|\mathbf{12}|\breve{e}45|ee|e|eeeee|ee|\cdots & \\
 & \downarrow & \text{Move 1's.} \\
 & \cdots|ee5|ee\mathbf{2}5|4|1\breve{e}3|e\mathbf{2}|145|\breve{e}e|e|eeeee|ee|\cdots & \\
 & \downarrow & \text{Move 2's.} \\
 & \cdots|ee5|eee5|4|12\mathbf{3}|\breve{e}e|145|2e|e|eeeee|ee|\cdots & \\
 & \downarrow & \text{Move 3.} \\
 & \cdots|ee5|eee5|\mathbf{4}|12\breve{e}|3e|1\mathbf{4}5|2\breve{e}|e|eeeee|ee|\cdots & \\
 & \downarrow & \text{Move 4's.} \\
 & \cdots|ee\mathbf{5}|\breve{e}ee\mathbf{5}|\breve{e}|124|3e|1e\mathbf{5}|24|\breve{e}|eeeee|ee|\cdots & \\
 & \downarrow & \text{Move 5's.} \\
 & \cdots|eee|5\boldsymbol{eee}|5|124|3\boldsymbol{e}|1\boldsymbol{ee}|24|5|eeeee|ee|\cdots & \\
 & \downarrow & \text{Pack the } e\text{'s in each box to the left.} \\
t+1: & \cdots|eee|eee5|5|124|e3|ee1|24|5|eeeee|ee|\cdots &
\end{array}
$$

Figure 8. A generalized version of the example in figure 7.

$$
\boxed{
\begin{array}{l}
\text{Box-label: } \cdots \;\; 1 \quad\; 2 \;\; 3 \;\; 4 \;\; 5 \;\; 6 \;\; 7 \;\; 8 \cdots \\
\qquad\qquad\quad \cdots|ee5|e125|4|ee3|12|e45|ee|e|\cdots \\
\\
\Rightarrow \left\{
\begin{array}{l}
\boldsymbol{w} = \begin{pmatrix} 1 & 2 & 2 & 2 & 3 & 4 & 5 & 5 & 6 & 6 \\ 5 & 1 & 2 & 5 & 4 & 3 & 1 & 2 & 4 & 5 \end{pmatrix} \\
\boldsymbol{w}^* = \begin{pmatrix} 1 & 1 & 2 & 2 & 3 & 4 & 4 & 5 & 5 & 5 \\ 2 & 5 & 2 & 5 & 4 & 3 & 6 & 1 & 2 & 6 \end{pmatrix}
\end{array}
\right.
\end{array}
}
$$

Figure 9. Biword formulation for the generalized version.

example, inside a box of capacity 2, the two expressions $|ab|$ and $|ba|$ are considered as representing the same state). For convenience, we always rearrange the balls in one box in the order $e, 1, 2, \ldots, n$ so that e's are packed to the left.

Given a state of the generalized BBS, we scan the sequence from left to right in order to obtain the biword $\boldsymbol{w}$ (see figure 9 cf. Section 3.2). We also denote by (P, Q) the pair of tableaux assigned to w through the RSK correspondence. Notice that P is a tableau in which each entry is taken from the numbers $1, 2, \ldots, n$, and that Q is a tableau of the same shape in which the entries are integers.

The time evolution of the generalized BBS is then translated into the time evolution of the corresponding biword, and also, via the RSK correspondence, into the time evolution of the pair of tableaux (P, Q) of the same shape.

Theorem 5.1 *We regard the generalized BBS as the time evolution of the pairs of tableaux (P, Q) through the RSK correspondence in the way explained above. Then,*

1. *The P-symbol is a conserved quantity under the time evolution of the BBS.*
2. *The Q-symbol evolves independently of the P-symbol.*
3. *The time evolution of the Q-symbol is described by the box-label algorithm with a carrier.*

In the box-label algorithm for the time evolution of the Q-symbol, the initial state of the carrier is defined to be the multiset obtained from that of all possible box-labels, by removing the labels contained in the Q-symbol, each as many times as the number of the appearences in Q. In this algorithm, the carrier runs along the rows of the tableau Q from left to right, and bottom to top.

To prove Theorem 5.1 we can apply the same method as the standard version, and hence we omit the proof of the generalized version. In the following, we explain the third statement of Theorem 5.1, the box-label algorithm for the generalized version. We denote by L the sequence of all box-labels with each $j \in \mathbf{Z}$ repeated δ_j times:

$$L = (\ldots, 0^{\delta_0}, 1^{\delta_1}, 2^{\delta_2}, \ldots) = (\ldots, \overbrace{0, \ldots, 0}^{\delta_0}, \overbrace{1, \ldots, 1}^{\delta_1}, \overbrace{2, \ldots, 2}^{\delta_2}, \ldots).$$

In what follows, we use the parentheses ( ) for sequences of numbers with multiplicities. We define $C = (l_p, \ldots, l_q) = (l_i \mid a_i = e, i \in [p, q])$ to be the sequence obtained from L by removing the box-labels $l_{i_1}, l_{i_2}, \ldots, l_{i_N}$. Here we consider the interval $[p, q]$ again, similarly as Section 3.3: $p = \min\{i \in \mathbf{Z} \mid a_i \neq e\}$, $q = \max\{i \in \mathbf{Z} \mid a_i \neq e\} + N$. We then apply the carrier algorithm to the word $b = (l_{\sigma(i_1)}, l_{\sigma(i_2)}, \ldots, l_{\sigma(i_N)})$ of box-labels by taking C for the initial state of the carrier.

Example 8 We show the box-label algorithm (with a carrier) by taking the same example as in figure 8. We consider the generalized BBS in which the boxes with labels $1, 2, \ldots, 10$ have capacities $3, 4, 1, 3, 2, 3, 2, 1, 5, 2$, respectively ($\delta_1 = 3, \delta_2 = 4, \ldots, \delta_{10} = 2$). Since

$$\begin{aligned} L &= (\ldots, 1^3, 2^4, 3^1, 4^3, 5^2, 6^3, 7^2, 8^1, 9^5, 10^2 \ldots) \\ &= (\ldots, 1, 1, 1, 2, 2, 2, 2, 3, 4, 4, 4, 5, 5, 6, 6, 6, 7, 7, 8, 9, 9, 9, 9, 9, 10, 10, \ldots) \end{aligned}$$

and

$$\begin{aligned} w &= \begin{pmatrix} 1 & 2 & 2 & 2 & 3 & 4 & 5 & 5 & 6 & 6 \\ 5 & 1 & 2 & 5 & 4 & 3 & 1 & 2 & 4 & 5 \end{pmatrix} \\ &= \begin{pmatrix} l_3 & l_5 & l_6 & l_7 & l_8 & l_{11} & l_{12} & l_{13} & l_{15} & l_{16} \\ a_3 & a_5 & a_6 & a_7 & a_8 & a_{11} & a_{12} & a_{13} & a_{15} & a_{16} \end{pmatrix} \end{aligned}$$

we take $p = 3, q = 16 + 10 = 26$, and

$$\begin{aligned} C &= (l_4, l_9, l_{10}, l_{14}, l_{17}, l_{18}, \ldots, l_{26}) \\ &= (2, 4, 4, 6, 7, 7, 8, 9, 9, 9, 9, 9, 10, 10) \end{aligned}$$

for the initial state of the carrier. The figure on the next page indicates how the box-label algorithm with a carrier works to generate the time evolution from time t to $t + 1$. Notice that a carrier always has labels of available boxes.

$$\begin{aligned}
Cb &= (2, \check{4}, 4, 6, 7, 7, 8, 9, 9, 9, 9, 9, 10, 10)\mathbf{2}525436126 \\
&\approx 4(2, 2, 4, \check{6}, 7, 7, 8, 9, 9, 9, 9, 9, 10, 10)\mathbf{5}25436126 \\
&\approx 46(2, 2, \check{4}, 5, 7, 7, 8, 9, 9, 9, 9, 9, 10, 10)\mathbf{2}5436126 \\
&\approx 464(2, 2, 2, 5, \check{7}, 7, 8, 9, 9, 9, 9, 9, 10, 10)\mathbf{5}436126 \\
&\approx 4647(2, 2, 2, \check{5}, 5, 7, 8, 9, 9, 9, 9, 9, 10, 10)\mathbf{4}36126 \\
&\approx 46475(2, 2, 2, \check{4}, 5, 7, 8, 9, 9, 9, 9, 9, 10, 10)\mathbf{3}6126 \\
&\approx 464754(2, 2, 2, 3, 5, \check{7}, 8, 9, 9, 9, 9, 9, 10, 10)\mathbf{6}126 \\
&\approx 4647547(\check{2}, 2, 2, 3, 5, 6, 8, 9, 9, 9, 9, 9, 10, 10)\mathbf{1}26 \\
&\approx 46475472(1, 2, 2, \check{3}, 5, 6, 8, 9, 9, 9, 9, 9, 10, 10)\mathbf{2}6 \\
&\approx 464754723(1, 2, 2, 2, 5, 6, \check{8}, 9, 9, 9, 9, 9, 10, 10)\mathbf{6} \\
&\approx 4647547238(1, 2, 2, 2, 5, 6, 6, 9, 9, 9, 9, 9, 10, 10) = b'C'.
\end{aligned}$$

Here we mention some properties of the BBS with (P, Q)-formulation.

1. In the standard BBS, the P-symbol is a standard tableau, and each Q-symbol of the same shape as P-symbol, contains n different numbers.
2. In the advanced BBS, P-symbol is a general tableau, and each Q-symbol of the same shape as P-symbol contains n different numbers.
3. In the generalized BBS, P-symbol and each Q-symbol of the same shape as P-symbol are both general tableaux.

6. Summary based on examples

Example 9 We consider the following BBS in which the boxes with labels $1, 2, \ldots, 15$ have capacities 3, 4, 1, 3, 2, 3, 2, 1, 5, 2, 1, 6, 3, 15, 7, respectively.

box :	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*
time : 1	$ee5$	$e125$	4	$ee3$	12	$e45$	ee	e	$eeeee$	ee	e	$eeeeee$	eee
time : 2	eee	$eee5$	5	124	$e3$	$ee1$	24	5	$eeeee$	ee	e	$eeeeee$	eee
time : 3	eee	$eeee$	e	$e55$	14	$e23$	ee	e	$e1245$	ee	e	$eeeeee$	eee
time : 4	eee	$eeee$	e	eee	55	$e14$	23	e	$eeeee$	12	4	$eeeee5$	eee
time : 5	eee	$eeee$	e	eee	ee	$e55$	$e4$	1	$eee23$	ee	e	$eee124$	$ee5$

The above result is obtained either by the original algorithm or by the carrier algorithm (recall Proposition 3.2).

Example 10 We next consider the same BBS in the form of biwords.

$$\text{time : 1} \quad \begin{pmatrix} 1 & 2 & 2 & 2 & 3 & 4 & 5 & 5 & 6 & 6 \\ 5 & 1 & 2 & 5 & 4 & 3 & 1 & 2 & 4 & 5 \end{pmatrix}$$

$$\text{time : 2} \quad \begin{pmatrix} 2 & 3 & 4 & 4 & 4 & 5 & 6 & 7 & 7 & 8 \\ 5 & 5 & 1 & 2 & 4 & 3 & 1 & 2 & 4 & 5 \end{pmatrix}$$

$$\text{time : 3} \quad \begin{pmatrix} 4 & 4 & 5 & 5 & 6 & 6 & 9 & 9 & 9 & 9 \\ 5 & 5 & 1 & 4 & 2 & 3 & 1 & 2 & 4 & 5 \end{pmatrix}$$

$$\text{time : 4} \quad \begin{pmatrix} 5 & 5 & 6 & 6 & 7 & 7 & 10 & 10 & 11 & 12 \\ 5 & 5 & 1 & 4 & 2 & 3 & 1 & 2 & 4 & 5 \end{pmatrix}$$

$$\text{time : 5} \quad \begin{pmatrix} 6 & 6 & 7 & 8 & 9 & 9 & 12 & 12 & 12 & 13 \\ 5 & 5 & 4 & 1 & 2 & 3 & 1 & 2 & 4 & 5 \end{pmatrix}$$

The corresponding dual biwords are given as follows:

$$\text{time : 1} \quad \begin{pmatrix} 1 & 1 & 2 & 2 & 3 & 4 & 4 & 5 & 5 & 5 \\ 2 & 5 & 2 & 5 & 4 & 3 & 6 & 1 & 2 & 6 \end{pmatrix}$$

$$\text{time : 2} \quad \begin{pmatrix} 1 & 1 & 2 & 2 & 3 & 4 & 4 & 5 & 5 & 5 \\ 4 & 6 & 4 & 7 & 5 & 4 & 7 & 2 & 3 & 8 \end{pmatrix}$$

$$\text{time : 3} \quad \begin{pmatrix} 1 & 1 & 2 & 2 & 3 & 4 & 4 & 5 & 5 & 5 \\ 5 & 9 & 6 & 9 & 6 & 5 & 9 & 4 & 4 & 9 \end{pmatrix}$$

$$\text{time : 4} \quad \begin{pmatrix} 1 & 1 & 2 & 2 & 3 & 4 & 4 & 5 & 5 & 5 \\ 6 & 10 & 7 & 10 & 7 & 6 & 11 & 5 & 5 & 12 \end{pmatrix}$$

$$\text{time : 5} \quad \begin{pmatrix} 1 & 1 & 2 & 2 & 3 & 4 & 4 & 5 & 5 & 5 \\ 8 & 12 & 9 & 12 & 9 & 7 & 12 & 6 & 6 & 13 \end{pmatrix}$$

In the above, we can check that the time evolution of the bottom rows is also determined by the box-label algorithm (Recall Proposition 3.3). Notice that the initial state of the carrier for the box-label algorithm should be given by

$$C = (1^1, 2^1, 4^2, 6^1, 7^2, 8^1, 9^5, 10^2, 11^1, 12^6, 13^3, 14^{15}, 15^7).$$

Example 11 We finally consider the same BBS in terms of the pairs of tableaux (P, Q). The P-symbol

$$P = \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 2 & 4 & 5 \\ \hline 2 & 3 \\ \cline{1-2} 4 & 5 \\ \cline{1-2} 5 \\ \cline{1-1} \end{array}$$

is conserved under the time evolution of the BBS. The entries (numbers) of this P-symbol are identified with the colors of the balls. The time evolution of the Q-symbol is given as in the figure below.

$$Q_1 = \begin{ytableau} 1 & 2 & 2 & 6 & 6 \\ 2 & 3 \\ 4 & 5 \\ 5 \end{ytableau} \qquad Q_2 = \begin{ytableau} 2 & 3 & 4 & 7 & 8 \\ 4 & 4 \\ 5 & 7 \\ 6 \end{ytableau} \qquad Q_3 = \begin{ytableau} 4 & 4 & 6 & 9 & 9 \\ 5 & 5 \\ 6 & 9 \\ 9 \end{ytableau}$$

$$Q_4 = \begin{ytableau} 5 & 5 & 7 & 12 & 13 \\ 6 & 6 \\ 7 & 10 \\ 10 \end{ytableau} \qquad Q_5 = \begin{ytableau} 6 & 6 & 9 & 12 & 13 \\ 7 & 9 \\ 8 & 12 \\ 12 \end{ytableau}$$

Acknowledgments

The author would like to thank Professors M. Noumi and Y. Yamada for valuable discussions and kind interest. She is the most grateful to the referee for many helpful suggestions for revising this paper.

References

1. K. Fukuda, M. Okado, and Y. Yamada, "Energy functions in box ball systems," *Internat. J. Modern Phys. A* **15**(9) (2000), 1379–1392.
2. W. Fulton, *Young Tableaux*, London Mathematical Society Student Texts, vol. 35, Cambridge University Press, 1997.
3. G. Hatayama, A. Kuniba, and T. Takagi, "Soliton cellular automata associated with finite crystals," *Nulclear Phys. B* **577** (2000), 619–645.
4. D.E. Knuth, *The Art of Computer Programming*, vol. 3, *Sorting and Searching*, Addison-Wesley Series in Computer Science and Information Processing. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1973.
5. I.G. Macdonald, *Symmetric Functions and Hall Polynomials*, 2nd edn., Oxford University Press, 1995.
6. D. Takahashi, "On some soliton systems defined by using boxes and balls," in *Proceedings of the International Symposium on Nonlinear Theory and Its Applications (NOLTA '93)*, 1993, pp. 555–558.
7. D. Takahashi and J. Matsukidaira, "Box and ball system with a carrier and ultradiscrete modified KdV equation," *J. Phys. A* **30** (1997), L733–L739.
8. D. Takahashi and J. Satsuma, "A soliton cellular automaton," *J. Phys. Soc. Jpn.* **59** (1990), 3514–3519.
9. T. Tokihiro, A. Nagai, and J. Satsuma, "Proof of solitonical nature of box and ball systems by means of inverse ultra-discretization," *Inverse Problems* **15**(6) (1999), 1639–1662.
10. M. Torii, D. Takahashi, and J. Satsuma, "Combinatorial representation of invariants of a soliton cellular automaton," *Physica* **D 92** (1996), 209–220.