



HAL
open science

A generalization of the self-dual induction to every interval exchange transformation

Sébastien Ferenczi

► **To cite this version:**

Sébastien Ferenczi. A generalization of the self-dual induction to every interval exchange transformation. *Annales de l'Institut Fourier*, 2014, 64 (5), pp.1947-2002. hal-02120141

HAL Id: hal-02120141

<https://hal.science/hal-02120141>

Submitted on 5 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



ANNALES

DE

L'INSTITUT FOURIER

Sébastien FERENCZI

A generalization of the self-dual induction to every interval exchange transformation

Tome 00, n° 0 (XXXX), p. 1-56.

http://aif.cedram.org/item?id=AIF_XXXX__00_0_0_0

© Association des Annales de l'institut Fourier, XXXX, tous droits réservés.

L'accès aux articles de la revue « Annales de l'institut Fourier » (<http://aif.cedram.org/>), implique l'accord avec les conditions générales d'utilisation (<http://aif.cedram.org/legal/>). Toute reproduction en tout ou partie cet article sous quelque forme que ce soit pour tout usage autre que l'utilisation à fin strictement personnelle du copiste est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

cedram

*Article mis en ligne dans le cadre du
Centre de diffusion des revues académiques de mathématiques
<http://www.cedram.org/>*

A GENERALIZATION OF THE SELF-DUAL INDUCTION TO EVERY INTERVAL EXCHANGE TRANSFORMATION

by Sébastien FERENCZI

ABSTRACT. — We generalize to all interval exchanges the induction algorithm defined by Ferenczi and Zamponi for a particular class. Each interval exchange corresponds to an infinite path in a graph whose vertices are certain unions of trees we call castle forests. We use it to describe those words obtained by coding trajectories and give an explicit representation of the system by Rokhlin towers. As an application, we build the first known example of a weakly mixing interval exchange outside the hyperelliptic and rotations Rauzy classes.

RÉSUMÉ. — Nous généralisons à tous les échanges d'intervalles l'algorithme d'induction défini par Ferenczi et Zamponi pour une classe particulière. Chaque échange d'intervalles correspond à un chemin infini dans un graphe dont les sommets sont certaines unions d'arbres que nous appelons des forêts de châteaux. Nous l'utilisons pour décrire les mots obtenus en codant les trajectoires, et donner une représentation explicite du système par des tours de Rokhlin. Comme application, nous construisons le premier exemple connu d'échange d'intervalles faiblement mélangeant en-dehors de la classe de Rauzy hyper-elliptique et de celle des rotations.

1. Preliminaries

Interval exchanges were originally introduced by Oseledec [33], following an idea of Arnold [2], see also Katok and Stepin [26]; an exchange of k intervals, denoted throughout this paper by \mathcal{I} , is given by a probability vector of k lengths together with a permutation π on k letters; the unit interval is partitioned into k subintervals of lengths $\alpha_1, \dots, \alpha_k$ which are rearranged by \mathcal{I} according to π . It was Rauzy [35] who first defined an algorithm of renormalization for interval exchanges, now called *Rauzy induction*, which

Keywords: Dynamical systems, interval exchanges, symbolic dynamics.

Math. classification: 37B10, 68R15.

generalizes the Euclid algorithm of continued fraction approximation and coincides with it for $k = 2$.

The Rauzy induction, further developed by Veech [38], and modified by Zorich [43] and more recently by Marmi, Moussa and Yoccoz [31], had a tremendous success in solving many problems associated with interval exchanges. These inductions are also a fundamental tool in the study of the space of moduli of Riemann surfaces, and the various strata of its unit tangent bundle, through what has been a basic object of interest for the last 25 years, the *Teichmüller flow* on a stratum. Consider the translation surface obtained by gluing opposite parallel sides of a polygon: to study the Teichmüller flow applied to this given surface, the Rauzy induction chooses an initial segment of an horizontal separatrix and follows its vertical separatrix till it intersects this initial segment, in order to obtain an interval exchange as induced map; then it considers shorter and shorter initial segments. But a basic flaw is that we only consider one horizontal separatrix; the *da Rocha induction* [30] considers all the horizontal separatrices and one vertical separatrix, and its duality with the Rauzy induction appears in the natural extension of the induction process. The trouble with both procedures is that they destroy the symmetry of the geometrical situation, by giving a special role to one of the separatrices; because of that, each foliation admits several descriptions, and the relative position of the separatrices is not taken into account.

In [20] we describe a new induction algorithm for interval exchanges (following a preliminary version for three intervals [17] [18] [21]), which is proved in [11], also [21] for $k = 3$, to be *self-dual* for the duality mentioned above, giving the same role to each horizontal and vertical separatrix (this notion of duality is defined in [21] and [11] see also [24], but it is indeed the same as the one defined by Schweiger in Chapter 21 of the book [36]). At each stage we induce \mathcal{I} on a disjoint union of $k - 1$ intervals, each one containing a discontinuity of \mathcal{I}^{-1} and having its extremities on negative orbits of discontinuities of \mathcal{I} . This ensures that each of our intervals is the cylinder of a so-called *bispecial* word of the trajectories. As in the case of the other inductions, it is described by an infinite path in a finite graph, whose vertices are the states of the induction.

Unfortunately, the algorithm in [20] is defined only when the permutation is the *symmetric* one, $i \rightarrow k + 1 - i$, $1 \leq i \leq k$, and thus the results in [20] can be applied only to interval exchanges in the *hyperelliptic* Rauzy class. At long last, the present paper gives an algorithm (indeed, several

ones) generalizing the self-dual induction for every interval exchange satisfying Keane's i.d.o.c. condition, whatever its (primitive) permutation. The aim of the present paper is first to make this general induction work, and then to focus on the full information it gives on the word-combinatorial and dynamical properties of the system. The geometry lying behind the induction is investigated in [11] for the hyperelliptic case, and is studied in its full generality in [12], where it is proved, among others, that in general the self-duality does not work.

As in the hyperelliptic case the new induction is neither unique nor straightforward to implement; as there is no canonical order between the parameters to be changed, there will be *decisions* to make, as in the problem of induction of a train-track [34], which is not solved in the general case. Thus we propose several algorithms, which create the same induction sub-intervals though at different speeds. For the Rauzy induction and its multiplicative accelerations, the states are defined by permutations; in our case, each state corresponds to a description of *Rokhlin towers* for the induced map, in terms of an abstract object called a *castle forest*; their properties allow us to implement the induction, through one of several possible deterministic algorithms. The fundamental step in the process consists in surgery on trees to get the castle forest in the next state. The sequence of castle forests, seen as a path in the *graph of graphs*, defines inductively the bispecial words of the trajectories; the constructive sequence of Rokhlin towers which it describes gives a complete knowledge of the dynamical system from the measure-theoretic point of view.

The main interest of the new induction lies in its capacity of building examples, which has been already exploited in [19], [20], [23], [15], [10], [7]; these examples were by necessity limited to the hyperelliptic class, except one in Section 5.2 of [20] which indeed anticipated the present paper in a particular case; it happens that, as far as we know, every interval exchange built in the literature in order to get some interesting properties belongs to the hyperelliptic Rauzy class, whatever the way it is built, see for example the discussion in Section 4.3 below; the other classes seem much more difficult to study, partially but not only because, except for the less interesting rotations class, they do not exist for small numbers of intervals. Thus the present induction algorithm, after being used to generalize some results known in the hyperelliptic case, such as a partial result on repetitions of words, and an *S-adic* presentation, with a characterization of those interval exchanges whose trajectories are self-similar and generated by a primitive substitution rule, allows us to break new ground by building explicitly a

family of weakly mixing interval exchanges in the Rauzy class corresponding to the component $\mathcal{H}_{\text{odd}}(2g - 2)$ for every $g \geq 3$, which constitute to our knowledge the first known such examples outside the hyperelliptic and rotations classes.

This research was partially supported by ANR project GeoDyn. The author wishes to thank the referee for suggesting substantial improvements in the presentation.

1.1. Interval exchanges

Throughout the paper, *intervals* are semi-open as $[a, b[$. For any question about interval exchanges, we refer the reader to the surveys [41], [42], [16].

DEFINITION 1.1. — A k -interval exchange \mathcal{I} with probability vector $(\alpha_1, \alpha_2, \dots, \alpha_k)$, and permutation π is defined by

$$\mathcal{I}x = x + \sum_{\pi^{-1}(j) < \pi^{-1}(i)} \alpha_j - \sum_{j < i} \alpha_j.$$

when x is in the interval

$$\Delta_i = \left[\sum_{j < i} \alpha_j, \sum_{j \leq i} \alpha_j \right].$$

We denote by β_i , $1 \leq i \leq k - 1$, the i -th discontinuity of \mathcal{I}^{-1} , namely $\beta_i = \sum_{\pi^{-1}(j) \leq \pi^{-1}(i)} \alpha_j$, while γ_i is the i -th discontinuity of \mathcal{I} , namely $\gamma_i = \sum_{j \leq i} \alpha_j$, we define also $\gamma_0 = 0$, $\gamma_k = 1$. Then Δ_i is the interval $[\gamma_{i-1}, \gamma_i[$ if $1 \leq i \leq k$.

Warning: Roughly half the texts on interval exchanges re-order the subintervals by π^{-1} ; the present definition corresponds to the following ordering of the $\mathcal{I}\Delta_i$: from left to right, $\mathcal{I}\Delta_{\pi(1)}, \dots, \mathcal{I}\Delta_{\pi(k)}$.

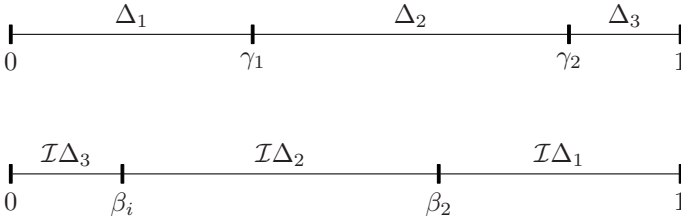


Figure 1.1. A 3-interval exchange with $\pi i = 3 - i$.

DEFINITION 1.2. — \mathcal{I} satisfies the infinite distinct orbit condition or i.d.o.c. of Keane [27] if the $k-1$ negative orbits $\{\mathcal{I}^{-n}\gamma_i\}_{n \geq 0}$, $1 \leq i \leq k-1$, of the discontinuities of \mathcal{I} are infinite disjoint sets.

As is proved in [27], the i.d.o.c. condition implies that \mathcal{I} has no periodic orbit and is *minimal*: every orbit is dense. If π is *primitive*, that is $\pi(\{1, \dots, j\}) \neq \{1, \dots, j\}$ for every $1 \leq j \leq k-1$, the i.d.o.c. condition is (strictly) weaker than the *total irrationality*, where the only rational relation satisfied by α_i , $1 \leq i \leq k$, is $\sum_{i=1}^k \alpha_i = 1$.

DEFINITION 1.3. — For every point x in $[0, 1[$, its trajectory is the infinite sequence $(x_n)_{n \in \mathbb{N}}$ defined by $x_n = i$ if $\mathcal{I}^n x$ falls into Δ_i , $1 \leq i \leq k$.

DEFINITION 1.4. — The induced map of a map T on a set Y is the map $y \rightarrow T^{r(y)}y$ where, for $y \in Y$, $r(y)$ is the smallest $r \geq 1$ such that $T^r y$ is in Y (in all cases considered in this paper, $r(y)$ is finite).

Rauzy classes are equivalence classes of primitive permutations whose full definition (through the Rauzy induction) is not relevant to the present paper, and can be found in [41]. Their link with connected components of strata in the moduli space of abelian differentials is described in [29][44][9]. Among the Rauzy classes of permutations on $\{1, \dots, k\}$, two particular ones are the *hyperelliptic* class which contains the *symmetric* permutation $i \rightarrow k+1-i$, $1 \leq i \leq k$, and the *rotations* class which contains the circular permutation $1 \rightarrow k$, $i \rightarrow i-1$, $2 \leq i \leq k$. These two classes are the same for $k=3$, distinct for $k \geq 4$. The class corresponding to the component $\mathcal{H}_{\text{odd}}(2g-2)$ is defined for $2g = k \geq 6$ and disjoint from the two previous classes.

1.2. Word combinatorics

We look at finite words on a finite alphabet $\mathcal{A} = \{1, \dots, k\}$. A word $w_1 \cdots w_t$ has length $|w| = t$ (not to be confused with the length of a corresponding interval). The *empty word* is the unique word of length 0. The *concatenation* of two words w and w' is denoted by ww' .

DEFINITION 1.5. — A word $w = w_1 \cdots w_t$ occurs at place i in a word $v = v_1 \cdots v_s$ or an infinite sequence $v = v_1 v_2 \cdots$ if $w_1 = v_i, \dots, w_t = v_{i+t-1}$. We say that w is a factor of v . The *empty word* is a factor of any v . Prefixes and suffixes are defined in the usual way.

DEFINITION 1.6. — A language L over \mathcal{A} is a set of words such if w is in L , all its factors are in L , aw is in L for at least one letter a of \mathcal{A} , and wb is in L for at least one letter b of \mathcal{A} .

A language L is minimal if for each w in L there exists n such that w occurs in each word of L with n letters.

The language $L(u)$ of an infinite sequence u is the set of its finite factors.

DEFINITION 1.7. — A language L being fixed, for a word w we call arrival set of w and denote by $A(w)$ the set of all letters x such that xw is in L , and call departure set of w and denote by $D(w)$ the set of all letters x such that wx is in L .

A word w is called right special, resp. left special if $\#D(w) > 1$, resp. $\#A(w) > 1$. If w is both right special and left special, then w is called bispecial.

DEFINITION 1.8. — The symbolic dynamical system associated to a language L is the one-sided shift $S(x_0x_1x_2\cdots) = x_1x_2\cdots$ on the subset X_L of $\mathcal{A}^{\mathbb{N}}$ made with the infinite sequences such that for every $t < s$, $x_t\cdots x_s$ is in L .

For a word $w = w_1\cdots w_s$ in L , the cylinder $[w]$ is the set $\{x \in X_L; x_0 = w_1, \dots, x_{s-1} = w_s\}$.

If the interval exchange \mathcal{I} is minimal, all its trajectories have the same finite factors, whose set is called the language $L(\mathcal{I})$, and this language is minimal; the arrival sets, departure sets, special words, depend on the language and not on the individual trajectories; in the sequel, \mathcal{I} being fixed, all these objects are those defined by any trajectory of \mathcal{I} . If there is no periodic orbit, every word w is a factor of a bispecial word; hence the bispecial words determine the finite factors of the trajectories, and thus the symbolic dynamical system $X_{L(\mathcal{I})}$.

The following result will be used throughout the paper:

LEMMA 1.9. — For any word w in $L(\mathcal{I})$, $[w]$ is an interval. $D(w)$ is an integer interval (meaning a set of the form $\{k, k+1, \dots, k+a\}$) and $A(w)$ is the image by π of an integer interval. The sets $[wd]$, $d \in D(w)$, resp. $\mathcal{I}[\pi(a)w]$, $a \in \pi^{-1}A(w)$, constitute a partition of $[w]$ into subintervals, positioned from left to right according to the increasing order of the d , resp. a .

Proof. — Suppose w is of length t ; then $[wd] = [w] \cap \mathcal{I}^{-t}[\gamma_{d-1}, \gamma_d[$ and d is in $D(w)$ if and only if this is not empty. Thus we get, by induction on t , that $[w]$ is an interval of continuity of \mathcal{I}^t , and the result on $D(w)$ as \mathcal{I} and its iterates are growing on their intervals of continuity. As for $A(w)$, $\mathcal{I}[aw] = [w] \cap \mathcal{I}[\gamma_{a-1}, \gamma_a[= [w] \cap [\beta_{\pi^{-1}a-1}, \beta_{\pi^{-1}a}[$ (with $\beta_{\pi^{-1}a-1}$ replaced by 0 if $a = \pi 1$, $\beta_{\pi^{-1}a}$ replaced by 1 if $a = \pi k$), and a is in $A(w)$ if and only if this is not empty. \square

2. The new induction in the case of alternate discontinuities

DEFINITION 2.1. — Let \mathcal{I} be a k -interval exchange; it has alternate discontinuities if $\beta_i < \gamma_i$ for each $1 \leq i \leq k-1$ and $\gamma_i < \beta_{i+1}$ for each $1 \leq i \leq k-2$.

The combinatorial consequence of alternate discontinuities, coming from the proof of Lemma 1.9, is that in the language, after $\pi 1$ there is always 1; after πj , $j-1$ or j , $2 \leq j \leq k$. Thus $A(w)$ and $D(w)$ have at most two elements, for words of length one hence for every word. The example in Figure 1.1 above is in this case.

DEFINITION 2.2. — For $i \neq \pi 1$, we denote by $i_m < i_p = i_m + 1$ the two letters which can follow it. For $i \neq k$, we denote by i_- and i_+ the two letters which can precede it, denoted so that $\pi^{-1}i_+ = \pi^{-1}i_- + 1$.

2.1. Special intervals

Our aim is to build the bispecial words of the language $L(\mathcal{I})$; to each of these corresponds an interval $[w]$. However, we may have $w \neq w'$ but $[w] = [w']$:

PROPOSITION 2.3. — w is a left special word of length t if and only if $[w]$ is an interval of continuity of \mathcal{I}^t containing exactly one point β_i in its interior; w is a right special word of length t if and only if $[w]$ is an interval of continuity of \mathcal{I}^t containing exactly one point $\mathcal{I}^{-t}\gamma_j$ in its interior; if w is left special, $[w] = [w']$ for one bispecial word $[w']$, which is the longest word v such that $[v] = [w]$.

Proof. — w is left special if and only if $[w]$ intersects two intervals $\mathcal{I}\Delta_j$ and $\mathcal{I}\Delta_{j'}$, thus if and only if $[w]$ contains a β_i in its interior. This β_i is unique because of the alternate discontinuities. Similarly, $w = w_1 \cdots w_t$ is right special if and only if $\mathcal{I}^t[w]$ intersects two intervals Δ_j and $\Delta_{j'}$, thus if and only if $\mathcal{I}^t[w]$ has a γ_i in its interior, and only one because of the alternate discontinuities. If w is not right special, then $[w] = [w']$ for some w' of length $t+1$; if there is no right special w' such that $[w] = [w']$, then $[w] = [w^{(n)}]$ for an infinite sequence of non right special words of length n , thus $[w]$ does not contain any $\mathcal{I}^{-s}\gamma_h$ in its interior, which contradicts minimality. As w' is right special, no $[w'a]$ can be the full interval E , thus w' is unique and is indeed the longest possible v . \square

We shall first focus on building those intervals $[w]$ corresponding to left (or bi-) special words, leaving the building of the words themselves for a second stage.

DEFINITION 2.4. — *An interval E is a special interval if $E = [w]$ for at least one left special word, or equivalently for one bispecial word.*

For a special interval E containing one point β_i in its interior, we define

- $\gamma(E)$ is the first element $\mathcal{I}^{-m}\gamma_j$, $m > 0$, $1 \leq j \leq k-1$, which falls in the interior of E , where $\mathcal{I}^{-m'}\gamma_{j'}$ is after $\mathcal{I}^{-m}\gamma_j$ if $m' > m$,
- $E_- = E \cap [0, \beta_i[$, $E_+ = E \cap [\beta_i, 1[$,
- $E_m = E \cap [0, \gamma(E)[$, $E_p = E \cap [\gamma(E), 1[$,

A family of special intervals is a family of $k-1$ disjoint special intervals E_i , $1 \leq i \leq k-1$, such that β_i is in the interior of E_i for each i .

$\gamma(E)$ exists by minimality; it is unique because of the alternate discontinuities, as E cannot contain more than one $\mathcal{I}^{-s}\gamma_j$, $1 \leq j \leq k-1$. for a given s ; it is different from β_i by the i.d.o.c. condition.

COROLLARY 2.5. — *Let E be a special interval, containing β_i in its interior. Then the maximal special strict subinterval of E is the one of the two subintervals E_m and E_p which contains β_i in its interior.*

Proof. — This is a consequence of the characterization in Proposition 2.3. □

At the beginning, our definition of alternate discontinuities ensures that the words of length one $1, 2, \dots, k-1$ are left special, thus we have a family of special intervals $E_{i,0} = \Delta_i$, $1 \leq i \leq k-1$. Corollary 2.5 gives us the recipe to build further special intervals: we need to build the points where the negative orbits of the discontinuities of \mathcal{I} approximate the discontinuities of \mathcal{I}^{-1} . As there are $k-1$ discontinuities β_i , we shall build a nested sequence of $k-1$ intervals, which, as will be proved in Proposition 2.25, converge to the β_i , namely

$$E_{i,n} = [\beta_i - l_{i,n}, \beta_i + r_{i,n}] \quad 1 \leq i \leq k-1,$$

so that the $E_{i,n}$ are the intervals containing β_i , and whose endpoints are the successive $\mathcal{I}^{-m}\gamma_j$ which fall closest to β_i . *The number n of the stage will be omitted whenever it is not necessary*: when we go from one stage to the next, $E_{i,n}$ will be E_i and $E_{i,n+1}$ will be E'_i , or else “the new E_i ”.

l_i and r_i are called the *half-lengths* of the interval E_i and the induction will operate on these parameters.

By Corollary 2.5, the smallest possible $E'_i \neq E_i$ is the one of the two intervals $E_{i,m}$ and $E_{i,p}$ which contains β_i . It will often prove necessary

to wait before cutting E_i ; indeed, at any given stage we shall put $E'_i = E_i$ for all i outside a set defined in Definition 2.13 below and called a *decision*. The main problem is indeed to define a sequence of decisions which makes the induction work and is non-trivial: it is solved in [20] for some permutations, and will be solved for all permutations in the remainder of Section 2. There are many possible decisions, which define algorithms described in Section 2.7; each algorithm is one of the possible ways to build the sequence of intervals we want. These algorithms are indeed *induction algorithms* because the intervals E'_i are built from the E_i by using S , the induced map of \mathcal{I} on the set $\cup_{i=1}^{k-1} E_i$. Indeed, all the information about \mathcal{I} is given by the S at all stages; these maps S defined on disjoint unions of intervals are the effective object of our study, and we can completely forget the interval exchange \mathcal{I} , which could always be retrieved from the initial S when needed. Note that S' , the induced map of \mathcal{I} on $\cup_{i=1}^{k-1} E'_i$, is also the induced map of S on $\cup_{i=1}^{k-1} E'_i$.

2.2. Induction castles and castle forests

A fundamental tool for studying any induced map is the building of Rokhlin towers.

DEFINITION 2.6. — *In a measure-theoretic system (X, T, μ) , a Rokhlin tower of basis Y is a collection of disjoint measurable sets called levels $Y, TY, \dots, T^{h-1}Y$.*

Our map S is unusual as \mathcal{I} is induced on a disjoint union of intervals; the appropriate tool for describing its structure is the induction castle introduced in [23]: following [8], we say that a union of towers is a *castle* (the Ornstein school used the words *stacks* and *gadgets* instead of towers and castles).

DEFINITION 2.7. — *Given a k -interval exchange \mathcal{I} , and a family of special intervals E_i , let S be the induced map of \mathcal{I} on $E_1 \cup \dots \cup E_{k-1}$. The induction castle of the E_i is the unique partition of $[0, 1[$ into levels $T^r Y_{i,t}$, $1 \leq i \leq k-1$, $1 \leq t \leq e_i$, $0 \leq r \leq h_{i,t} - 1$, where*

- *each interval E_i is partitioned into e_i subintervals $Y_{i,t}$, $1 \leq t \leq e_i$, numbered by t from left to right,*
- *$S Y_{i,t}$ is a subinterval of $E_{g_{i,t}}$, and on $Y_{i,t}$ $S = \mathcal{I}^{h_{i,t}}$.*

An induction castle is indeed a union of Rokhlin towers, each tower being made with the levels $T^r Y_{i,t}$, $0 \leq r \leq h_{i,t} - 1$, thus the bases of the towers partition the intervals E_i . The e_i are finite by compactness, but in general

each of the $k - 1$ intervals could be partitioned in many subintervals; only for interval exchange transformations and the type of intervals chosen shall we be able to bound these numbers. Note that, by definition, if some of our intervals E_i are adjacent, the top of the towers have to be split, somewhat artificially, by the pre-image of the border points, as in the right tower of Figure 2.1 below.

For our families of special intervals, the induction castles have the following properties:

LEMMA 2.8. — *The $Y_{i,t}$, $1 \leq t \leq e_i$, correspond to a partition of E_i by points of the form $\mathcal{I}^{-m}\gamma_j$, $m \geq 0$, which can be ordered by increasing values of m . $SY_{i,t}$ is either $E_{g_{i,t,-}}$ or $E_{g_{i,t,+}}$ (see Definition 2.4).*

Proof. — The towers are made by continuity intervals up to time s , which at time $s + 1$ are split because they meet either a discontinuity or the pre-image of an endpoint of an E_i ; thus on the top of a tower there is an $\mathcal{I}^{-m}\gamma_j$, thus the endpoints of the $Y_{i,t}$ must also be such points; and the condition of alternate discontinuities allows to order those which are inside the same E_i . And the partition of any E_g by the different $SY_{i,t}$ which fall into it must be by points β_j , while there is only one such point inside E_g . \square

Example 1. — We consider a three-interval exchange with the symmetric permutation $1 \rightarrow 3, 2 \rightarrow 2, 3 \rightarrow 1$, and alternate discontinuities, see Figure 1.1 above; the special intervals in the initial stage are $E_1 = [0, \gamma_1[= [1]$, $E_2 = [\gamma_1, \gamma_2[= [2]$. The induced map S determines the induction castle in Figure 2.1 (where we have added the $SY_{i,t}$ as dotted lines above the towers). The map S is then defined as the map sending each subinterval in the bottom to the corresponding dotted subinterval in the top, each move from one level to the next being made by \mathcal{I} . A further induction castle will be shown in Figure 2.10 below.

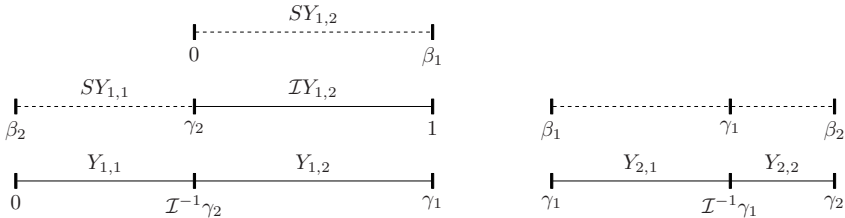


Figure 2.1. Induction castle 1.0.

We shall now describe the induction castles through combinatorial objects, which constitute a variant of the *castle graphs* used in [23].

DEFINITION 2.9. — *A tree is a connected graph without loops. All trees we consider, except the trees of relations of Section 2.7, are directed. A leaf is a vertex with no outgoing edge, a root is a vertex with no incoming edge, a node is a vertex with at least two outgoing edges. Our directed trees are pictured in the plane with edges going upward; thus the natural notion of before, resp. after, is equivalent to below, resp. above. This embedding defines also natural notions of left and right on edges out of a common node; then we say that a leaf is left of another leaf if it is after the left edge issuing from the last common node of the paths from the root to the two leaves. An edge is single if it does not start from a node. The roots, edges and leaves may be labeled, the label of a path is the concatenation of the labels of its successive edges.*

DEFINITION 2.10. — *A castle forest G is the disjoint union of $k - 1$ directed trees, numbered from 1 to $k - 1$, with the following properties:*

- *all vertices are roots, nodes or leaves, a root may also be a node; each tree has one root,*
- *there are $2k - 2$ leaves labeled by l_j and r_j , $1 \leq j \leq k - 1$,*
- *a node has two issuing edges, denoted as the left and right one,*
- *a tree whose root is not a node has one single edge,*
- *in a tree with nodes, a leaf l_j , resp r_j , is at the end of a right, resp. left, edge,*
- *there is no strict subset J of $\{1, \dots, k - 1\}$ such that the trees i , $i \in J$, have only leaves l_j or r_j , $j \in J$.*

The parenthesized train-track equalities of G are the equalities expressing that, for $1 \leq i \leq k - 1$, $l_i + r_i$ is the sum of the labels of the leaves of the tree i , with parentheses around the sum of the leaves which are after each node except the root.

For a castle forest G , the subset Ω_G of \mathbb{R}^{2k-2} consists of the vectors of $0 < l_i < 1$, $0 < r_i < 1$, $1 \leq i \leq k - 1$, satisfying the train-track equalities of G . An allowed castle forest G is a castle forest such that Ω_G is nonempty.

The parenthesized train-track equalities give a complete description of the castle forest; its actual description as a forest will become useful when we put labels on the edges and roots, see Section 2.4 below. Note that the labels l_i and r_i of the leaves are formal parameters, which shall later be associated to half-lengths of intervals.

DEFINITION 2.11. — *The castle forest of a family of special intervals is built from their induction castle in the following way:*

- whenever $e_i = 1$, the tree i has one single edge,
- each node in the tree i corresponds to one of the points $\mathcal{I}^{-m}\gamma_j$ partitioning E_i (excluding the endpoints of E_i),
- if $e_i > 1$, the root of the tree i corresponds to the first (in the increasing order of m) of the points $\mathcal{I}^{-m}\gamma_j$ partitioning E_i ,
- from each node $\mathcal{I}^{-m_0}\gamma_{j_0}$ in the tree i , there is a left edge, going to the first (in the increasing order of m) node for which the point $\mathcal{I}^{-m}\gamma_j$ lies (strictly) left of the point $\mathcal{I}^{-m_0}\gamma_{j_0}$, or to a leaf if there is no such node,
- from each node $\mathcal{I}^{-m_0}\gamma_{j_0}$ in the tree i , there is a right edge, going to the first (in the increasing order of m) node for which the point $\mathcal{I}^{-m}\gamma_j$ lies (strictly) right of the point $\mathcal{I}^{-m_0}\gamma_{j_0}$, or to a leaf if there is no such node,
- the t -th leaf from the left in the tree i receives the label l_g , resp. r_g , if $SY_{i,t}$ is $E_{g,-}$, resp. $E_{g,+}$.

Thus the induction castle over each subinterval has as many towers as leaves in the corresponding tree. The castle forest for Example 1 will be built, together with its labels, in Section 2.4, and is in Figure 2.6 below, but let us consider a more convoluted example, though it was not actually built from an interval exchange.

Example 2. — If the part of the induction castle lying over E_1 is as in Figure 2.2, then the castle tree 1 is shown in Figure 2.3, where for each node we have mentioned to which point it corresponds. The corresponding parenthesized train-track equality is $l_1 + r_1 = r_7 + ((r_8 + l_9) + l_2)$.

More examples will be found in Section 2.4 below.

PROPOSITION 2.12. — *The castle forest of a family of $k - 1$ special intervals is an allowed castle forest, and its train-track equalities are satisfied by the actual half-lengths of the E_i .*

Proof. — We have to prove all the properties in Definition 2.10. Some of them are included in Definition 2.11 taking into account Lemma 2.8. We look at the remaining ones.

The points corresponding to nodes determine the partition of E_i into subintervals $Y_{i,t}$, which are associated with leaves of the tree i : they are intervals of continuity of S , sent by S onto some $E_{j,-}$, or $E_{j,+}$.

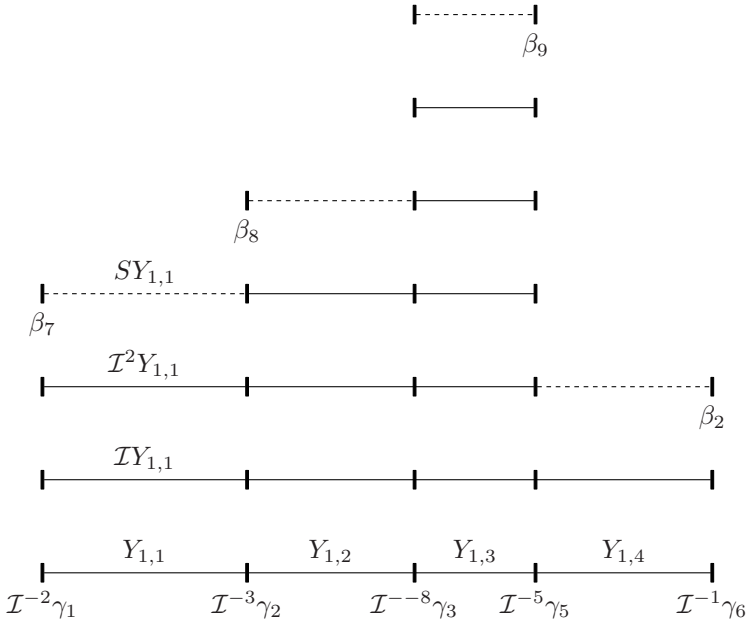


Figure 2.2. Induction castle 2.0 over E_1 .

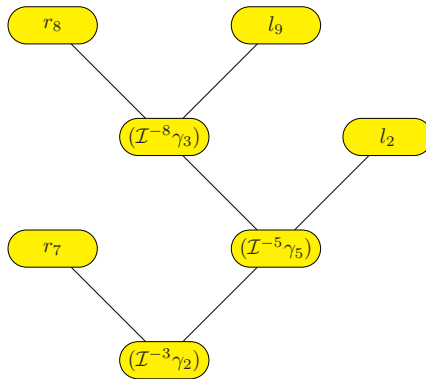


Figure 2.3. Tree 1 in castle forest 2.0.

The forest is allowed as the vector of half-lengths l_i and r_i is in Ω_G : these satisfy the train-track equalities, as we consider the partitions of E_i into intervals of continuity of S^{-1} and S .

By definition of the castle forest, each l_i or r_i is the label of only one leaf. Thus if we make the sum of the $k - 1$ train-track equalities, every l_i and r_i , which appear on the left and are nonzero (because β_i is in the interior of E_i), must appear on the right. Thus there are $2k - 2$ leaves.

The last property in the definition of castle forests is satisfied by minimality.

As for the last but one property, suppose a leaf l_j is the end of a left edge; then the endpoints of the associated subinterval are $\mathcal{I}^{-s_1}\gamma_{t_1}$ to the left, $\mathcal{I}^{-s_2}\gamma_{t_2}$ to the right, with $s_2 > s_1$, while this interval is sent by some \mathcal{I}^t onto an interval with endpoints $\mathcal{I}^{-s_3}\gamma_{t_3}$ to the left, β_j to the right, and this contradicts the i.d.o.c. condition. A similar reasoning works in the opposite case. \square

2.3. Induction step, choices and decisions

At the initial stage, as we have seen $E_{i,0} = \Delta_i$, $1 \leq i \leq k - 1$. We can now define the initial castle forest; there are two situations, for which examples can be found in castle forests 1.0 and 3.0 below:

- if $\pi^{-1}k = 1$: in tree πi , $2 \leq i \leq k - 1$, from the root there is a left edge to a leaf r_{i-1} , and a right edge to a leaf l_i ; in tree 1, from the root there is a left edge to a leaf r_{k-1} , and a right edge to a leaf l_1 ;
- if $\pi^{-1}k = j \neq 1$, in the initial castle forest: in tree πi , $2 \leq i \leq k - 1$, $i \neq j$, from the root there is a left edge to a leaf r_{i-1} , and a right edge to a leaf l_i ; in tree πk , from the root there is a left edge to a leaf r_{k-1} , and a right edge to a node, from which there is a left edge to a leaf r_{j-1} , and a right edge to a leaf l_j ; in tree $\pi 1$, there is a single edge to a leaf l_1 .

We suppose that at some stage we have a family of special intervals E_i , with half-lengths l_i , r_i , and its castle forest.

DEFINITION 2.13. — *At any given stage, the greedy decision is the set H of i such that the root of tree i is a node. An allowed decision is any nonempty subset of H . If i is in F , the new special interval E'_i is the one of the intervals $E_{i,m}$ and $E_{i,p}$ which contains β_i ; if i is not in F , $E'_i = E_i$.*

The greedy decision is the maximal set of i such that E_i can be cut at that stage; it corresponds to the i such that some $\mathcal{I}^{-m}\gamma_j$ appears inside E_i in the induction castle of the E_i at that stage. For others i , of course some $\mathcal{I}^{-m}\gamma_j$ exists inside E_i by minimality, but there is not enough information in the picture to determine it, and for cutting E_i we have to wait until it is synchronized again with the others.

DEFINITION 2.14. — For a castle forest, let H be the greedy decision. A choice is an application c of H into $\{-, +\}$. A choice c and an allowed decision F define a linear map C_F from \mathbb{R}^{2k-2} to itself by $C_F(l_i, r_i, 1 \leq i \leq k-1) = (l'_i, r'_i, 1 \leq i \leq k-1)$ with

- if $i \in F$ and $c(i) = -$, $l'_i = l_i$, $r'_i = r_i - U_{i,2}$,
- if $i \in F$ and $c(i) = +$, $l'_i = l_i - U_{i,1}$, $r'_i = r_i$,
- if $i \notin F$, $l'_i = l_i$, $r'_i = r_i$,

where $U_{i,1}$, resp. $U_{i,2}$, is the sum of the labels of the leaves which are after the left (resp. right) edge out of the root of tree i .

PROPOSITION 2.15. — For a family of special intervals, for any i in the greedy decision H , we define the choice $c(i)$ to be $+$ if $l_i - U_{i,1} = U_{i,2} - r_i$ is positive, $-$ otherwise, for $U_{i,j}$ defined as above. Then β_i is in $E_{i,p}$, resp. $E_{i,m}$, if and only if $c(i) = +$, resp. $-$. For any allowed decision F , the half-lengths of the corresponding E'_i are given by $(l'_i, r'_i, 1 \leq i \leq k-1) = C_F(l_i, r_i, 1 \leq i \leq k-1)$.

Proof. — From the definition of the castle forest, the first $\mathcal{I}^{-s}\gamma_t$ which falls into E_i is the one which creates a decomposition of E_i according to the first node, if there is one. Thus for i in H , this node is the root, the length of $E_{i,m}$ is $U_{i,1}$ and the length of $E_{i,p}$ is $U_{i,2}$, while by definition the length of $E_{i,-}$ is l_i and the length of $E_{i,+}$ is r_i , hence the result; note that $l_i - U_{i,1} = U_{i,2} - r_i$ is not zero by the i.d.o.c. condition. \square

COROLLARY 2.16. — The choice c defined above is allowed, which means that $C_H(\Omega_G)$ has a nonempty intersection with the positive open cone of \mathbb{R}^{2k-2} .

Proof. — This comes from Proposition 2.15, using the i.d.o.c. condition. \square

PROPOSITION 2.17. — The castle forest of the E'_i built as in Proposition 2.15 with an allowed decision F is built from the castle forest of the E_i by making for all i in F , successively in any order, the surgery on trees detailed below if $c(i) = -$, the surgery deduced from the one detailed below by exchanging left and right, l and r , if $c(i) = +$.

Let n be the node or leaf at the end of the left edge out of the root of tree i ;

- if n is a node, the part of the former tree i beyond n becomes the new tree i , including n which becomes its root; the part of the tree i lying below n or right of the root is cut away, and put on top of leaf r_i , which is in a tree h (if $h = i$, the new tree i). The vertex n is duplicated: in

the new tree h , the former node n (in the added part) becomes a leaf labeled r_i , and the former leaf r_i becomes a node;

- if n is a leaf labeled r_j , a new tree i is made with a single edge and a leaf labeled r_j , while the whole former tree i is put on top of leaf r_i in its tree h (if $h = i$, the new tree i). In the new tree h the former leaf r_i becomes a node, and the former leaf r_j (in the added part) is re-labeled r_i .

In both cases, if the former leaf r_j was the only leaf in tree h , then in the new tree h the single edge leading to it is deleted, and the new node becomes the root.

Proof. — When i is in F and $c(i) = -$, E_i is cut to become $E'_i = E_{i,m}$, the part of the induction castle which lies above $E_{i,p}$ is cut and pasted on top of the tower such that $SY_{h,t} = E_{i,+}$, and the surgery on trees just mimics the effects of this cut and paste operation, translated by Definition 2.11. Note that, as $c(i) = -$, $E_{i,+}$ is strictly longer than $E_{i,p}$, thus, as S is measure-preserving, the leaf r_i cannot be in the part of the former tree i which has just been cut away. \square

From this construction, we see that single edges may disappear or be created, but non-single edges are conserved by the induction step, though they may be displaced. Thus their numbers is always the same as in the initial state, namely $2k - 2$.

Another way to build the new castle forest, possibly more “algorithmic” or efficient, would be to work only on parenthesized train-track equalities: by building the new r'_i and l'_i with Proposition 2.15, then expressing the r_i and l_i in function of the r'_i and l'_i and inputting these values into the old train-track equalities, we get all the new ones simultaneously.

Evolution of Example 2. — We look at the example in Figures 2.2–2.3; then 1 is in the greedy decision. Suppose first that β_1 is in $Y_{1,t}$, for $t = 2, 3$ or 4. Then $c(1) = +$, and take any decision containing 1; we are in the first case above, with left and right exchanged; suppose the leaf l_1 is in tree 4, and is not its single leaf; then Figure 2.4 shows the new tree 1 and what is added to tree 4, with the former leaf l_1 shown between parentheses.

Suppose now β_1 is in $Y_{1,1}$, and take any decision containing 1. Then $c(1) = -$, we are in the second case above; suppose the leaf r_1 is in tree 5, and is its single leaf; then Figure 2.5 shows the new trees 1 and 5, with the former leaf r_1 shown between parentheses.

Again, we refer the reader to Section 2.4 for more examples.

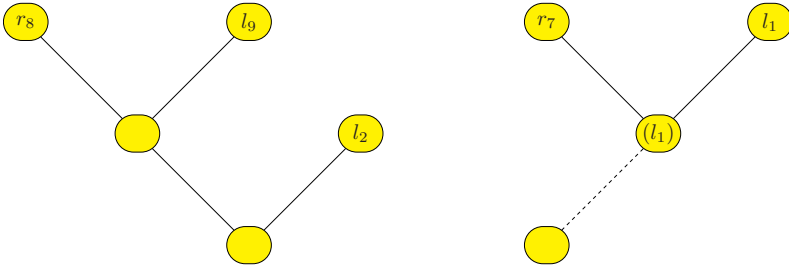


Figure 2.4. Trees 1 and 4 in castle forest 2.1.a.

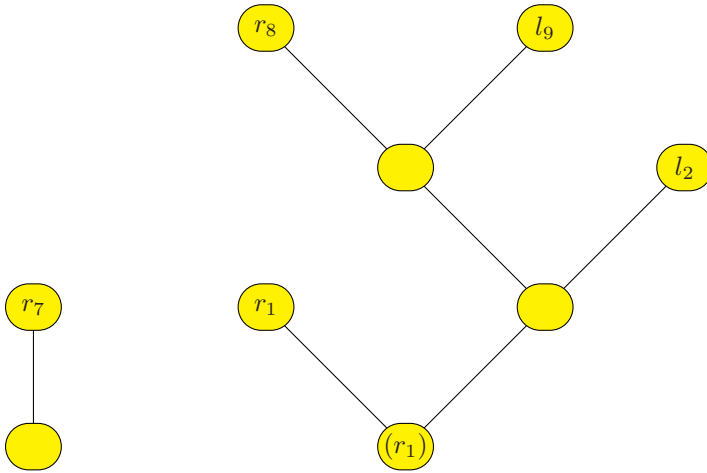


Figure 2.5. Trees 1 and 5 in castle forest 2.1.b.

2.4. Induction families and bispecial words

We shall now build the actual bispecial words.

PROPOSITION 2.18. — For every t and every $1 \leq i \leq k - 1$ there is one left special word w of length t beginning with i , which is a prefix of \mathcal{O}_i , the positive trajectory of β_i , $1 \leq i \leq k - 1$. Then the shortest bispecial word containing w as a prefix is the longest of all the possible words v such that $[v] = [w]$.

Proof. — This comes from the proof of Proposition 2.3. □

By Proposition 2.3, each E_i we have built defines uniquely a bispecial word w_i such that $E_i = [w_i]$; unfortunately, when we build the family

$E_{i,n}$, we are *not* able to retrieve immediately the corresponding family of bispecial words. This is due to the usual synchronization problem: if one $E_{i,n}$ is much smaller than another $E_{j,n}$, the corresponding $w_{j,n}$ will be a factor of $w_{i,n}$, and this is an obstacle in building the next $w_{i,n+1}$. The non-trivial algorithm of [20] managed to produce at each stage a family of words w_i which were bispecial and not factors of one another. Here, for a general permutation, we achieve a somewhat weaker result: at each stage, we are able to produce some $w_{i,n}$ which are (at least) left special, and achieve some degree of synchronization.

DEFINITION 2.19. — *An induction family W is a family of $k - 1$ words in $L(\mathcal{I})$, w_i with first letter i , $1 \leq i \leq k - 1$, such that*

- *if w_i is a factor of w_j for $i \neq j$, w_i is a strict suffix of w_j ,*
- *the w_i are left special,*
- *the w_i which do not have any w_j as a strict factor are right special,*
- *there are no right special factors strictly sandwiched between a w_i and a w_j .*

To see what it means, look at the initial state when $\pi 1 = h \neq k$, see castle forest 3.0 below for a picture. Then, because of the alternate discontinuities, the words $1, 2, \dots, k - 1$ are left special, and all of them except h are right special; the only word extending h to the left is $h1$. Thus $w_i = i$ do not constitute an induction family, but if we put $w_h = h1$ instead of $w_h = h$, we do get an induction family: at this stage we don't know if w_h is right special, but as it contains w_1 as a suffix, it is allowed (though not obliged) to be only left special. Note that now no way of extending w_h or any w_i while keeping the same E_i is apparent from our knowledge.

Getting back to our family of $E_{i,n}$ built by induction, the corresponding bispecial $w_{i,n}$ will be built inductively by using *return words*.

DEFINITION 2.20. — *Given a family of q different words $W = \{w_1, \dots, w_q\}$, a (suffix) return word of W is any (possibly empty) word v such that*

- *either $w_i v = v' w_j$, for some i and j and a nonempty word v' , and $w_i v$ contains no words of W except as factors of its prefix w_i or its suffix w_j ,*
- *or $w_i v$ is a strict suffix of w_j , and $w_i v$ contains no words of W except as factors of its prefix w_i .*

Note that the relation $w_i v = v' w_j$ defines a suffix return word v , but also a *prefix return word* v' . For an induction family W , the second item of Definition 2.20 applies to words w_i which are strict suffixes of w_j , thus,

when applicable, produces only the empty word as a return word v , and indeed if we delete from W the words w_j which contain words w_i as strict suffixes, the return words of the smaller family are deduced from those of W by deleting the empty one (if it is there).

What we shall do now recursively is to build induction families $W_n = \{w_{1,n}, \dots, w_{k-1,n}\}$ such that $E_{i,n} = [w_{i,n}]$. For that, we shall label the castle forests. Namely, the edges of the castle trees are labeled with words which will give the return words of W after the labels of edges are concatenated into labels of paths. Also, for convenience we shall label the root of tree i , in the castle forest of the E_i , with the word w_i .

PROPOSITION 2.21. — *While we build inductively family of special intervals E_i , we can build by the same process induction families W , and labels on the edges of the castle forest of the E_i such that*

- $E_i = [w_i]$,
- the tree i has one single edge whenever w_i contains some w_j in W as a strict suffix, and then this edge has an empty label; the only edges with empty labels are single edges;
- if a node in tree i has a path from the root with label v , the last letter of v being z , then in the induction castle it corresponds to the point $\mathcal{I}^{-s}\gamma_t$ in E_i , where s is the length of w_iv , and γ_t is the point separating Δ_{z_m} and Δ_{z_p} (see Definition 2.2); then w_iv is right special, the left, resp. right edge from the node has a label beginning with z_m , resp. z_p ,
- v is the (possibly empty) label of a path from the root of tree i to a leaf l_j , resp r_j , if and only if v is a return word of W where $w_iv = v'w_j$, v' ends with the letter j_- , resp j_+ , and w_j is the longest word w_h for which $w_iv = v''w_h$ for a nonempty v'' .

The bispecial words themselves do not appear in general in the labels of paths, see castle forest 1.3 below, the canonical way to get them is to build them inductively using the return words as in Proposition 2.22 below. Note also that the labels of paths are not exactly the codings of the successive levels hit by points along the towers, see Section 2.5 below.

Example 1 again. — This is the very case which is described in full details in Section 2.3 of [20]: in the words of the language $L(\mathcal{I})$, after 3 there is always 1; after 2, 1 or 2; after 1, 2 or 3; thus 1 and 2 are the bispecial words at the initial stage. This defines the labeled castle forest in Figure 2.6, with train-track equalities $l_1 + r_1 = r_2 + l_1$, $l_2 + r_2 = r_1 + l_2$.

We check that indeed the labels on the edges are the (suffix) return words of the family of words $\{1, 2\}$: from 1 these are 2, leading to 2, and 31 leading

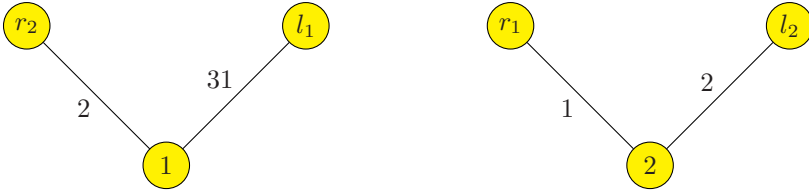


Figure 2.6. Labeled castle forest 1.0.

to 1, which correspond respectively to $E_{1,m}$ being sent by S onto $E_{2,+}$, $E_{1,p}$ being sent by S onto $E_{1,-}$.

We proceed now to do the building in the general case. In the initial state, the left special words of length 1 are all the letters except k ;

- if $\pi^{-1}k = 1$: the initial induction family is $W_0 = \{1, \dots, k-1\}$; in tree πi , $2 \leq i \leq k-1$, the left edge is labeled $(\pi i)_m = i-1$, and the right edge, is labeled $(\pi i)_p = i$; in tree 1, the left edge is labeled $k-1$, and the right edge is labeled $k1$;
- if $\pi^{-1}k = j \neq 1$, $\pi 1 = h \neq k$ the initial induction family W_0 is made with $w_i = i$, $1 \leq i \leq k-1$, $i \neq h$, $w_h = h1$; in tree πi , $2 \leq i \leq k-1$, $\pi i \neq j$, the left edge is labeled $i-1$ if $i-1 \neq h$, $h1$ if $i-1 = h$, and the right is labeled i if $i \neq h$, $h1$ if $i = h$; in tree πk , the left edge is labeled $k-1$, and the right edge is labeled k ; from the node following that edge, the left edge is labeled $j-1$ if $j-1 \neq h$, $h1$ if $j-1 = h$, and the right edge is labeled j if $j \neq h$, $h1$ if $j = h$; in tree h , the single edge has an empty label.

Example 3. — We consider now a case with which the algorithms of [20], [21] cannot deal: a three-interval exchange with permutation $1 \rightarrow 2$, $2 \rightarrow 3$, $3 \rightarrow 1$ and alternate discontinuities. In the language, after 2 there is always 1; after 3, 1 or 2; after 1, 2 or 3; thus 1 and 2 are left special words. At the initial stage $E_1 = [0, \alpha_1[= [1]$, $E_2 = [\alpha_1, \alpha_1 + \alpha_2[= [2]$. The castle forest is in Figure 2.7.

The train-track equalities are $l_1 + r_1 = r_2 + (r_1 + l_2)$, $l_2 + r_2 = l_1$.

We turn back to the general case. We check that the initial induction families and labels do satisfy the properties of Proposition 2.21. We make the recursion hypothesis that this is still true at some stage, for a family of special intervals E_i and an induction family W .

PROPOSITION 2.22. — *Let F be an allowed decision. We modify the w_i and the labels of the edges of the castle forest of the E_i by making, for all*

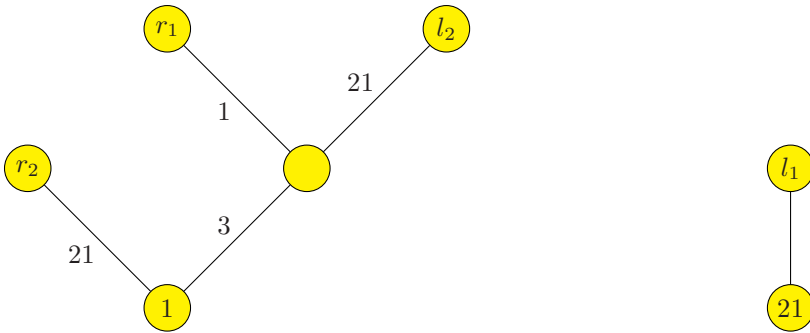


Figure 2.7. Labeled castle forest 3.0.

i in F , successively in any order, the changes detailed below if $c(i) = -$, the changes deduced from those detailed below by exchanging l and r , left edge and right edge, if $c(i) = +$:

- w_i , and every w_j which contains w_i , as a strict suffix, are extended to the right by the label v of the left edge from the root of tree i ,
- the labels of the edge leading to the leaf l_i , and of every edge leading to a leaf l_j or r_j where w_j contains w_i a strict suffix, are extended to the right by v ,
- when an edge is displaced by Proposition 2.17, it keeps the same label, possibly extended as in the previous item.

Then the new family W is an induction family corresponding to the family E'_i built in Proposition 2.17; together with the new edge labels, they satisfy the properties of Proposition 2.21. Moreover, there is no bispecial word strictly sandwiched between the old and new w_i for $1 \leq i \leq k - 1$.

Proof. — Suppose i is in F and $c(i) = -$. Then $w_i v$ is left special, where v is the label of the edge from the root w_i to the first node strictly on its left, or to the leaf on the left if there is no such node, but also, if v' is the label of the path arriving to the leaf l_i from a root w_j , then $w_j v'$ is not right special, and can only be extended to the right by v . The other properties follow from the definitions. □

Thus we get the recursion hypothesis at the next stage, which achieves the proof of Proposition 2.21.

Evolution of Example 1. — The greedy decision is $\{1, 2\}$; we suppose $l_1 > r_1$, $l_2 < r_2$, thus $c(1) = +$, $c(2) = -$. We take $F = \{1, 2\}$, though in the same situation in [20] [21], we would have taken $F = \{1\}$.

We follow Proposition 2.17: we make first the surgery determined by $2 \in F$, being in the second case of the proposition. We get the intermediate castle forest in Figure 2.8, which is also the castle forest we would get by the decision $\{2\}$: in the new induction family $w_1 = 1$, $w_2 = 21$, and the labels of the edges are built as in Proposition 2.22.

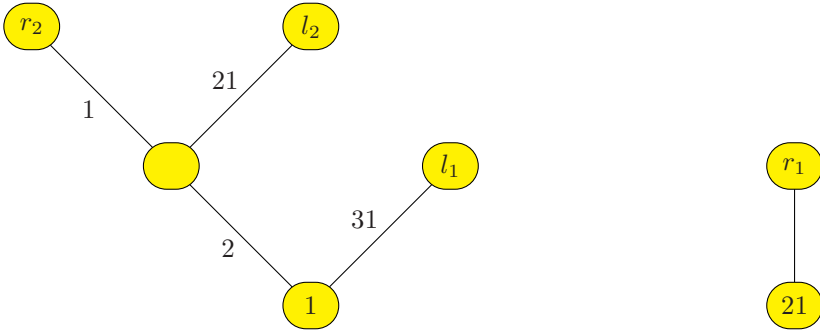


Figure 2.8. Labeled castle forest 1.1.

Then we make the surgery determined by $1 \in F$, thus w_1 is extended to 131, and also w_2 to 2131 as it contained w_1 .

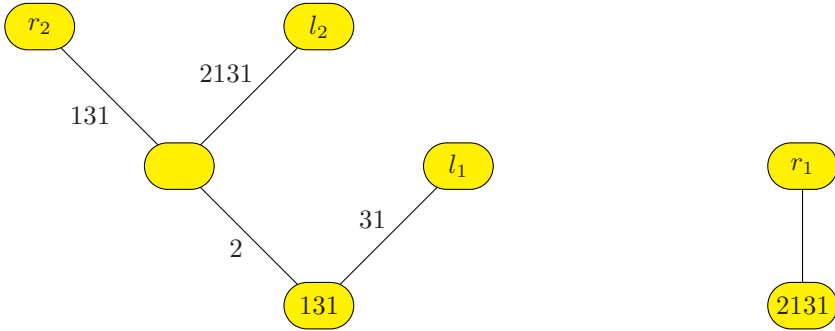


Figure 2.9. Labeled castle forest 1.2.

The new castle forest is again in the second case of Proposition 2.17, with left and right exchanged, but with $h = i = 1$: the tree 1 is first reduced to a single edge leading to a leaf l_1 , then the whole former tree 1 is put on top of this leaf, and the single edge deleted, thus we get exactly the same tree

1, but with new labels. The train-track equalities for both castle forests 1.1 and 1.2 are $l_1 + r_1 = (r_2 + l_2) + l_1$, $l_2 + r_2 = r_1$.

The return words of the family $W = \{131, 2131\}$ are the empty word (because 131 is a suffix of 2131), 2131, 22131 and 31; note that the nonempty ones form the return words of the family $W = \{131\}$. The edge from the root of tree 2 to the leaf r_1 still indicates that $SE_2 = E_{1,+}$ but its label is the empty word. We see that, while 131 is a bispecial word, 2131 is left special but at this stage we do not know whether it is right special or not. This example illustrates the last part of the last condition in Proposition 2.21: the paths labeled 2131 and 22131 from the root w_1 lead to a leaf corresponding to a subinterval of $[w_2]$, and not $[w_1]$.

Note that if we had made first the surgery determined by 1 and then by 2, we would have arrived to the same castle forest, with an intermediate one equal to castle forest 1.0, with different labels and $w_1 = 131$, $w_2 = 2$.

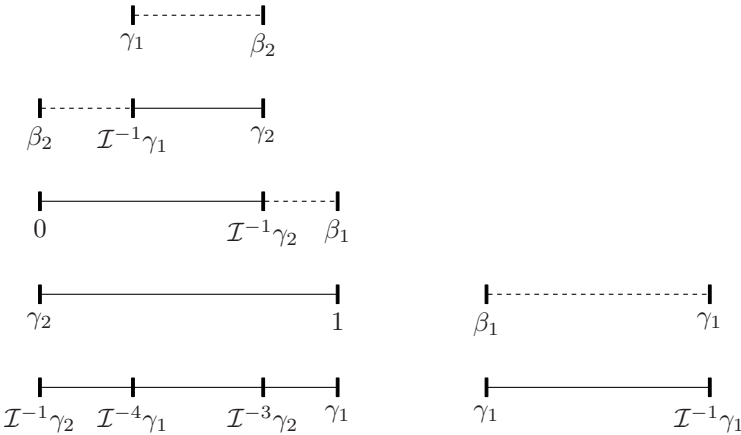


Figure 2.10. Induction castle 1.2.

Figure 2.10 shows the actual induction castle of the new family $E'_1 = [131]$, $E'_2 = [2131]$. In the interval E_2 at this stage appear no point $\mathcal{I}^{-s}\gamma_j$, though some must be there by minimality.

Further evolution of Example 1. — We suppose now that for our new intervals $r_1 > l_1$. The greedy decision is $\{1\}$, $c(1) = -$. We have to take $F = \{1\}$. Then the new w_1 is 1312 and the new castle forest, using the second case of Proposition 2.17, is in Figure 2.11.

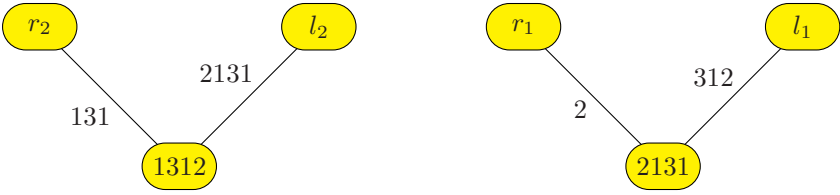


Figure 2.11. Labeled castle forest 1.3.

Thus 2131 was indeed bispecial (but it would not have been so if for the new intervals we had $l_1 > r_1$), and we are in a situation reached by the algorithms of [20] [21], though we got there in a different way. Note that the bispecial word 1312, which labels the root of tree 2 by convention, does not appear as a label of any edge or path in the castle forest.

Evolution of Example 3. — $\gamma(E_1)$ is $\mathcal{I}^{-1}\gamma_2$, the order between it and β_1 is given by the sign of $l_1 - r_2 = r_1 + l_2 - r_1 > 0$, thus $c(1) = +$ without any extra assumption. We take $F = \{1\}$, the greedy decision, and the new castle forest is in Figure 2.12.

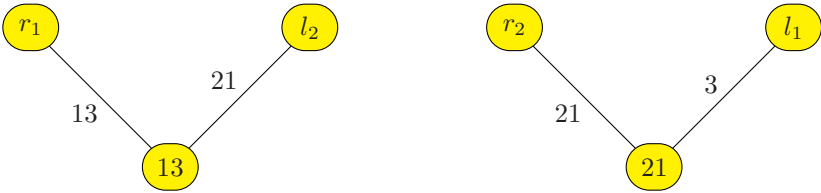


Figure 2.12. Labeled castle forest 3.1.

We arrive in a situation which can be dealt with by the algorithms of [20] [21], which reflects the fact that our permutation is in the same Rauzy class as the symmetric permutation.

2.5. Names of Rokhlin towers

The labels on the castle forest give extra information on Rokhlin towers, as they are linked to the coding of the levels by the partition into Δ_i .

DEFINITION 2.23. — *If X is equipped with a partition P such that each level $T^r Y$ is contained in one atom $P_{w(r)}$, the name of the tower is the word $w(0) \cdots w(h - 1)$.*

However, there is a technical problem with the towers used in Section 2.2; to understand it, look at Figures 2.9 and 2.10. Starting from the induction castle 1.2, we want to discard the tower of basis E_2 , corresponding to a tree with a single edge and empty label, and to an empty return word by the remarks after Definition 2.20; this is done by pasting it on top of the other towers; we get three towers over E_1 , and we check their names are, from left to right, 1312, 13122 and 13; these names do not appear as labels of paths in castle forest 1.2 as they are the *prefix* return words of the family of words $\{131\}$, see the remarks after Definition 2.20. In [23] we actually used the names of these towers; here, as we have been working with suffix return words, which are necessary to build the special words by successive extensions to the right, we prefer to change the towers so that their names are suffix return words, this will be done by using the cylinders for \mathcal{I}^{-1} , which amounts to replace each $[w_i]$ by a translate $\mathcal{I}^{|w_i|-1}[w_i]$. Then we split the towers so that their names are labels of edges instead of paths. Namely

PROPOSITION 2.24. — *At any stage, the $2k - 2$ nonempty labels of the edges in the castle forest are the names, for the partition P of $[0, 1[$ into $\{\Delta_1, \dots, \Delta_k\}$, of $2k - 2$ disjoint Rokhlin towers in the system $([0, 1[, \mathcal{I}, \mu)$ (for any invariant probability μ), filling the whole space, and whose levels are intervals.*

Proof. — At any given stage, let H be the greedy decision, $v_{i,j}$, $i \in H$, $1 \leq j \leq s_i$, the labels of all the paths from the root of tree i , $i \in H$, to the leaves. Then, each word $w_i v_{i,j}$, $i \in H$, $1 \leq j \leq s_i$, has only two factors of the form w_h for $h \in H$, one as a prefix and one as a suffix. Thus, if we denote by \bar{x} the word x read backwards, by looking in each trajectory of \mathcal{I}^{-1} at the occurrences of the \bar{w}_i , $i \in H$, we get $\sum_{i \in H} s_i$ disjoint Rokhlin towers for \mathcal{I}^{-1} whose bases are the cylinders $[\bar{v}_{i,j} \bar{w}_i]$, $i \in H$, $1 \leq j \leq s_i$; they form an induction castle thus their levels are intervals, and their names are the $\bar{v}_{i,j}$. By reversing the order of the levels, we get $\sum_{i \in H} s_i$ disjoint Rokhlin towers for \mathcal{I} whose names are the $v_{i,j}$. Then $v_{i,j} = v'_{i,j,1} \cdots v'_{i,j,t(i,j)}$, where the $v'_{i,j,h}$, $i \in H$, $1 \leq j \leq s_i$, $1 \leq h \leq t(i,j)$, are all the nonempty labels of edges, and thus $\sum_{i \in H, 1 \leq j \leq s_i} t(i,j) = 2k - 2$. Then we cut each tower with name $v_{i,j}$ into $t(i,j)$ disjoint towers with names $v'_{i,j,h}$, proving our proposition. \square

The nonempty labels of edges in the castle forest at stage $n + 1$ are concatenations of the nonempty labels at stage n ; this concatenation is made as in Proposition 2.22 (this would not be the case for labels of paths, and that is why we split the towers). By a standard ergodic argument,

the towers at stage $n + 1$ are made by *cutting and stacking* following the recursion rules giving their names by concatenating names of towers at stage n .

2.6. Structure theorem

We are now ready to consider infinite iterations of the induction. We check first that no decision can block the process, or equivalently that, at each stage we reach, there exists a nonempty decision. Indeed *the greedy decision is nonempty* as we know from Proposition 2.12 that there is always at least one tree with a node. Thus, whatever the allowed (thus nonempty) decision we take, we can iterate the induction infinitely many times.

PROPOSITION 2.25. — *By our induction, through any sequence of allowed decisions F_n , \mathcal{I} generates an infinite sequence of allowed castle forests and allowed choices $(G_n, c_n, n \in \mathbb{N})$, such that*

- (i) G_0 is defined in Section 2.3,
- (ii) G_{n+1} is deduced from G_n and c_n as in Proposition 2.17, with decision F_n ,
- (iii) $c_{n+1}(i) = c_n(i)$ whenever i is not in F_n ,
- (iv) or each i , i is in F_n with $c_n(i) = +$, resp. $-$, for infinitely many n .

The half-lengths $l_{i,n}$ and $r_{i,n}$ of the special intervals $E_{i,n}$ built by Definition 2.13 at each stage tend to zero when n goes to infinity.

Proof. — We just have to prove item iv and the last assertion. Suppose first that i is in F_n at infinitely many stages; then by construction, the left and right endpoints of $E_{i,n}$ are respectively $\mathcal{I}^{-a(n)}\gamma_{b(n)}$ and $\mathcal{I}^{-a'(n)}\gamma_{b'(n)}$, and there is no point $\mathcal{I}^{-x}\gamma_j$ inside $E_{i,n}$ for $1 \leq x \leq a(n) \vee a'(n)$, with $a(n) \vee a'(n) \rightarrow +\infty$. By minimality this implies that both half-lengths of $E_{i,n}$ tend to zero, and that $c(i)$ is $+$ and $-$ infinitely many times.

Let now J be the set of i which are in F_n at infinitely many stages; J is nonempty as the F_n are nonempty. After some stage N only the i in J are in F_n , and after some stage $N' \geq N$ no subtree of a tree i for i in J , can be pasted on a tree j for j in J^c (as there is a finite number of these subtrees, and they could not be pasted again on a tree i for i in J). Then if at stage N' some l_j , resp. r_j , for j in J^c , is a leaf of a tree i for i in J , at further stages this leaf can only be transferred between trees i for i in J , and at infinitely many stages it will be a leaf of the tree i_0 for some i_0 in J ; but then the train-track equalities imply that at infinitely many stages $n > N'$

$l_{i_0,n} + r_{i_0,n} \geq l_{j,n} = l_{j,N}$, resp. $l_{i_0,n} + r_{i_0,n} \geq r_{j,n} = r_{j,N}$, which contradicts the last paragraph.

Thus at stage N' the trees i , for i in J , have only leaves l_j or r_j for j in J , and thus $J = \{1, \dots, k-1\}$ by Proposition 2.12. \square

COROLLARY 2.26. — *The system $([0, 1[, \mathcal{I}, \mu)$ is of rank at most $2k - 2$ by intervals, see [14]: namely, for each f in $\mathcal{L}_2([0, 1])$ and each $\varepsilon > 0$ there exists $N(\varepsilon)$ such that for all $k > N(\varepsilon)$ there exists f_k , which satisfies $\int \|f - f_k\| d\mu < \varepsilon$ and is constant on each level of each of the $2k - 2$ Rokhlin tower of Proposition 2.24 at stage n .*

Proof. — These levels are intervals, whose lengths at stage n tend to zero, thus the σ -algebras generated by the levels of the towers at stage n converge to the full σ -algebra when n tends to infinity. \square

This result is not optimal as the rank by intervals is known to be at most k [14], but our towers are completely explicit and will be used in Section 4.3 below.

We have seen that given the map \mathcal{I} , the sequence of decisions, and thus the sequences of castle forests and choices, are not unique. But, for a given interval E_i , the first strict subinterval to be generated by Definition 2.13 is the same, independently of the decisions; as each interval is cut at infinitely many stages, every sequence of allowed decisions gives the same subintervals, though possibly not at the same stages; however, what a particular sequence of allowed decisions determines are the relations between the half-lengths of $E_{i,n}$ at the same stage n for different $1 \leq i \leq k-1$, and thus the sequence of castle forests depends strongly on the sequence of decisions.

We can now tell which words in an induction family are bispecial, though this needs information which will be available only at further stages of the process.

PROPOSITION 2.27. — *A word w_i in an induction family W is bispecial if and only if it satisfies the two equivalent conditions*

- *if (j_1, \dots, j_s) is the unique sequence such that $j_1 = i$, the tree j_h has a single edge going to a leaf $x_{h+1} = l_{j_{h+1}}$ or $x_{h+1} = r_{j_{h+1}}$ for $1 \leq h \leq s-1$, and the root of tree j_s is a node, then w_{j_h} is in the greedy decision with $c(i) = +$, resp. $-$, whenever $x_h = l_{j_h}$, resp. $x_h = r_{j_h}$, for $2 \leq h \leq s$,*
- *there exists a further stage of the induction such that $w'_i = w_i$ and the root of tree i is a node.*

All the bispecial words of $L(\mathcal{I})$ are the words $w_{i,n}$ defined by the induction in Section 2.4.

Proof. — If the second condition is satisfied, $w_i = w'_i$ is bispecial as it is already left special. If the first condition is not satisfied, at some further stage w_i is extended to the right in a unique way, and thus is not right special. The last assertion follows because of Proposition 2.22. \square

We can now show the converse of Proposition 2.25.

THEOREM 2.28. — *Any infinite sequence of castle forests, allowed choices and allowed decisions, satisfying items i to iv in Proposition 2.25, defines at least one k -interval exchange with permutation π and alternate discontinuities, satisfying the i.d.o.c. condition, which generates it as in Proposition 2.25.*

Proof. — The proof follows exactly the proof of Theorem 2.9 in [20]. Given the castle forests G_n , the choices c_n and the decisions F_n , we build words $w_{i,n}$, $1 \leq i \leq k-1$, $n \geq 0$, and labels for each edge, as in Section 2.4. By following the steps of the induction, we get that indeed the families W_n satisfy the conditions defining induction families, have the required castle forests, and thus, if the language L is the set of all the finite factors of the $w_{i,n}$, its bispecial words are given by Proposition 2.27.

We shall now show that L satisfies the conditions of Theorem 2 of [22]. Translated into the simpler context of alternate discontinuities, this states that a language L is the language of a k -interval exchange with permutation π and alternate discontinuities, satisfying the i.d.o.c. condition, if (and only if) it is *minimal* (meaning that for each m , there exists n such that every word of length m of L occurs in every word of length n of L), the words with two letters are $\pi 11$, $\pi i(i-1)$, πii , $2 \leq i \leq k$, and, if w is a nonempty bispecial word, there exist a and b such that $A(w) = \{\pi a, \pi(a+1)\}$, $D(w) = \{b, b+1\}$, there are three words of the form xwy for letters x, y and these are πawb , $\pi(a+1)w(b+1)$ and either $\pi(a+1)wb$ or $\pi aw(b+1)$.

Our language L satisfies the condition on words with two letters, and the condition on bispecial words. by construction, as can be checked on the castle forests.

The nontrivial condition to satisfy is minimality, and this is shown exactly as in [20]. The words of L are the factors of the $w_{i,n}$, or of the $k-1$ infinite sequences \mathcal{O}_i which begin with $w_{i,n}$ for all n . Let w be a word of L : w must be a suffix of infinitely many words of L , because every word of L can be extended infinitely many times to the left, thus w occurs infinitely often in at least one \mathcal{O}_i ; hence all the prefixes of \mathcal{O}_i occur infinitely often

in at least one \mathcal{O}_j , and if $j \neq i$, then we can drop \mathcal{O}_i and use only the others \mathcal{O}_j to generate L . Thus L is the union of the $L(V_i)$ for $d \leq k - 1$ one-sided infinite sequences V_i which are *recurrent*: each factor of V_i occurs at infinitely many places in V_i .

Let now $L' = \cup_{i=2}^d L(V_i)$; suppose there exist words w_1 in $L(V_1) \setminus L'$ and w' in $L' \setminus L(V_1)$; suppose w' is in $L(V_t)$: we check that $L(V_1)$ and $L(V_t)$ have at least a one-letter word in common, otherwise this contradicts the condition on the words of length 2 of L ; take a nonempty word w_0 in $L(V_1) \cap L(V_t)$; by the recurrence property, we can find a word $w_1 v_1 w_0 v'_1 w_1$ in $L_1 \subset L$, and a word $w' v_2 w_0 v'_2 w'$ in $L' \subset L$; but no word $w_1 v w_2$ or $w_2 v w_1$ is in L , as such a word is neither in $L(V_1)$ nor in L' , and this implies that there is a bispecial word of L , w_3 , such that two of the four possible words $x w_3 y$ are not in L , and this contradicts the condition on bispecial words. Hence either $L' \subset L(V_1)$, thus $L = L(V_1)$, or $L(V_1) \subset L'$, and then we can drop V_1 ; by iterating the process, we get that $L = L(V)$ for one recurrent infinite sequence V .

Let now w be a factor of V ; we look at the possible words v such that wv is right special; if two different such words v and v' have a common suffix z , then there exist two different words such that $v_1 z$ and $v_2 z$ are right special, and we find a bispecial word w_4 with four possible $x w_4 y$, which contradicts the condition on bispecial words. Thus there is at most one such possible v with last letter i , $1 \leq i \leq k - 1$ (and none with last letter k), and thus there are at most k ways of going from one occurrence of w in V to the next one. Hence any factor w occurs in V at infinitely many places with bounded gaps, and $L = L(V)$ is minimal.

Thus we have found \mathcal{I} such that $L = L(\mathcal{I})$, and, by following again the steps of the induction, we check the required sequence of castle forests, choices and decisions is one of the possible such sequences defined by \mathcal{I} . \square

Given the castle forests G_n , the choices c_n , and the decisions F_n , it follows from the above proof and [22] that a vector of lengths for an associated \mathcal{I} is given by $(\mu[1], \dots, \mu[k])$ for any invariant probability μ on the symbolic dynamical system X_L . Thus for one sequence there may be several corresponding k -interval exchanges \mathcal{I} with permutation π . The solution \mathcal{I} is unique if and only if \mathcal{I} is *uniquely ergodic* (it has a unique invariant probability measure); a famous result of Veech [39] and Masur [32] states that the set of $(\alpha_1, \dots, \alpha_k)$ in \mathbb{R}^{+k} for which \mathcal{I} defined by the vector $(\frac{\alpha_1}{\alpha_1 + \dots + \alpha_k}, \dots, \frac{\alpha_k}{\alpha_1 + \dots + \alpha_k})$ is uniquely ergodic has full Lebesgue measure; a mainly combinatorial proof of this result, quite in the spirit of the present paper, can be deduced from [5] and [4], see [16]. When the solutions are

non uniquely ergodic the analysis has been made in [28]: the possible vectors of lengths lies in a convex set \mathcal{S} , with extremal points $\alpha^1, \dots, \alpha^d$, $d \leq \lfloor \frac{k}{2} \rfloor$ [25], [38]; each α^i defines, by $\mu_i[j] = (\alpha^i)_j$, an ergodic invariant measure μ_i for every \mathcal{I} with vector of lengths $\alpha \in \mathcal{S}$ and permutation π ; for such an \mathcal{I} , all its invariant ergodic probabilities are the μ_i . For a given solution \mathcal{I} of vector of lengths α , the Lebesgue measure is such that $\mu[j] = \alpha_j$; the Lebesgue measure will be ergodic if and only if α is one of the α^i .

To know whether a choice c is allowed, the only way we know is by trial and error: we make the induction by following the rules of Propositions 2.15 and 2.17 with choice c and the greedy decision, getting a new castle forest G' , and check whether G' is allowed. For example, starting from the castle forest 1.3 above, we suppose that $c(1) = +$ and $c(2) = -$. This is clearly not allowed as this implies $l_1 > r_2$ and $r_2 > l_1$, but we can also check that in that case G' has the parenthesized relations $r_1 + l_1 = l_2$, $r_2 + l_2 = r_1$, thus G' has just two trees with a single edge and is not allowed.

2.7. Algorithms

If we want to make the induction process, we can define the decision as we want at each stage; however, it is better to be *deterministic*, in the sense that for a given castle forest and a given choice the decision is always the same. Thus an *algorithm of induction* is a way to associate a decision to each castle forest and choice. When we fix an algorithm, each map \mathcal{I} defines a unique sequence of castle forests, choices and decisions.

The *greedy algorithm* consists in taking the greedy decision at each stage. It works for every map \mathcal{I} with alternate discontinuities (this condition will be lifted in Section 3): we change all the i for which we have the necessary information.

The *trees of relations algorithms* comprise the one described extensively in [20], and two further variants used in [15] and [10], we give here a quick summary. A *tree of relations* (not to be confused with a castle tree) is a non-oriented and non-rooted tree with $k-1$ vertices labeled i , $1 \leq i \leq k-1$, and edges labelled $\hat{+}$, $\hat{-}$, or $\hat{=}$, such that two adjacent edges never have the same label. A tree of relations defines three bijections s , p , m , by

- $s(i)$ is the only j such that there is a $\hat{=}$ edge between i and j , or $s(i) = i$ if there is no such edge,
- $p(i)$ is the only j such that there is a $\hat{+}$ edge between $s(i)$ and j , or $p(i) = s(i)$ if there is no such edge,
- $m(i)$ is the only j such that there is a $\hat{-}$ edge between $s(i)$ and j , or $m(i) = s(i)$ if there is no such edge.

This defines a castle forest by the parenthesized train-track equalities $l_i + r_i = r_{m(i)} + l_{p(i)}$, $1 \leq i \leq k-1$. In this sense a tree of relations is a tree which hides a forest: for example castle forest 1.0 in Figure 2.6 is associated with the tree of relations $1 \hat{=} 2$, while castle forest 1.3 in Figure 2.11 is associated to $1 \hat{=} 2$, and castle forest 4.2 in Figure 3.4 below to $1 \hat{=} 3 \hat{=} 2$.

What is shown (though in other terms) in [20] is that, *if at one stage of an induction process we reach a castle forest associated to a tree of relations, then there exist decisions such that the castle forest at the next stage is associated to a tree of relations*, and thus our castle forests can stay associated to trees of relations at all further stages (hence every root is a node, every tree has two edges); in particular, if π is the symmetric permutation, the initial castle forest has this property. There are several possible such decisions: the ones used in [20] are not the greedy decisions of Definition 2.13, but indeed the greediest ones allowing to stay associated with trees of relations, and define what we call the *greedy-hyperelliptic algorithm*, while both the decisions in [15] and [10] are slightly slower; all these algorithms are slower than the greedy one, but create less castle forests, see Section 2.8 below. This stems from the fact that they correspond to special cases, namely interval exchanges which come from cross-sections of Veech translation surfaces linear flows, and hence belong to a set which is invariant under renormalization and therefore described by less castle graphs.

The *loops variant* is a possible modification from any given algorithm, for example the greedy one, as follows: for a given castle forest G and choice c , let H' be the set of i such that the root of tree i is a node, and, if $c(i) = +$, resp. $-$, at the end of the right, resp. left, edge out of this root, there is a leaf l_i , resp. r_i . If H' is empty, we just use the given algorithm; if H' is nonempty, we order H' , lexicographically for example, and *take the decision* $\{i_1\}$, where i_1 is the lowest element in H' . Then at the next stage we have the same castle forest G , and we iterate the process a finite number of times, until the choice of i_1 becomes the opposite to its previous choice; then we change i_2 , etc., until we reach a choice where H' is empty. Thus, in Example 1 of Section 2.3, starting from the castle forest 1.0, we take decision $\{1\}$ once as $c(1) = +$, coming to the same castle forest (with different labels), then $c(1) = -$ and we take whatever decision is allowed by our original algorithm, in that case $\{1, 2\}$ for either the greedy or the greedy-hyperelliptic one; we arrive at castle forest 1.3.

In every case, for a given castle forest and choice, we have complete freedom to choose the decision we want, and thus mix several algorithms.

In Section 2.8 we give examples where we choose to minimize the number of possible castle forests; we use the greedy-hyperelliptic algorithm (with the loops variant) wherever it is possible, and this corresponds exactly to permutations in the hyperelliptic Rauzy class; in other cases, we generally use the greedy algorithm with the loops variant, but use also some local ad hoc variants. It is not clear whether one can use less castle forests, as happens with the trees of relations algorithms, if starting from a general interval exchange.

All the above algorithms are *additive*, each interval being changed at most one time at each step; *multiplicative* algorithms can then be build by gluing several steps together, as is done for three intervals in [18], [21].

2.8. Graphs of graphs

Let an algorithm be given. Inspired by [20], we call *graph of graphs* the oriented graph whose vertices are all the possible allowed castle forests on $\{1, \dots, k-1\}$ which can be reached from the castle forest G_0 defined by a finite sequence of inductions associated to allowed choices and decisions determined by the algorithm; from a castle forest G , each choice c determines an edge from G to the new castle forest built with the decision F associated to G and c by the algorithm. This edge is called a *move* and is labelled by $c'(1) \cdots c'(k-1)$ where $c'(i) = c(i)$ if i is in F , $c'(i) = 0$ otherwise. Then the sequence of allowed castle forests in Proposition 2.25 and Theorem 2.28 corresponds to a path in the graph of graphs. Note that for the graphs of graphs in [20], their vertices were labeled by trees of relations, in contrast with castle forests in the present paper, but as the name is still appropriate we keep it, the second “graph” word referring either to castle forests or their underlying tree of relations if there is one; similarly, the castle forests can be replaced by the castle graphs of [23].

We proceed now to describe a graph of graphs for each exchange of up to 4 intervals with alternate discontinuities, that is $\beta_i < \gamma_i$ for each $1 \leq i \leq k-1$ and $\gamma_i < \beta_{i+1}$ for each $1 \leq i \leq k-2$. The opposite situation, where $\gamma_i < \beta_i$ for each $1 \leq i \leq k-1$ and $\beta_i < \gamma_{i+1}$ for each $1 \leq i \leq k-2$, can be deduced of it, after renumbering the E_i , by reversing the orientation (this means conjugation with the map $x \rightarrow 1-x$); other situations will be dealt with in Section 3 below.

We apologize to the reader: by lack of space we do not draw actual graphs of graphs; to define a castle forest, we use its parenthesized train-track equalities in an abbreviated form by listing their second members

from 1 to $k - 1$. Thus castle forest 1.0 above becomes $r_2 + l_1, r_1 + l_2$ as a shorter form of $l_1 + r_1 = r_2 + l_1, l_2 + r_2 = r_1 + l_2$.

For $k = 2$ the new induction coincides with the Rauzy induction, as mentioned in [20].

For $k = 3$ there are three primitive permutations. The most studied is $1 \rightarrow 3, 2 \rightarrow 2, 3 \rightarrow 1$. With either the greedy or the greedy-hyperelliptic algorithm, both with the loops variant, the graph of graphs we get is made with the three *recurrent* states (as indicated by the edges) of Figure 2.13, as in [21]; the initial state is the left one. The greedy-hyperelliptic algorithm without the loops variant would give the same states with two more loops, $--$ and $++$; the greedy algorithm without the loops variant would give 7 recurrent castle forests (including castle forest 1.1 or 1.2 above).

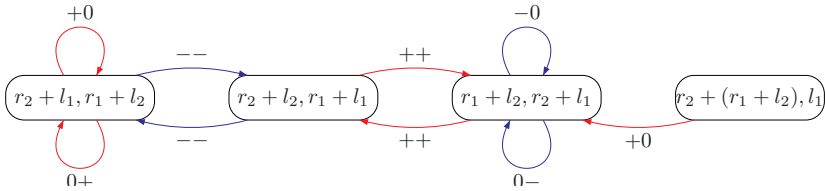


Figure 2.13. The graph of graphs for three intervals.

The permutation $1 \rightarrow 3, 2 \rightarrow 1, 3 \rightarrow 2$, with the greedy-hyperelliptic algorithm and loops variant, gives the same graph of graphs and the same moves, with $r_1 + l_2, r_2 + l_1$ as initial state.

For $1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 1$, the initial castle forest is the *transient* one on the right and the graph of graphs is the full Figure 2.13; we use the greedy-hyperelliptic algorithm, with the loops variant, after having left the transient state.

For $k = 4$ there are 13 primitive permutations, belonging to two Rauzy classes, the hyperelliptic class and the rotations class; we begin by the much-studied hyperelliptic class, for which we draw the graph in Figure 2.14 (which is not an actual graph of graphs, as will be seen below): it has nine recurrent and four transient states.

For the symmetric permutation $1 \rightarrow 4, 2 \rightarrow 3, 3 \rightarrow 2, 4 \rightarrow 1$, with the greedy-hyperelliptic algorithm and the loops variant, the graph of graphs, as described in [20] using trees of relations, and in [23] using castle graphs, consists of the nine recurrent castle forests of \mathcal{G} ; the initial state is in the middle of the middle (or third) row of \mathcal{G} .

For other permutations, we use always the greedy-hyperelliptic algorithm (with the loops variant) as soon as we are in a castle forest associated to a

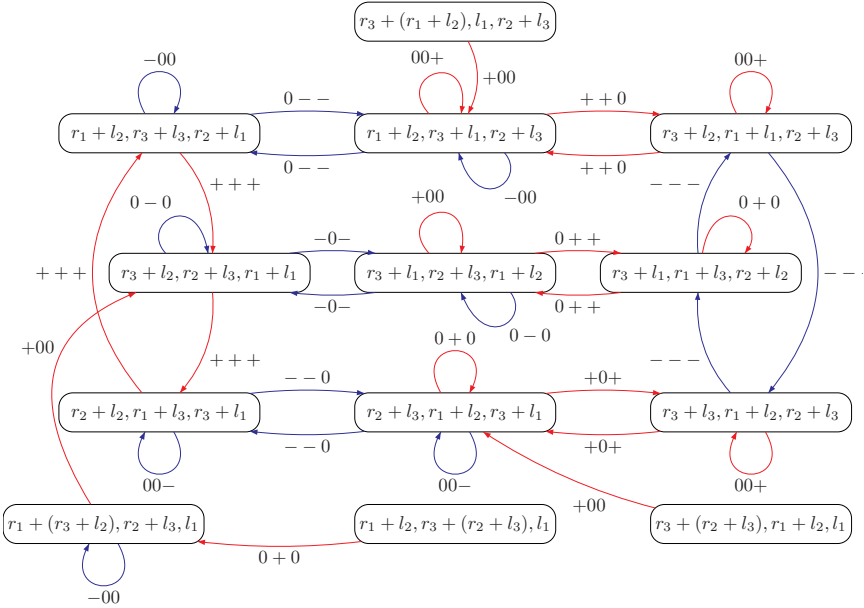


Figure 2.14. The graph \mathcal{G} .

tree of relations. Thus $1 \rightarrow 4, 2 \rightarrow 1, 3 \rightarrow 3, 4 \rightarrow 2$ and $1 \rightarrow 4, 2 \rightarrow 2, 3 \rightarrow 1, 4 \rightarrow 3$ give also the recurrent part of \mathcal{G} with initial states respectively in the middle of the second and fourth row of \mathcal{G} .

For $1 \rightarrow 3, 2 \rightarrow 1, 3 \rightarrow 4, 4 \rightarrow 2$, the initial state is in the middle of the fifth row of \mathcal{G} ; from this state, we choose the algorithm which imposes the decision $\{2\}$ (with the only allowed choice $c(2) = +$), going to the state on its left, in which we impose the decision $\{1\}$; we get the possible moves in the picture, the graph of graphs is made with the nine recurrent states of \mathcal{G} and two transient states.

For $1 \rightarrow 3, 2 \rightarrow 2, 3 \rightarrow 4, 4 \rightarrow 1$, we have the initial castle forest in the top row of \mathcal{G} . From this state, by imposing the decision $\{1\}$ (with the only allowed choice $c(1) = +$), we get the possible moves in the picture: the graph of graphs is made with the nine recurrent states of \mathcal{G} and one transient state.

For $1 \rightarrow 2, 2 \rightarrow 4, 3 \rightarrow 3, 4 \rightarrow 1$, we have the initial castle forest in the right of the fifth row of \mathcal{G} . From this state, by imposing the decision $\{1\}$ (with the only allowed choice $c(1) = +$), we get the possible moves in the

picture: the graph of graphs is made with the nine recurrent states of \mathcal{G} and one transient state.

Finally for $1 \rightarrow 2, 2 \rightarrow 4, 3 \rightarrow 1, 4 \rightarrow 3$, we have a transient initial castle forest $r_2 + l_3, l_1, r_3 + (r_1 + l_2)$. From this state, we impose the decision $\{3\}$; this gives a loop $00-$, and a move $00+$ to another transient castle forest $r_2 + (r_3 + l_3), l_1, r_1 + l_2$. From that state, by imposing again the decision $\{1\}$ (with the only allowed choice $c(1) = +$), we go to the castle forest $r_3 + l_3, r_2 + l_1, r_1 + l_2$, which is one of the nine recurrent states of the graph of graphs deduced from \mathcal{G} by exchanging the letters 2 and 3, with corresponding moves (this is the renumbering we get by reversing the orientation). Thus we have again nine recurrent and two transient states, but not those of \mathcal{G} .

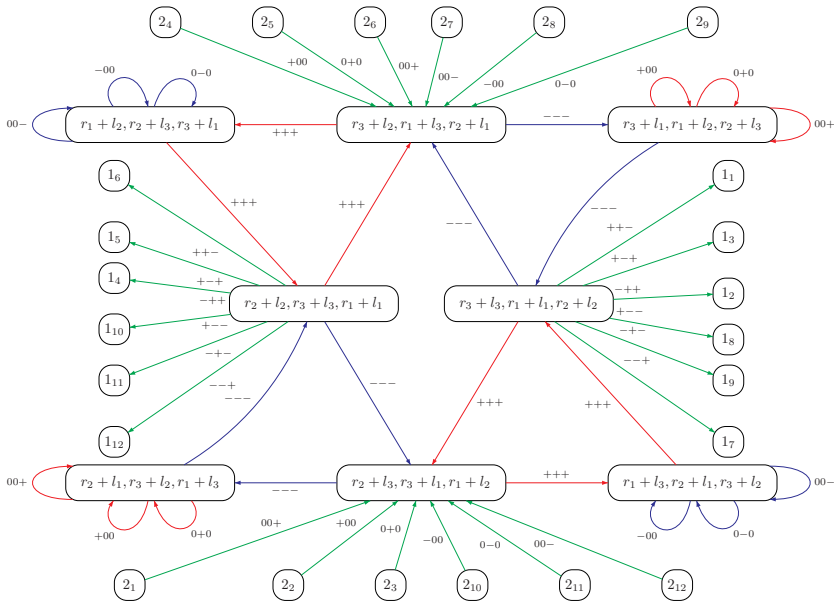


Figure 2.15. The central part of \mathcal{G}' .

We look now at the rotations class. We give here the full graph of graphs \mathcal{G}' for $1 \rightarrow 4, 2 \rightarrow 3, 3 \rightarrow 1, 4 \rightarrow 2$, for the greedy algorithm, with the loops variant and a local variant detailed below for states 2_1 to 2_{12} . In the initial stage, we have the castle forest $r_2 + l_3, r_3 + l_1, r_1 + l_2$; the eight castle forests where each tree has two edges, and all the moves concerning them, are shown in Figure 2.15.

There are 108 castle forests where some trees do not have two edges; we name them 1_j to 9_j , $1 \leq j \leq 12$; this terminology reflects the fact that the graph of graphs is invariant by all permutations on $\{1, 2, 3\}$ and by the exchange between l and r , thus, for each x in $\{1, 9\}$ x_2 , resp. x_3, x_4, x_5, x_6 are deduced from x_1 by replacing, on both sides of each parenthesized relations, $(1, 2, 3)$ by $(2, 3, 1)$, resp. $(3, 1, 2), (3, 2, 1), (1, 3, 2), (2, 1, 3)$, then x_{6+i} is deduced from x_i by exchanging l and r . We have

$$\begin{aligned} 1_1 &= l_3, (r_3 + (r_1 + l_2)) + l_1, r_2, \\ 2_1 &= l_3, r_3 + l_1, r_2 + (r_1 + l_2), \\ 3_1 &= r_3, l_1, (r_2 + (r_1 + l_3)) + l_2, \\ 4_1 &= r_3, (r_2 + (r_1 + l_3)) + l_1, l_2, \\ 5_1 &= (r_2 + (r_1 + l_2)) + l_3, r_3, l_1, \\ 6_1 &= l_3, r_3 + (r_1 + l_2), r_2 + l_1, \\ 7_1 &= l_3, r_3, (r_2 + (r_1 + l_2)) + l_1, \\ 8_1 &= l_3, (r_3 + l_1) + (r_1 + l_2), r_2, \\ 9_1 &= l_3, r_1 + (r_3 + l_2), r_2 + l_1. \end{aligned}$$

Here are now the allowed moves between forests where some trees do not have two edges, up to permutations on $\{1, 2, 3\}$ and exchanges on l and r (and thus on $+$ and $-$):

- from $1_1, 0-0$ leads to 6_1 ,
- from $2_1, 0--$ leads to 6_1 ; when $c(3) = +$ we impose decision $\{3\}$, getting the move in Figure 2.15,
- from $3_1, 00-$ leads to 9_2 ,
- from $4_1, 0+0$ leads to $3_1, 0-0$ to 8_7 ,
- from $5_1, -00$ leads to $3_{10}, +00$ to 7_1 ,
- from $6_1, 0--$ leads to $2_1, 0-+$ to $5_1, 0+-$ to $4_7, 0++$ to 6_4 ,
- from $7_1, 00-$ leads to $I2_1, 00+$ to 5_1 ,
- from $8_1, 0-0$ leads to $2_1, 0+0$ to 4_7 ,
- from $9_1, 0--$ leads to $3_7, 0-+$ to $7_4, 0+-$ to $8_1, 0++$ to 9_4 .

If we look now at other permutations in the rotations class, with the greedy algorithm and the same variants:

- $1 \rightarrow 4, 2 \rightarrow 1, 3 \rightarrow 2, 4 \rightarrow 3$, has \mathcal{G}' as graph of graphs, with initial castle forest $r_1 + l_2, r_2 + l_3, r_3 + l_1$,

$1 \rightarrow 4, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 1$, has \mathcal{G}' as graph of graphs, with initial castle forest $r_3 + l_1, r_1 + l_2, r_2 + l_3$,

$1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4, 4 \rightarrow 1$, has \mathcal{G}' as graph of graphs, with initial castle forest 2_2 ,

$1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 1, 4 \rightarrow 2$, has \mathcal{G}' as graph of graphs, with initial castle forest 9_4 .

For $1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 2, 4 \rightarrow 1$, we have the initial castle forest $Z = r_3 + (r_2 + l_3), r_1 + l_2, l_1$; then we impose the decision $\{1\}$ (with the only allowed choice $c(1) = +$), the move $+00$ leads to $r_1 + l_2, r_2 + l_3, r_3 + l_1$, and we get as graph of graphs the union of \mathcal{G}' and the initial transient castle forest Z . If we use the loops variant from the initial Z , with decision $\{2\}$ as long as $c(2) = -$, we get one more transient castle forest, $Z' = r_1 + (r_2 + l_2), l_3, r_3 + l_1$, and arrive in \mathcal{G}' through $r_2 + l_2, r_3 + l_3, r_1 + l_1$.

3. Induction without alternate discontinuities

3.1. Decompositions of intervals

Without the condition of alternate discontinuities, the situation is much more complicated in the initial stages, though, by minimality, after a finite number of stages we are back in a situation similar to those in Section 2, as will be explained in Section 3.4.

Namely, see Figure 3.1, an interval $E = [w]$ with a left special w may contain in its interior several consecutive discontinuities β_j , $i(E) \leq j \leq i(E) + a(E) - 2$ which cut it into $a(E)$ subintervals, $a(E) \geq 2$, denoted from left to right by $E_{A,t}$, $1 \leq t \leq a(E)$, where in the previous notations $E_- = E_{A,1}$, $E_+ = E_{A,a(E)}$. $E_{A,t}$ is also $\mathcal{I}[\pi(x+t-1)w]$, where the set $A(w)$ of Definition 1.7 is $\{\pi(x) \cdots \pi(x+a(E)-1)\}$ ($A(w)$ must have this form because of Lemma 1.9). If E is the interval E_i at a stage of the induction, then the numbering will be made so that $i(E_i) = i$ and the respective lengths of the subintervals are denoted by $u_{i,t}$, $1 \leq t \leq a_i$, or equivalently by the notations $l_i = u_{i,1}$, $r_i = u_{i,a_i}$, $u_{i,t} = u_{i+t-2}$ for $2 \leq t \leq a_i - 1$, where u_i is the distance from β_i to β_{i+1} ; the $u_{i,t}$ are the *partial lengths* of E_i and will constitute the parameters of the induction.

An interval $E = [w]$ with a right special w may contain in its interior several discontinuities $\mathcal{I}^{-s}\gamma_j$ for the same minimal s , which cut it into $d(E)$ subintervals, $d(E) \geq 2$, denoted from left to right by $E_{D,t}$, $1 \leq t \leq d(E)$, where in the previous notations $E_m = E_{D,1}$, $E_p = E_{D,d(E)}$. $E_{D,t}$ is also the cylinder $[w(x'+t-1)]$, where the set $D(w)$ of Definition 1.7 is

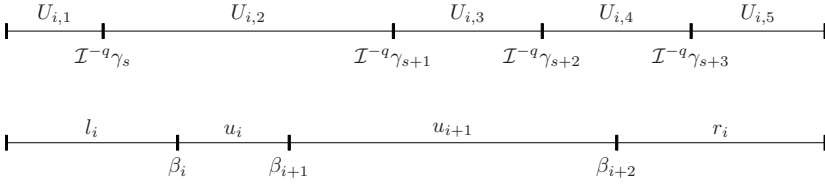


Figure 3.1. Decompositions of an interval E_i .

$\{x', \dots, x' + d(E) - 1\}$ ($D(w)$ must have this form because of Lemma 1.9). If E is the interval E_i at a stage of the induction, the respective lengths of these intervals are denoted by $U_{i,t}$, $1 \leq t \leq d_i$; the $U_{i,t}$ will be deduced from the parameters of the induction. Note that all these $\mathcal{I}^{-s}\gamma_j$ are different from all points β_i , $1 \leq i \leq k - 1$, because of the i.d.o.c. condition.

However, the distance between two different β_i or two different $\mathcal{I}^{-s}\gamma_j$ for the same s has a strictly positive lower bound; thus as soon as the length of the interval $[w]$ is small enough, $A(w)$ and $D(w)$ have at most two elements, like in the case of alternate discontinuities.

3.2. More general induction families and castle forests

We shall now formalize our induction in this general case; rather than re-writing a whole tedious theory which nobody would read, we shall stress the differences with the case of alternate discontinuities. The proofs are straightforward, using the methods of Sections 2.2 and 2.3.

Families of special intervals are indexed by $i \in K$, for a set $K \subset \{1, \dots, k - 1\}$, depending on the stage of the induction, where the set K increases along the stages, and is equal to $\{1, \dots, k - 1\}$ after a finite number of stages; at a given stage we have $\#K$ special intervals, and the notation we adopt throughout this section is to *number by i the interval E_i which contains the points β_j for $i \leq j \leq i'$* . Thus in general K is not an integer interval; of course other systems of notation are possible.

Definition 2.10 of a castle forest stays the same except that

- there are at most $2k - 2$ leaves, labeled by $u_{j,t}$, $j \in K$, $1 \leq t \leq a_j$;
- a node has at least two issuing edges, ordered from left to right;
- in a tree with nodes, a leaf l_j , resp r_j , is not at the end of the leftmost, resp. rightmost, edge issuing from a node.

In Lemma 2.8, the $\mathcal{I}^{-m}\gamma_j$ are not totally ordered, and $SY_{i,t}$ is some $E_{g(i,t),A,r(i,t)}$. Thus Definition 2.11 has to be replaced by

DEFINITION 3.1. — *The castle forest of a family of special intervals is deduced from its induction castle in the following way*

- whenever $e_i = 1$, the tree i has one single edge,
- each node in the tree i corresponds to one or several points $\mathcal{I}^{-m}\gamma_j$ partitioning E_i (excluding the endpoints of E_i) for one value of m ,
- if $e_i > 1$, the root of the tree i corresponds to all these points $\mathcal{I}^{-m}\gamma_j$ with the smallest m ,
- from each node in the tree i corresponding to $\mathcal{I}^{-m_0}\gamma_{j_t}$, $1 \leq t \leq s$, there are $s + 1$ edges, ordered from left to right by the order of the points $\mathcal{I}^{-m_0}\gamma_{j_t}$; the t -th edge from the left goes to a node corresponding to all the points $\mathcal{I}^{-m}\gamma_j$ lying between $\mathcal{I}^{-m_0}\gamma_{j_{t-1}}$ and $\mathcal{I}^{-m_0}\gamma_{j_t}$, (resp. left of $\mathcal{I}^{-m_0}\gamma_{j_1}$ if $t = 1$, right of $\mathcal{I}^{-m_0}\gamma_{j_s}$ if $t = s + 1$) for which m is the smallest, or to a leaf if there is no such node,
- the t -th leaf from the left in the tree i receives the label $u_{g,r}$, if $SY_{i,t} = E_{g,A,r}$

The train-track equalities express that for $i \in K$, $\sum_{t=1}^{a_i} u_{i,t}$ is the sum of the labels of the leaves of the tree i , with parentheses corresponding to nodes as before. The castle forest of a family of special intervals has $\sum_{i \in K} a_i$ leaves. Induction families are indexed by $i \in K$, and in general w_i does not begin with the letter i .

Proposition 2.21 is replaced by

PROPOSITION 3.2. — *While we build inductively a family of special intervals E_i , we can build by the same process induction families W , and labels on the edges of the castle forest of the E_i such that*

- $E_i = [w_i]$;
- the tree i has one single edge whenever w_i contains some w_j in W as a strict suffix, and then this edge has an empty label; the only edges with empty labels are single edges;
- a node corresponding to the word $w_i v$ has $d([w_i v])$ issuing edges, ordered from left to right according to the increasing order of the first letters of their label;
- if a node in tree i has a path from the root with label v , then in the induction castle it corresponds to $d([w_i v]) - 1$ points $\mathcal{I}^{-s}\gamma_{y+t}$ in the interior of E_i , where s is the length of the word $w_i v$ and the $y + t$, $0 \leq t \leq d([w_i v]) - 1$, form the set $D(w_i v)$;
- v is the (possibly empty) label of a path from a root w_i to a leaf $u_{j,t}$ if and only if v is a return word of W where $w_i v = v' w_j$, v' ends with the letter $\pi(x + t - 1)$, where $A(w_j)$ is the set $\{\pi(x), \dots, \pi(x + a_j - 1)\}$,

and w_j is the longest word w_h for which $w_i v = v'' w_h$ for a nonempty v'' .

Example 4. — We look at 4 intervals with the symmetric permutation $1 \rightarrow 4, 2 \rightarrow 3, 3 \rightarrow 2, 4 \rightarrow 1$, and suppose that $\gamma_1 < \gamma_2 < \beta_1 < \gamma_3 < \beta_2 < \beta_3$, thus the words of length 2 are $\{41, 42, 43, 33, 34, 14, 24\}$. The left special words with one letter are 3 and 4, which constitute our initial family; we number the special intervals by 1 and 2, so that $\beta_i \in E_i$, and get the initial castle forest in Figure 3.2.

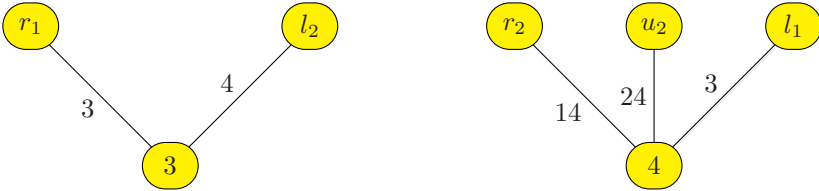


Figure 3.2. Labeled castle forest 4.0.

The interval $E_2 = [4]$ contains β_2 and β_3 , and is thus cut into three subintervals of lengths $l_2 = u_{2,1}, u_2 = u_{2,2}, r_2 = u_{2,3}$. It contains also $\mathcal{I}^{-1}\gamma_1$ and $\mathcal{I}^{-1}\gamma_2$, which are the two points corresponding to the node at the root of tree 2; they cut E_2 into three subintervals, whose lengths are deduced from the castle forest to be r_2, u_2, l_1 from left to right. The train-track equalities are $l_1 + r_1 = r_1 + l_2, l_2 + u_2 + r_2 = r_2 + u_2 + l_1$ (no parentheses in the last one as there is only one node).

Example 5. — We change now the permutation to $1 \rightarrow 4, 2 \rightarrow 3, 3 \rightarrow 1, 4 \rightarrow 2$, again with $\gamma_1 < \gamma_2 < \beta_1 < \gamma_3 < \beta_2 < \beta_3$; the words of length 2 are still $\{41, 42, 43, 33, 34, 14, 24\}$. We get the initial castle forest in Figure 3.3. The train-track equalities are $l_1 + r_1 = r_1 + l_2, l_2 + u_2 + r_2 = u_2 + r_2 + l_1$.

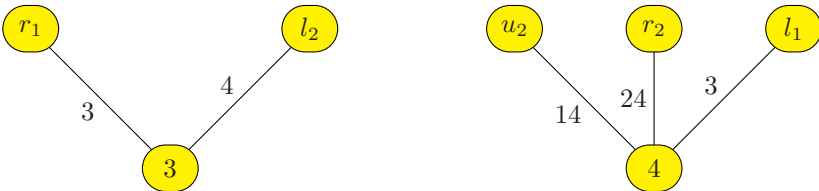


Figure 3.3. Labeled castle forest 5.0.

We shall now build simultaneously the families of special intervals and their induction families, amalgamating the generalizations of Sections 2.2, 2.3 and 2.4.

3.3. The general initial stage

DEFINITION 3.3. — *An initial state is a strict order on the $2k-2$ points β_i and γ_i , $1 \leq i \leq k-1$, such that $\beta_i < \beta_{i+1}$ and $\gamma_i < \gamma_{i+1}$ for $1 \leq i \leq k-2$.*

The initial state, together with the permutation π , determines the sets $A(w)$ and $D(w)$ for words of length 1, and hence the words of length 2. Namely,

- if $\gamma_{i-1} < \beta_j < \gamma_i$ for $e_-(i) \leq j \leq e_+(i)$, $A(i)$ is the set $\{\pi h, e_-(i) \leq h \leq e_+(i) + 1\}$; if there is no such j , $A(i)$ is made with the single letter πj , where j is the number of the first $\beta_q > \gamma_i$;
- if $\beta_{\pi^{-1}(i)-1} < \gamma_j < \beta_{\pi^{-1}i}$ for $b_-(i) \leq j \leq b_+(i)$, $D(i)$ is the set $\{j, b_-(i) \leq j \leq b_+(i) + 1\}$; if there is no such j , $D(i)$ is made with the single letter j , where j is the number of the first $\gamma_q > \beta_{\pi i}$.

The labeled castle forest G_0 is defined in several successive steps; the w_i in the initial induction family is used to label the root of tree i .

- (1) let \tilde{K} be the set of j such that $A(j)$ has at least two elements; for $j \in \tilde{K}$, let i be the smallest element of $A(j)$ and let $w_i = j$; the set of such i is our initial K ;
- (2) for each $1 \leq i \leq k-1$, we define a tree with root labeled i , and $\#D(i)$ edges, the t -th edge (from the left) being labeled j where j is the t -th letter of $D(j)$, and going to a leaf $u_{s,t'}$ where $i = \pi(e_-(j) + t' - 1)$ if j is in \tilde{K} , $t' = 0$ if j is not in \tilde{K} ;
- (3a) for each i in \tilde{K} , in the tree with root labeled i we replace each leaf $u_{j,0}$, such that the tree with root labeled j has a single edge going to a leaf $u_{j',t}$, by the leaf $u_{j',t}$, and extend to the right by j' the label of that single edge;
- (3b) for each i in \tilde{K} , in the tree with root i we replace each leaf $u_{j,0}$, such that the root j is a node, by a node, and after this node we put the tree with root j ;
- (4) the process in (3a) and (3b) is iterated if new leaves $u_{j,0}$ are thus added to trees with roots in \tilde{K} ; a finite number of iterations are undertaken until each such leaf has been replaced by a node;
- (5) for each $i \in \tilde{K}$, we relabel the root i by j , and the leaves $u_{i,t}$ by $u_{j,t}$, where $w_j = i$; the tree j in G_0 is the tree with root j , $j \in K$.

- (6) for each i such that the tree i has one single edge with label v , we replace w_i by w_iv , the label v by an empty label, and each label v' leading to a leaf $u_{i,t}$ by $v'v$.

The initial castle forest G_0 is allowed if the partial lengths of the initial intervals $[w_i]$, which are the $u_{i,t}$ for $i \in K$, $t > 0$, are strictly positive for at least one value of the probability vector of \mathcal{I} . These partial lengths constitute $\sum_{i \in K} a_i$ parameters, which satisfy the train-track equalities. In that case we say *the initial state is allowed*.

For example, for $1 \rightarrow 4$, $2 \rightarrow 3$, $3 \rightarrow 1$, $4 \rightarrow 2$, there are 20 initial states, among which we found 5 are not allowed, namely $\gamma_1 < \gamma_2 < \beta_1 < \beta_2 < \beta_3 < \gamma_3$, $\gamma_1 < \gamma_2 < \beta_1 < \beta_2 < \gamma_3 < \beta_3$, $\gamma_1 < \beta_1 < \gamma_2 < \gamma_3 < \beta_2 < \beta_3$, $\beta_1 < \beta_2 < \beta_3 < \gamma_1 < \gamma_2 < \gamma_3$, $\beta_1 < \gamma_1 < \gamma_2 < \gamma_3 < \beta_2 < \beta_3$. The castle forest for the first one would have only one tree, numbered 1, with a single edge to a leaf u_1 , with similar impossibilities for the others.

3.4. The general induction step and structure theorem

The greedy decision H is the set of i such that the root of tree i is a node; for i in H , $U_{i,t}$ is the sum of the labels of the leaves which are after the t -th edge (from the left) out of the root labeled w_i . An allowed decision F is again a nonempty subset of H .

A choice c associates to each $i \in K$ a c_i -uple of numbers $t_1(i) < \dots < t_{c_i}(i)$, where the previous notations $c(i) = -$, resp $c(i) = +$, correspond to the case where $c(i)$ consists only of $t_1(i) = 1$, resp. $t_1(i) = d_i$. Namely, the choice $c(i)$ is the set of all $1 \leq t \leq d_i$ such that $E_{i,D,t}$ contains at least one β_j , or equivalently such that

$$\sum_{s=1}^{t-1} U_{i,s} < \sum_{y=1}^q u_{i,y} < \sum_{s=1}^t U_{i,s}$$

for at least one q . Then, if t is not in $c(i)$, $E_{i,D,t}$ is included in one $E_{i,A,s}$ for $s = s(i,t)$ satisfying $\sum_{y=1}^{s-1} u_{i,y} < \sum_{q=1}^t U_{i,q} < \sum_{y=1}^s u_{i,y}$; if t is in $c(i)$, $E_{i,D,t}$ intersects all the $E_{i,A,s}$ for $s_-(i,t) \leq s \leq s_+(i,t)$, with $\sum_{y=1}^{s_- - 1} u_{i,y} < \sum_{q=1}^{t-1} U_{i,q} < \sum_{y=1}^{s_-} u_{i,y}$, $\sum_{y=1}^{s_+ - 1} u_{i,y} < \sum_{q=1}^t U_{i,q} < \sum_{y=1}^{s_+} u_{i,y}$.

Note that the values of $s(t)$, $s_-(t)$, $s_+(t)$ can be deduced from the knowledge of the set K and the choice c . For example, in Figure 3.1 above $c(i) = \{2, 4\}$, $s(i, 1) = 1$, $s_-(i, 2) = 1$, $s_+(i, 2) = 3$, $s(i, 3) = 3$, $s_-(i, 4) = 3$, $s_+(i, 4) = 4$, $s(i, 5) = 4$.

For i in F , we create a new interval E' for each t in $c(i)$, and there may be more than one: namely, $E_{i,D,t}$ becomes $E'_{j(i,t)}$, where $j(i,t)$ is the

smallest j such that β_j is in $E_{i,D,t}$. $E'_i = E_i$ if i is not in F . If i is in F and $t \in c(i)$, then for $j = j(i, t)$ the partial lengths of E'_j are given by $l'_j = \sum_{y=1}^{j(i,t)} u_{i,y} - \sum_{s=1}^{t-1} U_{i,s}$, $r'_j = \sum_{s=1}^t U_{i,s} - \sum_{y=1}^{j_+(i,t)} u_{i,y}$, $u'_{j,q} = u_{i,j(i,t)+q-1}$ for $2 \leq q \leq a'_j - 1$ with $a'_j = j_+(i, t) - j(i, t) + 2$. Thus each choice and decision define a set of integers K' numbering the new intervals, and a linear map C_F from \mathbb{R}^K to $\mathbb{R}^{K'}$; an allowed choice is defined by C_H as before.

For example, in Figure 3.1 above, if i is in F , then $j(i, 2) = i$, $j(i, 4) = i + 2$, the two new intervals are E'_i , of length $U_{i,2}$, and E'_{i+2} , of length $U_{i,4}$.

The castle forest of the E'_i built with an allowed decision F is built from the castle forest of the E_i by making for all i in F , successively in any order on i , the surgery on trees detailed below;

- for each t in $c(i)$, let n denote the node or leaf at the end of the t -th edge (from the left) issuing from the root of tree i ; if n is a node, the part of the former tree i beyond n is cut away and forms a new tree $j(i, t)$, including n which becomes its root; if n is a leaf labeled $u_{h,s}$, a new tree $j(i, t)$ is made with a single edge and a leaf labeled $u_{h,s}$;
- if t is in $c(i)$ and $t > 1$, the t -th edge out of the root of the former tree i , including n , is cut away and put on top of the former leaf $u_{i,s_-(i,t)}$, which is in a tree h (if h is one of the $j(i, t)$, the new tree $j(i, t)$); n (in the added part) becomes a leaf labeled $l_{j(i,t)}$; the former leaf $u_{i,s_-(i,t)}$ becomes a node; if it was the only leaf in tree h , then the single edge leading to it is deleted, and the new node becomes the root;
- if t is in $c(i)$ and $t < d_i$, the t -th edge out of the root of the former tree i , including n , is cut away if $t = 1$, resp. duplicated if $t > 1$, and it, resp. its new copy, is put on top of the former leaf $u_{i,s_+(i,t)}$, which is in a tree h' (if h' is one of the $j(i, t)$, the new tree $j(i, t)$); n (in the added part) becomes a leaf labeled $r_{j(i,t)}$; the former leaf $u_{i,s_+(i,t)}$ becomes a node; if it was the only leaf in tree h' , then the single edge leading to it is deleted, and the new node becomes the root;
- if t is not in $c(i)$, the t -th edge out of the root of the former tree i and everything beyond it, are cut away and put on top of the former leaf $u_{i,s(i,t)}$.

As for the labels, for all i in F ,

- if t is in $c(i)$, w_i is extended to the right by the label v of the t -th edge from the root of tree i , and renamed $w_{j(i,t)}$; every w_h which contains the former w_i as a strict suffix is extended to the right by the label v of the t -th edge from the root of tree i ; the labels of the edges leading to the leaves $u_{i,s}$ for $s_-(i, t) < s < s_+(i, t)$, also $s = s_-(i, t)$ if $t = 1$,

- $s = s_+(i, t)$ if $t = d_i$, and of every edge leading to a leaf $u_{h,s}$ where w_h contains the former w_i a strict suffix, are extended to the right by v ;
- when an edge is displaced or duplicated, it, and its new copy when it exists, keep the same label, possibly extended as in the previous item;
- the leaf r_i is relabeled r_j , where j is the number of the interval E' which contains β_{i+a_i-2} ; all other labels of leaves l_h, r_h, u_h which appear in the castle forest at that stage are unchanged, which determines the new numbers j and q for which $u_h = u_{j,q}$.

Evolution of Example 4. — We suppose $r_2 < l_2$ and $r_1 < l_1$, Then the interval $[\mathcal{I}^{-1}\gamma_1, \mathcal{I}^{-1}\gamma_2[$ contains β_2 and the interval $[\mathcal{I}^{-1}\gamma_2, 1[$ contains β_3 . We take the decision $\{1, 2\}$. We make first the surgery determined by 1, with $c(1) = \{2\}$, which agrees also with the rules of Section 2 and $c(i) = +$; this gives the intermediate castle forest in Figure 3.4.

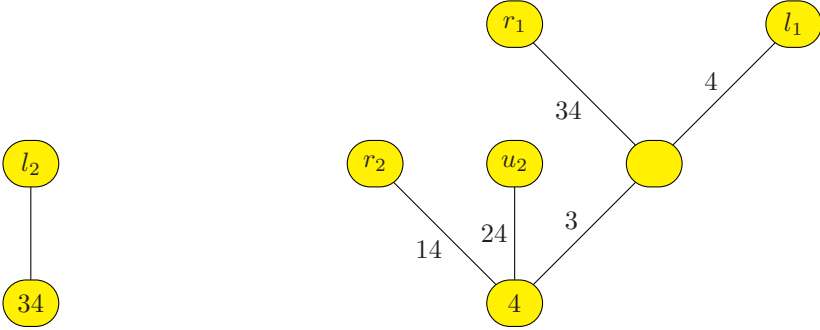


Figure 3.4. Labeled castle forest 4.1.

Then the surgery determined by 2 illustrates the new rules: $c(2) = \{2, 3\}$, both 424 and 43 are left special, they give the new w_2 and w_3 ; a new tree 2 is made with a single edge going to u_2 , and a new tree 3 with the edges going to r_1 and l_1 . Then the former edge going to r_2 is put on top of l_2 (in tree 1) as $s(2, 1) = 1$, and the leaf at its end is renumbered from r_2 to r_3 ; the former edge going to u_2 is put on top of l_2 as $s_-(2, 2) = 1$, and the leaf at its end is renumbered from u_2 to l_2 ; a copy of the former edge going to u_2 is put on top of u_2 (in the new tree 2) as $s_+(2, 2) = 2$, and the leaf at its end is renumbered from u_2 to r_2 ; the former edge going to the node is put on top of u_2 as $s_-(2, 3) = 2$, and receives a leaf l_3 .

The new castle forest is in Figure 3.5. The new train-track equalities are $l_1 + r_1 = r_3 + l_2$, $l_2 + r_2 = r_2 + l_3$, $l_3 + r_3 = r_1 + l_1$; these correspond to a

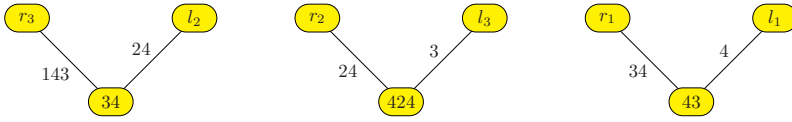


Figure 3.5. Labeled castle forest 4.2.

castle forest associated to the tree of relations $1 \hat{=} 3 \hat{+} 2$, and thus to a castle forest in a graph of graphs of the symmetric permutation in the case of alternate discontinuities.

Evolution of Example 5. — We suppose first $r_2 < l_2 < u_2$, $l_1 < r_1$, and take the decision $\{1, 2\}$. Then $c(2) = \{1, 3\}$, 414 and 43 are left special, 43 being then uniquely extended to 433, and we denote them by w_2 and w_3 . The new castle forest is in Figure 3.6. The new train-track equalities are $l_1 + r_1 = r_1 + l_2$, $l_2 + r_2 = r_2 + r_3 + l_3$, $l_3 + r_3 = l_1$. Note that here a tree 3 is added, thus a leaf u is replaced by two leaves l and r , but no edge is duplicated because the choices contain only extremal values (1 or d_i), thus the number of edges with nonempty labels becomes smaller than the number of leaves (an edge will be duplicated at a later stage).

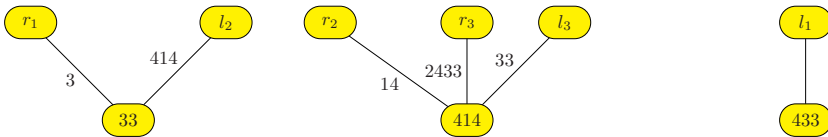


Figure 3.6. Labeled castle forest 5.1.a.

We go back to the initial state of Example 5, suppose $l_2 < u_2$, $l_2 < r_2$, $l_1 < r_1$, and take the decision $\{1, 2\}$. Then $c(2) = \{1, 2\}$, 414 and 424 are left special, and we denote them by w_2 and w_3 . The new castle forest is in Figure 3.7. The new train-track equalities are $l_1 + r_1 = r_1 + l_2$, $l_2 + r_2 = r_2 + l_3$, $l_3 + r_3 = r_3 + l_1$; these correspond to a castle forest in Figure 2.15, in a graph of graphs of the rotations class in the case of alternate discontinuities. There has been one duplicated edge, leading to leaves l_3 and r_3 .

Proposition 2.24 is still valid at any stage with, for the number of towers, $2k - 2$ replaced by the number of edges with nonempty labels, which is at most $\sum_{i \in K} a_i$ (but can be less as in castle forest 5.1.a above) and, as will

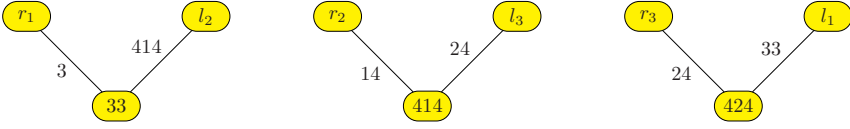


Figure 3.7. Labeled castle forest 5.1.b.

be shown below, is indeed $2k - 2$ ultimately. Then Proposition 2.25, Corollary 2.26, Theorem 2.28 are still valid in the general case (the condition that G_0 is allowed is nontrivial here, while it was always satisfied in the case of alternate discontinuities). Moreover, as in Proposition 2.25 the lengths of the $E_{i,n}$ tend to zero, by the remark at the end of Section 3.1, after a finite number of stages the induction families and castle forests satisfy all the properties of Sections 2.2, 2.3, 2.4 with the sole exception that the first letter of w_i is not i in general. In Proposition 2.27, the second condition is not changed while the first is replaced by: if (j_1, \dots, j_s) is the unique sequence such that $j_1 = i$, the tree root j_h has a single edge going to a leaf $x_{h+1} = u_{j_{h+1}, t_{h+1}}$ for $1 \leq h \leq s - 1$, and the root of tree j_s is a node, then, for $2 \leq h \leq s$, at the first ulterior stage for which j_h is in F , the leaf x_h is replaced by a node.

The greedy algorithm is still available; a (generalized) trees of relation algorithm is described completely in the little (if ever)-read Section 3 of [20], and the loops variant is carried naturally to the general case.

4. Applications of the new induction

4.1. Substitutions and S-adic presentation

Our algorithm of induction allows us to generalize Section 3.3 of [20]. We recall

DEFINITION 4.1. — A substitution τ is an application from an alphabet \mathcal{A} into the set \mathcal{A}^* of finite words on \mathcal{A} ; it extends naturally to a morphism of \mathcal{A}^* for the operation of concatenation. It is primitive if there exists k such that a occurs in $\tau^k b$ for any $a \in \mathcal{A}$, $b \in \mathcal{A}$. A fixed point of τ is an infinite sequence u with $\tau u = u$.

A sequence x in $\mathcal{A}^{\mathbb{N}}$ is primitive substitutive if $x = \psi(u)$, where u is a fixed point of a primitive substitution on an alphabet \mathcal{A}_0 , and ψ a map from \mathcal{A}_0 to \mathcal{A}^* , extended to a morphism for concatenation. A symbolic

system X_L is a primitive substitutive system if $L = L(x)$ for a primitive substitutive sequence x .

An immediate consequence of Proposition 2.24 is that we can generate all factors of the trajectories by the names of the Rokhlin towers, thus by the $2k - 2$ (possibly less in the initial stages) nonempty labels of the edges in the castle forests at stage n ; more precisely, by minimality, the symbolic system $X_{L(\mathcal{I})}$ is the shift on all sequences x such that for every $s < t$ there exists n such that $x_s \cdots x_t$ is a factor of any such label $L_{i,n}$ at some stage n . By Proposition 2.22 or its generalization in Section 3, the labels at stage $n + 1$ are given as concatenations of the labels at stage n , of the form $L_{i,n+1} = L_{j_1(i,n),n} \cdots L_{j_r(i,n),n}$ and, if we define the substitutions τ_n by $\tau_n(i) = j_1(i,n) \cdots j_r(i,n)$, then we check the well-known equality (see [13]) $L_{i,n} = \psi \circ \tau_0 \circ \cdots \circ \tau_{n-1}(i)$ where $\psi(j) = L_{j,0}$. This constitutes a presentation of $X_{L(\mathcal{I})}$ as an *adic system*, see for example [40]; we can also say that it is generated by the substitutions τ_n ; as by Proposition 2.22 τ_n of any symbol is a word of length at most k , they form a finite family of substitutions, and thus define $X_{L(\mathcal{I})}$ as an *S-adic system*, see [13]. And, exactly as in [20], we can show for the new induction the following result, which is “in the folklore” and holds if we replace our induction algorithm by any classical one, but we state it for sake of completeness.

PROPOSITION 4.2. — *Let \mathcal{I} be a k -interval exchange, satisfying the i.d.o.c. condition; then the symbolic system $X_{L(\mathcal{I})}$ is a primitive substitutive system if and only if any sequence of allowed castle forests, choices and decisions generated by an algorithm of induction is ultimately periodic.*

4.2. Repetitions of words

The following result is a partial answer to a question of Boshernitzan to the author (2008), which got the same positive answer in [20] for the hyperelliptic class, for any number of intervals and any initial condition. This questions interests word combinatorists, but also arithmeticians as it gives important informations on the approximation by rationals of the numbers whose either the expansion in base k or the continued fraction expansion is a trajectory of \mathcal{I} , see for example [1].

PROPOSITION 4.3. — *The language of any 4-interval exchange, satisfying the i.d.o.c. condition and the condition of alternate discontinuities, contains the word w for infinitely many different words w .*

Proof. — We have just to show it for the rotations class. In any castle forest, if v is the label of a path from a root w_i to a leaf l_i or r_i , we have $w_i v = v' w_i$; we check inductively that v is not longer than w_i , except at a finite number of initial stages, thus $w_i = v'' v$, and vv is in the language of \mathcal{I} . Thus we need only to prove that, for one algorithm of induction, any possible sequence of formal castle forests goes infinitely many times through castle forests with at least one path from a root w_i to a leaf l_i or r_i . Now, for the algorithm of Section 2.8, we check that the only non-transient castle forests without this property are the four of the central ones in figure 16 which have no loops around them, the 2_i and the 7_i , and that the allowed moves make it impossible to stay ultimately among these ones. \square

To eliminate the condition of alternate discontinuities, we need to show that, starting from any initial condition as defined in Section 3.3, the possible non-transient castle forests for $k = 4$ and π in the rotations class are all in the graph of graphs \mathcal{G}' of Section 2.8, possibly after a renumbering of the w_i . This is a particular case of the relations between graphs of graphs and Rauzy classes mentioned in Section 5: it could be proved combinatorially by looking at the 108 remaining initial states (this number can be reduced to 54 by using the map $x \rightarrow 1 - x$, and some of them are not allowed) and the way they can evolve until we get a castle forest where each node has at most two issuing edges, but this would take up too much space for not enough interest.

4.3. Weak mixing in an odd component

We recall that (X, T, μ) is weakly mixing if μ is ergodic and the operator $f \circ T$ in $\mathcal{L}^2(X, \mathbb{R}/\mathbb{Z})$ has no nonzero *eigenvalue* (denoted additively, $f \circ T = f + \zeta$). Avila and Forni [3] have proved that, for every given π not satisfying $i \equiv \pi i \pmod{k}$, for all $1 \leq i \leq k$, almost every k -interval exchange is weakly mixing; but, as far as we know, explicit constructions (as opposed to existence theorems) of weakly mixing k -interval exchanges have been made only for the symmetric permutation: the only ones we have been able to find in the literature are for $k = 3$ [26][19], $k = 4$ [23][37], and $k = 6$ [37], while in Theorem 13 of [15] we give a construction for every value of k , though it should be modified as indicated after the proof of Theorem 4.4 below. It would not be difficult to build weak mixing examples in the rotations class, provided our interval exchange is the induced map of a rotation but not a rotation itself: indeed for $k = 3$ the rotations class and the hyperelliptic class are the same, while for $1 \rightarrow 4, 2 \rightarrow 3, 3 \rightarrow 1, 4 \rightarrow 2$ we could build examples using the method below and the analysis in Section 2.8.

We shall now build a family of weakly mixing interval exchanges outside both the hyperelliptic and the rotations Rauzy classes, by choosing in [44] a permutation in the Rauzy class corresponding to the connected component with odd spin parity in the stratum $\mathcal{H}(2g - 2)$, for any given $g \geq 3$, and applying the general self-dual induction for this permutation. The idea of the proof is the same as in [15]: we kill any possible eigenvalue ζ by the usual *Chacon trick*, i.e. we find an integer Q such that both $Q\zeta$ and $(Q+1)\zeta$ are close to an integer, as in [6], see also [16]; this is achieved by ensuring first that the heights of two of the Rokhlin towers are coprime, then that these towers are cycled for a large enough number of times to allow the use of Bezout's relation; we conclude by checking that these two towers appear at least as a fixed proportion of all the towers at a further stage.

Following [44], we build a permutation π in $\mathcal{H}_{\text{odd}}(2g - 2)$. It is defined on $\{1, \dots, 2g\}$ by

- $\pi i = 2g + 1 - i$ for $i = 1, 2, \dots, 2g$,
- $\pi(2g - 1) = 2g - 2$,
- $\pi(2j + 1) = 2j + 1$ for $2 < 2j + 1 < 2g - 1$,
- $\pi(2j) = 2j - 2$ for $2 < 2j < 2g - 1$.

We suppose the condition of alternate discontinuities is satisfied, and start from the initial castle forest G_0 defined in Sections 2.3 and 2.4. After labeling, it consists in

- an edge from the root w_i to the leaf l_i for $i = 1, 3, 5, \dots, 2g - 3$;
- an edge from the root w_i to the leaf l_{i+2} for $i = 2, 4, 6, \dots, 2g - 4$;
- an edge from the root w_{2g-2} to the leaf l_{2g-1} ;
- an edge from the root w_{2g-1} to the leaf l_2 ;
- an edge from the root w_i to the leaf r_{i-1} for $i = 3, 5, \dots, 2g - 3$;
- an edge from the root w_i to the leaf r_{i+1} for $i = 2, 4, \dots, 2g - 4$;
- an edge from the root w_{2g-2} to the leaf r_{2g-2} ;
- an edge from the root w_{2g-1} to the leaf r_1 ;
- an edge from the root w_1 to the leaf r_{2g-1} .

For any castle forest where each tree has two edges, we denote by $P_{i,j}$, resp. $M_{i,j}$, the label of the edge from the root w_i to the leaf l_j , resp. r_j . Starting from such a castle forest, if we take the greedy decisions and $c(i)$ is $+$ for all i or $-$ for all i , the choice is allowed and in the new castle forest each tree has two edges. In such particular cases, when all trees have two edges, both in the former and the new castle forest, with our numbering of the edges, the rules of Proposition 2.22 have a simple expression: if $c(i) : -$, the unique label $M_{g,i}$ is not changed, while the unique label $P_{h,i}$ is extended to the right by the unique label $M_{i,j}$, giving a label which is the new $P_{h,j}$,

and mutatis mutandis if $c(i) = +$. Moreover, if in the initial castle forest there is a circuit of M edges, namely $M_{j_1, j_2}, M_{j_2, j_3}, \dots, M_{j_r, j_1}$, and if we make r consecutive inductions, with the greedy decisions and $c(i) = -$ for all i , then the unique label M_{g, j_1} is not changed, while the unique label P_{h, j_1} is extended to the right by the concatenation $M_{j_1, j_2} M_{j_2, j_3}, \dots, M_{j_r, j_1}$, giving a label which is the new P_{h, j_1} , and mutatis mutandis if we exchange P and M , $-$ and $+$.

In our initial castle forest G_0 , we have $P_{1,1} = (2g)1$, $P_{i,j} = j$ for all possible $(i, j) \neq (1, 1)$, $M_{i,j} = j$ for all possible (i, j) . We partition the edges into circuits of M edges and circuits of P edges: these are (up to circular permutations inside each circuit) $M_{2i+1, 2i}$, $M_{2i, 2i+1}$, for $3 \leq 2i+1 \leq 2g-3$; $M_{1, 2g-1}$, $M_{2g-1, 1}$; $M_{2g-2, 2g-2}$; $P_{2g-1, 2}$, $P_{2, 4}, \dots, P_{2g-4, 2g-2}$, $P_{2g-2, 2g-1}$; $P_{2i+1, 2i+1}$ for $1 \leq 2i+1 \leq 2g-3$. The M circuits have lengths 1 and 2, the P circuits have lengths 1 and g ; thus the discussion above implies that, if we start from G_0 and take the greedy decision at each stage,

- g consecutive choices $c(i) = +$, $1 \leq i \leq 2g-1$, are allowed and then, at successive stages, we go to castle forests $G_1, \dots, G_g = G_0$, where each tree has two edges;
- 2 consecutive choices $c(i) = -$, $1 \leq i \leq 2g-1$, are allowed and then, at successive stages, we go to castle forests $\tilde{G}_1, \tilde{G}_2 = G_0$, where each tree has two edges.

THEOREM 4.4. — *For any $g \geq 3$, one can construct recursively two sequences p_n and q_n such that the $2g$ -interval exchange with permutation π , defined as in Theorem 2.28 by the sequence of castle forests, choices, and decisions, made, from the initial stage, by successive runs of $2p_n$ choices $c(i) = -$, $1 \leq i \leq 2g-1$, gq_n choices $c(i) = +$, $1 \leq i \leq 2g-1$, for $n = 1, 2, \dots$, and a greedy decision at each stage, is weakly mixing for any invariant probability measure.*

Proof. — The p_n and q_n are chosen inductively.

At the beginning of a run of $2p_n$ negative choices, we have castle forest G_0 , with edges labeled $M_{i,j}$ and $P_{i,j}$ for the couples (i, j) described above. By the discussion before Theorem 4.4, the M edges do not change, while the P edge ending at i is extended by the concatenated labels of the edges forming the M circuit beginning at i , as many times as necessary; we get that at the beginning of the next run of gq_n positive choices, we have G_0 again with edges labeled $M_{i,j}$ and $P'_{i,j}$ with

$$\begin{aligned} P'_{2i+1, 2i+1} &= P_{2i+1, 2i+1} (M_{2i+1, 2i} M_{2i, 2i+1})^{p_n}, & 3 \leq 2i+1 \leq 2g-3, \\ P'_{2i, 2i+2} &= P_{2i, 2i+2} (M_{2i+2, 2i+3} M_{2i+3, 2i+2})^{p_n}, & 2 \leq 2i \leq 2g-6, \end{aligned}$$

$$\begin{aligned}
 P'_{1,1} &= P_{1,1}(M_{1,2g-1}M_{2g-1,1})^{p_n}, \\
 P'_{2g-1,2} &= P_{2g-1,2}(M_{2,3}M_{3,2})^{p_n}, \\
 P'_{2g-2,2g-1} &= P_{2g-2,2g-1}(M_{2g-1,1}M_{1,2g-1})^{p_n}, \\
 P'_{2g-4,2g-2} &= P_{2g-4,2g-2}M_{2g-2,2g-2}^{2p_n}.
 \end{aligned}$$

By a similar reasoning, at the beginning of the run of $2p_{n+1}$ negative choices, we have castle forest G_0 , with edges labeled $M'_{i,j}$ and $P'_{i,j}$ with

$$M'_{j,2i} = M_{j,2i}(P'_{2i,2i+2}P'_{2i+2,2i+4} \cdots P'_{2g-2,2g-1}P'_{2g-1,2}P'_{2,4} \cdots P'_{2i-2,2i})^{q_n}$$

for $2 \leq 2i \leq 2g - 2$ and corresponding j ,

$$\begin{aligned}
 M'_{1,2g-1} &= M_{1,2g-1}(P'_{2g-1,2}P'_{2,4} \cdots P'_{2g-4,2g-2} \cdot P'_{2g-2,2g-1})^{q_n}, \\
 M'_{j,2i+1} &= M_{j,2i+1}(P'_{2i+1,2i+1})^{gq_n}
 \end{aligned}$$

for $1 \leq 2i + 1 \leq 2g - 3$ and corresponding j .

Let x_n , resp. y_n , resp. X_n , resp. Y_n , be the length of the word $P_{1,1}$, resp. $P_{3,3}$, resp. $M_{1,2g-1}M_{2g-1,1}$, resp. $M_{3,2}M_{2,3}$, at the beginning of the run of $2p_n$ negative choices. We make the recursion hypotheses that x_n and y_n are coprime and $Y_n x_n - X_n y_n \neq 0$. The hypothesis is satisfied at the initial stage where $x_1 = 2, y_1 = 1, X_1 = Y_1 = 2$; we suppose now that it is satisfied for n , and shall choose p_n and q_n such that they will be satisfied for $n + 1$.

Namely, $x_{n+1} = x_n + p_n X_n, y_{n+1} = y_n + p_n Y_n$. Any common factor of x_{n+1} and y_{n+1} has to divide $Y_n x_{n+1} - X_n y_{n+1} = Y_n x_n - X_n y_n = Z_n \neq 0$, which is independent of p_n . Let \mathcal{D} be the set of all prime factors of Z_n , \mathcal{D}_1 the set of those factors which divide also x_n , \mathcal{D}_2 the set of the other factors. If d is in \mathcal{D}_2 and divides X_n , any choice of p_n ensures that d does not divide x_{n+1} ; if d is in \mathcal{D}_2 and does not divide X_n , d does not divide x_{n+1} for any p_n such that $p_n \equiv X_n^{-1}(z - x_n)$ modulo d , for any $z \not\equiv 0$ modulo d . Similarly if d is in \mathcal{D}_1 , and therefore does not divide y_n , either d does not divide y_{n+1} for any value of p_n , or this can be ensured by a congruence condition modulo d . Thus, by the Chinese remainder theorem, we can find infinitely many values of p_n such that no prime number d divides the three numbers Z_n, x_{n+1} and y_{n+1} , and this ensures that x_{n+1} and y_{n+1} are coprime. We also ask that p_n is large enough to have $|P_{i,j}| < \epsilon_n |P'_{i,j}|$ for all pairs (i, j) in the above recursion formulas, for some prescribed ϵ_n . Note that p_n depends only on the parameters $p_1, \dots, p_{n-1}, q_1, \dots, q_{n-1}$.

Thus for any n there exist positive integers U_n and V_n such that $|U_n x_n - V_n y_n| = 1$. As the value of x_{n+1} and y_{n+1} depend only on the parameters $p_1 \dots, p_n, q_1, \dots, q_{n-1}$, we can then choose q_n larger than $U_{n+1} \vee V_{n+1}$. We

also ask that q_n is large enough to have $|M_{i,j}| < \epsilon_n |M'_{i,j}|$ for all pairs (i, j) in the above recursion formulas. We finally ask that q_n is large enough to ensure that $Z_{n+1} \neq 0$: indeed, by construction $Z_{n+1} = Z_n + q_n \Xi_n$ for a Ξ_n independent of q_n , thus if $\Xi_n = 0$ any choice of q_n is good, and otherwise $q_n |\Xi_n| > |Z_n|$ is sufficient. We shall now prove that, with this choice of the p_n and q_n , \mathcal{I} is weakly mixing.

By Proposition 2.24, at each stage the space is filled by $4g - 2$ Rokhlin towers, whose names are the $M_{i,j}$ and $P_{i,j}$. The towers are made by cutting and stacking following the recursion rules above, thus in the tower with name $M'_{2g-1,1}$ at the beginning of the run of $2p_{n+1}$ negative choices, we see $(P'_{1,1})^{gq_n}$, when we read the name from level $|M_{2g-1,1}|$ to level $|M_{2g-1,1}| + gq_n |P'_{1,1}| - 1$. Let τ_n be the union of all these levels. For any point ω in τ_n , $\mathcal{I}^{x_n}\omega, \mathcal{I}^{2x_n+1}\omega, \dots, \mathcal{I}^{U_{n+1}x_{n+1}}\omega$ are in the same level of the tower with name $P'_{1,1}$ as ω . Similarly, in the name of $M'_{2,3}$ we see $(P'_{3,3})^{gq_n}$ from level $|M_{2,3}|$ to level $|M_{2,3}| + gq_n |P'_{3,3}| - 1$ let τ'_n be the union of all these levels. For any point ω in τ'_n , $\mathcal{I}^{y_{n+1}}\omega, \mathcal{I}^{2y_{n+1}}\omega, \dots, \mathcal{I}^{V_{n+1}y_{n+1}}\omega$ are in the same level of the tower with name $P'_{3,3}$ as ω .

Let μ be an invariant probability for \mathcal{I} , f be an eigenfunction for the eigenvalue ζ ; by Corollary 2.26, for each $\varepsilon > 0$ there exists $N(\varepsilon)$ such that for all $n > N(\varepsilon)$ there exists f_n , which satisfies $\int \|f - f_n\| d\mu < \varepsilon$ and is constant on each level of each tower at the beginning of the run of $2p_n$ negative choices (where $\|x\|$ denotes its distance to the nearest integer).

Thus for μ -almost every ω in τ_n , $f_n(\mathcal{I}^{U_{n+1}x_{n+1}}\omega) = f_n(\omega)$ while

$$f(\mathcal{I}^{U_{n+1}x_{n+1}}\omega) = \zeta U_{n+1}x_{n+1} + f(\omega);$$

we have

$$\begin{aligned} \int_{\tau_n} \|f_n \circ \mathcal{I}^{U_{n+1}x_{n+1}} - \zeta U_{n+1}x_{n+1} - f_n\| d\mu \\ = \int_{\tau_n} \|\zeta U_{n+1}x_{n+1}\| d\mu = \|\zeta U_{n+1}x_{n+1}\| \mu(\tau_n) \end{aligned}$$

and

$$\begin{aligned} \int_{\tau_n} \|f_n \circ \mathcal{I}^{U_{n+1}x_{n+1}} - \zeta U_{n+1}x_{n+1} - f_n\| d\mu \\ \leq \int_{\tau_n} \|f_n \circ \mathcal{I}^{U_{n+1}x_{n+1}} - f \circ \mathcal{I}^{U_{n+1}x_{n+1}}\| d\mu + \int_{\tau_n} \|f_n - f\| d\mu < 2\varepsilon. \end{aligned}$$

Thus $\mu(\tau_n) \|\zeta U_{n+1}x_{n+1}\| < 2\varepsilon$, and similarly $\mu(\tau'_n) \|\zeta V_{n+1}y_{n+1}\| < 2\varepsilon$; as $U_{n+1}x_{n+1} - V_{n+1}y_{n+1} = \pm 1$, we shall conclude that $\zeta = 0$, and thus get

the weak mixing, if we can prove that $\mu(\tau_n)$ and $\mu(\tau'_n)$ are bounded away from 0.

For this, we need first to check that all the lengths of the $M_{i,j}$, resp. $P_{i,j}$, are comparable at the beginning of the same run of positive or negative choices; at the initial stage they are equal to 1 except $|P_{1,1}| = 2$, and, because the recursion formulas have all the same number of terms, at all subsequent stages the ϵ_n can be chosen such that $c_1|P_{i,j}| \leq |P_{i',j'}| \leq c_2|P_{i,j}|$, $c_1|M_{i,j}| \leq |M_{i',j'}| \leq c_2|M_{i,j}|$ at the beginning of the same run for all i, j, i', j' .

Thus the strings $(P'_{1,1})^{gq^n}$ fill (about) all the length of $M_{2g-1,1}$ after the next runs, thus (because of the recursion formulas and the comparability of the lengths of the $P_{i,j}$) at least a fixed proportion κ of the lengths of $P_{1,1}$ and $P_{2g-2,2g-1}$ after the next runs, thus, in the same way, a proportion at least $\kappa\kappa'$ of the lengths of $M_{2g-1,1}$, $M_{1,2g-1}$ and all the $M_{j,2i}$ after the next runs, thus a proportion at least $\kappa^2\kappa'$ of the lengths of all the $P_{i,j}$ after the next runs, thus a proportion at least $\kappa^2(\kappa')^2$ of the lengths of all the $M_{i,j}$ and $P_{i,j}$ after the next runs, and these are the names of $4g - 2$ Rokhlin towers filling all the space. This implies that $\mu(\tau_n) \geq \kappa^2(\kappa')^2$, and a similar reasoning works for $\mu(\tau'_n)$. \square

We apologize to the readers of [15], as in its Theorem 13 mentioned above there is a gap in the published proof: with its notations, which are essentially the same as in the proof of Theorem 4.4 above, not only we failed to ensure that Z is nonzero, but indeed it is 0 at the first stage. The shortest way to correct that is to change the pair of names whose lengths will be coprime: in the case when $n = 2p + 1$, these should not be $P_{k,p+1,p+1}$ and $P_{k,p,p+2}P_{k,p+2,p}$, but $P_{k,p+1,p+1}$ and $P_{k,1,1}$, then Z is nonzero at the first stage and the proof of Theorem 4.4 above can be carried out, mutatis mutandis; when $n = 2p$ we can take $P_{k,1,1}$ and (among others) $P_{k,p,p+1}P_{k,p+1,p}$, but we have to weaken the recursion hypothesis, replacing the two coprime lengths by two lengths which have no common divisor except 2; an imitation of the proof of Theorem 4.4, ensuring this is satisfied, implies that the system has no eigenvalue except maybe $\zeta = \frac{1}{2}$; then the fact that the name $M_{k,p,p}$ has always an odd length and is a return word of a single word completes the proof of weak mixing.

5. Open questions

We would like to know under which conditions a given allowed castle forest may be the castle forest of an induction family for some \mathcal{I} ; this depends

on the permutation, but also the initial state as defined in Section 3.3, the algorithm used, see Section 2.7, and whether we allow it to be transient as in some examples in Section 2.8; we do not know a general answer to the question, the castle forests we do use are those which can be reached by an induction algorithm from some well-defined initial castle forests.

A related problem is to find an algorithm allowing us, as is the case for the hyperelliptic class, to use only castle forests where each tree has two edges, possibly with some additional transient castle forests. We have not been successful in the general case: indeed, with the algorithm of Section 2.8 it is easy to build interval exchanges in the rotations class which define a sequence of castle forests where ultimately there is always at least one tree with more than two edges.

For $k \geq 5$, Boshernitzan's question on words ww remains open outside the hyperelliptic class.

The relations between graphs of graphs and Rauzy classes, examples of which appear in Section 2.8, are tackled in [12].

BIBLIOGRAPHY

- [1] B. ADAMCZEWSKI & Y. BUGEAUD, "Transcendence and Diophantine approximation", in *Combinatorics, automata and number theory*, Encyclopedia Math. Appl., vol. 135, Cambridge Univ. Press, Cambridge, 2010, p. 410-451.
- [2] V. I. ARNOL'D, "Small denominators and problems of stability of motion in classical and celestial mechanics", *Uspehi Mat. Nauk* **18** (1963), no. 6 (114), p. 91-192.
- [3] A. AVILA & G. FORNI, "Weak mixing for interval exchange transformations and translation flows", *Ann. of Math. (2)* **165** (2007), no. 2, p. 637-664.
- [4] M. BOSHERNITZAN, "A unique ergodicity of minimal symbolic flows with linear block growth", *J. Analyse Math.* **44** (1984/85), p. 77-96.
- [5] ———, "A condition for minimal interval exchange maps to be uniquely ergodic", *Duke Math. J.* **52** (1985), no. 3, p. 723-752.
- [6] ———, "A condition for weak mixing of induced IETs", in *Dynamical systems and group actions*, Contemp. Math., vol. 567, Amer. Math. Soc., Providence, RI, 2012, p. 53-65.
- [7] J. BOURGAIN, "On the correlation of the Moebius function with rank-one systems", *J. Anal. Math.* **120** (2013), p. 105-130.
- [8] S. D. CRUZ & L. F. C. DA ROCHA, "A generalization of the Gauss map and some classical theorems on continued fractions", *Nonlinearity* **18** (2005), no. 2, p. 505-525.
- [9] V. DELECROIX, "Cardinality of Rauzy classes", *Ann. Inst. Fourier (Grenoble)* **63** (2013), no. 5, p. 1651-1715.
- [10] ———, "Divergent trajectories in the periodic wind-tree model", *J. Mod. Dyn.* **7** (2013), no. 1, p. 1-29.
- [11] V. DELECROIX & C. ULCIGRAI, "Diagonal changes for surfaces in hyperelliptic components", to appear in *Geometriae Dedicata*, [arXiv:1310.1052](https://arxiv.org/abs/1310.1052).

- [12] S. FERENCZI, “Diagonal changes for every interval exchange transformation”, preprint, <http://iml.univ-mrs.fr/~ferenczi/fid.pdf>.
- [13] ———, “Rank and symbolic complexity”, *Ergodic Theory Dynam. Systems* **16** (1996), no. 4, p. 663-682.
- [14] ———, “Systems of finite rank”, *Colloq. Math.* **73** (1997), no. 1, p. 35-65.
- [15] ———, “Billiards in regular $2n$ -gons and the self-dual induction”, *J. Lond. Math. Soc. (2)* **87** (2013), no. 3, p. 766-784.
- [16] ———, “Combinatorial methods for interval exchange transformations”, *Southeast Asian Bull. Math.* **37** (2013), no. 1, p. 47-66.
- [17] S. FERENCZI, C. HOLTON & L. Q. ZAMBONI, “Structure of three interval exchange transformations. I. An arithmetic study”, *Ann. Inst. Fourier (Grenoble)* **51** (2001), no. 4, p. 861-901.
- [18] ———, “Structure of three-interval exchange transformations. II. A combinatorial description of the trajectories”, *J. Anal. Math.* **89** (2003), p. 239-276.
- [19] ———, “Structure of three-interval exchange transformations III: ergodic and spectral properties”, *J. Anal. Math.* **93** (2004), p. 103-138.
- [20] ———, “Structure of K -interval exchange transformations: induction, trajectories, and distance theorems”, *J. Anal. Math.* **112** (2010), p. 289-328.
- [21] S. FERENCZI & L. F. C. DA ROCHA, “A self-dual induction for three-interval exchange transformations”, *Dyn. Syst.* **24** (2009), no. 3, p. 393-412.
- [22] S. FERENCZI & L. Q. ZAMBONI, “Languages of k -interval exchange transformations”, *Bull. Lond. Math. Soc.* **40** (2008), no. 4, p. 705-714.
- [23] ———, “Eigenvalues and simplicity of interval exchange transformations”, *Ann. Sci. Éc. Norm. Supér. (4)* **44** (2011), no. 3, p. 361-392.
- [24] K. INOUE & H. NAKADA, “On the dual of Rauzy induction”, preprint.
- [25] A. B. KATOK, “Invariant measures of flows on orientable surfaces”, *Dokl. Akad. Nauk SSSR* **211** (1973), p. 775-778.
- [26] A. B. KATOK & A. M. STEPIN, “Approximations in ergodic theory”, *Uspehi Mat. Nauk* **22** (1967), no. 5 (137), p. 81-106, translated in *Russian Math. Surveys* **22** (1967), p. 76-102.
- [27] M. KEANE, “Interval exchange transformations”, *Math. Z.* **141** (1975), p. 25-31.
- [28] ———, “Non-ergodic interval exchange transformations”, *Israel J. Math.* **26** (1977), no. 2, p. 188-196.
- [29] M. KONTSEVICH & A. ZORICH, “Connected components of the moduli spaces of Abelian differentials with prescribed singularities”, *Invent. Math.* **153** (2003), no. 3, p. 631-678.
- [30] A. O. LOPES & L. F. C. DA ROCHA, “Invariant measures for Gauss maps associated with interval exchange maps”, *Indiana Univ. Math. J.* **43** (1994), no. 4, p. 1399-1438.
- [31] S. MARMI, P. MOUSSA & J.-C. YOCCOZ, “The cohomological equation for Roth-type interval exchange maps”, *J. Amer. Math. Soc.* **18** (2005), no. 4, p. 823-872 (electronic).
- [32] H. MASUR, “Interval exchange transformations and measured foliations”, *Ann. of Math. (2)* **115** (1982), no. 1, p. 169-200.
- [33] V. I. OSELEDEC, “The spectrum of ergodic automorphisms”, *Dokl. Akad. Nauk SSSR* **168** (1966), p. 1009-1011.
- [34] R. C. PENNER & J. L. HARER, *Combinatorics of train tracks*, Annals of Mathematics Studies, vol. 125, Princeton University Press, Princeton, NJ, 1992, xii+216 pages.

- [35] G. RAUZY, “Échanges d’intervalles et transformations induites”, *Acta Arith.* **34** (1979), no. 4, p. 315-328.
- [36] F. SCHWEIGER, *Ergodic theory of fibred systems and metric number theory*, Oxford Science Publications, The Clarendon Press, Oxford University Press, New York, 1995, xiv+295 pages.
- [37] Y. G. SINAI & C. ULCIGRAI, “Weak mixing in interval exchange transformations of periodic type”, *Lett. Math. Phys.* **74** (2005), no. 2, p. 111-133.
- [38] W. A. VEECH, “Interval exchange transformations”, *J. Analyse Math.* **33** (1978), p. 222-272.
- [39] ———, “Gauss measures for transformations on the space of interval exchange maps”, *Ann. of Math. (2)* **115** (1982), no. 1, p. 201-242.
- [40] A. M. VERSHIK & A. N. LIVSHITS, “Adic models of ergodic transformations, spectral theory, substitutions, and related topics”, in *Representation theory and dynamical systems*, Adv. Soviet Math., vol. 9, Amer. Math. Soc., Providence, RI, 1992, p. 185-204.
- [41] M. VIANA, “Dynamics of interval exchange maps and Teichmüller flows”, preliminary manuscript available from <http://w3.impa.br/~viana/out/ietf.pdf>.
- [42] J.-C. YOCOZ, “Échanges d’intervalles”, Cours au Collège de France, available from <http://www.college-de-france.fr/site/jean-christophe-yoccoz/>, 2005.
- [43] A. ZORICH, “Finite Gauss measure on the space of interval exchange transformations. Lyapunov exponents”, *Ann. Inst. Fourier (Grenoble)* **46** (1996), no. 2, p. 325-370.
- [44] ———, “Explicit Jenkins-Strebel representatives of all strata of abelian and quadratic differentials”, *J. Mod. Dyn.* **2** (2008), no. 1, p. 139-185.

Manuscrit reçu le 10 janvier 2013,
accepté le 8 juillet 2013.

Sébastien FERENCZI
Institut de Mathématiques de Marseille
CNRS - UMR 7373
Case 907 - 163 av. de Luminy
F13288 Marseille Cedex 9 (France)
ssferenczi@gmail.com