Bijections between Łukasiewicz walks and generalized tandem walks

Frédéric Chyzak

INRIA
Palaiseau, France
frederic.chyzak@inria.fr

Karen Yeats

Combinatorics and Optimization
University of Waterloo
Waterloo, Canada
kayeats@uwaterloo.ca

Abstract

In this article, we study the enumeration by length of several walk models on the square lattice. We obtain bijections between walks in the upper half-plane returning to the x-axis and walks in the quarter plane. A recent work by Bostan, Chyzak, and Mahboubi has given a bijection for models using small north, west, and south-east steps. We adapt and generalize it to a bijection between half-plane walks using those three steps in two colours and a quarter-plane model over the symmetrized step set consisting of north, north-west, west, south, south-east, and east. We then generalize our bijections to certain models with large steps: for given $p \geqslant 1$, a bijection is given between the half-plane and quarter-plane models obtained by keeping the small south-east step and replacing the two steps north and west of length 1 by the $p + 1$ steps of length p in directions between north and west. This model is close to, but distinct from, the model of generalized tandem walks studied by Bousquet-Mélou, Fusy, and Raschel.

Mathematics Subject Classifications: 05C88, 05C89

1 Introduction

Tandem walks are walks on the square lattice $\mathbb{Z}^2$ that start at the origin and are confined to the quarter plane, with all steps taken in the set

$$S_1 = \{(-1, 0), (0, 1), (1, -1)\},$$

that is, the set consisting of the small steps west, north, and south-east. The name originates from queuing theory: tandem walks indeed represent the behaviour of two M/M/1 queues in tandem. See for example [10, Section 4.7]. Tandem walks are one simple model in a larger family of models of walks on the quarter plane with a prescribed

step set. In 2010, Bousquet-Mélou and Mishna [5] identified 79 essentially different models of walks with step sets consisting of nontrivial steps in $\{-1, 0 + 1\}^2$. These models have since then received a vast amount of attention in the combinatorial literature. From the 79, another class of interest to the present work is obtained by symmetrizing the step set with respect to the line $x = y$, leading to a generalization of tandem walks to walks over the step set

$$S_{1,\text{sym}} = S_1 \cup \{(0, -1), (1, 0), (-1, 1)\},$$

that is, the six small steps west, north, south-east, south, east, and north-west. Both classes are known to have algebraic generating series when enumerated by length (the number of steps). For the former, Gouyou-Beauchamps gave in 1989 an implicit bijection in terms of Motzkin walks and standard Young tableaux [11]. For the latter, Bousquet-Mélou and Mishna gave in 2010 an explicit algebraic calculation of the generating series, by an averaging of a functional equation satisfied by the series [5]. In addition, the last two authors remarked that the symmetric duplication of the step set used to define the second class is likely to be mirrored, in a manner not yet explained at the time, by the factor 2^n that they observed in their enumeration. They asked for a bijective explanation of the phenomenon.

While the bijection suggested by Gouyou-Beauchamps [11] proceeds by composing several known bijections, more recent works by Eu and collaborators [8, 9] introduced a more direct transformation that operates by searching for patterns on the walk viewed as a word and successively rewriting their occurrences until none is found. The patterns used are arbitrarily distant pairs of matching parentheses. Very recently, Bostan, Chyzak, and Mahboubi revisited the problem to get bijections described by single-pass algorithms [3].[1] They developed an approach inspired by the theory of automata: the algorithm for each of the two converse bijections consists of a single unidirectional traversal of the input walk, read step by step by a transducer that additionally maintains two nonnegative integer counters[2]. The evolution of the counters follows very local rules. This more explicit presentation of the bijection also allows for following information about the final position.

The present work adapts the use of automata in [3] to study two models that generalize tandem walks. On the one hand (Section 4), a suitable interlacing of two copies of the bijection between Motzkin walks and tandem walks provides a bijection between bicoloured Motzkin walks and 6-step walks in the quarter plane. This leads to a combinatorial explanation to the 2^n factor for the symmetric step-set model. On the other hand (Section 3), proceeding beyond Motzkin walks and tandem walks, we introduce walks with large nonnegative steps. Given a nonnegative integer p, we consider the class of walks of the upper half plane that use the steps $(1, i)$ for $-1 \leqslant i \leqslant p$ and return to the x-axis, which we call *p-Łukasiewicz walks*. We also introduce the class of walks of the quarter plane that use the small step $(1, -1)$ and the large steps $(-i, p - i)$ for $0 \leqslant i \leqslant p$, which we call *p-tandem walks*. Our principal result is to obtain a bijection between these two

---

[1]In the absence of a written text yet, one can consult [7] for an earlier, related presentation.

[2]The original presentation in [3] uses automata with infinitely many states and no auxiliary memory, but, in view of the generalization to come, we prefer an equivalent presentation with finitely many states and counters that encode the numbering of the states considered in [3].

classes (Theorem 6).

Łukasiewicz walks have been studied for a long time in several forms. They were initially introduced in logic by Łukasiewicz, see, e.g., the book by Rosenbloom [15], and introduced to combinatorics seemingly for the first time by Raney [14]. Traditionally, they are defined as words that correspond to walks confined to the upper half plane, with the exception of their very last step, which has to be $(1, -1)$ and passes below the axis. Keeping the last SE step implicit makes a bijection to our class more direct, and our p-Łukasiewicz walks have been studied for instance by Brak, Iliev, and Prellberg under the name of "$(0, p)$-restricted Łukasiewicz paths" [6], as models of polymer adsorption. The traditional form is better suited to studying their recursive, tree-like structure, and applications to random sampling and random generation of trees are to be found for instance in [2].

Although we claim a proximity of our work to automata theory, we will not use more than its vocabulary in most of the present article. We will indeed describe the procedure implementing our bijections in a more relaxed way, hiding their automatic content in a presentation by "boxes" (see Sections 2–4). The interested reader will wait until Section 5 to see a formal comparison of our presentation to classical automata theory.

A combinatorial explanation of the factor 2^n in the symmetric step-set model was already given by Yeats [17]. Her approach shares similarities with the work by Eu, Fu, Hou, and Hsu for the simpler tandem model. For comparison's sake, we provide it here as well, together with a generalization to p-tandem walks (Section 6).

Remarkably enough, similar bijective results were recently obtained by Bousquet-Mélou, Fusy, and Raschel [4]. There, the authors use a recent bijection between the class of all tandem walks for all p and the class of so-called marked bipolar-oriented planar maps, by Kenyon, Miller, Sheffield, and Wilson [12]. Our bijections are different from those based on maps, with no clearly studied potential connection so far. This is discussed further in Section 7.1.

We see three main benefits to our bijections. First of all, they are described algorithmically with only a single pass through the input word. This makes it clear they have short, fast implementations. The single pass algorithmic description leads to nicer, clearer proofs than the approach of [17], and also made the generalization to larger step sets possible. Then, after the fact, the bijections for larger step sets could also be given an alternate explanation in the spirit of [17]. This gives us two ways, each with different benefits, to understand all of our bijections.

1.1 General setup. Notation

All of our bijections are described algorithmically and proceed in a similar fashion: to compute the image of a given word $w = w_1 \ldots w_n$, a counter c is introduced with values associated to the prefixes of w, and an output word $\bar{w} = \bar{w}_1 \ldots \bar{w}_n$ is computed as the image of w by computing $(\bar{w}_i, c_i)$ from (w_{i-1}, c_{i-1}) for successive values of i. The value of $(\bar{w}_i, c_i)$ depends on (w_{i-1}, c_{i-1}) in a deterministic way, according to a piecewise definition given by possible transitions. We describe such transitions by boxes in what follows.

Steps in 2-dimensional step sets may be depicted by arrows, which then represent vectors with integer coordinates. Step sets made of such steps are named using the letter S. Step sets for 1-dimensional walks are subsets of $\mathbb{Z}$. They are named using the letter Σ. We will freely identify such a step set Σ with the set of vectors $\{1\} \times \Sigma$. For future reference, we summarize the use of these alphabets in the following list, together with the specific classes of words that we will build on them. For this description, given an alphabet A, A^* denotes as usual the set of finite words on A, including the empty word ϵ, and we write $w' \leqslant w$ when w' is a prefix of w, and $|w|_a$ for the number of occurrences of the letter a in w.

- $\Sigma_1 = \{-1, 0, +1\} \simeq \{\searrow, \rightarrow, \nearrow\}$. Motzkin words are those words on Σ_1 whose prefixes contain at least as many up steps as down steps:

$$\mathcal{M} = \{w \in \Sigma_1^* : |w|_{\nearrow} = |w|_{\searrow} \text{ and } \forall w',\ w' \leqslant w \Rightarrow |w'|_{\nearrow} \geqslant |w'|_{\searrow}\}.$$

- $S_1 = \{\uparrow, \leftarrow, \searrow\}$.
    - Tandem words are classically those words on S_1 giving rise to a walk in the quarter plane. We will also call them quarter-plane 1-tandem words:

    $$\mathcal{Q} = \{w \in S_1^* : \forall w',\ w' \leqslant w \Rightarrow |w'|_{\uparrow} \geqslant |w'|_{\searrow} \geqslant |w'|_{\leftarrow}\}.$$

    - By relaxing the quarter-plane constraint, but forcing the final height to be 0, we obtain what we will call half-plane 1-tandem words:

    $$\mathcal{H} = \{w \in S_1^* : |w|_{\uparrow} = |w|_{\searrow} \text{ and } \forall w',\ w' \leqslant w \Rightarrow |w'|_{\uparrow} \geqslant |w'|_{\searrow}\}.$$

    The bijection $\mathcal{M} \simeq \mathcal{H}$ is obvious.

- $\{1, 2, 3\}$. We restrict the classical class of Yamanouchi words (using any letters from $\mathbb{N}^*$) to those words using only the first three integers:

$$\mathcal{Y}_3 = \{w \in \{1, 2, 3\}^* : \forall w',\ w' \leqslant w \Rightarrow |w'|_1 \geqslant |w'|_2 \geqslant |w'|_3\}.$$

The bijection $\mathcal{Y}_3 \simeq \mathcal{Q}$ is obvious.

- $S_{1,\text{sym}} = \{\uparrow, \leftarrow, \searrow, \downarrow, \rightarrow, \nwarrow\}$. By augmenting the step set of classical tandem walks by symmetry, we obtain a superset $\mathcal{Q}^{\text{sym}}$ of $\mathcal{Q}$:

$$\mathcal{Q}^{\text{sym}} = \{w \in S_{1,\text{sym}}^* : \forall w',\ w' \leqslant w \Rightarrow$$
$$|w'|_{\uparrow} + |w'|_{\nwarrow} \geqslant |w'|_{\searrow} + |w'|_{\downarrow} \text{ and } |w'|_{\nwarrow} + |w'|_{\rightarrow} \geqslant |w'|_{\leftarrow} + |w'|_{\searrow}\}.$$

- $\Sigma_{1,\text{bicol}} = \{-\mathbf{1}, \mathbf{0}, +\mathbf{1}, -\mathbb{1}, \mathbb{0}, +\mathbb{1}\} \simeq \{\nearrow, \rightarrow, \searrow, \nearrow, \rightarrow, \searrow\}$. For a bicolouring of Motzkin words, we use the two colours "solid" and "striped":

$$\mathcal{M}^{\text{bicol}} = \{w \in \Sigma_{1,\text{bicol}}^* : |w|_{\nearrow} + |w|_{\nearrow} = |w|_{\searrow} + |w|_{\searrow} \text{ and}$$
$$\forall w',\ w' \leqslant w \Rightarrow |w'|_{\nearrow} + |w'|_{\nearrow} \geqslant |w'|_{\searrow} + |w'|_{\searrow}\}.$$

- $\Sigma_p = \{-1, 0, 1, \ldots, p\}$. Our p-Łukasiewicz words generalize Motzkin words by allowing longer up steps:

$$\mathcal{L}_p = \{w \in \Sigma_p^* : |w|_1 + 2|w|_2 + \cdots + p|w|_p = |w|_{-1}$$
$$\forall w', \ w' \leqslant w \Rightarrow |w'|_1 + 2|w'|_2 + \cdots + p|w'|_p \geqslant |w'|_{-1}\}.$$

  In particular, $\mathcal{L}_1 = \mathcal{M}$.

- $S_p = \{\boxed{\uparrow}, \ldots, \boxed{\leftarrow}, \ldots, \boxed{\leftarrow}, \boxed{\lrcorner}\}$. Our quarter-plane p-tandem words generalize classical tandem words by allowing longer steps into the north-west quadrant:

$$\mathcal{T}_p = \{w \in S_p^* : \forall w', \ w' \leqslant w \Rightarrow$$
$$\textstyle\sum_{j=0}^p (p-j)|w'|_{(-j,p-j)} \geqslant |w'|_{(1,-1)} \geqslant \sum_{j=0}^p j|w'|_{(-j,p-j)}\}.$$

  In particular, $\mathcal{T}_1 = \mathcal{Q}$.

- $A_p := \{a_{\ell,m} : 0 \leqslant \ell + m \leqslant p - 1\}$. We will use the class A_p^* of unrestricted finite words on A_p to represent generalized counters (stack contents).

1.2 Outline of the paper. Contributions

Section 2 discusses the case $p = 1$. It is bibliographical, along the lines of [3, 7]. (But we prefer the present description in terms of counters over the equivalent description in terms of infinite automata developed in [3].) There, we provide a bijection $\mathcal{H} \simeq \mathcal{Q}$ between half-plane and quarter-plane words in the 1-tandem case. The bijection is essentially the bijection $\mathcal{M} \simeq \mathcal{Y}_3$ between Motzkin words and (restricted) Yamanouchi words, owing to the trivial relations $\mathcal{M} \simeq \mathcal{H}$ and $\mathcal{Y}_3 \simeq \mathcal{Q}$. This description is given without proofs, as it can be viewed as a special case of the case of general p. This new result is a bijection $\mathcal{L}_p \simeq \mathcal{T}_p$ for $1 \leqslant p \in \mathbb{N}$. It is developed in full in Section 3. For the proofs, we introduce *parameters* attached to the constructions, and we study their variations and invariants. We then develop the 6-step model and the corresponding bijection $\mathcal{M}^{\mathrm{bicol}} \simeq \mathcal{Q}^{\mathrm{sym}}$ in a more sketchy way in Section 4, providing just what has to be adapted to obtain complete proofs. Alternate interpretations follow, via explicit automata in Section 5 and via raisings of steps in Section 6.

Although we have developed the automata approach of the present article as a more intuitive explanation of the phenomenon for some of us, other readers may prefer the approach by raising followed in [17]. To save those readers from waiting until Section 6, we work out in Section 1.3 an example of execution of the automata and raising algorithms to transform an explicit quarter-plane walk to the corresponding Motzkin walk. We provide this comparison without any formalism and any proof of the bijections. We thus propose two reads of the article. Most readers will consider a linear read of it, possibly skipping Section 1.3 and postponing its read until after reading Section 6. On the other hand, readers already aware of the raising approach will probably prefer reading the worked example in Section 1.3 before jumping directly to Section 6 and returning to Section 2 when inclined to do so.

The interested reader will find a Julia implementation of the algorithms described in this article, both on Chyzak's web page and as ancillary files to our arXiv preprint.

1.3 Worked example

The difference between the traditional approach by raising and our new approach by transducers is most striking in the divergence of their intermediate calculations when converting an input walk in the quarter plane with steps from $\{\boxed{\uparrow},\boxed{\leftarrow},\boxed{\searrow}\}$ into a walk in the upper half plane using the same steps and returning to the x-axis, or, equivalently, into a Motzkin walk using steps from $\{\boxed{\searrow},\boxed{\rightarrow},\boxed{\nearrow}\}$. Both approaches realize the map ψ that we will define and describe using transducers in Section 3, and that we will describe using raising in Section 6.1.

In order to transform a quarter-plane walk, like the one at the bottom of column (b), the traditional approach is to read its steps from beginning to end, inducing the prefixes that can be seen from top to bottom in that column, and to progressively generate the corresponding Motzkin walks in column (a). To maintain the Motzkin property, this induces the so-called raising of an earlier step when the added step would otherwise go below the x-axis. Such raising instances are marked by round arrows and shift the rest of the walk vertically by one. The global transformation returns the final Motzkin walk, given at the bottom of column (a).

The approach introduced in the present work is to read the quarter-plane walk backwards, thus considering the suffixes that appear from bottom to top in column (c). This family of walks has no obvious property with regard to a quarter plane or even a half plane, and so we have not bothered to translate them in any way, so as to have the origin as final vertex for example. The new transformation is depicted in column (d), which shows suffixes of the final Motzkin walk. Observe that those suffixes are not altered after they have been produced in any other way than prepending another step. In the end, the completed Motzkin walks obtained by both algorithms and drawn at the bottom of column (a) and top of column (d) are the same. If we insist on writing upper-half-plane walks in terms of the same allowed steps as for the quarter-plane walks, we obtain in column (e) the suffixes of an upper-half-plane walk returning to the x-axis. The fact that the complete walk, at the top of column (e), returns to the origin is specific to the example, in relation to the quarter-plane walk ending on the main diagonal.

2 Three-step model

We start with the three-step model, to obtain a bijection $\mathcal{H} \simeq \mathcal{Q}$, which we realize as two converse bijective maps $\phi : \mathcal{H} \to \mathcal{Q}$ and $\mathcal{Q} \to \mathcal{H}$. This implicitly contains the bijection $\mathcal{M} \simeq \mathcal{Y}_3$, which is the foundation of the case $p = 1$ in the literature.

Following the strategy announced in the introduction, we introduce the transitions that we use to define the forward map ϕ and the backward map ψ. This is done by the 6 boxes on the right of Figure 2 and the 5 boxes on the right of Figure 3. We describe the meaning of the boxes using the notation introduced by the box (TR) on their left. There:

- $\boxed{a}$ and $\boxed{\bar{a}}$ stand for letters from the alphabet $S_1 := \{\boxed{\uparrow},\boxed{\leftarrow},\boxed{\searrow}\}$;

- μ is the y-coordinate of the letter a, that is, the vertical displacement caused by a,

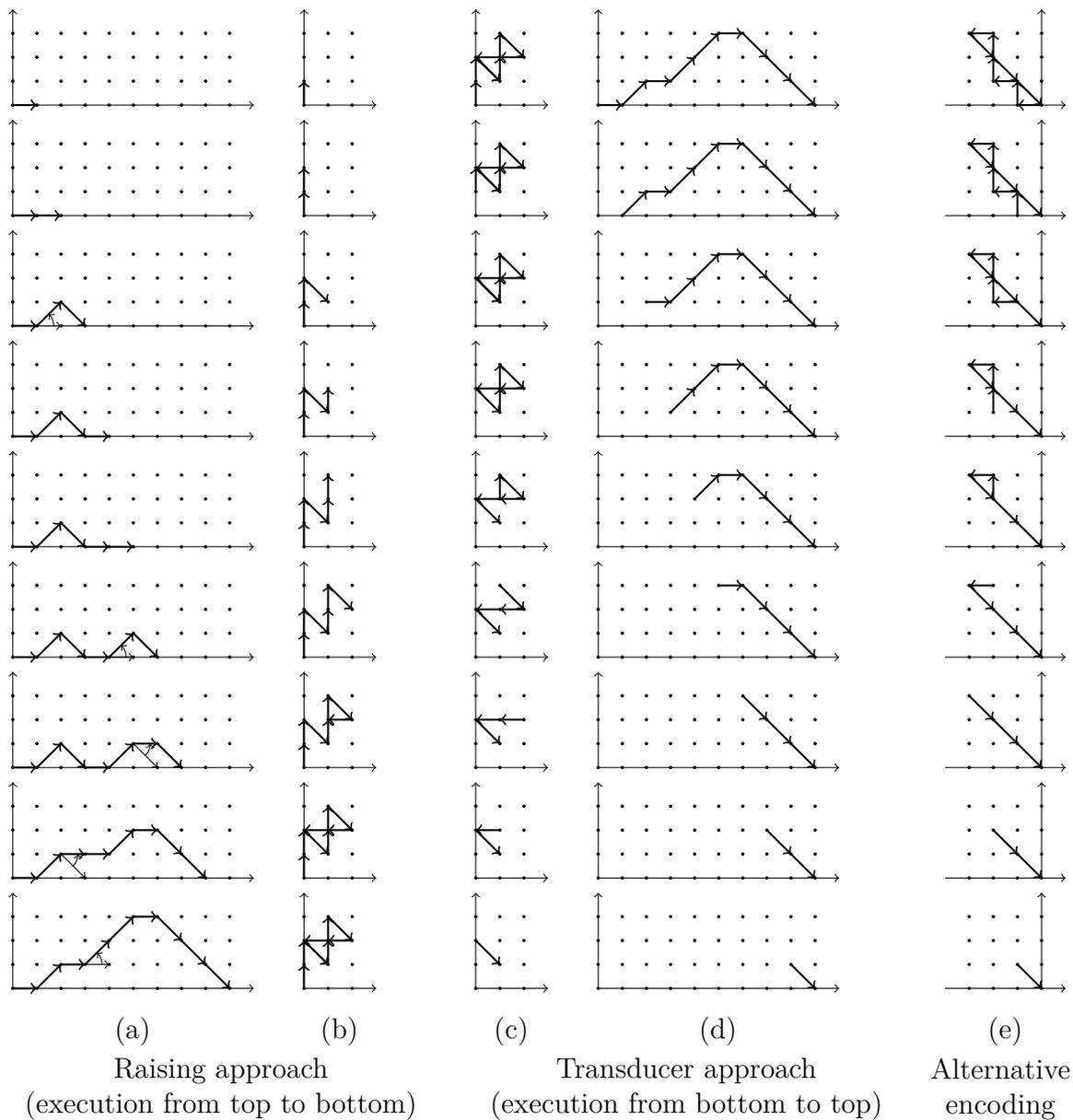|				
(a)	(b)	(c)	(d)	(e)
Raising approach (execution from top to bottom)		Transducer approach (execution from bottom to top)		Alternative encoding

Figure 1: Computing the Motzkin walk associated with a given quarter-plane walk: comparison of the traditional raising approach and the new transducer approach. *Raising algorithm*: input on the right (b), output on the left (a). These columns (a) and (b) display the same data as Figure 16. This exemplifies the algorithm provided formally as Figure 15 in Section 6.1, with the exception that the raisability marks are not shown here. *Transducer approach*: input on the left (c), output on the right (d). This algorithm is provided formally in Section 2 (map ψ) and generalized to long up steps in Section 3 (map Ψ_p). Column (e) provides the walks homologous to those of column (d) under the trivial bijection that exchanges $\boxed{\uparrow}$ and $\boxed{\nearrow}$, and $\boxed{\leftarrow}$ and $\boxed{\rightarrow}$.

(TR)	(U1)	(L2)	(L1)	(D3)	(D2)	(DE)
a	↑	←	←	↘	↘	↘
μ	$+1$	0	0	-1	-1	-1
m	↗	→	→	↘	↘	↘
$h_L\ h_R$ / $v_L\ v_R$	$\alpha\ \ \alpha$ / $\beta\ \ \beta+1$	$\alpha\ \ \alpha+1$ / $\beta+1\ \ \beta$	$\alpha\ \ \alpha$ / $0\ \ 0$	$\alpha+1\ \ \alpha$ / $\beta\ \ \beta$	$0\ \ 0$ / $\beta+1\ \ \beta$	$0\ \ -$ / $0\ \ -$
χ	1	2	1	3	2	err.
$\bar{a}$	↑	↘	↑	←	↘	
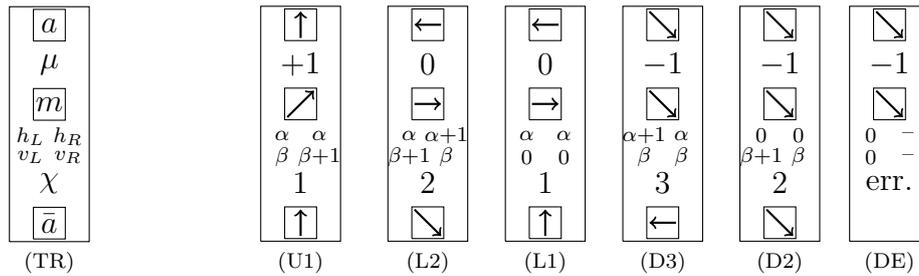
Figure 2: Transitions used to define the forward map ϕ: the letter $\boxed{m}$ is read from the input Motzkin word w; the letter χ is written to the output Yamanouchi word $\bar{w}$. At the top, the rows $\boxed{a}$ and μ present alternate encodings of $\boxed{m}$ under obvious bijections. At the bottom, the row $\boxed{\bar{a}}$ presents an alternate encodings of χ. A transition named (Xχ) reads a letter $\boxed{m}$ indicated by X: respectively ↗, →, ↘, if X stands for U(p), L(evel), D(own). With the exception of (DE), all transitions (Xχ) output the letter χ.

with value μ in $\{-1, 0, +1\}$;

- $\boxed{m}$ stands for a letter from the alphabet $\Sigma_1 := \{↗, →, ↘\}$;

- (h_L, v_L) and (h_R, v_R) are pairs of counters, from $\mathbb{N}^2$ (henceforth, we will simply say "counter" for short, instead of "pair of counters");

- χ stands for an integer from $\{1, 2, 3\}$.

The forward map ϕ converts a walk of the 3-step model $\mathcal{H}$ in the upper half plane to a walk of the 3-step model $\mathcal{Q}$ in the quarter plane. To ease its definition, the map is defined as a partial function of any word from S_1^*. This is convenient because the construction potentially leads to an error. The map is defined in terms of the 6 forward transitions (U1) to (DE) listed in Figure 2. Any application of (U1) to (D2) consumes a letter $\boxed{a}$ of the 3-step model as well as a counter $(h_L, v_L) \in \mathbb{N}^2$, and produces a new counter (h_R, v_R) together with a letter $\boxed{\bar{a}}$ of the 3-step model. The values of the counters (h_L, v_L) and (h_R, v_R) in the boxes are given as patterns, where α and β have to be viewed as matching nonnegative integers. So, for example, if the input step is ← and the counters are (h_L, v_L), then we use transition (L2) if $v_L > 0$ and (L3) if $v_L = 0$. The special transition (DE) results in an error, instead of producing a new counter and a new letter. The triples of consumed letters $\boxed{a}$ and patterns for h_L and v_L over all 6 boxes of Figure 2 cover all possible input cases from $S_1 \times \mathbb{N}^2$.

Given a word $w = w_1 \dots w_n \in S_1^*$ of length n, the forward transform $\phi(w)$ is now defined as follows. Set $(h_0^{\mathrm{LR}}, v_0^{\mathrm{LR}}) = (0, 0)$. For i from 1 to n, find the only forward transition that can be applied to the letter w_i and the counter $(h_{i-1}^{\mathrm{LR}}, v_{i-1}^{\mathrm{LR}})$. If this is (DE), then ϕ is not defined at w. Otherwise, apply the transition to set $(h_i^{\mathrm{LR}}, v_i^{\mathrm{LR}})$ to the returned counter and $\bar{w}_i$ to the returned letter. If the loop terminates without using the transition (DE), then $\phi(w)$ is defined as $\bar{w}_1 \dots \bar{w}_n$.
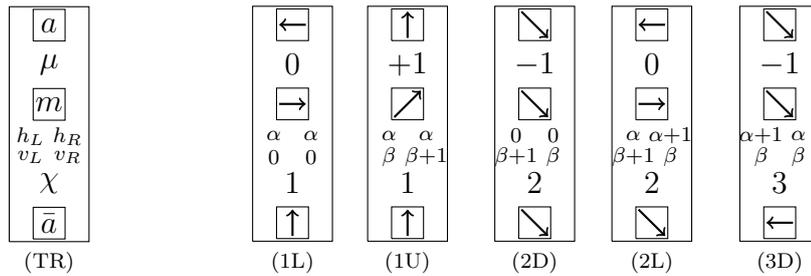
(TR)	(1L)	(1U)	(2D)	(2L)	(3D)
$\boxed{a}$	←	↑	↘	←	↘
μ	0	+1	−1	0	−1
$\boxed{m}$	→	↗	↘	→	↘
$\begin{smallmatrix}h_L & h_R\\ v_L & v_R\end{smallmatrix}$	$\begin{smallmatrix}\alpha & \alpha\\ 0 & 0\end{smallmatrix}$	$\begin{smallmatrix}\alpha & \alpha\\ \beta & \beta+1\end{smallmatrix}$	$\begin{smallmatrix}0 & 0\\ \beta+1 & \beta\end{smallmatrix}$	$\begin{smallmatrix}\alpha & \alpha+1\\ \beta+1 & \beta\end{smallmatrix}$	$\begin{smallmatrix}\alpha+1 & \alpha\\ \beta & \beta\end{smallmatrix}$
χ	1	1	2	2	3
$\boxed{\bar{a}}$	↑	↑	↘	↘	←

Figure 3: Transitions used to define the backward map ψ. Each transition named (χX) is the reverse of the transition named $(X\chi)$ in Figure 2. The letter χ is read from the input Yamanouchi word $\bar{w}$; the letter $\boxed{m}$ is written to the output Motzkin word w. Alternate representations are as described in that figure.

The backward map ψ is defined similarly using the backward transitions (1L) to (3E) listed in Figure 3, with the added simplicity that no error can occur. Applications of backward transitions consume a letter $\boxed{\bar{a}}$ and a counter (h_R, v_R), and produce a counter (h, v) together with a letter $\boxed{a}$. Again, the triples $(\boxed{\bar{a}}, h_R, v_R)$ cover all possible input cases from $S_1 \times \mathbb{N}^2$. (For other models in future sections, we will not list the backward transitions separately since they are the same boxes interpreted in reverse.)

Given a word $\bar{w} = \bar{w}_1 \ldots \bar{w}_n \in S_1^*$ of length n, the backward transform $\psi(\bar{w})$ is now defined as follows. Set $(h_n^{\mathrm{RL}}, v_n^{\mathrm{RL}}) = (0, 0)$. For i from n down to 1, find the only backward transition that can be applied to the letter $\bar{w}_i$ and the counter $(h_i^{\mathrm{RL}}, v_i^{\mathrm{RL}})$, so as to define a letter w_i and a counter $(h_{i-1}^{\mathrm{RL}}, v_{i-1}^{\mathrm{RL}})$. Then $\psi(\bar{w})$ is defined as $w_1 \ldots w_n$.

One can prove that, after calculating some $\phi(w)$, the last counter $(h_n^{\mathrm{LR}}, v_n^{\mathrm{LR}})$ is $(0, 0)$ if and only if w is a Motzkin word. Likewise, after calculating some $\psi(\bar{w})$, the last counter $(h_0^{\mathrm{RL}}, v_0^{\mathrm{RL}})$ is $(0, 0)$ if and only if $\bar{w}$ is a quarter-plane word. Also, one can prove that if some $(h_i^{\mathrm{LR}}, v_i^{\mathrm{LR}}, \bar{w}_i)$ is the result of applying the forward transition (U1), then the only backward transition that can be applied to this result, in a right to left process having this specific letter $\bar{w}_i$ at position i, is (1U). Like properties hold for all other natural pairs of forward and backward transitions. The proof that ϕ and ψ are inverse bijections between Motzkin words of $\mathcal{M}$ and quarter-plane words of $\mathcal{Q}$ follows from this local reversibility property. We do not exhibit those proofs here as they will be seen as special cases of the case of a general p in Section 3.

Because of an interpretation specific to the case $p = 1$ and in view of the generalization to larger p in Section 3, the input and output letters are also given in the boxes in different alphabets. Three-step walks in the upper half plane are in bijection with Motzkin words, by replacing each ↑ with a ↗, each ← with a →, and leaving all ↘'s unchanged. The letter $\boxed{m}$ corresponding to $\boxed{a}$ is given in the boxes, together with its vertical displacement μ. Here, the letters are those for Motzkin walks, implying $-1 \leqslant \mu \leqslant 1$, but the bijection in Section 3 will use $-1 \leqslant \mu \leqslant p$, so this notation will be preferred. On the other hand, the constraint on a 3-step walk to remain in the quarter plane leads to each prefix of such a 3-step walk having more ↑'s than ↘'s and more ↘'s than ←'s. Replacing ↑, ↘, and ← respectively with 1, 2, and 3 puts walks in the quarter plane in bijection
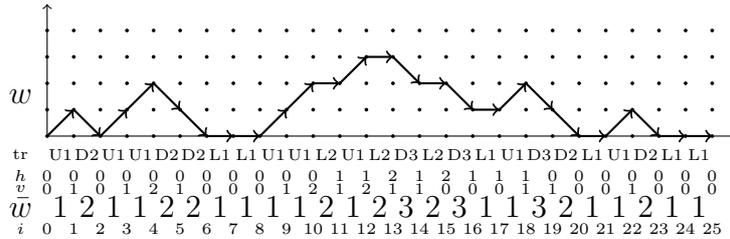
Figure 4: An example of a Motzkin walk w (letters $\boxed{m}$) and of a Yamanouchi word $\bar{w}$ (letters χ), both of length 25, associated with one another by the maps ϕ and ψ. The row marked 'tr' indicates the transitions used by the LR process, computing $\bar{w}$ from w. For the computation of w from $\bar{w}$ by the RL process, each transition is replaced by its reverse form, according to Figure 2.
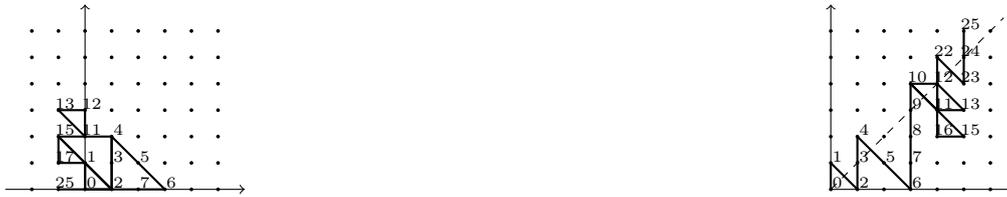

Figure 5: Motzkin walk of Figure 4 interpreted as an upper-half-plane walk (left) and Yamanouchi word of Figure 4 interpreted as a quarter-plane walk (right).

with Yamanouchi words. The bijection for $p = 1$ is thus essentially a bijection between Motzkin words and Yamanouchi words using only 1s, 2s, and 3s.

Example 1. Figure 4 is an example of the bijection, expressed as a relation between a Motzkin walk and a Yamanouchi word. The letters $w_i = \boxed{m}$ (at the top) have been glued into a walk w. The corresponding letters $\bar{w}_i = \chi$ are given at the bottom, below the counters h and v. The index i on the final row mark the "time" in the algorithmic transformations. Specifically, the value of h, respectively, of v, in the same column as a value of i are the values of h_i^{LR} or h_i^{RL}, respectively, of v_i^{LR} or v_i^{RL}, in our previous description. Pay attention to how a value $\chi = \bar{w}_i$ appears between times $i-1$ and i.

Interpreted as a bijection between walks in the 3-step models, the bijection maps the upper-half-plane walk on the left of Figure 5 to the quarter-plane walk on the right. To help the reader understand the order of these tangled walks, we have put the time of first reach close to each position. This corresponds to the value of i in Figure 4. Additionally, for the upper-half-plane walk, 8, 20 and 23 coincide with 2, 9 with 3, 10 with 4, 14 with 11, 16, 19 and 22 with 1, 18 with 15, 21 and 24 with 0; for the quarter-plane walk, 14, 17 and 20 coincide with 11, 19 with 10, 18 and 21 with 12.

An interesting property is made apparent by this example. Focus on those positions between times i and $i+1$ for which $\boxed{m} = \boxed{\rightarrow}$ in the Motzkin walk and for which $\chi = 1$ in the Yamanouchi word. In other words, consider $i \in \{6, 7, 16, 20, 23, 24\}$, but not $i \in \{10, 12, 14\}$. Then, we observe that the suffix of the quarter-plane walk for the position at that time i never goes back to a lower abscissa than that of its vertex marked i. (On

the other hand, this is not true for 10, 12, 14.) See the first point of Proposition 7.

The reader in need of an early intuition about our bijections between models with long steps can compare Figure 4 (three-step model, $p = 1$) and Figure 11 ($p = 5$).

3 p-Tandem Walks

In this section, we generalize the bijections of Section 2 to general $p \geqslant 1$.

3.1   Construction of maps between Σ_p^* and S_p^*

For the description of the bijection for general p, we proceed to introduce transitions akin to those for the case $p = 1$, but with two kinds of generalizations needed.

- The letters $\boxed{\nearrow}$, $\boxed{\rightarrow}$, and $\boxed{\searrow}$, used to encode Motzkin words for $p = 1$ and also represented by their y-coordinates (that is, their vertical displacements), respectively as $+1$, 0, -1, need to be generalized to a set of $p + 2$ letters μ, named by numbers from -1 to p.

- The letters $\boxed{\uparrow}$, $\boxed{\leftarrow}$, and $\boxed{\searrow}$, used for the 3-step quarter-plane model, need to be generalized to another set of $p+2$ letters $\bar{a}$: the $p+1$ vectors $(-\ell, p-\ell)$ for $0 \leqslant \ell \leqslant p$, together with the vector $(1, -1)$. In the boxes of Figure 6, the parameter $\bar{\mu}$ is the y-coordinate of $\bar{a}$. It also varies from -1 to p and could be used as another encoding.

- The parameter $h \in \mathbb{N}$ of the case $p = 1$ needs to be generalized to a word H over the alphabet A_p consisting of the $p(p-1)/2$ letters $a_{\ell,m}$ for $0 \leqslant \ell + m \leqslant p-1$. For $p = 1$, this alphabet reduces to a single letter $a_{0,0}$, so that the values of H for $p = 1$ are in natural bijection with the possible values of $h \in \mathbb{N}$. This explains why a mere integer was sufficient in Section 2.

We could find no better way to name the transitions in Figure 6 than simply naming them from (T1) to (T8). We describe their meaning using the notation introduced by the box (TG). There:

- μ is the y-coordinate of a letter $(1, \mu)$ of a Łukasiewicz word, that is, the vertical displacement caused by it, with value in $\Sigma_p := \{-1, 0, 1, \ldots, p\}$;

- (H_L, v_L) and (H_R, v_R) are generalized counters, from $A_p^* \times \mathbb{N}$ where A_p is the alphabet $\{a_{\ell,m} : 0 \leqslant \ell + m \leqslant p - 1\}$;

- $\boxed{\bar{a}}$ stands for a letter from the alphabet $S_p := \{\boxed{\uparrow}, \ldots, \boxed{\leftarrow}, \ldots, \boxed{\leftarrow}, \boxed{\lrcorner}\}$;

- $\bar{\mu}$ also stands for a letter from the alphabet Σ_p.

As in the definition of the forward map ϕ in Section 2, those transitions are used in a forward manner to produce a counter $(H_i^{\mathrm{LR}}, v_i^{\mathrm{LR}})$ and a letter $\bar{w}_i$ from a given letter w_i and a counter $(H_{i-1}^{\mathrm{LR}}, v_{i-1}^{\mathrm{LR}})$. The procedure, given in Figure 7, explains how transitions are
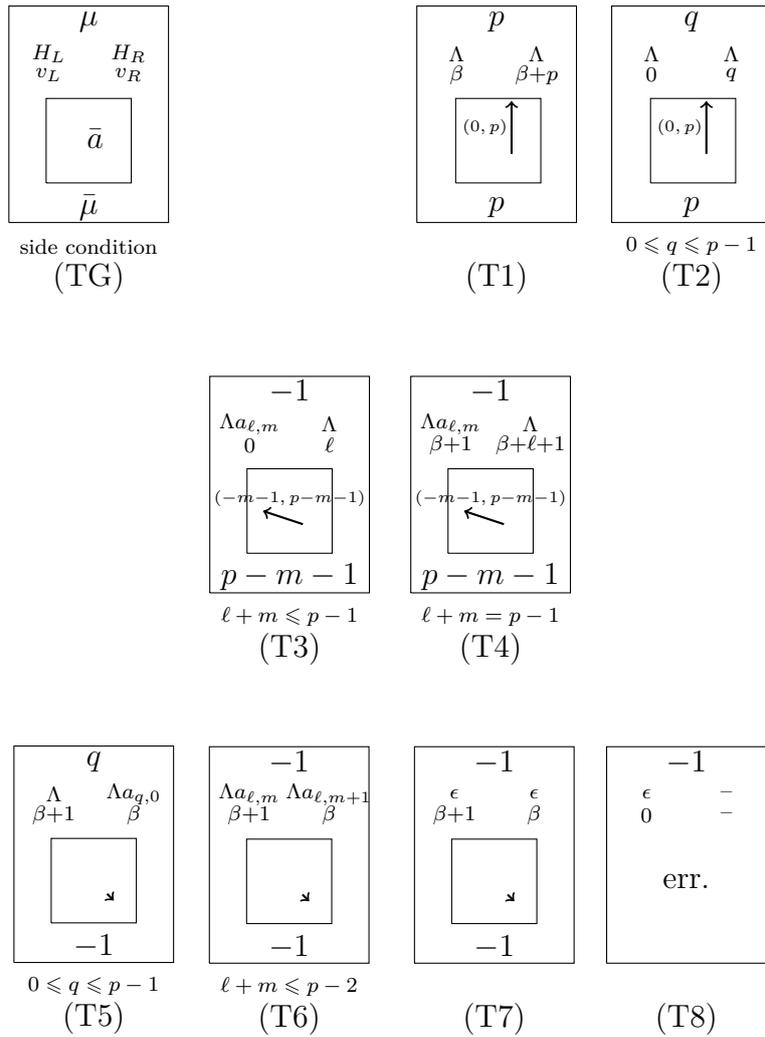
Figure 6: Transitions used for the bijections for general p. When $p = 1$, (T3) and (T6) disappear, and (T1), (T2), (T4), (T5), (T7), (T8), respectively, become (U1), (L1), (D3), (L2), (D2), (DE), respectively, in Figure 2.

selected unambiguously. We prove that, given an input $(w_i, H_{i-1}^{\mathrm{LR}}, v_{i-1}^{\mathrm{LR}})$ from $\Sigma_p \times A_p^* \times \mathbb{N}$, this procedure returns an output $(H_i^{\mathrm{LR}}, v_i^{\mathrm{LR}}, \bar{w}_i)$ from $A_p^* \times \mathbb{N} \times S_p$, unless it applies (T8). By inspection:

- $\bar{w}_i$ is either $(0, p)$, or $(1, -1)$, or, at steps (3)(b) and (4)(b)(i), $(-m-1, p-m-1)$ for $0 \leqslant m \leqslant p - 1 - \ell$. In all cases, $\bar{w}_i$ is in Σ_p.

- v_i^{LR} is the sum of one or two nonnegative integers, except at steps (2)(b), (4)(a), and (4)(b)(ii), where is it $v_{i-1}^{\mathrm{LR}} - 1$, for positive v_{i-1}^{LR}. In all cases, v_i^{LR} is in $\mathbb{N}$.

- H_i^{LR} is always a prefix of H_{i-1}^{LR}, possibly augmented by a letter b: at step (2)(b), by $b = a_{w_i,0}$ when $0 \leqslant w_i \leqslant p - 1$; at step (4)(b)(ii), by $b = a_{\ell,m+1}$ when $0 \leqslant \ell + m \leqslant p - 2$. In all cases, H_i^{LR} is in A_p^*.

Input: $(w_i, H_{i-1}^{\mathrm{LR}}, v_{i-1}^{\mathrm{LR}}) \in \Sigma_p \times A_p^* \times \mathbb{N}$, where w_i is the letter input in the context of counters equal to $(H_{i-1}^{\mathrm{LR}}, v_{i-1}^{\mathrm{LR}})$

Output: $(H_i^{\mathrm{LR}}, v_i^{\mathrm{LR}}, \bar{w}_i) \in A_p^* \times \mathbb{N} \times S_p$, where $\bar{w}_i$ is the letter output while changing the counters to the value $(H_i^{\mathrm{LR}}, v_i^{\mathrm{LR}})$

---

1. if $w_i = p$, apply (T1) to return $(H_i^{\mathrm{LR}}, v_i^{\mathrm{LR}}) = (H_{i-1}^{\mathrm{LR}}, v_{i-1}^{\mathrm{LR}} + p)$ and $\bar{w}_i = (0, p)$,

2. if $w_i \geqslant 0$, then
   (a) if $v_{i-1}^{\mathrm{LR}} = 0$, apply (T2) to return $(H_i^{\mathrm{LR}}, v_i^{\mathrm{LR}}) = (H_{i-1}^{\mathrm{LR}}, w_i)$ and $\bar{w}_i = (0, p)$,
   (b) otherwise, apply (T5) to return $(H_i^{\mathrm{LR}}, v_i^{\mathrm{LR}}) = (H_{i-1}^{\mathrm{LR}} a_{w_i,0}, v_{i-1}^{\mathrm{LR}} - 1)$ and $\bar{w}_i = (1, -1)$,

3. if $v_{i-1}^{\mathrm{LR}} = 0$, then
   (a) if $H_{i-1}^{\mathrm{LR}} = \epsilon$, apply (T8) and quit on error,
   (b) otherwise, write $H_{i-1}^{\mathrm{LR}} = H' a_{\ell,m}$, then apply (T3) to return $(H_i^{\mathrm{LR}}, v_i^{\mathrm{LR}}) = (H', \ell)$ and $\bar{w}_i = (-m-1, p-m-1)$,

4. otherwise,
   (a) if $H_{i-1}^{\mathrm{LR}} = \epsilon$, apply (T7) to return $(H_i^{\mathrm{LR}}, v_i^{\mathrm{LR}}) = (\epsilon, v_{i-1}^{\mathrm{LR}} - 1)$ and $\bar{w}_i = (1, -1)$,
   (b) otherwise, write $H_{i-1}^{\mathrm{LR}} = H' a_{\ell,m}$, then
      i. if $\ell + m = p - 1$, apply (T4) to return $(H_i^{\mathrm{LR}}, v_i^{\mathrm{LR}}) = (H', v_{i-1}^{\mathrm{LR}} + \ell)$ and $\bar{w}_i = (-m-1, p-m-1)$,
      ii. otherwise, apply (T6) to return $(H_i^{\mathrm{LR}}, v_i^{\mathrm{LR}}) = (H' a_{\ell,m+1}, v_{i-1}^{\mathrm{LR}} - 1)$ and $\bar{w}_i = (1, -1)$.

Figure 7: One step of the LR process (ith step).

As in the case $p = 1$ of Section 2, we next define a (partial) map Φ_p from Σ_p^* to S_p^*. Given an input word $w = w_1 \dots w_n$ from Σ_p^*, iterating the process above from the initial counter $(H_0^{\mathrm{LR}}, v_0^{\mathrm{LR}}) = (\epsilon, 0)$ results either in an error, or in a word $\bar{w} = \bar{w}_1 \dots \bar{w}_n$. In the former case, Φ_p remains undefined at w, while in the latter case, $\Phi_p(w)$ is defined as $\bar{w}$.

Likewise, as in the definition of the backward map ψ in Section 2, the transitions (T1) to (T8) in Figure 6 are used in a backward manner to produce a counter $(H_{i-1}^{\mathrm{RL}}, v_{i-1}^{\mathrm{RL}})$ and a letter w_i from a given letter $\bar{w}_i$ and a counter $(H_i^{\mathrm{RL}}, v_i^{\mathrm{RL}})$. We provide the procedure in Figure 8 to explain how transitions, meant to be applied backwards, are selected unambiguously. We prove that, given an input $(H_i^{\mathrm{RL}}, v_i^{\mathrm{RL}}, \bar{w}_i)$ from $A_p^* \times \mathbb{N} \times S_p$, this procedure returns an output $(w_i, H_{i-1}^{\mathrm{RL}}, v_{i-1}^{\mathrm{RL}})$ from $\Sigma_p \times A_p^* \times \mathbb{N}$. By inspection:

Input: $(H_i^{\mathrm{RL}}, v_i^{\mathrm{RL}}, \bar{w}_i) \in A_p^* \times \mathbb{N} \times S_p$, where $\bar{w}_i$ is the letter input in the context
　　　　of counters equal to $(H_i^{\mathrm{LR}}, v_i^{\mathrm{LR}})$

Output: $(w_i, H_{i-1}^{\mathrm{RL}}, v_{i-1}^{\mathrm{RL}}) \in \Sigma_p \times A_p^* \times \mathbb{N}$, where w_i is the letter output while
　　　　changing the counters to the value $(H_{i-1}^{\mathrm{LR}}, v_{i-1}^{\mathrm{LR}})$

---

1. if $\bar{w}_i = (0, p)$, then
   (a) if $v_i^{\mathrm{RL}} \leqslant p - 1$, apply (T2) to return $(H_{i-1}^{\mathrm{RL}}, v_{i-1}^{\mathrm{RL}}) = (H_i^{\mathrm{RL}}, 0)$ and $w_i = v_i^{\mathrm{RL}}$,
   (b) otherwise, apply (T1) to return $(H_{i-1}^{\mathrm{RL}}, v_{i-1}^{\mathrm{RL}}) = (H_i^{\mathrm{RL}}, v_i^{\mathrm{RL}} - p)$ and $w_i = p$,

2. if $\bar{w}_i \neq (1, -1)$, write $\bar{w}_i = (\bar{\mu} - p, \bar{\mu})$ for $\bar{\mu} = p - m - 1$, then
   (a) if $v_i^{\mathrm{RL}} =: \ell \leqslant \bar{\mu}$, apply (T3) to return $(H_{i-1}^{\mathrm{RL}}, v_{i-1}^{\mathrm{RL}}) = (H_i^{\mathrm{RL}} a_{\ell,m}, 0)$ and $w_i = -1$,
   (b) apply (T4) to return $(H_{i-1}^{\mathrm{RL}}, v_{i-1}^{\mathrm{RL}}) = (H_i^{\mathrm{RL}} a_{\bar{\mu},m}, v_i^{\mathrm{RL}} - \bar{\mu})$ and $w_i = -1$,

3. if $H = \epsilon$, apply (T7) to return $(H_{i-1}^{\mathrm{RL}}, v_{i-1}^{\mathrm{RL}}) = (\epsilon, v_i^{\mathrm{RL}} + 1)$ and $w_i = -1$,

4. otherwise, write $H_i^{\mathrm{RL}} = H' a_{\ell,m}$, then
   (a) if $m = 0$, apply (T5) to return $(H_{i-1}^{\mathrm{RL}}, v_{i-1}^{\mathrm{RL}}) = (H', v_i^{\mathrm{RL}} + 1)$ and $w_i = \ell$,
   (b) otherwise, apply (T6) to return $(H_{i-1}^{\mathrm{RL}}, v_{i-1}^{\mathrm{RL}}) = (H' a_{\ell,m-1}, v_i^{\mathrm{RL}} + 1)$ and $w_i = -1$.

Figure 8: One step of the RL process $((n + 1 - i)$th step).

- w_i is either p or -1, or, at steps (1)(a) and (4)(a), an integer between 0 and $p - 1$. In all cases, w_i is in Σ_p.

- v_{i-1}^{RL} is either 0 or $v_i^{\mathrm{RL}} + 1$, or, at steps (1)(b) and (2)(b), of the form $v_i^{\mathrm{RL}} - b$ in situations where $v_i^{\mathrm{RL}} \geqslant b$. In all cases, v_{i-1}^{RL} is in $\mathbb{N}$.

- H_{i-1}^{RL} is always a prefix of H_i^{RL}, possibly augmented by a letter b: at step (2)(a), by $b = a_{\ell,m}$ when $0 \leqslant \ell + m \leqslant \bar{\mu} + m = p - 1$; at step (2)(b), by $b = a_{\bar{\mu},m}$ when $0 \leqslant \bar{\mu} + m = p - 1$; at step (4)(b), by $b = a_{\ell,m-1}$ when $m \geqslant 1$ and $1 \leqslant \ell + m \leqslant p - 1$. In all cases, H_{i-1}^{RL} is in A_p^*.

As in the case $p = 1$ of Section 2, we next define a map Ψ_p from S_p^* to Σ_p^* as follows. Given an input word $\bar{w} = \bar{w}_1 \ldots \bar{w}_n$ from S_p^*, iterating the process above from the initial counter $(H_n^{\mathrm{RL}}, v_n^{\mathrm{RL}}) = (\epsilon, 0)$ results in a word $w = w_1 \ldots w_n$, and $\Psi_p(\bar{w})$ is defined as w.

In Figure 9, we provide again, this time in usual set notation, the input and output sets of the forward transitions (T1) to (T8) and of the corresponding backward transitions. The careful reader will realize that the eight sets in the middle column form a partition of

	Input: (μ, H_L, v_L) from $\Sigma_p \times A_p^* \times \mathbb{N}$	Output: $(H_R, v_R, \bar\mu)$ from $A_p^* \times \mathbb{N} \times S_p$
(T1)	$\{p\} \times A_p^* \times \mathbb{N}$	$A_p^* \times (p + \mathbb{N}) \times \{(0,p)\}$
(T2)	$\{0,\dots,p-1\} \times A_p^* \times \{0\}$	$A_p^* \times \{0,\dots,p-1\} \times \{(0,p)\}$
(T3)	$\{-1\} \times (A_p^* \smallsetminus \{\epsilon\}) \times \{0\}$	$A_p^* \times \bigcup_{j=1}^p \{0,\dots,p-j\} \times \{(-j,p-j)\}$
(T4)	$\{-1\} \times \bigcup_{\ell+m=p-1} A_p^* a_{\ell,m} \times (1+\mathbb{N})$	$A_p^* \times \bigcup_{j=1}^p (p-j+1+\mathbb{N}) \times \{(-j,p-j)\}$
(T5)	$\{0,\dots,p-1\} \times A_p^* \times (1+\mathbb{N})$	$\bigcup_{\ell=0}^{p-1} A_p^* a_{\ell,0} \times \mathbb{N} \times \{(1,-1)\}$
(T6)	$\{-1\} \times \bigcup_{\ell+m\leqslant p-2} A_p^* a_{\ell,m} \times (1+\mathbb{N})$	$\bigcup_{\ell=0}^{p-2} \bigcup_{m=1}^{p-1-\ell} A_p^* a_{\ell,m} \times \mathbb{N} \times \{(1,-1)\}$
(T7)	$\{-1\} \times \{\epsilon\} \times (1+\mathbb{N})$	$\{\epsilon\} \times \mathbb{N} \times \{(1,-1)\}$
(T8)	$\{-1\} \times \{\epsilon\} \times \{0\}$	

Figure 9: Partitioning the input and output sets of the forward and backward transitions. For each column, the sets in the 8 rows partition the set given in the header.
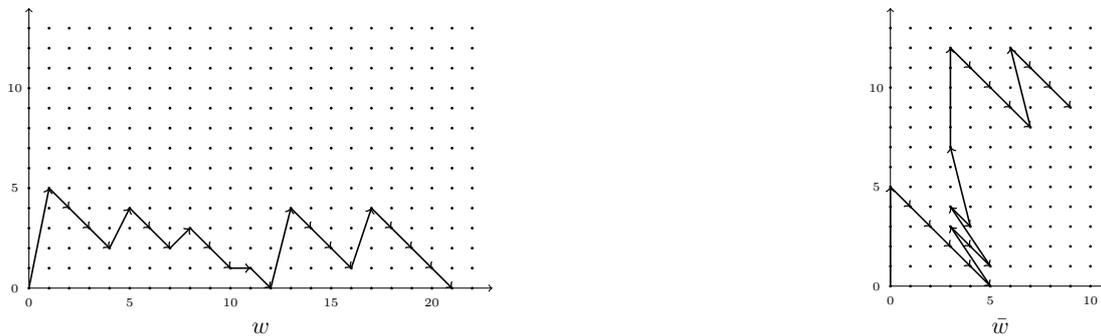

Figure 10: An example of a Łukasiewicz walk w (left) and of the corresponding p-tandem word $\bar w$ (right), for $p = 5$ and length 21.

$\Sigma_p \times A_p^* \times \mathbb{N}$, while the seven sets in the right-hand column form a partition of $A_p^* \times \mathbb{N} \times S_p$. Thus, the forward and backward procedures described above induce inverse bijections between $(\Sigma_p \times A_p^* \times \mathbb{N}) \smallsetminus \{(-1, \epsilon, 0)\}$ and $A_p^* \times \mathbb{N} \times S_p$.

This local reversibility property and the constructions of Φ_p and Ψ_p immediately make it possible to identify simple sets of words on which the restriction of Φ_p has a left inverse given by Ψ_p, and on which the restriction of Ψ_p has a left inverse given by Φ_p. This is made precise in the following proposition.

Proposition 2. *If $w \in \Sigma_p^*$ is in the domain of definition of Φ_p, that is, if (T8) is never used in the forward construction applied to w, and if additionally this construction terminates with $(H_n^{LR}, v_n^{LR}) = (\epsilon, 0)$, thus defining some $\bar w = \Phi_p(w) \in S_p^*$, then Ψ_p is well defined at $\bar w$, with the property $\Psi_p(\bar w) = w$, that is, $\Psi_p(\Phi_p(w)) = w$.*

Symmetrically, given $\bar w \in S_p^*$, if the backward construction applied to $\bar w$ terminates with $(H_0^{RL}, v_0^{RL}) = (\epsilon, 0)$, thus defining some $w = \Psi_p(\bar w) \in \Sigma_p^*$, then Φ_p is well defined at w, with the property $\Phi_p(w) = \bar w$, that is, $\Phi_p(\Psi_p(\bar w)) = \bar w$.

Example 3. Figure 10 provides an example for $p = 5$: on the left side, the 5-Łukasiewicz

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
w	5	D	D	D	2	D	D	1	D	D	0	D	4	D	D	D	3	D	D	D	D	
tr	T1	T7	T7	T7	T5	T6	T3	T5	T6	T3	T5	T3	T2	T7	T7	T7	T5	T3	T7	T7	T7	
H	ϵ	ϵ	ϵ	ϵ	ϵ	$a_{2,0}$	$a_{2,1}$	ϵ	$a_{1,0}$	$a_{1,1}$	ϵ	$a_{0,0}$	ϵ	ϵ	ϵ	ϵ	ϵ	$a_{3,0}$	ϵ	ϵ	ϵ	ϵ
v	0	5	4	3	2	1	0	2	1	0	1	0	0	4	3	2	1	0	3	2	1	0
$\bar w$	5	D	D	D	D	D	3	D	D	3	D	4	5	D	D	D	D	4	D	D	D	
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
		2																				
		3	3											14								
		4	4	4										15	15				19			
		5	5	5	5			8						16	16	16			20	20		
V	ϵ	6	6	6	6	6	ϵ	9	9	ϵ	11	ϵ	ϵ	17	17	17	17	ϵ	21	21	21	ϵ

Figure 11: Transduction corresponding to Figure 10 (top rows), augmented with the stack to be used in Section 6.4 (bottom row). The letters μ in row w are the input to the LR process as well as the output from the RL process. They give the Łukasiewicz walk w as the succession of steps $(1, \mu)$, with D standing for -1. The letters $\bar\mu$ in row $\bar w$ are the output from the LR process, the input to the RL process. They give the p-tandem word $\bar w$ as the succession of steps $(\bar\mu - 5, \bar\mu)$, or $(1, -1)$ if $\bar\mu = $ D. The transitions used are indicated by row tr. How they transform the generalized counters H and v is provided by two more rows. Along this run, H remains either the empty word ϵ or is a word consisting of a single letter $a_{\ell,m}$. The last row provides the evolution of the stack V of the augmented RL transducer process of Section 6.4, with pushes to the top of the depicted columns (ϵ denoting the empty stack). Observe that the length of V constantly matches v.

walk

$$w = (5, -1, -1, -1, 2, -1, -1, 1, -1, -1, 0, -1, 4, -1, -1, -1, 3, -1, -1, -1, -1),$$

where each letter μ represents the step $(1, \mu)$; on the right side, the corresponding 5-tandem walk, obtained by applying Φ_5,

$$\bar w = (5, -1, -1, -1, -1, -1, 3, -1, -1, 3, -1, 4, 5, -1, -1, -1, -1, 4, -1, -1, -1),$$

where each letter $\bar\mu$ represents the step $(\bar\mu - 5, \bar\mu)$ if $\bar\mu \geqslant 0$, and $(1, -1)$ if $\bar\mu = -1$. Figure 11 provides the transitions used and the evolution of the counters H and v along the construction of the 5-tandem walk from the 5-Łukasiewicz walk.

3.2 Parameters depending on the constructions, and their variations

The counters (H, v) of Section 3.1 (whether it be in LR or in RL form) are needed explicitly for the constructions of Φ_p and Ψ_p. In Section 3.3, we will introduce further parameters that could be computed, along with H and v, over the course of the constructions. Yet, these parameters are only needed for proving properties of the constructions, which is why they are not made part of the definition of the construction in the way that H and v are. A notational difficulty in proofs is that we need to compare the counters, outputs, and

parameters as computed by the left-to-right process Φ_p with those computed by the right-to-left process Ψ_p, and in fact several sorts of variations in the parameters. We introduce such variations in abstract in the present section, before defining specific parameters in Section 3.3. (We will freely use the abbreviations LR and RL to refer to the two processes.)

We start with notation to distinguish the counters (H, v) and the words produced by the two processes. Given a p-Lukasiewicz word $w \in \Sigma_p^n$, for any integer i satisfying $1 \leqslant i \leqslant n$ we denote by $H_{\mathrm{LR},i}^{(w)}$ and $v_{\mathrm{LR},i}^{(w)}$ the values of H and v obtained by the LR process after its ith step, that is, after using the letter w_i from the input word w, and by $\bar{w}_{\mathrm{LR},i}^{(w)}$ the letter output at this step. Similarly, given a quarter-plane word $\bar{w} \in S_p^n$, for any integer i satisfying $0 \leqslant i \leqslant n-1$ we denote by $H_{\mathrm{RL},i}^{(\bar{w})}$ and $v_{\mathrm{RL},i}^{(\bar{w})}$ the values of H and v obtained by the RL process after its $(n-i)$th step, that is, after using the letter $\bar{w}_{i+1}$ from the input word $\bar{w}$, and by $w_{\mathrm{RL},i+1}^{(\bar{w})}$ the letter output at this step. To get total maps over $0 \leqslant i \leqslant n$, we also set $H_{\mathrm{LR},0}^{(w)} = H_{\mathrm{RL},n}^{(\bar{w})} = \epsilon$ and $v_{\mathrm{LR},0}^{(w)} = v_{\mathrm{RL},n}^{(\bar{w})} = 0$. (Note that $(H_{\mathrm{LR},i}^{(w)}, v_{\mathrm{LR},i}^{(w)})$ and $(H_{\mathrm{RL},i}^{(\bar{w})}, v_{\mathrm{RL},i}^{(\bar{w})})$ were respectively denoted $(H_i^{\mathrm{LR}}, v_i^{\mathrm{LR}})$ and $(H_i^{\mathrm{RL}}, v_i^{\mathrm{RL}})$ in Section 3.1.)

The parameters we are interested in depend on the words input to the constructing processes, in Σ_p^* for LR and in S_p^* for RL, on the words output by the same processes, in S_p^* for LR and in Σ_p^* for RL, as well as on the counter (H, v) at each step in the process. So, we formally introduce a *parameter* q as a tuple $(\alpha, \bar{\alpha}, \sigma_1, \sigma_2)$ of monoid morphisms with integer values. That is, we assume that the four maps satisfy formulas

$$\alpha(w) = \sum_{j=1}^{i} \alpha(w_j), \quad \bar{\alpha}(\bar{w}) = \sum_{j=1}^{i} \bar{\alpha}(\bar{w}_j), \quad \sigma_1(H) = \sum_{j=1}^{i} \sigma_1(\eta_i), \quad \sigma_2(v) = v\sigma_2(1).$$

for all $i \in \mathbb{N}$, $w = w_1 \dots w_i \in \Sigma_p^i$, $\bar{w} = \bar{w}_1 \dots \bar{w}_i \in S_p^i$, $H = \eta_1 \dots \eta_i \in A_p^i$, and $v \in \mathbb{N}$. After fixing a length n for words, we associate two sequences to the parameter q, which we name the *values of q along the LR and RL processes*, respectively:

- For a p-Lukasiewicz word $w \in \Sigma_p^n$ and $0 \leqslant i \leqslant n$, the value $q_{\mathrm{LR},i}^{(w)}$ of q obtained after i steps in the construction Φ_p is given, assuming no error occurs, as

$$q_{\mathrm{LR},i}^{(w)} = \alpha(w_1 \dots w_i) + \bar{\alpha}(\bar{w}_{\mathrm{LR},1}^{(w)} \dots \bar{w}_{\mathrm{LR},i}^{(w)}) + \sigma_1(H_{\mathrm{LR},i}^{(w)}) + \sigma_2(v_{\mathrm{LR},i}^{(w)}).$$

- Similarly, for a quarter-plane word $\bar{w} \in S_p^n$ and $0 \leqslant i \leqslant n$, the value $q_{\mathrm{RL},i}^{(\bar{w})}$ of q obtained after $n-i$ steps in the construction Ψ_p is given as

$$q_{\mathrm{RL},i}^{(\bar{w})} = \alpha(w_{\mathrm{RL},i+1}^{(\bar{w})} \dots w_{\mathrm{RL},n}^{(\bar{w})}) + \bar{\alpha}(\bar{w}_{i+1} \dots \bar{w}_n) - \sigma_1(H_{\mathrm{RL},i}^{(\bar{w})}) - \sigma_2(v_{\mathrm{RL},i}^{(\bar{w})}).$$

  (Negative signs may look counter-intuitive. They relate to RL's operating transitions backwards.)

Observe that $q_{\mathrm{LR},i}^{(w)}$ may be undefined, as Φ_p may lead to error. It will be our goal to relate $q_{\mathrm{LR},i}^{(w)}$ and $q_{\mathrm{RL},i}^{(\bar{w})}$ when w and $\bar{w}$ are properly related by Φ_p and Ψ_p (and for suitable parameters q).

Our plan is next to describe how parameters vary along the processes LR and RL, and how (LR and RL) variations in parameters decompose into variations across transitions. We first define two kinds of variation operators on parameters:

- For a p-Łukasiewicz word $w \in \Sigma_p^n$ and $0 \leqslant i \leqslant j \leqslant n$, the LR-variation in q is defined as
$$\Delta_{i \to j}^{(w)} q = q_{\mathrm{LR},j}^{(w)} - q_{\mathrm{LR},i}^{(w)},$$
which is equal to
$$\alpha(w_{i+1} \dots w_j) + \bar{\alpha}(\bar{w}_{\mathrm{LR},i+1}^{(w)} \dots \bar{w}_{\mathrm{LR},j}^{(w)}) + \sigma_1(H_{\mathrm{LR},j}^{(w)}) - \sigma_1(H_{\mathrm{LR},i}^{(w)}) + \sigma_2(v_{\mathrm{LR},j}^{(w)} - v_{\mathrm{LR},i}^{(w)}).$$

- Similarly, for a quarter-plane word $\bar{w} \in S_p^n$ and $0 \leqslant i \leqslant j \leqslant n$, the RL-variation in q is defined as
$$\Delta_{i \leftarrow j}^{(\bar{w})} q = q_{\mathrm{RL},i}^{(\bar{w})} - q_{\mathrm{RL},j}^{(\bar{w})},$$
which is equal to
$$\alpha(w_{\mathrm{RL},i+1}^{(\bar{w})} \dots w_{\mathrm{RL},j}^{(\bar{w})}) + \bar{\alpha}(\bar{w}_{i+1} \dots \bar{w}_j) + \sigma_1(H_{\mathrm{RL},j}^{(\bar{w})}) - \sigma_1(H_{\mathrm{RL},i}^{(\bar{w})}) + \sigma_2(v_{\mathrm{RL},j}^{(\bar{w})} - v_{\mathrm{RL},i}^{(\bar{w})}).$$

An extreme case is obtained when i and j differ by 1, in which case
$$\Delta_{i-1 \to i}^{(w)} q = \alpha(w_i) + \bar{\alpha}(\bar{w}_{\mathrm{LR},i}^{(w)}) + \sigma_1(H_{\mathrm{LR},i}^{(w)}) - \sigma_1(H_{\mathrm{LR},i-1}^{(w)}) + \sigma_2(v_{\mathrm{LR},i}^{(w)} - v_{\mathrm{LR},i-1}^{(w)})$$
and
$$\Delta_{i-1 \leftarrow i}^{(\bar{w})} q = \alpha(w_{\mathrm{RL},i}^{(\bar{w})}) + \bar{\alpha}(\bar{w}_i) + \sigma_1(H_{\mathrm{RL},i}^{(\bar{w})}) - \sigma_1(H_{\mathrm{RL},i-1}^{(\bar{w})}) + \sigma_2(v_{\mathrm{RL},i}^{(\bar{w})} - v_{\mathrm{RL},i-1}^{(\bar{w})}).$$

Given two (independent) words w and $\bar{w}$ of same length n, it can happen that the ith step of the LR process at w and the $(n+1-i)$th step of the RL process at $\bar{w}$ use the same transition (in reverse forms). That is, a possible situation is the existence of a tuple $\tau = (\mu, H_L, v_L, H_R, v_R, \bar{a})$ satisfying
$$w_i = w_{\mathrm{RL},i}^{(\bar{w})} = \mu, \qquad H_{\mathrm{LR},i-1}^{(w)} = H_{\mathrm{RL},i-1}^{(\bar{w})} = H_L, \quad v_{\mathrm{LR},i-1}^{(w)} = v_{\mathrm{RL},i-1}^{(\bar{w})} = v_L,$$
$$H_{\mathrm{LR},i}^{(w)} = H_{\mathrm{RL},i}^{(\bar{w})} = H_R, \qquad v_{\mathrm{LR},i}^{(w)} = v_{\mathrm{RL},i}^{(\bar{w})} = v_R, \qquad \bar{w}_{\mathrm{LR},i}^{(w)} = \bar{w}_i = \bar{a},$$
in which case $\Delta_{i-1 \to i}^{(w)} q$ and $\Delta_{i-1 \leftarrow i}^{(\bar{w})} q$ are equal. In such a situation, we call τ a *transition instance* (or more specifically an instance of a named transition). We can then define the *variation in q across τ* as the common value
$$\Delta_\tau q = \alpha(\mu) + \bar{\alpha}(\bar{a}) + \sigma_1(H_R) - \sigma_1(H_L) + \sigma_2(v_R - v_L).$$

Obviously, $\Delta_\tau q$ depends on the 6-tuple τ, but neither on the location i of its use, nor on the context of the entire p-Łukasiewicz walk or p-tandem quarter-plane walk being processed. Furthermore, both $\Delta_{i \to j}^{(w)} q$ and $\Delta_{i \leftarrow j}^{(\bar{w})} q$ decompose as sums of $\Delta_\tau q$'s, for the successive τ's set up by LR or RL.

3.3 Specific parameters and some of their invariants

The specific parameters we are about to define are of two types: first, parameters on counters; second, parameters on input and output words. Our proofs will require parameters that are linear combinations of the two types.

For the parameters on counters, we introduce three monoid morphisms from A_p^* to $\mathbb{N}$. Namely, we define $|\cdot|$, wt_ℓ, and wt_m, by, respectively, $|a_{\ell,m}| = 1$, $\mathrm{wt}_\ell(a_{\ell,m}) = \ell + 1$, and $\mathrm{wt}_m(a_{\ell,m}) = m + 1$ for each $a_{\ell,m} \in A_p$. We obtain three parameters, named respectively λ, ζ, and ξ, by setting α, $\bar{\alpha}$, and σ_2 to the zero map, and defining σ_1 by, respectively,

$$\sigma_1(H) = |H|, \qquad \sigma_1(H) = \mathrm{wt}_\ell(H), \qquad \sigma_1(H) = \mathrm{wt}_m(H).$$

A fourth parameter depending on the counters, named v, is obtained by setting σ_2 to the identity and α, $\bar{\alpha}$, and σ_1 to the zero map. We state the following proposition with no reference to parameters, but it clearly expresses constraints for the first three parameters above to cancel.

Proposition 4. *For any $H \in A_p^*$, the following four properties are equivalent: $|H| = 0$; $\mathrm{wt}_\ell(H) = 0$; $\mathrm{wt}_m(H) = 0$; $H = \epsilon$.*

The simplest parameters depending on input and output words we use are given as coordinates of the vertices of the walks. A first parameter, z, is obtained by setting $\bar{\alpha} = \sigma_1 = \sigma_2 = 0$ and by defining $\alpha(\mu) = \mu$ for any letter $\mu \in \Sigma_p$. Two parameters, $\bar{x}$ and $\bar{y}$, respectively, are obtained by setting $\alpha = \sigma_1 = \sigma_2 = 0$ and by defining $\bar{\alpha}(\bar{a})$ to be the first coordinate $\bar{a}_1$, respectively the second coordinate $\bar{a}_2$ of any letter $\bar{a} = (\bar{a}_1, \bar{a}_2) \in S_p$. In particular, for any p-Łukasiewicz word $w \in S_p^n$ we have the formulas

$$(\bar{x}_{\mathrm{LR},i}^{(w)}, \bar{y}_{\mathrm{LR},i}^{(w)}) = \sum_{1 \leqslant j \leqslant i} \bar{w}_{\mathrm{LR},j}^{(w)}, \qquad z_{\mathrm{LR},i}^{(w)} = \sum_{1 \leqslant j \leqslant i} w_j,$$

so that the $(i, z_{\mathrm{LR},i}^{(w)})$ form a walk that remains in the upper half plane, and the $(\bar{x}_{\mathrm{LR},i}^{(w)}, \bar{y}_{\mathrm{LR},i}^{(w)})$ form a (free) p-tandem walk associated to $\bar{w}_{\mathrm{LR},1}^{(w)} \ldots \bar{w}_{\mathrm{LR},n}^{(w)} \in \Sigma_p^n$. Another parameter, named $\bar{k}$, measures how far the p-tandem walk is from the main diagonal, and is obtained by setting $\alpha = \sigma_1 = \sigma_2 = 0$ and by defining $\bar{\alpha}(\bar{a})$ to be the difference $\bar{a}_2 - \bar{a}_1$, for any letter $\bar{a} = (\bar{a}_1, \bar{a}_2) \in S_p$. Specifically,

$$\bar{k}_{\mathrm{LR},i}^{(w)} = \bar{y}_{\mathrm{LR},i}^{(w)} - \bar{x}_{\mathrm{LR},i}^{(w)} = \sum_{j=1}^{n} \left(\bar{w}_{\mathrm{LR},j}^{(w)} \right)_2 - \left(\bar{w}_{\mathrm{LR},j}^{(w)} \right)_1.$$

Note that, across one step in the LR process, the parameter $\bar{k}$ either increases by p, if the letter $\bar{w}_{\mathrm{LR},j}^{(w)}$ is one of the $(-j, p-j)$ for $0 \leqslant j \leqslant p$, or decreases by 2, if it is $(1, -1)$. We mimic this definition on the walk associated to w, by setting now $\bar{\alpha} = \sigma_1 = \sigma_2 = 0$ and by defining the monoid morphism α from Σ_p^* to $\mathbb{N}$ defined by $\alpha(\mu) = p$ if $0 \leqslant \mu \leqslant p$ and $\alpha(-1) = -2$. This defines a parameter, named k. Two additional parameters, which we will need specifically for the analysis of the RL process, keep track of the cumulative

amount by which the inequalities given below in (2) are strict. They can be understood visually by noting that under most transitions, $\bar{y}$ and v move in parallel, as do $\bar{x}$ and ξ. The parameters we want, named $\bar{r}$ and $\bar{s}$ below, are the shifts separating the two parallel functions in each case. More precisely, define a parameter $\bar{r}$ by setting

$$\alpha(\mu) = 0, \quad \bar{\alpha}(\bar{a}) = \bar{a}_1, \quad \sigma_1(H) = -\operatorname{wt}_m(H), \quad \sigma_2(v) = 0.$$

Similarly, define a parameter $\bar{s}$ by setting

$$\alpha(\mu) = 0, \quad \bar{\alpha}(\bar{a}) = \bar{a}_2, \quad \sigma_1(H) = 0, \quad \sigma_2(v) = -v.$$

Thus, the following formulas hold for $\bar{w} \in S_p^*$:

$$\bar{r}_{\mathrm{RL},i}^{(\bar{w})} = \bar{x}_{\mathrm{RL},i}^{(\bar{w})} - \xi_{\mathrm{RL},i}^{(\bar{w})}, \qquad \bar{s}_{\mathrm{RL},i}^{(\bar{w})} = \bar{y}_{\mathrm{RL},i}^{(\bar{w})} - v_{\mathrm{RL},i}^{(\bar{w})}. \tag{1}$$

It will prove convenient in what follows to write identities like the last one in a more compact, while more general, form. To this end, remember that our parameters are defined through morphisms to $\mathbb{Z}$. Thus, the operations in $\mathbb{Z}$ of addition, subtract, and multiplication by an integer, transpose in an obvious manner into operations on morphisms, then into operations on parameters. For instance, we could have defined $\bar{r}$ and $\bar{s}$ as differences of parameters, via the formulas $\bar{r} = \bar{x} - \xi$ and $\bar{s} = \bar{y} - v$. These conveniently suggest the identities (1), as well as similar identities for the RL process and for variations.

The variations along the walks in the parameters introduced so far depend only on the transition instance $\tau = (\mu, H_L, v_L, H_R, v_R, \bar{\mu})$, so we can define and calculate Δ_τ for each of them. As it turns out, the variations actually do not depend on the specific transition instance τ, but simply on the transition, one of (T1) to (T8). The results are given in the table in Figure 12.

Proposition 5. *For each τ and $\Delta = \Delta_\tau$, the following inequalities and equalities hold:*

$$\Delta\bar{x} \geqslant \Delta\xi, \quad \Delta\bar{y} \geqslant \Delta v, \tag{2}$$

$$\Delta z = \Delta\zeta + \Delta v, \quad \Delta k - \Delta\bar{k} = (p+2)\Delta\lambda, \tag{3}$$

$$\Delta\bar{x} = \Delta\bar{r} + \Delta\xi, \quad \Delta\bar{y} = \Delta\bar{s} + \Delta v. \tag{4}$$

Additionally, the inequalities (2) *are strict in precisely the following circumstances:*

- $\Delta_\tau\bar{x} > \Delta_\tau\xi$ *only if τ is an instance of (T7), implying $H_L = H_R = \epsilon$;*

- $\Delta_\tau\bar{y} > \Delta_\tau v$ *only if τ is an instance of (T2) or (T3), implying $v_L = 0$.*

Proof. By inspection of the table. (The equalities of (4) are really just restatements of the definitions of $\bar{r}$ and $\bar{s}$.) $\qquad \square$

Since these quantities depend only on the transitions, we also have that for given $w \in \Sigma_p^*$, $\bar{w} \in S_p^*$, and $0 \leqslant i \leqslant j \leqslant n$, the variations $\Delta_{i \to j}^{(w)}$ and $\Delta_{i \leftarrow j}^{(\bar{w})}$ are sums of Δ_τ's operating on the letters of w and $\bar{w}$. It follows by induction that the two equalities and two inequalities in (2) and (3) still hold for $\Delta = \Delta_{i \to j}^{(w)}$ and $\Delta = \Delta_{i \leftarrow j}^{(\bar{w})}$.

Observe that $\Delta\bar{r}$ and $\Delta\bar{s}$ are both nonnegative for all choices of Δ as a Δ_τ, a $\Delta_{i \to j}^{(w)}$, or a $\Delta_{i \leftarrow j}^{(\bar{w})}$, provided $i < j$.

τ inst. of	(T1)	(T2)	(T3)	(T4)	(T5)	(T6)	(T7)
$\Delta_\tau \bar{x}$	0	0	$-m-1$	$-m-1$	1	1	1
$\Delta_\tau \bar{y}$	p	p	$p-m-1$	$p-m-1$	-1	-1	-1
$\Delta_\tau z$	p	q	-1	-1	q	-1	-1
$\Delta_\tau v$	p	q	ℓ	ℓ	-1	-1	-1
$\Delta_\tau \lambda$	0	0	-1	-1	1	0	0
$\Delta_\tau \zeta$	0	0	$-\ell-1$	$-\ell-1$	$q+1$	0	0
$\Delta_\tau \xi$	0	0	$-m-1$	$-m-1$	1	1	0
$\Delta_\tau k$	p	p	-2	-2	p	-2	-2
$\Delta_\tau \bar{k}$	p	p	p	p	-2	-2	-2
$\Delta_\tau \bar{r}$	0	0	0	0	0	0	1
$\Delta_\tau \bar{s}$	0	$p-q$	$p-m-1-\ell$	0	0	0	0
hypothesis		$p-1\geqslant q$		$p-m-1=\ell$	$p-m-2\geqslant\ell$		
			$p-m-1\geqslant\ell$		$p-1\geqslant q$		

Figure 12: Variations in the parameters across the transitions defined in Figure 6 for the bijections between p-Łukasiewicz and p-tandem walks.

3.4 Properties of the constructed walks

Theorem 6. *The map Φ_p is a bijection between p-Łukasiewicz walks and p-tandem walks that preserves the total number of steps, maps k to $\bar{k}$, and maps the number of -1 steps to the number of $\searrow$ steps. The map Ψ_p is the inverse bijection.*

Proof. We first prove that Φ_p maps prefixes of p-Łukasiewicz walks to quarter-plane walks and that applying Φ_p to a p-Łukasiewicz walk in its entirety results in $(v, H) = (0, \epsilon)$ at the end of the algorithm.

To this end, again take $w \in \Sigma_p^n$ and $\bar{w} = \Phi_p(w)$, both of length n.

Note that we have the initial conditions $\bar{x}_{\mathrm{LR},0}^{(w)} = \bar{y}_{\mathrm{LR},0}^{(w)} = z_{\mathrm{LR},0}^{(w)} = v_{\mathrm{LR},0}^{(w)} = k_{\mathrm{LR},0}^{(w)} = \bar{k}_{\mathrm{LR},0}^{(w)} = \lambda_{\mathrm{LR},0}^{(w)} = 0$. By (2) we have for all i that $\Delta_{i-1\to i}^{(w)}\bar{x} \geqslant \Delta_{i-1\to i}^{(w)}\xi$. Summing this with the initial conditions we get that for all i

$$\bar{x}_{\mathrm{LR},i}^{(w)} \geqslant \xi_{\mathrm{LR},i}^{(w)}.$$

Arguing likewise from (2) and (3) we have

$$\bar{y}_{\mathrm{LR},i}^{(w)} \geqslant v_{\mathrm{LR},i}^{(w)}, \qquad z_{\mathrm{LR},i}^{(w)} = \zeta_{\mathrm{LR},i}^{(w)} + v_{\mathrm{LR},i}^{(w)}, \qquad k_{\mathrm{LR},i}^{(w)} - \bar{k}_{\mathrm{LR},i}^{(w)} = (p+2)\lambda_{\mathrm{LR},i}^{(w)}.$$

A first consequence is that $\bar{y}_{\mathrm{LR},i}^{(w)}$ and $\bar{x}_{\mathrm{LR},i}^{(w)}$ are nonnegative for all i, as they are both bounded below by lengths, and so the p-tandem walk associated to $\bar{w}$ remains in the quarter plane. Second, $z_{\mathrm{LR},i}^{(w)}$ is the sum of two nonnegative integers, and more precisely $z_{\mathrm{LR},i}^{(w)} = 0$ if and only if $(\zeta_{\mathrm{LR},i}^{(w)}, v_{\mathrm{LR},i}^{(w)}) = (0,0)$. As a consequence, by Proposition 4, w is a p-Łukasiewicz word if and only if the construction of $\bar{w} = \Phi_p(w)$ ends with $H = \epsilon$ and $v = 0$. Additionally, if w is a p-Łukasiewicz word, then $w = \Psi_p(\Phi_p(w))$ by the remark at the end of Section 3.1. Finally, also $\lambda_{\mathrm{LR},n}^{(w)} = 0$, because $H_{\mathrm{LR},n}^{(w)} = \epsilon$, and so

$$k_{\mathrm{LR},n}^{(w)} = \bar{k}_{\mathrm{LR},n}^{(w)}. \tag{5}$$

Next we will prove that Ψ_p maps quarter-plane walks in S_p^* to suffixes of p-Łukasiewicz walks and that we have $(H, v) = (\epsilon, 0)$ at the end of the algorithm. To this end, now take $\bar{w} \in S_p^*$ and $w = \Psi_p(\bar{w})$, both of length n.

By (3) we have for all i that

$$\Delta_{i \leftarrow i+1}^{(\bar{w})} z = \Delta_{i \leftarrow i+1}^{(\bar{w})} \zeta + \Delta_{i \leftarrow i+1}^{(\bar{w})} v.$$

Next, summing we get that $z_{\mathrm{RL},i}^{(\bar{w})}$ is a sum of nonnegative terms and so is nonnegative for all i. Thus every suffix of w remains in the upper half plane.

Next consider $\bar{r}_{\mathrm{RL},i}^{(\bar{w})}$ and $\bar{s}_{\mathrm{RL},i}^{(\bar{w})}$. All the entries for $\bar{r}$ and $\bar{s}$ in the table are nonnegative and so $\bar{r}_{\mathrm{RL},i}^{(\bar{w})} \geqslant \bar{r}_{\mathrm{RL},j}^{(\bar{w})}$ for $i \leqslant j$, and likewise for $\bar{s}_{\mathrm{RL},i}^{(\bar{w})}$. In other words, $\bar{r}_{\mathrm{RL},i}^{(\bar{w})}$ and $\bar{s}_{\mathrm{RL},i}^{(\bar{w})}$ are weakly decreasing as functions of i. Consequently $\bar{r}_{\mathrm{RL},0}^{(\bar{w})}$ is the maximum value of $\{\bar{r}_{\mathrm{RL},i}^{(\bar{w})} : 0 \leqslant i \leqslant n\}$. Let j be maximal such that $\bar{r}_{\mathrm{RL},j}^{(\bar{w})} = \bar{r}_{\mathrm{RL},0}^{(\bar{w})}$. As observed in the previous section, $\bar{r}_{\mathrm{RL},i}^{(\bar{w})}$ can only change under transition (T7) and the stack must be empty for this transition. Thus the choice of j implies by Proposition 4 that $\xi_{\mathrm{RL},j}^{(\bar{w})} = 0$. By definition of $\bar{r}$, for the remainder of the algorithm, from index j down to index 0, $\bar{x}_{\mathrm{RL},j'}^{(\bar{w})}$ and $\xi_{\mathrm{RL},j'}^{(\bar{w})}$ move in parallel with a fixed shift of $\bar{r}_{\mathrm{RL},0}^{(\bar{w})}$.

At the end of the algorithm, Ψ_p has consumed all of $\bar{w}$ from its end to its beginning, and so we have $\bar{x}_{\mathrm{RL},0}^{(\bar{w})} = 0$. Since $\bar{w}$ is a quarter-plane walk, $\bar{x}_{\mathrm{RL},j}^{(\bar{w})} \geqslant 0$, and by the previous paragraph $\bar{x}_{\mathrm{RL},j}^{(\bar{w})} - \xi_{\mathrm{RL},j}^{(\bar{w})} = \bar{r}_{\mathrm{RL},0}^{(\bar{w})} = \bar{x}_{\mathrm{RL},0}^{(\bar{w})} - \xi_{\mathrm{RL},0}^{(\bar{w})}$. We get that

$$\xi_{\mathrm{RL},0}^{(\bar{w})} = \bar{x}_{\mathrm{RL},0}^{(\bar{w})} - \bar{x}_{\mathrm{RL},j}^{(\bar{w})} + \xi_{\mathrm{RL},j}^{(\bar{w})} = -\bar{x}_{\mathrm{RL},j}^{(\bar{w})} \leqslant 0.$$

However m-weight is always nonnegative and hence $\xi_{\mathrm{RL},0}^{(\bar{w})} = 0$ and so, by Proposition 4, H is empty at the end of the process.

We can make an analogous argument with $\bar{y}$ in place of $\bar{x}$ and v in place of $\mathrm{wt}_m(H)$. Specifically, now let j be maximal such that $\bar{s}_{\mathrm{RL},j}^{(\bar{w})} = \bar{s}_{\mathrm{RL},0}^{(\bar{w})}$. Since $\bar{s}$ can only change under transitions (T2) and (T3), $v_{\mathrm{RL},j}^{(\bar{w})} = 0$. For the remainder of the algorithm, from index j down to index 0, $\bar{y}_{\mathrm{RL},j'}^{(\bar{w})}$ and $v_{\mathrm{RL},j'}^{(\bar{w})}$ move in parallel with a fixed shift of $\bar{s}_{\mathrm{RL},0}^{(\bar{w})}$. We have $\bar{y}_{\mathrm{RL},0}^{(\bar{w})} = 0$, and, since $\bar{w}$ is a quarter-plane walk, $\bar{y}_{\mathrm{RL},j}^{(\bar{w})} \geqslant 0$. Additionally, by the definition of j, $\bar{y}_{\mathrm{RL},j}^{(\bar{w})} - v_{\mathrm{RL},j}^{(\bar{w})} = \bar{y}_{\mathrm{RL},0}^{(\bar{w})} - v_{\mathrm{RL},0}^{(\bar{w})}$ and so we get $v_{\mathrm{RL},0}^{(\bar{w})} = -\bar{y}_{\mathrm{RL},j}^{(\bar{w})} \leqslant 0$. However, v is always nonnegative so $v_{\mathrm{RL},0}^{(\bar{w})} = 0$.

From all this we conclude that $(H, v) = (\epsilon, 0)$ after applying Ψ_p to $\bar{w}$ and that $z_{\mathrm{RL},0}^{(\bar{w})} = 0$. Thus w is a p-Łukasiewicz word and $\bar{w} = \Phi_p(\Psi_p(\bar{w}))$.

Therefore for each n, Φ_p and Ψ_p are mutually inverse bijections between p-Łukasiewicz walks of length n and quarter-plane walks in S_p^n.

Finally, by (4), $k_{\mathrm{RL},i}^{(\bar{w})} - \bar{k}_{\mathrm{RL},i}^{(\bar{w})} = (p+2)\lambda_{\mathrm{RL},i}^{(\bar{w})}$, and $\lambda_{\mathrm{RL},0}^{(\bar{w})} = 0$ and so $k_{\mathrm{RL},0}^{(\bar{w})} = \bar{k}_{\mathrm{RL},0}^{(\bar{w})}$. Remembering (5), this tells us that Φ_p and Ψ_p, convert k to $\bar{k}$ and vice versa. Additionally, since -1 (respectively $\searc
box$) is the only step that decreases k (respectively $\bar{k}$) and all the other steps increase k (respectively $\bar{k}$) by the same amount $p+2$, we also obtain that Φ_p and Ψ_p take the number of -1 steps to the number $\searcbox$ steps in the walks and vice versa. $\qquad\square$

Many interesting consequences can be read off of the observations we have made so far. We collect a few in the following proposition.

Proposition 7.

- *A lower bound on the minimum y-value of the suffixes of the quarter-plane walk is obtained as follows: each time transition (T2) or transition (T3) with $p - m - 1 > \ell$ is used, the lower bound on the minimum for the suffix from after this step to the end is increased at least by $p - q > 0$ (for (T2)) or $p - m - 1 - \ell > 0$ (for (T3)).*

- *A lower bound on the minimum x-value of the suffixes of the quarter-plane walk is obtained as follows: each time transition (T7) is used, the lower bound on the minimum for the suffix from after this step to the end is increased at least by 1.*

- *If a p-Łukasiewicz walk w returns to the x-axis after i letters, thus factoring into two p-Łukasiewicz walks, write $w = w_1 w_2$ for a left factor of length i. Then $\Phi_p(w)$ is the concatenation of $\Phi_p(w_1)$ and $\Phi_p(w_2)$.*

Note that in the case $p = 1$, transition (T3) does not exist and transition (T2) becomes (L1). Consequently, the first item of the proposition explains the observation at the end of Example 1.

Yeats [17, Proposition 2.4] has given the result analogous to the second point in the six-step model. The converse to the third point is also true, but this is easier to see from the raising description and so the proof will be postponed to Section 6.

Proof. From Proposition 5, we have $\Delta \bar{y} = \Delta \bar{s} + \Delta v$. Intuitively, this means that $\bar{y}$ and v move in parallel except when $\bar{s}$ is nonzero. From the table in Figure 12 we see that $\bar{s}$ is nonzero precisely for transition (T2) and transition (T3) with $p - m - 1 > \ell$, and when $\bar{s}$ is nonzero it is positive. Finally, by the Łukasiewicz property, $v \geqslant 0$, therefore $\bar{s}$ gives a lower bound as described in the first item of the statement.

Similarly, from Proposition 5, we have $\Delta \bar{x} = \Delta \bar{r} + \Delta \xi$, so intuitively $\bar{x}$ and ξ move in parallel except with $\bar{r}$ is nonzero. From the table in Figure 12 we see that $\bar{r}$ is nonzero precisely for transition (T7) where it has value 1. Finally, ξ is a weight on H and so is always nonnegative, therefore $\bar{r}$ gives a lower bound as described in the second item of the statement.

For the third point, since w_1 is a p-Łukasiewicz walk, then as in the proof of Theorem 6, we have $(v, H) = (0, \epsilon)$ after i steps of the LR process for Φ_p. Consequently, when the process moves beyond length i it cannot remember anything from the prefix, as the only internal state is v and H, both of which have been returned to their initial situation. Therefore the same steps are outputted when Φ_p acts on the steps of w_2 as part of w as when Φ_p acts on w_2 as a p-Łukasiewicz walk in its own right. The second point follows. $\quad \square$

4 Six-step model

Let us consider now the 6-step model, that is we return to the case $p = 1$ and consider augmenting the step set with the steps' flips through the line $y = x$.

Figure 13 provides the forward transitions with their flips for the 6-step model. They include a first copy of the rules (U1) to (DE) of the model for $p = 1$ in the top row and
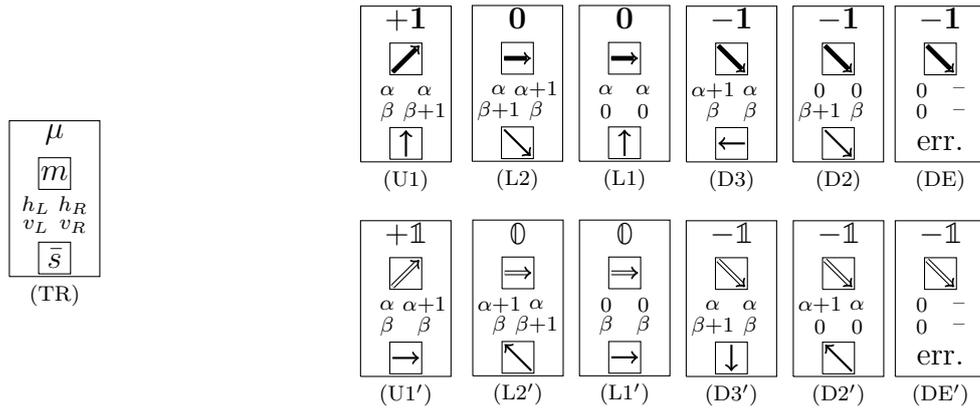
Figure 13: Transitions used for the 6-step model. The first row is a copy of Figure 2 with "solid" input letters. The second row is a copy of Figure 2 with "striped" input letters, output letters flipped around the main diagonal, and roles of counters exchanged.

	(U1)	(U1′)	(L2)	(L2′)	(L1)	(L1′)	(D3)	(D3′)	(D2)	(D2′)
$\Delta\bar{x}$	0	1	1	−1	0	1	−1	0	1	−1
$\Delta\bar{y}$	1	0	−1	1	1	0	0	−1	−1	1
Δz	1	1	0	0	0	0	−1	−1	−1	−1
Δh	0	1	1	−1	0	0	−1	0	0	−1
Δv	1	0	−1	1	0	0	0	−1	−1	0
Δk	1	1	1	1	1	1	−2	−2	−2	−2
$\Delta\bar{k}$	1	−1	−2	2	1	−1	1	−1	−2	2
$\Delta\bar{r}$	0	0	0	0	0	1	0	0	1	0
$\Delta\bar{s}$	0	0	0	0	1	0	0	0	0	1

Figure 14: Variations in the parameters across the transitions defined in Figure 13 for the 6-step model.

a second copy of the same rules with output letters flipped around the main diagonal in the bottom row. To distinguish between the two sets, the input letters now come in two colours. The interpretation of the notation used in the boxes is the following: μ is from $\{-\mathbf{1}, \mathbf{0}, +\mathbf{1}, -\mathbb{1}, \mathbb{0}, +\mathbb{1}\}$; $\boxed{m}$ stands for a letter from the alphabet $\Sigma_{1,\text{bicol}} :=$ $\{\nearrow, \rightarrow, \searrow, \nearrow, \Rightarrow, \searrow\}$; (h_L, v_L) and (h_R, v_R) are counters, again from $\mathbb{N}^2$; $\boxed{\bar{s}}$ stands for a letter from $S_{1,\text{sym}} = \{\uparrow, \leftarrow, \searrow, \downarrow, \rightarrow, \nwarrow\}$. We will call the steps $\nearrow, \rightarrow, \searrow$ *solid* and the steps $\nearrow, \Rightarrow, \searrow$ *striped*, and the walks made of these steps we will call *bicoloured*.

Define $\Phi_{1,\text{sym}}$ and $\Psi_{1,\text{sym}}$ by the following transitions as in Section 2 or Section 3.1.

Inspecting the transitions we see that the forwards and backwards input sets of the transitions partition the forwards and backwards input spaces and the outputs and parameters lie in the correct sets. We can define the same parameters as in Section 3.3 and capture how these parameters change in the table in Figure 14.

Inspecting the table we see

$$\Delta z = \Delta h + \Delta v, \qquad \Delta \bar{x} = \Delta h + \Delta \bar{r}, \qquad \Delta \bar{y} = \Delta v + \Delta \bar{s},$$
$$\Delta \bar{x} \geqslant \Delta h \quad \text{and} \quad \Delta \bar{x} > \Delta h \implies h = 0,$$
$$\Delta \bar{y} \geqslant \Delta v \quad \text{and} \quad \Delta \bar{y} > \Delta v \implies v = 0.$$

Then, using these equalities and inequalities, the proof that the maps give a bijection between bicoloured Motzkin paths and quarter-plane walks in $S_{1,\text{sym}}$ proceeds as in the unsymmetrized case.

This bijection answers a question of Bousquet-Mélou and Mishna (see [5, Section 7.1, item 2]); this is encapsulated in the following theorem. The question was first answered in a more complicated way in [17]. We will use the notation $\mathcal{M}(n)$ and $\mathcal{M}^{\text{bicol}}(n)$ for the classes of Motzkin and bicoloured Motzkin paths with n steps, $\mathcal{Q}(n)$ for 1-tandem walks with n steps, $\mathcal{Q}^{\text{sym}}(n)$ for quarter-plane walks with n steps each in $S_{1,\text{sym}}$ and $2^{[n]}$ for the powerset of $\{1, 2, \dots, n\}$.

Theorem 8. *The map $\Phi_{1,sym}$ is a length-preserving bijection between bicoloured Motzkin paths and quarter-plane walks with steps in $S_{1,sym}$. The map $\Psi_{1,sym}$ is the inverse bijection.*

With the notation as described above, the chain of bijections

$$\mathcal{Q}^{sym}(n) \xrightarrow{\Psi_{1,sym}} \mathcal{M}^{bicol}(n) \cong \mathcal{M}(n) \times 2^{[n]} \xrightarrow{\Phi_1 \times id} \mathcal{Q}(n) \times 2^{[n]}$$

explains combinatorially why $|\mathcal{Q}^{sym}(n)| = 2^n |\mathcal{Q}(n)|$.

To say the same thing in a slightly different way, if f is the map forgetting the colouring then $\Phi_1 \circ f \circ \Psi_{1,\text{sym}}$ gives a combinatorially defined 2^n-to-1 map from quarter-plane walks with steps in $S_{1,\text{sym}}$ to quarter-plane walks with steps in S_1 explaining $|\mathcal{Q}^{\text{sym}}(n)| = 2^n |\mathcal{Q}(n)|$.

Our answer to the question of Bousquet-Mélou and Mishna, while now fully combinatorial, still goes through Motzkin paths as intermediate objects. We can nevertheless understand some things about the composition directly without this intermediary. Following the definition of $\Psi_{1,\text{sym}}$ we see that steps $\boxed{\uparrow}$, $\boxed{\searrow}$, and $\boxed{\leftarrow}$ map to solid letters and $\boxed{\rightarrow}$, $\boxed{\nwarrow}$, and $\boxed{\downarrow}$ map to striped letters. Carrying solid and striped as "colours" though Φ_1 to obtain coloured quarter plane walks in S_1, we see that output steps which came from input steps in S_1 have the solid colour while output steps which came from the reflection of S_1 have the striped colour. So the colours in the output remember which half of the step set the input steps came from.

From the definition of the maps one can immediately observe that $\Psi_{1,\text{sym}}$ restricted to those 6-step walks which happen to only use steps of S_1 agrees with Ψ_1 after colouring the output of Ψ_1 solid. Additionally, $\Psi_{1,\text{sym}}$ restricted to those 6-step walks which only use steps not in S_1 agrees with Ψ_1 applied to the reflection of the walk after colouring the output of Ψ_1 striped. This was part of the initial motivation: we were looking for a generalization that extended Ψ_1 and that also extended Ψ_1 applied to the reflection for walks purely not using S_1 steps. Returning to the context of the question of question of Bousquet-Mélou and Mishna this implies that $\Phi_1 \circ f \circ \Psi_{1,\text{sym}}$ is the identity on those walks with only steps in S_1 and is reflection through $y = x$ on those walks with no steps in S_1.

5 Alternate interpretation via automata

It was said in the introduction that the bijections we introduce in the present work take their origin in automata theory. Yet, no explicit automaton is provided in Sections 2–4, where we used a notation by "boxes". In this section, we show that boxes are just a rephrasing of the transitions in an automaton, the only essential difference being that a single one of our boxes will usually pack a number of transitions of an automaton. We refer the reader to [1, 13, 16] for a detailed introduction to automata theory, and we follow the first of these references for the terminology. More accurately, our maps ϕ, ψ, Φ_p, Ψ_p, $\Phi_{1,\text{sym}}$, $\Psi_{1,\text{sym}}$ turn out to be the *transductions realized* by *pushdown transducers* (see [1, Section 5.4]), with the slight extension that our automata/transducers use two stacks (not just one).

We now give an explicit description of the transducer to transform Motzkin walks to tandem walks (case $p = 1$). We fix an input alphabet $A = \Sigma_1 = \{\boxed{\nearrow}, \boxed{\rightarrow}, \boxed{\searrow}\}$, an output alphabet $B = S_1 = \{\boxed{\uparrow}, \boxed{\leftarrow}, \boxed{\searrow}\}$, a finite set of states $Q = \{q\}$ (in fact a singleton), a set of accepting states $F = Q$ (that is, we want all states to be accepting), a stack alphabet common to both stacks $Z_1 = Z_2 = \{o, \iota\}$, an internal starting configuration $i = (q, o, o) \in Q \times Z_1^* \times Z_2^*$ (of the associated pushdown machine), and a set of internal accepting configurations $K = F \times \{\epsilon\} \times \{\epsilon\} \subset Q \times Z_1^* \times Z_2^*$. We will also shortly fix a finite set of transition rules $\gamma \subset (A \cup \{\epsilon\}) \times Q \times Z_1 \times Z_1^* \times Z_2 \times Z_2^* \times Q \times B^*$, after which collecting those sets into the tuple $\mathbb{T} = (A, Q, Z_1, Z_2, B, i, K, \gamma)$ will describe a pushdown transducer over A. As is traditional, an automaton is pictorially described as an oriented graph whose vertices are the elements of Q (states), with transition rules given by a labeling of edges. This labeling takes the form "s; t_1/T_1, t_2/T_2; s'", which really denotes a 6-tuple $(s, t_1, T_1, t_2, T_2, s') \in A \times Z_1 \times Z_1^* \times Z_2 \times Z_2^* \times B$. In our transducers, all possible transitions are from state q to itself. A computation by the underlying pushdown machine thus consists of a succession of transitions. Each transition operates in a well-defined manner. If the machine inputs a letter $s \in A$ when the internal configuration has t_1 as top letter of the first stack and t_2 as top letter of the second stack, then it replaces the top letter t_1 by the word T_1 and the top letter t_2 by the word T_2, before outputting the letter $s' \in B$. We will make sure that no stack will ever become empty and that their contents encode natural integers in unary: some $m \in \mathbb{N}$ will correspond to a stack consisting of an o (at the bottom), followed by m copies of ι (towards the top). More precisely, the first, respectively second, stack will represent the counter h, respectively the counter v. So the variations of counters will be be encoded in one of three forms: ι/ϵ for a decrement by 1; o/o and ι/ι for no variation; $o/o\iota$ and $\iota/\iota\iota$ for an increment by 1. This leads to a set γ containing the following 11 transition rules:

- $\boxed{\nearrow}$; $o/o\iota$, o/o; $\boxed{\uparrow}$, $\boxed{\nearrow}$; $o/o\iota$, ι/ι; $\boxed{\uparrow}$, $\boxed{\nearrow}$; $\iota/\iota\iota$, o/o; $\boxed{\uparrow}$, $\boxed{\nearrow}$; $\iota/\iota\iota$, ι/ι; $\boxed{\uparrow}$,

- $\boxed{\rightarrow}$; o/o, o/o; $\boxed{\uparrow}$, $\boxed{\rightarrow}$; o/o, ι/ι; $\boxed{\uparrow}$, $\boxed{\rightarrow}$; ι/ϵ, $o/o\iota$; $\boxed{\searrow}$, $\boxed{\rightarrow}$; ι/ϵ, $\iota/\iota\iota$; $\boxed{\searrow}$,

- $\boxed{\searrow}$; ι/ϵ, o/o; $\boxed{\searrow}$, $\boxed{\searrow}$; o/o, ι/ϵ; $\boxed{\leftarrow}$, $\boxed{\searrow}$; ι/ι, ι/ϵ; $\boxed{\leftarrow}$.

Observe the absence of a transition rule of the form $\boxed{\searrow}$; o/T_1, o/T_2; s', which would correspond to the box with an error in Section 2: not listing any such transition rule

makes a computation on input $\searrow$ and internal configuration (q, o, o) impossible by the underlying pushdown machine. The reader will easily identify the transduction realized by the machine just introduced as the map ϕ of Section 2.

For general p as well as for the reflected $p = 1$ case, we change some of the ingredients defining the previous transducer $\mathbb{T}$. In the first situation, the input alphabet, output alphabet, and the alphabet for the first stack are

$$A = \Sigma_p = \{-1, 0, 1, \ldots, p\}, \qquad B = S_p = \{\boxed{\uparrow}, \ldots, \boxed{\leftarrow}, \ldots, \boxed{\leftarrow}, \boxed{\llcorner}\},$$
$$Z_1 = \{o\} \cup A_p = \{o\} \cup \{a_{\ell,m} : 0 \leqslant \ell + m \leqslant p - 1\}.$$

In the second situation, the input and output alphabets are

$$A = \Sigma_{1,\mathrm{bicol}} = \{\boxed{\nearrow}, \boxed{\rightarrow}, \boxed{\searrow}, \boxed{\nearrow}, \boxed{\Rightarrow}, \boxed{\searrow}\}, \qquad B = S_{1,\mathrm{sym}} = \{\boxed{\uparrow}, \boxed{\leftarrow}, \boxed{\searrow}, \boxed{\downarrow}, \boxed{\rightarrow}, \boxed{\nwarrow}\},$$

and the alphabet for the first stack is unchanged. We will not go here into the details of the corresponding redefinitions for the sets γ of transition rules. We only provide one example, the encoding of the rule (T6): this single box gets expanded into the $(p-1)p/2$ transition rules $-1; a_{\ell,m}/a_{\ell,m+1}, \iota/\epsilon; -1$, one for each pair (ℓ, m) satisfying $\ell + m \leqslant p - 2$.

6   Alternate interpretation via raising

So far, we have provided descriptions of algorithms by transducers that map several classes of quarter-plane walks starting at the origin to related classes of walks in the upper half plane returning to the x-axis. Those maps are described in terms of the alphabet and notation given in Section 1.1, which we repeat here for convenience and to make the present section more independent from the rest. Specifically:

- In Section 2, the map ψ (also denoted Ψ_1) converts an input (classical) tandem walk with steps from $S_1 = \{\boxed{\uparrow}, \boxed{\leftarrow}, \boxed{\searrow}\}$ into a Motzkin walk using steps from $\Sigma_1 = \{-1, 0, +1\} \simeq \{\boxed{\searrow}, \boxed{\rightarrow}, \boxed{\nearrow}\}$.

- In Section 3, the map Ψ_p converts an input (general) p-tandem walk with steps from $S_p = \{\boxed{\uparrow}, \ldots, \boxed{\leftarrow}, \ldots, \boxed{\leftarrow}, \boxed{\llcorner}\}$ into a p-Łukasiewicz walk using steps from $\Sigma_p = \{-1, 0, 1, \ldots, p\}$.

- In Section 4, the map $\Psi_{1,\mathrm{sym}}$ converts an input walk with steps from the symmetrized step set $S_{1,\mathrm{sym}} = \{\boxed{\uparrow}, \boxed{\leftarrow}, \boxed{\searrow}, \boxed{\downarrow}, \boxed{\rightarrow}, \boxed{\nwarrow}\}$ into a bicoloured Motzkin walk using steps from $\Sigma_{1,\mathrm{bicol}} = \{\mathbf{-1}, \mathbf{0}, \mathbf{+1}, -\mathbb{1}, \mathbb{0}, +\mathbb{1}\} \simeq \{\boxed{\nearrow}, \boxed{\rightarrow}, \boxed{\searrow}, \boxed{\nearrow}, \boxed{\Rightarrow}, \boxed{\searrow}\}$, with colours that we name "solid" and "striped".

There is another interpretation of our maps, $\psi, \Psi_p, \Psi_{1,\mathrm{sym}}$, from quarter-plane walks to Motzkin, Łukasiewicz, and bicoloured Motzkin walks. This other interpretation builds the output half-plane walk by, at each stage of the construction, adding a new step at the end and sometimes raising a past step. In this interpretation, stopping the process after dealing with a prefix of a quarter-plane walk results in exactly the action of the full

process on that shorter-length walk. Thus the smaller words constructed along the way will not be prefixes of the ultimately constructed half-plane walk, and the approach can no longer be implemented in a single pass. However, it allows us to define the map ψ by reading the quarter-plane walk from left to right instead of from right to left. Additionally it has a prefix property that is reminiscent of the work by Bousquet-Mélou, Fusy, and Raschel [4]: the process really constructs a sequence of half-plane walks that all return to the x-axis, one for each length up to the total length of the input quarter-plane walk.

In the $p = 1$ case and the reflected $p = 1$ case, the same maps as those we constructed in Sections 2 and 4 have been defined (albeit in a difficult-to-read way) using this raising perspective in [17]. The extension of this raising idea to the $p > 1$ case is new and will be presented in Section 6.4.

The decision of which past step to raise requires knowing whether past steps can be raised. This cannot be determined simply by looking at past steps, as for instance for $p = 1$ a past step 0 could have been raised from -1 in which case it is no longer available to be raised or it may have been originally added to the word as a 0 in which case it has not yet been raised and is available for raising. To keep track of this information, over the course of the construction each Motzkin, Łukasiewicz, or bicoloured Motzkin step carries the additional information of whether it can still be raised, and then this additional data is discarded at the end of the construction. For the $p = 1$ and the flipped $p = 1$ cases, it is not necessary to carry this extra information provided the algorithm is allowed to look back at previous steps in the quarter-plane walk. This is discussed further below.

In the $p > 1$ case we additionally need to keep track of a pairing between certain steps. Again this information is discarded at the end of the algorithm.

6.1 Raising in the 3-step model

To make the idea clear first in the simplest case, consider the 3-step model one more time. It is clearer and easier to generalize to $p > 1$ if, for the duration of the algorithm, we augment the output alphabet so as to include the raisability of the steps. Thus for the duration of the algorithm we will have 5 possible letters in the output word. These will be $\boxed{\rightarrow}$ for a $\boxed{\rightarrow}$ in the Motzkin path that is not raisable and $\boxed{\overset{\bullet}{\rightarrow}}$ for one that is raisable. Similarly there will be $\boxed{\searrow}$ for a $\boxed{\searrow}$ in the Motzkin path that is not raisable and $\boxed{\overset{\bullet}{\searrow}}$ for one that is raisable. The fifth letter in the alphabet will be $\boxed{\nearrow}$ since such steps are never raisable. At the end of the algorithm we forget the extra information and so project onto the smaller alphabet $\{\boxed{\nearrow}, \boxed{\rightarrow}, \boxed{\searrow}\}$ in order to get a Motzkin path.

Given a quarter-plane walk in S_1^* written as a sequence of steps $\bar{a}_1 \bar{a}_2 \ldots \bar{a}_n$, the procedure in Figure 15 computes a sequence of words, M_0 to M_n, by successively obtaining each M_i (of length i) from $\bar{a}_i$ and M_{i-1}. The parsings of M_{i-1} required in (2)b and (2)c are always possible by the quarter-plane property. Note that steps added in case (2)c cannot be raised and no step can be raised more than once.

In the language of [17] a step that is raisable is *marked* and a step that is not raisable is *unmarked*. Alternately, we can determine whether a step can be raised without enlarging the alphabet with $\bullet$ by looking at which $\bar{a}_j$ created the step. This can be done in the

1. Set $M_0 := \epsilon$.

2. For i from 1 to n:

   (a) If $\bar{a}_i = \boxed{\uparrow}$, then $M_i := M_{i-1}\boxed{\overset{\bullet}{\to}}$.

   (b) If $\bar{a}_i = \boxed{\searrow}$ then write $M_{i-1} = w\boxed{\overset{\bullet}{\to}}w'$ where w' is free of $\boxed{\overset{\bullet}{\to}}$, and set $M_i := w\boxed{\nearrow}w'\boxed{\overset{\bullet}{\searrow}}$.

   (c) If $\bar{a}_i = \boxed{\leftarrow}$ then write $M_{i-1} = w\boxed{\overset{\bullet}{\searrow}}w'$ where w' is free of $\boxed{\overset{\bullet}{\searrow}}$ and set $M_i := w\boxed{\to}w'\boxed{\searrow}$.

3. Remove all $\bullet$ from M_n and return the result.

Figure 15: Algorithm by raisings in the 3-step model [17].

following way: Let $m_j^{(i)}$ be the jth step in M_i (with no $\bullet$), then

- if $m_j^{(i)} = \boxed{\to}$ and $\bar{a}_j = \boxed{\uparrow}$ then $m_j^{(i)}$ can still be raised while if $m_j^{(i)} = \boxed{\to}$ and $\bar{a}_j = \boxed{\searrow}$ then $m_j^{(i)}$ has been raised and so can no longer be raised;

- if $m_j^{(i)} = \boxed{\searrow}$ and $\bar{a}_j = \boxed{\searrow}$ then $m_j^{(i)}$ can still be raised while if $m_j^{(i)} = \boxed{\searrow}$ and $\bar{a}_j = \boxed{\leftarrow}$ then $m_j^{(i)}$ can never be raised; and

- if $m_j^{(i)} = \boxed{\nearrow}$ then $\bar{a}_j = \boxed{\uparrow}$ and $m_j^{(i)}$ has been raised and so can no longer be raised.

It turns out that this procedure is an alternate realization of the map ψ from Section 2. See Figure 16 and the related Figure 1 for an example.

It is useful to sketch a proof that the map defined above is ψ because this explains how the idea of raising relates to the transitions that define ψ. To know what step will be in location i in the final output, it suffices to know what step $\bar{a}_i$ is and to know whether or not the letter first output to this location i will have to be raised at some later stage. Whether or not it will be raised is determined by which steps come later in the input, when read forward. In particular every $\bar{a}_j = \boxed{\searrow}$ must raise a step that came from some $\bar{a}_i = \boxed{\uparrow}$ with $i < j$ and every $\bar{a}_j = \boxed{\leftarrow}$ must raise a step that came from some $\bar{a}_i = \boxed{\searrow}$ with $i < j$. In both cases they raise the most recent such steps. If we read the quarter-plane walk in the reverse direction, that is, backwards, we already have the information about whether m_i will need to be raised or not when we get to it. Specifically, we only need to count $\boxed{\leftarrow}$ steps and $\boxed{\searrow}$ steps (these cause raisings) and subtract these amounts from the counts for $\boxed{\searrow}$ and $\boxed{\uparrow}$ steps, respectively (these get raised). This is exactly what h and v are doing in the original definition of ψ. The cases based on h and v give the raised and unraised options for the different input steps.
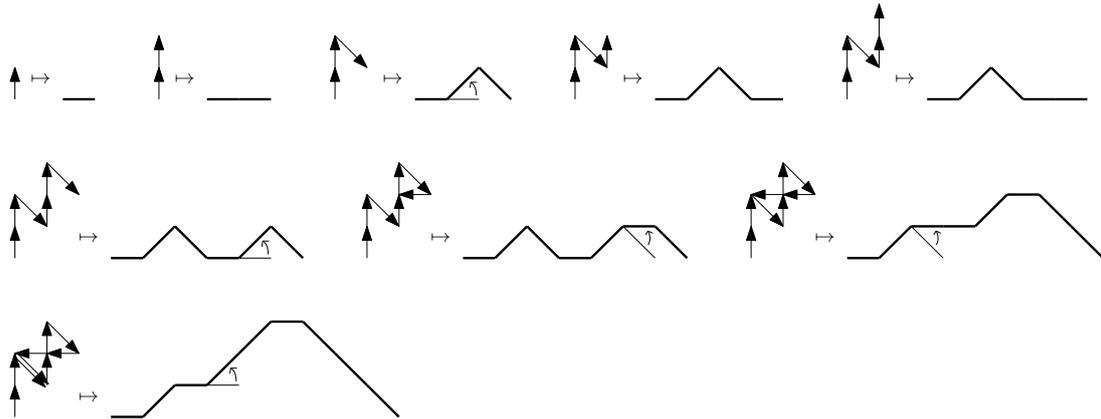
Figure 16: An example of the raising process describing the map ψ on the 3-step model, applied to the word $\bar{w} = \boxed{\uparrow}\boxed{\uparrow}\boxed{\seararrow}\boxed{\uparrow}\boxed{\uparrow}\boxed{\searrow}\boxed{\leftarrow}\boxed{\leftarrow}\boxed{\searrow}$. Note that we could represent all the Motzkin paths in a manner similar to Figure 22: in informal words, each Motzkin path is 'above' all previous ones. Compare with columns (a) and (b) in Figure 1.

The proof in [17] that the map is bijective is both more complicated and more unpleasant than the proofs of this paper, illustrating one of the benefits of our transition-based single pass construction. None-the-less the raisings can be a helpful reformulation for visualizing the map, and explains how to implement ψ while stepping *forward* through the quarter-plane walk, rather than *backward* as with the RL transducer algorithm.

6.2 Freedom of pairing and comparison with Eu, Fu, Hou, and Hsu

Observe that knowing whether or not a step will be raised is sufficient to determine the output word, but it is less information than what is given by the raising algorithm. The raising algorithm additionally pairs each raised step with the step that raised it. The left-hand side of Figure 18 shows the pairing obtained on the example worked out in Figure 16. This leaves open the possibility that other algorithms could be described, which give the same output but a different pairing. In fact, in [9], Eu, Fu, Hou, and Hsu give a description of the bijection by an algorithm giving a different pairing. See also [8].

Their algorithm (Algorithm B of [9, Section 2.2]) from the quarter-plane walk takes three passes each running right to left. This algorithm views the quarter-plane walk as a word in S_1^* and modifies it in place to obtain a Motzkin path. Because of the in-place modification, we need disjoint input and output alphabets, so we will write the Motzkin paths as words in $\{-1, 0, 1\}^*$ for the purposes of describing this algorithm. This yields the algorithm in Figure 17.

As in the raising description, $\boxed{\uparrow}$ steps become 0, which may or may not then be raised to 1 by a later $\boxed{\searrow}$ step, $\boxed{\searrow}$ steps themselves become -1, which may or may not be raised to 0 by a later $\boxed{\leftarrow}$ step, and $\boxed{\leftarrow}$ steps themselves always become -1. By making the three separate passes Eu, Fu, Hou, and Hsu's algorithm does not need to distinguish between those 0 steps that have been raised and those that have not, as first all the unraised 0s

---

Input: a quarter-plane walk written as a word in S_1^*

Output: a Motzkin walk written as a word in $\{-1, 0, 1\}^*$

---

1. (First pass) Moving right to left convert each $\boxed{\uparrow}$ step to 0

2. (Second pass) Moving from right to left, for each $\boxed{\searrow}$ find the closest preceding 0 and convert the pair $(0, \boxed{\searrow})$ to $(1, -1)$.

3. (Third pass) Moving from right to left, for each $\boxed{\leftarrow}$ find the closest preceding -1 and convert the pair $(-1, \boxed{\leftarrow})$ to $(0, -1)$.

4. Output the resulting word in $\{-1, 0, 1\}^*$.

Figure 17: Algorithm by Eu, Fu, Hou, and Hsu in the 3-step model [9].

are generated, then in the second pass some are raised, and finally in the last pass the raised 0s are generated.

Their algorithm also builds the same map ψ because, similarly to the situation in the raising algorithm, to know what step will be in position i in the output it suffices to know what the input step in position i is and whether or not this step will get converted in the second or third pass. To know whether or not a step will be converted, as we read from right to left through the word, we only need to count $\boxed{\searrow}$ steps and $\boxed{\leftarrow}$ steps (these cause conversions in the second and third passes) and subtract from the counts for each $\boxed{\uparrow}$ and $\boxed{\searrow}$ steps, respectively, that we meet (these get converted). This is again exactly what h and v are doing in the original definition of ψ. However, the particular pairing given by the three passes is different from the one given in the raising description.

To see the different pairing, consider the example of Figure 16. As a word the quarter-plane walk is

$$\boxed{\uparrow}, \boxed{\uparrow}, \boxed{\searrow}, \boxed{\uparrow}, \boxed{\uparrow}, \boxed{\searrow}, \boxed{\leftarrow}, \boxed{\leftarrow}, \boxed{\searrow}.$$

After the first pass we have

$$0, 0, \boxed{\searrow}, 0, 0, \boxed{\searrow}, \boxed{\leftarrow}, \boxed{\leftarrow}, \boxed{\searrow}.$$

After the second pass we have

$$0, 1, -1, 0, 0, -1, \boxed{\leftarrow}, \boxed{\leftarrow}, -1,$$

and after the third pass we have

$$0, 1, 0, 1, 1, 0, -1, -1, -1,$$

where the right-hand side of Figure 18 shows the pairings. Observe that the pairings are different though the Motzkin path is the same.
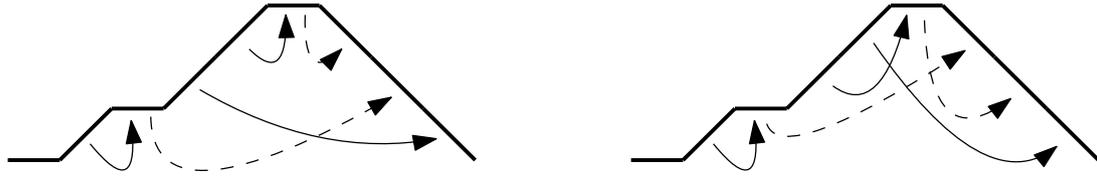
Figure 18: The example of Figure 16 showing the different pairings from the raising algorithm (left) and Eu, Fu, Hou, and Hsu's algorithm (right). Pairings indicated with solid lines come from (2)b in the raising algorithm or the second pass of Eu, Fu, Hou, and Hsu's algorithm while pairings indicated with dashed lines come from (2)c or the third pass.

The raising algorithm gives the pairing that is noncrossing when restricted to those pairings from step (2)b and when restricted to those from (2)c. Eu, Fu, Hou, and Hsu's algorithm gives the pairing that takes the closest step from right to left and so is in some sense maximally crossing in each pass.

6.3 Raising in the six-step model

Passing to the six-step model from the raising perspective is very much the same, see [17] for details. Motzkin steps coming from ⬅ and ⬇ can never be raised, but Motzkin steps coming from the other four quarter-plane steps can be raised. However they fall into two classes (do they go away from the x-axis or do they go away from the y-axis). Likewise all the steps except ⬆ and ➡ must cause raisings of earlier steps. Again they fall into two classes (do they go towards the x-axis or do they go towards the y-axis). This corresponds to the fact that in the transition-based description of the bijection for the six-step model we only need to keep track of two parameters rather than the four that one might naively expect from doubling the number of parameters in the three-step case. Additionally a step raises whichever step in the correct category was most recent with no regard to which kind of step in that category it was. This means that steps and flipped steps interact.

The algorithmic definition of the map has been given as [17, Definition 2.1]. We reproduce it in Figure 19. Given a quarter-plane walk in $S_{1,\mathrm{sym}}^*$ written as a sequence of steps $\bar{a}_1\bar{a}_2\ldots\bar{a}_n$, the procedure in the figure computes a sequence of words, M_0 to M_n, by successively obtaining each M_i (of length i) from $\bar{a}_i$ and M_{i-1}. Again, for the duration of the algorithm, we need to augment our output alphabet by $\bullet$ to indicate raisability.

In [17] the bicolouring is given by red and black where solid here is red there and striped here is black there. The proof that this defines the same map as in Section 4 is analogous to that of the 3-step model.

Input: a quarter-plane walk written as a word $\bar{a}_1\bar{a}_2\ldots\bar{a}_n$ in $S_{1,\text{sym}}^*$

Output: a bicoloured Motzkin walk written as a word in $\Sigma_{1,\text{bicol}}^*$

---

1. Set $M_0 := \epsilon$.

2. For i from 1 to n:

   (a) If $\bar{a}_i = \boxed{\uparrow}$ then $M_i := M_{i-1}\,\text{↠•}$.

   (b) If $\bar{a}_i = \boxed{\rightarrow}$ then $M_i := M_{i-1}\,\text{⇒•}$.

   (c) If $\bar{a}_i = \boxed{\searrow}$ then either $M_{i-1} = w\,\text{↠•}\,w'$ or $M_{i-1} = w\,\text{↘•}\,w'$ with w' free of both ↠• and ↘•. Then set $M_i := w\,\text{↗}\,w'\,\text{↘•}$ if $M_{i-1} = w\,\text{↠•}\,w'$ and $M_i := w\,\text{⇒}\,w'\,\text{↘•}$ otherwise.

   (d) If $\bar{a}_i = \boxed{\nwarrow}$ then either $M_{i-1} = w\,\text{⇒•}\,w'$ or $M_{i-1} = w\,\text{↘•}\,w'$ with w' free of both ⇒• and ↘•. Then set $M_i := w\,\text{↗}\,w'\,\text{↘•}$ if $M_{i-1} = w\,\text{⇒•}\,w'$ and $M_i := w\,\text{→}\,w'\,\text{↘•}$ otherwise.

   (e) If $\bar{a}_i = \boxed{\leftarrow}$ then either $M_{i-1} = w\,\text{⇒•}\,w'$ or $M_{i-1} = w\,\text{↘•}\,w'$ with w' free of both ⇒• and ↘•. Then set $M_i := w\,\text{↗}\,w'\,\text{↘}$ if $M_{i-1} = w\,\text{⇒•}\,w'$ and $M_i := w\,\text{→}\,w'\,\text{↘}$ otherwise.

   (f) If $\bar{a}_i = \boxed{\downarrow}$ then either $M_{i-1} = w\,\text{↠•}\,w'$ or $M_{i-1} = w\,\text{↘•}\,w'$ with w' free of both ↠• and ↘•. Then set $M_i := w\,\text{↗}\,w'\,\text{↘}$ if $M_{i-1} = w\,\text{↠•}\,w'$ and $M_i := w\,\text{⇒}\,w'\,\text{↘}$ otherwise.

3. Remove all $\bullet$ from M_n and return the result.

Figure 19: Algorithm by raisings in the 6-step model [17].

6.4 Raising for $p > 1$

For $p > 1$ we can move beyond [17] and still give a raising interpretation for Ψ_p. There are additional complications compared to the $p = 1$ cases as steps may be raised more than once, may be frozen from further raising by later steps, and may pass back their raising to a previous step (that is, at a position to its left). A first, rough description of the process follows, before we devote the rest of the section to formal details, culminating with the result that a raising algorithm (given in Figure 20) and the RL transducer algorithm compute the same object (Theorem 13). Reading forwards through the quarter-plane walk, take a letter $\bar{a}$ in the latter to the smallest Łukasiewicz letter μ that is possible under some transition for that quarter-plane letter $\bar{a}$. Most of the time this will break the Łukasiewicz property. Then raise enough previous steps to re-obtain the property. The steps to raise are recent ones that came from quarter-plane steps that went away from whichever axis $\bar{a}$ goes towards and that have not already been maximally raised.

6.4.1 Raisability and modality of raising

As for the case $p = 1$, we will keep track of raisability by augmenting the alphabet used during the run of our algorithm, and discarding the raisability information at the end of it. *Raisability* is now an integer parameter $j \geqslant 0$, indicating how many times the step can be raised (each time by 1, as we will see). However, we need to keep track of more than just the raisability of a step, namely we need to keep track of which kind of future steps it can be raised by and whether it can be raised itself or has to pass its raising on to a previous "proxy" step. We will encode this *modality* of raising with a second integer parameter $t \geqslant -1$. Depending on modality, steps separate into three sorts. Steps with modality $t = -1$ can be raised themselves, but only by one of the p steps of the form $(-m, p - m)$ for $m \geqslant 1$ (those having some strict west component). Steps with modality $t = 0$ can be raised themselves, but only by a step $(1, -1)$. Steps with modality $t \geqslant 1$ can only be raised by proxy, by a step $(1, -1)$, and the value of t then indicates the distance to the proxy step. (Observe that, conversely, a step $(0, p)$ never causes a raising.)

We arrive at an alphabet of letters that we denote $\overset{j,t}{(\mu)}$. For example, $\overset{4,-1}{(0)}$ represents a 0 letter from the Łukasiewicz alphabet that can be raised 4 times, by a step with strict west component and so has no proxy, while $\overset{1,3}{(-1)}$ is a letter -1 that can be raised once, by a south-east step and has the letter three positions before it as its proxy. (Beware that what we will soon define as *a step getting raised* is marked by the decrease of its raisability j, and not necessarily by the increase of its nominal value μ.)

Given a quarter-plane walk in S_p^* written as a sequence of steps $\bar{a}_1 \bar{a}_2 \dots \bar{a}_n$, the procedure in Figure 20 computes a sequence of words, L_0 to L_n, by successively obtaining each L_i (of length i) from $\bar{a}_i$ and L_{i-1}. Observe that the factorization in the second point of (2)b is always possible because the only case yielding $t > 0$, case (2)c, also places some $\overset{0,-1}{(\mu')}$ to its left, which then can only be altered into some $\overset{0,-1}{(\mu'')}$ for $\mu'' \geqslant \mu'$. Also, (2)c can be rephrased as follows: find the m most recent raisable letters with modality -1, raise the mth oldest one, and remove the raisability of the m letters for the rest of the process. Furthermore, the $m - 1$ more recent ones are permanently frozen as their raisability is 0 and they cannot be the proxy steps for any future step. The mth most recent of the letters that were found is different. Its raisability has also been set to 0 but it is a proxy. To avoid ambiguities we will use the following language with respect to raising:

- In cases (2)c and (2)b we say that the step at position i *causes a raising*.

- In case (2)b, in both the $t = 0$ and $t > 0$ cases we say that the letter $\overset{j,t}{(\mu)}$ *gets raised*, becoming $\overset{j-1,t}{(\mu + 1)}$ and $\overset{j-1,t}{(\mu)}$ respectively, and that the obtained letter *was raised by* the step at position i. The $t > 0$ case is a raising that is passed on to a proxy step, precisely $\overset{0,-1}{(\mu')}$, but it still counts as a raising for μ not for μ'.

- In case (2)c we say that the letter $\overset{j_1,-1}{(\mu_1)}$ *gets raised*, becoming $\overset{0,-1}{(\mu_1 + 1)}$, and that the

---

Input: a quarter-plane p-tandem walk written as a word $\bar{a}_1\bar{a}_2\ldots\bar{a}_n$ in S_p^*

Output: a p-Łukasiewicz walk written as a word in Σ_p^*

---

1. Set $L_0 := \epsilon$.

2. For i from 1 to n:

   (a) If $\bar{a}_i = (0, p)$ then $L_i := L_{i-1}\overset{p,0}{(0)}$.

   (b) If $\bar{a}_i = (1, -1)$ then write $L_{i-1} = w\overset{j,t}{(\mu)}w'$ where $j > 0$ and $t \geqslant 0$ and where w' has no letters with raisability > 0 and modality $\geqslant 0$.

      - If $t = 0$ then $L_i := w\overset{j-1,0}{(\mu + 1)}w'\overset{1,-1}{(-1)}$.
      - If $t > 0$ then rewrite $L_{i-1} = w''\overset{0,-1}{(\mu')}w'''\overset{j,t}{(\mu)}w'$ where w''' has length $t - 1$. Then $L_i := w''\overset{0,-1}{(\mu' + 1)}w'''\overset{j-1,t}{(\mu)}\,w'\overset{1,-1}{(-1)}$.

   (c) If $\bar{a}_i = (-m, p - m)$ with $p \geqslant m \geqslant 1$ then write

   $$L_{i-1} = w_0\overset{j_1,-1}{(\mu_1)}w_1\overset{j_2,-1}{(\mu_2)}w_2 \cdots w_{m-1}\overset{j_m,-1}{(\mu_m)}w_m$$

   where, for each $\ell > 0$, j_ℓ is > 0 and w_ℓ is free of steps with raisability > 0 and modality -1. Then

   $$L_i := w_0\overset{0,-1}{(\mu_1 + 1)}w_1\overset{0,-1}{(\mu_2)}w_2 \cdots w_{m-1}\overset{0,-1}{(\mu_m)}w_m\overset{p-m,d}{(-1)}$$

   where $d = |w_m| + |w_{m-1}| + \cdots + |w_1| + m = i - 1 - |w_0|$.

3. Remove the raisability and modality information from L_n, that is, map each $\overset{j,t}{(\mu)}$ to a plain μ, and return the result.

Figure 20: Algorithm by raisings for general $p \geqslant 1$.

obtained letter *was raised by* the step at position i. We say the $\overset{j_\ell,-1}{(\mu_\ell)}$ for $1 \leqslant \ell \leqslant m$ *get frozen by* the step at position i, and this also does not count as a raising.

Note that the raisability of a letter is never increased and only letters with positive raisability can get raised.

Example 9. We continue Example 3 by providing in Figures 21 and 22 an example of execution of the raising algorithm on the same 5-tandem walk $\bar{w}$. The forward process exemplified in Figure 21, with input $\bar{w}$ (at the top) and output w (at the bottom), should be compared to the RL transducer process in Figure 11, with same input (at the bot-

```
 w̄ | 5    D    D    D    D  | D    3    D    D    3  | D    4    5    D    D  | D    D    4    D    D  | D
    | a    b1   b1   b1   b1 | b1   c    b2   b2   c  | b2   c    a    b1   b1 | b1   b1   c    b2   b2 | b2
 ---+------------------------+-----------------------+-----------------------+-----------------------+----
       5,0
  1   (0)
       4,0  1,-1
  2   (1)  (D)
       3,0  1,-1 1,-1
  3   (2)  (D)  (D)
       2,0  1,-1 1,-1 1,-1
  4   (3)  (D)  (D)  (D)
       1,0  1,-1 1,-1 1,-1 1,-1
  5   (4)  (D)  (D)  (D)  (D)
       0,0  1,-1 1,-1 1,-1 1,-1 | 1,-1
  6   (5)  (D)  (D)  (D)  (D)  | (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 3,2
  7   (5)  (D)  (D)  (D)  (0)  | (D)  (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 2,2  1,-1
  8   (5)  (D)  (D)  (D)  (1)  | (D)  (D)  (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 1,2  1,-1 1,-1
  9   (5)  (D)  (D)  (D)  (2)  | (D)  (D)  (D)  (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 1,2  0,-1 0,-1 3,2
 10   (5)  (D)  (D)  (D)  (2)  | (D)  (D)  (0)  (D)  (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 1,2  0,-1 0,-1 2,2  | 1,-1
 11   (5)  (D)  (D)  (D)  (2)  | (D)  (D)  (1)  (D)  (D)  | (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 1,2  0,-1 0,-1 2,2  | 0,-1 4,1
 12   (5)  (D)  (D)  (D)  (2)  | (D)  (D)  (1)  (D)  (D)  | (0)  (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 1,2  0,-1 0,-1 2,2  | 0,-1 4,1  5,0
 13   (5)  (D)  (D)  (D)  (2)  | (D)  (D)  (1)  (D)  (D)  | (0)  (D)  (0)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 1,2  0,-1 0,-1 2,2  | 0,-1 4,1  4,0  1,-1
 14   (5)  (D)  (D)  (D)  (2)  | (D)  (D)  (1)  (D)  (D)  | (0)  (D)  (1)  (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 1,2  0,-1 0,-1 2,2  | 0,-1 4,1  3,0  1,-1 1,-1
 15   (5)  (D)  (D)  (D)  (2)  | (D)  (D)  (1)  (D)  (D)  | (0)  (D)  (2)  (D)  (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 1,2  0,-1 0,-1 2,2  | 0,-1 4,1  2,0  1,-1 1,-1 | 1,-1
 16   (5)  (D)  (D)  (D)  (2)  | (D)  (D)  (1)  (D)  (D)  | (0)  (D)  (3)  (D)  (D)  | (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 1,2  0,-1 0,-1 2,2  | 0,-1 4,1  1,0  1,-1 1,-1 | 1,-1 1,-1
 17   (5)  (D)  (D)  (D)  (2)  | (D)  (D)  (1)  (D)  (D)  | (0)  (D)  (4)  (D)  (D)  | (D)  (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 1,2  0,-1 0,-1 2,2  | 0,-1 4,1  1,0  1,-1 1,-1 | 1,-1 0,-1 4,1
 18   (5)  (D)  (D)  (D)  (2)  | (D)  (D)  (1)  (D)  (D)  | (0)  (D)  (4)  (D)  (D)  | (D)  (0)  (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 1,2  0,-1 0,-1 2,2  | 0,-1 4,1  1,0  1,-1 1,-1 | 1,-1 0,-1 3,1  1,-1
 19   (5)  (D)  (D)  (D)  (2)  | (D)  (D)  (1)  (D)  (D)  | (0)  (D)  (4)  (D)  (D)  | (D)  (1)  (D)  (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 1,2  0,-1 0,-1 2,2  | 0,-1 4,1  1,0  1,-1 1,-1 | 1,-1 0,-1 2,1  1,-1 1,-1
 20   (5)  (D)  (D)  (D)  (2)  | (D)  (D)  (1)  (D)  (D)  | (0)  (D)  (4)  (D)  (D)  | (D)  (2)  (D)  (D)  (D)
       0,0  1,-1 1,-1 1,-1 0,-1 | 0,-1 1,2  0,-1 0,-1 2,2  | 0,-1 4,1  1,0  1,-1 1,-1 | 1,-1 0,-1 1,1  1,-1 1,-1 | 1,-1
 21   (5)  (D)  (D)  (D)  (2)  | (D)  (D)  (1)  (D)  (D)  | (0)  (D)  (4)  (D)  (D)  | (D)  (3)  (D)  (D)  (D)  | (D)
 ---+------------------------+-----------------------+-----------------------+-----------------------+----
  w | 5    D    D    D    2  | D    D    1    D    D  | 0    D    4    D    D  | D    3    D    D    D  | D
```

Figure 21:   Example of execution of the raising algorithm on the 5-tandem walk $\bar{w}$ of Figure 10 (right). The letters $\bar{\mu}$ in row $\bar{w}$ (input) give the p-tandem word $\bar{w}$ as the succession of steps $(\bar{\mu} - 5, \bar{\mu})$, or $(1, -1)$ if $\bar{\mu} = \mathrm{D}$. The letters μ in row w (output) give the Łukasiewicz walk w as the succession of steps $(1, \mu)$, with D standing again for -1. The second row indicates the case used in the algorithm: 'a' for case (2)a; 'b1' and 'b2' for the first and second item of case (2)b; 'c' for case (2)c. The rows numbered from 1 to 21 provide the successive Łukasiewicz words L_i constructed by the algorithm.
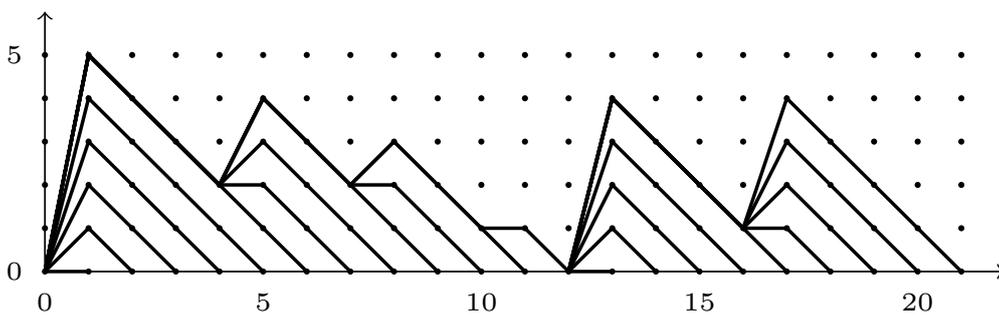

Figure 22:   Joint graphical representation of the successive Łukasiewicz words L_i constructed by the raising algorithm in the run of Figure 21. Each walk L_i is above all L_j for $j < i$ and the sequence ends with the Łukasiewicz walk w of Figure 10 (left).

input step $\bar{a}_j$	small step $(1, -1)$	west long step $(-p, 0)$	other long step with $y > 0$
variation in v	$+1$	0	$\leqslant 0$, in $\{-y, \ldots, 0\}$
transitions	(T5), (T6), (T7)	(T3) or (T4), when $m = p - 1$	(T3) or (T4), when $m < p - 1$
case	(2)b	(2)c with $m = p$	(2)a or (2)c with $m < p$
output modality	-1	> 0 with raisability 0	$\geqslant 0$
role in (2)b	cause one raising	none	output may get raised up to y times

Figure 23: Impact of the x-coordinate of input steps $\bar{a}_j$ in the flow of the RL transducer process and in the flow of the raising algorithm. Transitions have to be understood in reverse form, and variation in v is signed according to the RL flow.

tom) and output (at the top). The Łukasiewicz words L_i constructed are first shown in there precise form with raisability and modality in Figure 21, and next as plain walks in Figure 22.

To make the connection between the raising algorithm and the transducer algorithm more precise in Theorem 13 we need a few lemmas. We gather simple properties in Figures 23 and 24, then give the more elaborate Lemmas 11 and 12. For the rest of the section, let $\bar{w}$ be a quarter-plane walk in S_p^* with letters $\bar{a}_j$.

Immediate properties of the RL process and of the raising algorithm can be derived from the two algorithm descriptions. They are expressed in terms of the nature of steps in the input word $\bar{w}$, by trichotomies according to their x-coordinate and y-coordinate, respectively. These properties are gathered as tables in Figures 23 and 24 (only their last rows are less immediate and will be proved in lemmas). The trichotomy with respect to y matches exactly the case distinction in the raising algorithm (into (2)a, (2)b, (2)c); it also matches a partitioning of the transitions. In contrast, the trichotomy with respect to x does not directly match the computational case distinctions in our algorithms, but further refinements of case (2)c and of (T3) and (T4) provide a suitable partitioning. (We could have gathered the two tables by a partitioning into four columns, at the cost of some repetition.)

6.4.2 Augmented RL transducer process: a more elaborate stack V

For the purposes of the first lemma, we will reinterpret the parameter v from the RL transducer algorithm as the length of a new stack V added to the algorithm. The letters allowed on V will be integers representing positions in the input word $\bar{w}$. Thus it will make sense to speak of which letter or transition instance put a particular element onto V

input step $\bar{a}_j$	small step $(1, -1)$	north long step $(0, p)$	other long step with $x < 0$
effect on H	$a_{\ell,m'} \mapsto a_{\ell,m'-1},$ pop $a_{\ell,0}$, or do nothing	none	push $a_{\ell,|x|-1}$
transitions	(T6), (T5), or (T7) respectively	(T1) or (T2)	(T3) or (T4)
case	(2)b	(2)a	(2)c
output modality	-1	0	> 0
role in (2)c	output gets frozen, raised or is never touched by (2)c	none	cause one raising and potentially freezings

Figure 24: Impact of the y-coordinate of input steps $\bar{a}_j$ in the flow of the RL transducer process and in the flow of the raising algorithm. Transitions have to be understood in reverse form, and alteration of H is according to the RL flow.

and which took it off, and the stack elements will keep track of these indices. Specifically, for the rest of this section, we augment the RL process with the stack V and we modify the reverses of the transitions (T1) to (T7). In the augmented RL process, the reverses of (T5), (T6), and (T7) push i onto the stack V whenever those transitions are used at stage $n + 1 - i$, that is, when dealing with the input letter $\bar{a}_i$. Moreover, the reverses of (T1), (T2), (T3), and (T4) pop respectively p, q, ℓ, and ℓ elements from the stack V. (Those numbers are defined in the transition descriptions.)

Example 10. We consider again Example 3 and in particular the evolution of counters along LR and RL transductions provided by Figure 11. As represented by the row for V, the augmented RL transducer process successively pushes 21, 20, and 19 to the stack V by three instances of (T7), before an instance of (T3) pops all three integers, emptying the stack. Four similar cycles of successive pushes before emptying the stack occur in the rest of the run. Other examples involving (T2) would similarly empty the stack; on the other hand, examples with longer runs of successive pushes could lead to the use of (T4) or (T1), causing pops of several integers without fully emptying the stack.

6.4.3 Relation between steps with positive y-coordinates, the stack V, and raisings

Lemma 11. *Consider an index j in $\bar{w}$ for which $\bar{a}_j$ has positive y-coordinate. When processing $\bar{a}_j$, the RL transducer algorithm either*

- *leaves the stack V unchanged, in which case the letter in position j is never raised in the raising algorithm, or*

- *takes off at least one element from it, in which case the steps (with indices $> j$) that put on those elements taken off by $\bar{a}_j$ are precisely the steps that raise the letter in position j in the raising algorithm.*

Proof. Consider the trichotomy of input steps given by the $(1, -1)$ steps, the $(-p, 0)$ steps, and the steps with positive y-coordinate. We will now to describe, in relation to this trichotomy, which transitions in the RL transducer algorithm and which cases in the raising algorithm deal with the steps. We will also describe how the steps affect V in the RL process and are involved in raisings of steps of modality $\geqslant 0$ in the raising algorithm. The table in Figure 23 captures facts, all of which can be immediately derived from the two algorithm descriptions, with the exception of the final row, which requires additional explanation. This row is addressing how the input steps are involved in raisings of steps with modality $\geqslant 0$, which can occur only in case (2)b (as a raising requires raisability > 0). All $(1, -1)$ steps are dealt with by case (2)b and vice versa; each time exactly one raising of a step of modality $\geqslant 0$ is caused (this raising may or may not involve a proxy, but that is of no concern for this lemma). The step $(-p, 0)$ has no involvement with raisings of steps of modality $\geqslant 0$ as the raisings it causes are on steps of modality -1 and its output step is unraisable from its creation. The steps with positive y-coordinate have output steps of modality $\geqslant 0$ with nonzero raisability: each such output step starts with raisability equal to the y value of the input step and may get raised any number of times between 0 and this value. Steps with positive y do cause raisings but only on steps of modality -1 which are not of present concern.

Therefore the input steps that raise steps of nonnegative modality (that is, those dealt with by case (2)b) and the steps that put an element on V (that is, those that increase v) are the same, namely the $(1, -1)$ steps. Furthermore the steps with positive y-coordinate are the only ones which may take elements off V and are also the only ones whose corresponding outputs may get raised by step (2)b.

It remains to show that the these classes of steps are associated to each other as described in the statement of the lemma. This is a parenthesis matching problem. Let each step with positive y-coordinate correspond to as many open parentheses as its y-coordinate and each $(1, -1)$ correspond to a close parenthesis. In this way the input word is associated to a string of parentheses. Note that there may be more open parentheses in this string than close parentheses. The final thing we need to prove is that the RL process and the raising process both induce a natural matching of these parentheses, both being in fact the same matching. By remembering what input letter has introduced each parenthesis, this matching induces an association between input letters, each step with positive y-coordinate being potentially linked to several $(1, -1)$ more rightwards. Specifically, this association will be the one described in the statement of the lemma.

Consider first the RL transducer process. Reading the string of parentheses from right to left in parallel to the RL process, a close parenthesis corresponds to adding an element to V, open parentheses to removing elements of V provided V is not empty yet. By inspecting the transition rules, we see that in the RL process, when v decreases it always decreases by as much as it can (namely by the minimum of its current value and the y-coordinate of the input step). Consequently the parentheses are matched naturally from right to left in a last in first out manner, with the additional rule that unmatchable opens are skipped when they are met. The result of this is the parenthesis matching obtained by forcing all extra opens to be as leftwards as possible while retaining the possibility

to match all closes. Furthermore, the links between input letters naturally induced from the parenthesis matching are such that any step $\bar{a}_j$ with positive y-coordinate is linked to exactly those input steps in the RL process that added elements of V that $\bar{a}_j$ later removes. The stack V remains unchanged precisely when the linking induced from the parenthesis matching does not link $\bar{a}_j$ to any other steps.

Now consider the raising process. The reader can check that the raisability of the output letter at position j after dealing with the input letter $\bar{a}_i$ is equal to the number of unmatched open parentheses in the group originating from $\bar{a}_j$ when we consider the left-to-right last in first out parentheses-matching algorithm after all groups originating from $\bar{a}_1, \ldots, \bar{a}_i$ have been dealt with. This can be verified by a step-by-step comparison or a straightforward induction. This gives the same parenthesis matching as the RL transducer process. Furthermore, the links between input letters naturally induced from the parenthesis matching are such that any step $\bar{a}_j$ with positive y-coordinate is linked to exactly those input steps which raised the step at position j. The step at position j is never raised precisely when the linking induced from the parenthesis matching does not link $\bar{a}_j$ to any other steps.

The concluding sentences of the previous two paragraphs together prove the lemma.
□

6.4.4 Relation between steps with negative x-coordinates, the stack H, and freezings of raisings

Letters with negative x-coordinates cause a push onto H in the RL transducer algorithm and no other letters cause a push onto H. Let $a_{\ell,m-1}$ be a letter that is pushed onto H. Observe that the only way for $a_{\ell,m-1}$ to be taken off the stack is by $m-1$ (T6) transition instances and then a (T5) transition instance (not necessarily consecutively). Note also that in the raising algorithm the case that deals with letters with negative x-coordinate is case (2)c. These simple observations are refined in the next lemma.

Lemma 12. *Let j be an index in $\bar{w}$ for which $\bar{a}_j$ has negative x-coordinate, so that some letter $a_{\ell,m-1}$ is pushed on H at step j in the RL process. Then, the indices where the $m-1$ (T6) transition instances occur are precisely the indices of the letters that are permanently frozen by case (2)c for $\bar{a}_j$, and the index where the (T5) instance occurs is the index of the letter raised by $\bar{a}_j$.*

Additionally, indices where (T7) transition instances occur correspond to indices where the output step in the raising algorithm has modality -1 but is never raised or frozen.

Proof. The relevant trichotomy of input letters for this lemma is based on x-coordinate rather than y-coordinate. A table again helps to collect the key information that can be read off the definitions of the two algorithms, and is given in Figure 24. The final row addresses how the input steps and their outputs are involved in raisings and freezings of steps of modality -1. These raisings and freezings occur only in case (2)c. As we will prove in what follows, the three possibilities in the bottom left entry of the table correspond respectively with the three possibilities in each of the first and second left entries of the table.

Again we have a parenthesization problem. Associate every $(1, -1)$ step in the input to an open parenthesis. Associate every input step with x-coordinate $x < 0$ with $|x|$ close parentheses. The input word is again associated to a string of parentheses and again there may be more open parentheses in this string than close parentheses. The RL process and the raising process give natural matchings of these parentheses, and the final claim is that they both give the same matching, and that specifically, the matching is as described in the statement of the lemma.

Consider first the RL transducer process. Reading the string of parentheses from right to left in parallel to the RL process, every time an element $a_{\ell, |x|-1}$ is added to the stack H, $|x|$ close parentheses appear in the parenthesis string, and close parentheses can appear in no other way. Every time the second coordinate of the top of H is decreased by 1 as well as every time the top of H is popped, there is an open parenthesis that matches the most recent unmatched close parenthesis; additionally, each instance of transition (T7), which can only occur when H is empty, introduces one open parenthesis, which remains unmatched.

We proceed to justify that such a matching is always possible. Recall that the m-weight of $a_{\ell, m'-1}$ is m' (see Section 3.3). Thus, because the RL transducer process makes no other kind of alteration to H, at any given point in the RL process, $\mathrm{wt}_m(H)$ is the number of unmatched close parentheses at the corresponding point in the parenthesis string. In particular the matching is always possible. Also, the open parentheses from (T7) steps only occur when H is empty, that is, when there are no unmatched close parentheses.

Therefore, the RL transducer algorithm matches the parentheses from right to left in a last in first out manner with the additional rule that unmatchable opens are skipped when they are met. Furthermore, the links between input letters naturally induced from the parenthesis matching are such that any step $\bar{a}_j$ with negative x-coordinate is linked to exactly those steps that modify (via (T6)) or pop (via (T5)) the element of H that $\bar{a}_j$ pushed.

Now consider the raising process. The reader can verify by a straightforward induction that for a $(1, -1)$ input step at some position j, the output letter from this step, after dealing with the input letter $\bar{a}_i$ for $i \geqslant j$, is raisable if and only if the open parenthesis from the $(1, -1)$ step is unmatched when we consider the left-to-right last in first out parentheses-matching algorithm after all parentheses originating from $\bar{a}_1, \ldots, \bar{a}_i$ have been dealt with. Thus the raising algorithm matches the parentheses in a left to right, last in first out manner. This gives the same parenthesis matching as the RL transducer process. Observe that the unmatched open parentheses correspond to output letters from $(1, -1)$ input steps which are still raisable at the end of the algorithm. Since such output steps are raisable at most once and freezing also sets the raisability to 0, these are steps which were never frozen or raised. Furthermore, the links between input letters naturally induced from the parenthesis matching are such that any step $\bar{a}_j$ with negative x-coordinate is linked to exactly those steps that were either frozen or raised by $\bar{a}_j$ by step (2)c.

6.4.5 Equivalence between the raising and transducer processes

The concluding sentences of the previous two paragraphs prove the first statement of the lemma. The characterization of the unmatched open parentheses in the two algorithms implies the second statement of the lemma. □

Theorem 13. *The raising algorithm in Figure 20 and the RL transducer algorithm produce the same output.*

Proof. We prove the result for the augmented RL transducer algorithm, which has the same output as the unmodified RL transducer algorithm. The proof considers each position j independently and goes by a case analysis on the input letter $\bar{a}_j$:

- Suppose $\bar{a}_j$ is $(0, p)$. Case (2)a stipulates that the first letter output by the raising algorithm at position j is 0 and the second point of case (2)b stipulates that when this step is raised its output letter increases by 1 for each raising. Thus the final output by the raising algorithm at position j is the number of times the step in position j is raised. On the other hand, in the RL transducer algorithm, transitions (T1) and (T2) together make the output at j be the number of elements of V that are removed at step j. Therefore Lemma 11 implies that the final output of both algorithms at position j is the same.

- Suppose $\bar{a}_j$ is $(1, -1)$. We refine the case analysis according the the transition used at that position j by the RL process:

    - If $\bar{a}_j$ triggers a (T6) instance, then the output letter in this position by the RL process is -1, and by Lemma 12 the output letter in this position in the raising process was frozen and so is also -1 (by steps (2)b and step (2)c, along with the fact that Lemma 12 accounts for all transitions that affect H).

    - If $\bar{a}_j$ triggers a (T5) then Lemma 12 also implies that the output by the raising algorithm at that position j is turned into a proxy step after dealing with some position $i \geqslant j$. The step at position i is the step that originally put on the stack letter that the (T5) instance removed. The output of this (T5) instance is the value of the first index of this stack letter, but this is the number of elements taken off of V by $\bar{a}_i$. Lemma 11 implies that this number is the same as the number of steps that raised the output letter at position i. Hence by the nature of proxy raising, this number is the final output value at position j. Therefore both algorithms also agree in this case.

    - If $\bar{a}_j$ triggers a (T7) then the RL transducer algorithm outputs a -1; in the raising algorithm, the output step at position j first appears as a -1 by case (2)b, while by Lemma 12 it is never raised, and so it is -1 in the final output.

- Suppose $\bar{a}_j$ has a negative x-coordinate. All these cases yield -1 as output steps in both algorithms.

Therefore, the description of Ψ_p by raisings agrees with the transducer description. $\quad\square$

Overall, the raising description is more complicated than the transition-based description of Ψ_p.

This is a good moment to return to Proposition 7. The (T2) and (T3) steps with $p - m - 1 > \ell$ described in the first point of the proposition are the steps that could be raised but do not get raised by the end of the raising algorithm. This gives an intuitive explanation for why they correspond to an increased bound on the abscissa for the quarter-plane walk: if the quarter-plane walk were extended to return to lower x-values, then sufficiently many steps must be present in the p-Łukasiewicz walk that can still be raised to correspond to these extension steps. The extra capacity for raising corresponds to the bound on the abscissa, in the precise manner given by Proposition 7.

Additionally, the converse of the third item of Proposition 7 is also true. To see this, suppose the quarter-plane walk $\bar{w}$ is a concatenation of $\bar{w}_1$ and $\bar{w}_2$. By the raising description of Ψ_p, once all the steps of $\bar{w}_1$ have been processed we have $\Psi_p(\bar{w}_1)$. Since $\bar{w}_2$ is also a quarter-plane walk, Ψ_p applied to $\bar{w}_2$ is well-defined. This implies that whenever raising occurs while working through $\bar{w}_2$, the steps to be raised must remain within $\bar{w}_2$. Therefore the same holds with Ψ_p applied to $\bar{w}_2$ as it occurs within $\bar{w}$. Therefore $\Psi_p(\bar{w})$ is the concatenation of $\Psi_p(\bar{w}_1)$ and $\Psi_p(\bar{w}_2)$, each of which are p-Łukasiewicz walks. Hence there is a return to the x-axis at the concatenation point, giving the converse to the second point of Proposition 7.

7 Discussion

7.1 Comparison to the approach by oriented planar maps

The work by Bousquet-Mélou, Fusy, and Raschel [4] studies a larger class of generalized tandem walks, with (infinite) step set

$$\{\searembox\} \cup \bigcup_{p \geqslant 1} S_p,$$

and it provides a length-preserving involution σ of this class that also preserves the number of $\searembox$ steps and, for each p, the total number of steps in S_p. More importantly, this involution exchanges the differences $a = x_{\text{start}} - x_{\text{min}}$ and $d = y_{\text{end}} - y_{\text{min}}$ and preserves the differences $b = y_{\text{start}} - y_{\text{min}}$ and $c = x_{\text{end}} - x_{\text{min}}$ (with obvious notation). By a suitable restriction, namely, forcing the number of steps from the S_p to be 0 except for a single p and considering walks with $a = d = 0$ on the one hand (half plane), and walks with $a = b = 0$ on the other hand (quarter plane), this induces a bijection between $\mathcal{L}_p$ and $\mathcal{T}_p$. In particular, the involution σ maps each prefix of a given $\bar{w} \in \mathcal{T}_p$ to a word of $\mathcal{L}_p$. This makes it impossible for the sequence of images in $\mathcal{L}_p$ to be the prefixes of the image of the full word $\bar{w}$. Thus, σ breaks the principal requirement of our construction, and our bijections Φ_p and Ψ_p are no restrictions of the involution σ. Further, our bijection preserves $y_{\text{end}} - x_{\text{end}}$, which has no clear behaviour under σ.

It remains of interest to determine if σ could be implemented by an automaton similar to those we introduced. In this regard, it is interesting to note that the quantities exchanged by the involution σ involve the records $x_{\min}$ and $y_{\min}$, which cannot be modeled without extending the notion of parameters that we introduced in Section 3.2: one needs to augment the "arithmetic" of parameters to allow comparisons.

7.2 Symmetrization beyond $p = 1$

The generalization from the $p = 1$ case to the symmetrized $p = 1$ case is surprisingly nice; we keep track of the very same counters and simply slot in the extra transitions. One might have expected a need for four counters, doubling counters as well as steps, but two suffices. Having made this generalization the obvious next step would be to generalize $p > 1$ to some sort of symmetrized situation. Unfortunately, we do not know how to do this.

Take $S_{p,\mathrm{sym}}$ to be the union of S_p and the reflection of S_p in the line $y = x$. The first question is what sets of walks ought be in bijection. We would expect the generalized $\Psi_{p,\mathrm{sym}}$ to have quarter-plane walks in $S_{p,\mathrm{sym}}^*$ as its domain. The most obvious answers for the image of $\Psi_{p,\mathrm{sym}}$ would be either bicoloured Łukasiewicz walks or half-plane walks ending on the x-axis in $S_{p,\mathrm{sym}}^*$. The latter suggestion is in view of the bijection between Łukasiewicz walks and half-plane walks ending on the x-axis in S_p^*. However, direct counting of small walks in these sets show that neither of these are equinumerous with quarter-plane walks in $S_{p,\mathrm{sym}}^*$.

Leaving this problem aside for the moment, we could try to generalize the transitions in the hopes that the image itself will become clear in due time. Analogously to the relationship between the $p = 1$ transitions and the symmetrized $p = 1$ transitions, we would expect that restricting to only the flipped steps as input should give us something equivalent to the original map but with differently coloured output steps and with the parameters swapped. Thus we expect that transitions for a $\Psi_{p,\mathrm{sym}}$ map would have two stacks as parameters and would specialize to Ψ_p in two ways. Unfortunately, neither considering the symmetries of the transitions nor working through many small examples have shown how this could be done.

Consequently, we are left without a clear idea of how to generalize our maps to a symmetrized $p > 1$ situation.

7.3 Standard Young tableaux of bounded height

In their work [9], Eu, Fu, Hou, and Hsu gave an algorithm for the bijection when $p = 1$ that is similar in spirit to the algorithm by raising. See our comments in Section 6.2. This bijection, between (usual) Motzkin walks and Yamanouchi words as we described it in Section 2, really is a bijection between Motzkin walks and standard Young tableaux of height at most 3. But the true contribution of [9] was to generalize it to a bijection between standard Young tableaux of height at most d and a class of coloured Motzkin walks that they introduced. Their "d-Motzkin walks" are Motzkin walks whose up and down steps may appear in d-colours, with an additional constraint between the occurrences

of the colours that they describe by inequalities. The bijection works by treating and removing the colours one after the other. This makes it very tempting to try and apply the transducer approach to d-Motzkin walks. Our attempts have failed so far. If one conceives the transducer approach applied to (usual) Motzkin walks as a means to reschedule the raising transformations so that they can be applied as soon as a raised step is considered, instead of postponing to when the process needs it to restore a Motzkin property, then a transducer algorithm for bounded height would develop such a rescheduling for each colour, and the d transducers thus obtained would then have to be merged into a single one, leading to further rescheduling. It would be great to see such an algorithm.

Acknowledgments

References

[1] Jean-Michel Autebert, Jean Berstel, and Luc Boasson. Context-free languages and pushdown automata. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of formal languages, Vol. 1*, pages 111–174. Springer, Berlin, 1997. https://doi.org/10.1007/978-3-642-59136-5.

[2] Olivier Bodini, Julien David, and Philippe Marchal. Random-bit optimal uniform sampling for rooted planar trees with given sequence of degrees and applications. In *Algorithms and discrete applied mathematics*, volume 9602 of *Lecture Notes in Comput. Sci.*, pages 97–114. Springer, 2016. https://doi.org/10.1007/978-3-319-29221-2_9

[3] Alin Bostan, Frédéric Chyzak, and Assia Mahboubi. A bijection between classes of North/West/South-East walks. In preparation.

[4] Mireille Bousquet-Mélou, Éric Fusy, and Kilian Raschel. Plane bipolar orientations and quadrant walks. 61 pages, arXiv:1905.04256, 2019.

[5] Mireille Bousquet-Mélou and Marni Mishna. Walks with small steps in the quarter plane. In *Algorithmic probability and combinatorics*, volume 520 of *Contemp. Math.*, pages 1–39. Amer. Math. Soc., Providence, RI, 2010. https://doi.org/10.1090/conm/520/10252.

[6] R. Brak, G. K. Iliev, and T. Prellberg. An infinite family of adsorption models and restricted Lukasiewicz paths. *J. Stat. Phys.*, 145(3):669–685, 2011. https://doi.org/10.1007/s10955-011-0306-8.

[7] Frédéric Chyzak. Un algorithme en ligne pour une bijection entre des classes de marches Nord – Ouest – Sud-Est. Slides for a talk at "Journées Combinatoires Franco-Vancouvéroises" (Vancouver), July 2016. https://specfun.inria.fr/chyzak/Talks/jcfv1.pdf.

[8] Sen-Peng Eu. Skew-standard tableaux with three rows. *Adv. in Appl. Math.*, 45(4):463–469, 2010. https://doi.org/10.1016/j.aam.2010.03.004.

[9] Sen-Peng Eu, Tung-Shan Fu, Justin T. Hou, and Te-Wei Hsu. Standard Young tableaux and colored Motzkin paths. *J. Combin. Theory Ser. A*, 120(7):1786–1803, 2013. https://doi.org/10.1016/j.jcta.2013.06.007.

[10] Guy Fayolle, Roudolf Iasnogorodski, and Vadim Malyshev. *Random walks in the quarter-plane*, volume 40 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, 1999. https://doi.org/10.1007/978-3-642-60001-2.

[11] Dominique Gouyou-Beauchamps. Standard young tableaux of height 4 and 5. *European J. Combin.*, 10(1):69–82, 1989. https://doi.org/10.1016/S0195-6698(89)80034-4.

[12] Richard Kenyon, Jason Miller, Scott Sheffield, and David B. Wilson. Bipolar orientations on planar maps and SLE_{12}. *The Annals of Probability*, 47(3):1240–1269, 2019. https://doi.org/10.1214/18-aop1282.

[13] M. Lothaire, editor. *Combinatorics on words*. Cambridge University Press, 1997. https://doi.org/10.1017/cbo9780511566097.

[14] George N. Raney. Functional composition patterns and power series reversion. *Trans. Amer. Math. Soc.*, 94:441–451, 1960. https://doi.org/10.2307/1993433.

[15] Paul C. Rosenbloom. *The Elements of Mathematical Logic*. Dover Publications, Inc., New York, N. Y., 1950.

[16] Jacques Sakarovitch. *Elements of automata theory*. Cambridge University Press, 2009. https://doi.org/10.1017/cbo9781139195218.

[17] Karen Yeats. A bijection between certain quarter plane walks and Motzkin paths. 9 pages, arXiv:1412.1504, 2014.