Order
DOI 10.1007/s11083-017-9427-2

A Classification of Countable Lower 1-transitive Linear Orders

Silvia Barbina[1] · Katie Chicot[1]

Received: 8 November 2015 / Accepted: 28 April 2017

Abstract This paper contains a classification of countable lower 1-transitive linear orders. This is the first step in the classification of countable 1-transitive trees given in Chicot and Truss (2009): the notion of lower 1-transitivity generalises that of 1-transitivity for linear orders, and it is essential for the structure theory of 1-transitive trees. The classification is given in terms of *coding trees*, which describe how a linear order is fabricated from simpler pieces using concatenations, lexicographic products and other kinds of construction. We define coding trees and show that a coding tree can be constructed from a lower 1-transitive linear order $(X, \leqslant)$ by examining all the invariant partitions on X. Then we show that a lower 1-transitive linear order can be recovered from a coding tree up to isomorphism.

Keywords Countable linear order · Transitive tree · Lower 1-transitivity · Classification

1 Introduction

This paper extends a body of classification results for countably infinite ordered structures, under various homogeneity assumptions. As background we mention that Morel [7] classified the countable 1-transitive linear orders, of which there are $\aleph_1$, Campero-Arena and Truss [2] extended this classification to *coloured* countable 1-transitive linear orders, and

The results in this paper form part of the second author's PhD thesis at the University of Leeds, which was supported by EPSRC grant EP/H00677X/1.

✉ Katie Chicot
k.m.chicot@open.ac.uk

Silvia Barbina
silvia.barbina@open.ac.uk

[1] School of Mathematics and Statistics, The Open University, Walton Hall, Milton Keynes MK7 6AA, UK

Published online: 27 June 2017

Droste [5] classified the countable 2-transitive trees. The work of Droste was later generalised by Droste, Holland and Macpherson [6] to a classification of all countable 'weakly 2-transitive' trees – there are $2^{\aleph_0}$ non-isomorphic such trees. The goal of this paper, and of [3], is to extend this last classification result to a considerably richer class, by working under a much weaker symmetry hypothesis, namely 1-transitivity.

We first define the terminology used above and later. A **tree** is a partial order in which any two elements have a common upper bound and the upper bounds of any element are linearly ordered. A relational structure is said to be k**-transitive** if for any two isomorphic k-element substructures there is an automorphism taking the first to the second. For partial orders, there is a notion, called *weak 2-transitivity*, that generalises that of 2-transitivity: a partial order is weakly 2-transitive if for any two 2-element chains there is an automorphism taking the first to the second (but not necessarily for 2-element antichains).

A weaker notion still is that of *1-transitivity*, often called, more simply, *transitivity*. The classification of countable 1-transitive trees is considerably more involved than that of the weakly-2-transitive trees, and it rests on the classification of countable lower 1-transitive linear orders — the subject of this paper.

Definition 1.1 A linear order $(X, \leqslant)$ is **lower 1-transitive** if

$$(\forall a, b \in X)\ \{x \in X : x \leqslant a\} \cong \{x \in X : x \leqslant b\}.$$

An example of a lower 1-transitive, not 1-transitive linear order is ω^*, (that is, ω reversed). It is easy to see that any branch (that is, maximal chain) of a 1-transitive tree must be lower 1-transitive, though it is not necessarily 1-transitive.

The natural relation of equivalence between lower 1-transitive linear orders is *lower isomorphism*, rather than isomorphism.

Definition 1.2 Two linear orders, $(X, \leqslant)$ and $(Y, \leqslant)$ are **lower isomorphic** if

$$(\forall a \in X)(\forall b \in Y)\ \{x \in X : x \leqslant a\} \cong \{y \in Y : y \leqslant b\}.$$

When this happens, we write $(X, \leqslant) \cong_l (Y, \leqslant)$.

We shall use standard interval notation from now on where appropriate, for example

$$\begin{aligned}(-\infty, a] &:= \{x \in X : x \leqslant a\}, \\ (a, b] &:= \{x \in X : a < x \leqslant b\}, \\ [a, b] &:= \{x \in X : a \leqslant x \leqslant b\}.\end{aligned}$$

With this notation, the isomorphisms in the above definitions may then be written more succinctly as $(-\infty, a] \cong (-\infty, b]$.

The classification of countable lower 1-transitive linear orders is rather involved and so the current paper is devoted entirely to this, and the resultant classification of countable 1-transitive trees is deferred to [3].

A principal feature of the classifications of coloured 1-transitive countable linear orders ([1, 2]) and of 1-transitive trees [3], is the use of *coding trees* to describe the construction of the orderings. In these papers, and in what follows, coding trees play a totally different role from that of the 1-transitive trees which are classified in [3]: they are classifiers, rather than structures being classified. Section 2 of this paper contains the definition of coding tree and related notions. The main work of the paper is in Section 3, where we show how to construct a coding tree from a linear order. Section 4 then describes how to recover a linear order from a coding tree. The main theorem is Theorem 3.7, which, in conjunction with Theorems 4.2 and 4.3, gives our classification.

In order to give the flavour of the classification, we conclude this introduction with some examples of lower 1-transitive linear orders. First, some notation and terminology are needed.

Let $(A, \leqslant)$, $(B, \leqslant)$ be linear orders; for convenience, we often omit the order symbol. Then $A.B$ denotes the lexicographic product of A and B, where for $(a, b), (a', b') \in A \times B$, $(a, b) \leqslant (a', b')$ if and only if $a < a'$, or $a = a'$ and $b \leqslant b'$. Also, $A + B$ denotes A followed by B, that is, the disjoint union of A and B with $a < b$ for all $a \in A$ and $b \in B$. We write $\dot{\mathbb{Q}}$ for $\mathbb{Q} + \{+\infty\}$. If A is a linear order, then A^* denotes the ordering with the same domain and the reverse order. If $n \in \mathbb{N} \cup \{\aleph_0\}$, then $\mathbb{Q}_n$ is the Fraïssé generic n-coloured linear order, that is, the countable dense linear order coloured by n colours $c_0, \ldots, c_{n-1}$ and such that between any two distinct points there is a point of each colour. Likewise, $\dot{\mathbb{Q}}_n$ is $\mathbb{Q}_n + \{+\infty\}$, where the point $+\infty$ is also coloured by any of the c_i, or indeed any other colour. If $Y_0, \ldots, Y_{n-1}$ are linear orders, then $\mathbb{Q}_n(Y_0, \ldots, Y_{n-1})$ denotes the ordering obtained by replacing each point of $\mathbb{Q}_n$ coloured c_i by a convex copy of Y_i (with the natural induced ordering). We say that $\mathbb{Q}_n(Y_0, \ldots, Y_{n-1})$ is the **$\mathbb{Q}_n$-combination** of $Y_0, \ldots, Y_n$. If $n = \aleph_0$, we write $\mathbb{Q}_{\aleph_0}(Y_0, Y_1, \ldots)$.

The simplest countable lower 1-transitive linear orders are singletons, then ω^* and $\mathbb{Z}$ (which are lower isomorphic), and $\mathbb{Q}$ and $\dot{\mathbb{Q}}$ (which are also lower isomorphic). These orders are the basic building blocks for our constructions. We obtain new lower 1-transitive linear orders by concatenating and taking lexicographic products of existing ones. More precisely, Theorem 4.3 implies that if A and B are any lower 1-transitive linear orders which are lower isomorphic, then $\omega^*.A + B$ is lower 1-transitive.

For example, the lower isomorphism class of $\mathbb{Z}.\mathbb{Z}$ (that is, the class of linear orders that are lower isomorphic to $\mathbb{Z}.\mathbb{Z}$) consists of $\mathbb{Z}.\mathbb{Z}$, which by convention we write as $\mathbb{Z}^2$, $\omega^*.\mathbb{Z} + \mathbb{Z}$ and $\omega^*.\mathbb{Z} + \omega^*$. Note that we can concatenate $\omega^*.\mathbb{Z}$ with either $\mathbb{Z}$ or ω^* and the resulting linear order will still be lower 1-transitive. This is because ω^* has a right-hand endpoint and because $\mathbb{Z}$ and ω^* are lower isomorphic. Also note that $\omega^*.A + A \cong \omega^*.A$. We use the former form to streamline subsequent definitions in the paper. A yet more complex lower isomorphism class is that of $\mathbb{Z}^3$, which includes $\omega^*.\mathbb{Z}^2 + \mathbb{Z}^2$, $\omega^*.\mathbb{Z}^2 + \omega^*.\mathbb{Z} + \mathbb{Z}$ and $\omega^*.\mathbb{Z}^2 + \omega^*.\mathbb{Z} + \omega^*$.

Theorem 4.3 gives another construction of lower 1-transitive linear orders from existing ones. This construction involves the building block $\mathbb{Q}$: the linear order $\mathbb{Q}_n(Y_0, \ldots, Y_{n-1})$ (possibly with $n = \aleph_0$) is lower 1-transitive provided the Y_i are lower isomorphic to each other. Moreover, as above, $\mathbb{Q}_n(Y_0, \ldots, Y_{n-1}) + Y$ is lower 1-transitive provided Y and the Y_i are all lower isomorphic to each other. A simple example is $X = \mathbb{Q}_2(\omega^*, \mathbb{Z})$, which is countable and lower 1-transitive. Its lower isomorphism class also includes $\mathbb{Q}_2(\omega^*, \mathbb{Z}) + \mathbb{Z}$ and $\mathbb{Q}_2(\omega^*, \mathbb{Z}) + \omega^*$.

2 Coding Trees

This section introduces coding trees, which carry all the relevant information about lower 1-transitive linear orders.

First, recall that a tree $(T, \leqslant)$ is **Dedekind-MacNeille complete** if its maximal chains are Dedekind-complete in the usual sense, and if any two incomparable elements have a least upper bound. In fact, this is a special case of a general notion for partial orders, and the basics are given, for example, in Chapter 7 of [4]. Any tree $(T, \leq)$ has a unique (up to isomorphism over T) Dedekind-MacNeille completion, that is, a minimal Dedekind-MacNeille complete tree containing it, which is obtained as follows. If $A \subseteq T$ then A^{u} denotes the set of upper bounds of A and A^{l} the set of lower bounds, that is,

$$A^{\mathrm{u}} := \{x \in T : (\forall a \in A)\ (x \geqslant a)\}, \quad \text{and}$$

$$A^{\mathrm{l}} := \{x \in T : (\forall a \in A)\ (x \leqslant a)\}.$$

A subset $A \neq \emptyset$ is an **ideal** of T if $(A^{\mathrm{u}})^{\mathrm{l}} = A$. If x is any vertex of T, then the set $\mathrm{I}(x) := \{y \in T : y \leqslant x\}$ is an ideal of T. The Dedekind-MacNeille completion of T is the set $\mathrm{I}^D(T)$ of the ideals of T ordered by inclusion. It is easy to see that T embeds in $\mathrm{I}^D(T)$ via the map which takes $x \in T$ to $\mathrm{I}(x) \in \mathrm{I}^D(T)$.

Definition 2.1 If $(T, \leqslant)$ is a tree and $x \in T$, then a **child** of x is some y such that $y < x$ and there is no $z \in T$ with $y < z < x$. If x is a child of y then y is a **parent** of x. We write $\mathrm{child}(x)$ for the set of children of x. A **leaf** of $(T, \leqslant)$ is some $x \in T$ such that there is no $y \in T$ with $y < x$. We write $\mathrm{leaf}(T)$ for the set of leaves of $(T, \leqslant)$.

A **levelled tree** is a tree $(T, \leqslant)$ together with a partition, π, of T into maximal antichains, called **levels**, such that

(i) π is linearly ordered by $\ll$ so that $x \leqslant y$ in T implies that the level containing x is below the level containing y in the $\ll$ ordering;
(ii) if x and y are both children of z, then $x \sim_\pi y$.

A **leaf-branch** B of a (levelled) tree $(T, \leqslant)$ is a maximal chain of T which contains a leaf.

The supremum of two incomparable points (which exists in the Dedekind-MacNeille completion of T, even if not in T itself) is called a **ramification point**.

If $x \in T$ then the relation $\asymp_x$ on $\{y \in T : y < x\}$ given by

$$a \asymp_x b \ \text{ if there is} c \in T \ \text{ such that} a, b \leqslant c < x$$

is an equivalence relation. The equivalence classes are called **cones** at x.

Definition 2.2 A tree is **labelled** if each vertex is labelled by one of the symbols $\mathbb{Z}$, ω^*, $\mathbb{Q}$, $\dot{\mathbb{Q}}$, $\mathbb{Q}_n$, $\dot{\mathbb{Q}}_n$ (for $2 \leqslant n \leqslant \aleph_0$), $\{1\}$ (singleton), or lim.

Isomorphisms between labelled trees are required to preserve the labelling.

Definition 2.3 Let $(T, \leqslant)$ be a levelled tree. Let $x \in T$ and let $\lhd$ be a linear order on $\mathrm{child}(x)$. If x is labelled by one of ω^*, $\dot{\mathbb{Q}}$ and $\dot{\mathbb{Q}}_n$, the **right** child of x is the child which is greatest under the $\lhd$ ordering. All the remaining children are **left** children. If x is labelled by one of $\mathbb{Z}$, $\mathbb{Q}$ or $\mathbb{Q}_n$, we consider all its children to be **left** children.

A **left tree** of a vertex is a partially ordered set consisting of a left child of the given vertex together with the descendants of the left child, with the induced structure of levels and labels. The **left forest** of a vertex is the partially ordered set consisting of the left trees of that vertex.

Two forests are **isomorphic** provided the subtrees rooted at the greatest elements in each forest can be put into one-to-one correspondence in such a way that they are isomorphic as labelled trees.

Thus, an isomorphism between two forests preserves the levelling and the labelling, but it is not required to preserve the $\lhd$ ordering among children.

Definition 2.4 A **coding tree** has the form $(T, \leqslant, \triangleleft, \varsigma, \ll)$ where

1. T is a levelled tree with a greatest element, the root. The tree ordering is $\leqslant$, $\triangleleft$ is a linear ordering on the set of children of each parent and $\ll$ is the ordering of the levels.
2. There are countably many leaves.
3. Every vertex is a leaf or is above a leaf.
4. T is Dedekind-MacNeille complete.
5. The vertices are labelled by ς, the labelling function, which assigns to the vertices one of the following labels: $\mathbb{Z}$, ω^*, $\mathbb{Q}$, $\dot{\mathbb{Q}}$, $\mathbb{Q}_n$, $\dot{\mathbb{Q}}_n$ (for $2 \leqslant n \leqslant \aleph_0$), $\{1\}$ (singleton), or lim.
6. For any two vertices x_i and x_j on the same level, $\varsigma(x_i)$ is lower-isomorphic to $\varsigma(x_j)$ or $\varsigma(x_i) = \varsigma(x_j) = \text{lim}$.
7. If x and y are not siblings, then they are $\triangleleft$-incomparable.
8. For any vertex x of the tree:

   – if $\varsigma(x) = \mathbb{Z}$ or $\mathbb{Q}$ then x has exactly one child;
   – if $\varsigma(x) = \omega^*$ or $\dot{\mathbb{Q}}$ then x has two children;
   – if $\varsigma(x) = \mathbb{Q}_n$ then x has n children;
   – if $\varsigma(x) = \dot{\mathbb{Q}}_n$ then x has $n + 1$ children;
   – if $\varsigma(x) = \{1\}$ then x is a leaf and has no children;
   – if $\varsigma(x) = \text{lim}$ then there is only one cone at x (so x is not a leaf and has no children).

9. At each given level of T, the left forests of vertices at that level are all isomorphic in the sense of Definition 2.3.
10. If x is a parent vertex and y_0, y_1 are two of its left children, then the subtrees with roots y_0, y_1 are not isomorphic.

We illustrate Definition 2.4 in Fig. 1, where we give the coding trees for the lower 1-transitive linear orders in the lower isomorphism class of $\mathbb{Z}^3$, that is, $\mathbb{Z}^3$, $\omega^*.\mathbb{Z}^2 + \mathbb{Z}^2$, $\omega^*.\mathbb{Z}^2 + \omega^*.\mathbb{Z} + \mathbb{Z}$ and $\omega^*.\mathbb{Z}^2 + \omega^*.\mathbb{Z} + \omega^*$.

A full explanation of how to recover a linear order from a coding tree is given in Section 4. However, in finite cases, such as those illustrated in Fig. 1, it is possible to give an informal description of how to read a linear order off its coding tree: we can start at the root and proceed recursively through the tree. When at a vertex with label $\mathbb{Z}$ or $\mathbb{Q}$, we take the lexicographic product of the label with the linear order encoded by the subtree rooted at the child of the vertex. When at a vertex labelled ω^*, respectively $\dot{\mathbb{Q}}$, we take the lexicographic

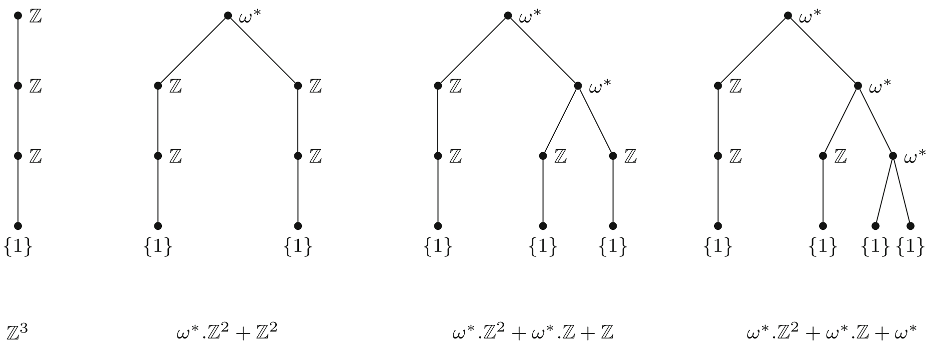

Fig. 1 Coding trees for lower 1-transitive linear orders in the lower isomorphism class of $\mathbb{Z}^3$

product of ω^*, respectively $\mathbb{Q}$, and the linear encoded by the subtree rooted at the left child of the vertex and we concatenate this with the linear order encoded by the subtree rooted at the right child. When the vertex is labelled $\mathbb{Q}_n$ or $\dot{\mathbb{Q}}_n$, we take $\mathbb{Q}_n$-combination of the linear orders encoded by the subtrees rooted at the left children of the vertex, and in the case of $\dot{\mathbb{Q}}_n$ we additionally concatenate this with the linear order encoded by the tree rooted at the right child.

In general, however, the linear order cannot be recovered from its coding tree in a recursive way, because Definition 2.4 does not imply that the levels of a coding tree are well ordered or conversely well ordered. Consider the example in Fig. 2.

In this tree, there are countably many levels of vertices labelled $\dot{\mathbb{Q}}_2$. The leaf-branches are maximal chains which eventually constantly descend through the right children of $\dot{\mathbb{Q}}_2$, that is, they only contain finitely many vertices that are left children. No other branches are leaf-branches. In order for this tree to be a coding tree, we need that for every parent vertex with left children y_0 and y_1, the subtrees rooted at y_0 and y_1 are not isomorphic. This is the case if there are non-isomorphic linear orders below the level labelled lim, while 8 in Definition 2.4 is still satisfied. Under these assumptions, this tree is a coding tree, yet it is neither well founded nor conversely well founded.

To help understand the linear orders represented by coding trees of this form, consider the example where the non-isomorphic linear orders below the level labelled lim are all lower-isomorphic to $\mathbb{Z}^\omega$. One way to think of such linear orders is as having the form

$$\mathbb{Q}_2(\square, \square) + \square$$

where each argument is obtained by iterating the construction

$$\mathbb{Q}_2(\square, \square) + \lozenge$$

countably many times, and where the third arguments in the construction (the orders that are concatenated at the end) are lower-isomorphic to $\mathbb{Z}^\omega$.

Examples of this kind are the reason why we need *expanded coding trees* to recover a lower 1-transitive linear order from a coding tree.

Expanded coding trees are closely related to coding trees and they are defined next. In place of a labelling function on vertices, expanded coding trees carry, as part of the structure, a total ordering on the set of children of each vertex. In general, a coding tree and the corresponding expanded coding tree do not have the same vertex set. For instance, a point of the expanded coding tree corresponding to a point labelled $\dot{\mathbb{Q}}$ in the coding tree will have infinitely many children in the expanded coding tree. All the children but the last one are associated with the left child in the coding tree. The idea is that a lower 1-transitive linear order $(X, \leqslant)$ lives on the set of leaves of the expanded coding tree, so the expanded coding tree facilitates the transition between coding tree and encoded order.

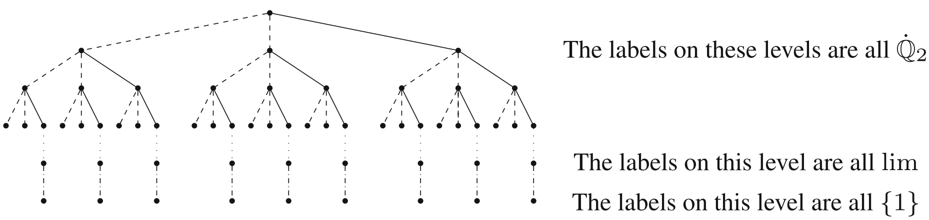

Fig. 2 A coding tree that is neither well founded nor conversely well founded

Definition 2.5 An **expanded coding tree** is a structure of the form $(E, \leqslant, \ll, \lhd)$ where:

1. E is a levelled tree with a greatest element, the root, denoted by r. The tree ordering is $\leqslant$, $\ll$ is the ordering of the levels and $\lhd$ is the ordering on the children of each parent vertex.
2. $(E, \lhd)$ is a partial ordering consisting of a disjoint union of chains of the form $(\text{child}(x), \lhd)$ for some $x \in T$. Moreover, if $x \lhd y$, then x is level with y.
3. $(E, \leqslant)$ has at most countably many leaves.
4. Every vertex of $(E, \leqslant)$ is a leaf or is above a leaf.
5. $(E, \leqslant)$ is Dedekind-MacNeille complete.
6. If a vertex has any children, then their $\lhd$-order type is one of $\mathbb{Z}$, ω^*, $\mathbb{Q}$, $\dot{\mathbb{Q}}$, $\mathbb{Q}_n$ or $\dot{\mathbb{Q}}_n$ for $2 \leqslant n \leqslant \aleph_0$.
7. Any two vertices x and x' on the same level are either both parent vertices, or they are both leaves, or they both have exactly one cone below them. If x and x' are both parent vertices, then $(\text{child}(x), \lhd) \cong_l (\text{child}(x'), \lhd)$.
8. For any parent vertex x of the tree, one of the following holds:
   (i) the $\lhd$-order type of $\text{child}(x)$ is $\mathbb{Z}$, $\mathbb{Q}$, ω^* or $\dot{\mathbb{Q}}$ and the left trees rooted at the children of x are all isomorphic, or
   (ii) the children of x are densely ordered by $\lhd$ and the trees rooted at the children of x fall into $n \geqslant 2$ isomorphism classes and this makes them isomorphic to $\mathbb{Q}_n$ (for $2 \leqslant n \leqslant \aleph_0$), or
   (iii) the left children are as in (ii) above, and x has a right child and this makes $(\text{child}(x), \lhd)$ order-isomorphic to $\dot{\mathbb{Q}}_n$.
9. At each given level of E the left forests (see Definition 2.3) from that level are order-isomorphic (meaning that $\leqslant$, $\ll$ and $\lhd$ are preserved).

In 8(ii), we mean that if the elements of $\text{child}(x)$ are coloured according to the isomorphism type of the trees below them, then the corresponding coloured linear order (with respect to $\lhd$) is isomorphic to $\mathbb{Q}_n$; likewise in 8(iii).

As in [2], we define a map which associates an expanded coding tree to a coding tree.

Definition 2.6 Let $(T, \leqslant, \varsigma, \ll, \lhd)$ be a coding tree, and $(E, \leqslant, \ll, \lhd)$ be an expanded coding tree. We say that E is **associated** with T via ϕ if there is a function $\phi : E \to T$ which takes the root of E to the root of T, each leaf of E to some leaf of T, and such that

(i) $v_1 \leqslant v_2 \implies \phi(v_1) \leqslant \phi(v_2)$,
(ii) ϕ induces an order-isomorphism from the set of levels of E (ordered by $\ll$) to the set of levels of T.
(iii) for each vertex v of E, ϕ maps $\{u \in E : u \leqslant v\}$ onto $\{u \in T : u \leqslant \phi(v)\}$, and for any leaf l of E, ϕ maps $[l, r]$ onto $[\phi(l), \phi(r)]$,
(iv) for each parent vertex v of E, one of the following holds:
- $\varsigma(\phi(v)) = \mathbb{Z}, \omega^*, \mathbb{Q}, \dot{\mathbb{Q}}$, and this is the order type of the children of v under $\lhd$;
- $\varsigma(\phi(v)) = \mathbb{Q}_n, \dot{\mathbb{Q}}_n$ (for $2 \leqslant n \leqslant \aleph_0$) and for any left children u, u' of v, if the trees rooted at u and u' are isomorphic then $\phi(u) = \phi(u')$;
- $\varsigma(\phi(v)) = \lim$ if v is neither a parent nor a leaf (in which case v has just one cone);
- $\varsigma(\phi(v)) = \{1\}$ if v is a leaf.

The map ϕ is said to be an **association map** between T and E.

We are now in a position to say explicitly how a tree encodes a linear order. If E is an expanded coding tree, there is a natural linear order on leaf(E), which we denote by $\lhd^*$, defined as follows: if x, y are leaves, then $x \lhd^* y$ if there are $x', y' \in E$ with $x \leqslant x'$, $y \leqslant y'$, and $x' \lhd y'$.

Definition 2.7 The coding tree $(T, \leqslant, \varsigma, \ll, \lhd)$ **encodes** the linear order $(X, \leqslant)$ if there is an expanded coding tree E associated with T such that X is order-isomorphic to the set of leaves of E ordered by $\lhd^*$, that is $(X, \leqslant) \cong (\text{leaf}(E), \lhd^*)$.

3 Construction of a Coding Tree from a Linear Order

In this section we show that any countable and lower 1-transitive $(X, \leqslant)$ is encoded by a suitable coding tree. We first define the tree I of *invariant partitions* of $(X, \leqslant)$ (see Definition 3.1 below) and we show that I is in fact an expanded coding tree. We then define a coding tree to which I is associated and that encodes $(X, \leqslant)$.

Definition 3.1 An **invariant partition** of X is a partition π that partitions X into convex subsets, called **parts**, which is invariant under lower isomorphisms of $(X, \leq)$ into itself. That is, for any $a, b \in X$, any order isomorphism $f : (-\infty, a] \to (-\infty, b]$, and any $x, y \leqslant a$,

$$x \sim_\pi y \iff f(x) \sim_\pi f(y).$$

The proof of the next lemma is left to the reader.

Lemma 3.2 *If $(X, \leqslant)$ is a countable lower 1-transitive linear order and π is an invariant partition of X, then $X/_{\sim_\pi}$ is also a countable lower 1-transitive linear order with the ordering induced by $(X, \leqslant)$.*

Definition 3.3 Let π_i, π_j be invariant partitions of $(X, \leqslant)$. We say that π_i is a **refinement** of π_j if every element of π_j is a union of members of π_i.

Lemma 3.4 *Given any two nontrivial distinct invariant partitions π_1, π_2 of X into convex subsets of X, one is a refinement of the other, and moreover π_1 and π_2 have no part in common.*

Proof Let $\sim_1, \sim_2$ be the equivalence relations defining π_1, π_2 respectively. We want to show that

$$(\forall x, y \in X)(x \sim_1 y \Rightarrow x \sim_2 y) \vee (\forall x, y \in X)(x \sim_2 y \Rightarrow x \sim_1 y).$$

Suppose both disjuncts are false. Then there are x, y, u, v such that

- $x \sim_1 y$ and $x \nsim_2 y$, and
- $u \nsim_1 v$ and $u \sim_2 v$.

We may assume that $x < y$ and $u < v$. Let $f : (-\infty, y] \to (-\infty, v]$ be an isomorphism. Then $f(x) < v$ and $f(x) \sim_1 v$. Moreover, we must have $u < f(x)$, otherwise $u \sim_1 v$ by convexity. So $u < f(x) < v$, and therefore $f(x) \sim_2 v$ by convexity. However, $x \nsim_2 y$ implies $f(x) \nsim_2 v$, which is a contradiction.

Without loss of generality, assume that π_1 is a refinement of π_2. We want to show that $\pi_1 \cap \pi_2 = \varnothing$. So suppose for a contradiction that there is $p \in \pi_1 \cap \pi_2$, and let $x, y \in X$

be such that $x \nsim_1 y$ and $x \sim_2 y$. Pick $z \in p$ and let $g : (-\infty, y] \to (-\infty, z]$ be an isomorphism. Then $g(x) \sim_2 z$, and since $p \in \pi_1 \cap \pi_2$, we have $g(x) \sim_1 g(y)$, contradicting our choice of x and y. □

The family I of all parts of invariant partitions of X is partially ordered by inclusion. This allows us to define a levelled tree structure on I.

Definition 3.5 For a lower 1-transitive linear order $(X, \leq)$, the **invariant tree** associated with X is the levelled tree I whose vertices are parts in the invariant partitions of X ordered by $\subseteq$ in such a way that

(i) each level is an invariant partition of X
(ii) the leaves are the singletons $\{x\}$ for $x \in X$
(iii) every invariant partition of X into convex subsets of X is represented by a level of vertices in I.

Lemmas 3.2 and 3.4 ensure that for any countable, lower 1-transitive linear order, the family I is a levelled tree, thereby justifying the description *the* invariant tree. We remark that I has a root since X is itself lower 1-transitive and a convex subset of X. Moreover, the parts of any invariant partition of X are lower isomorphic and lower 1-transitive.

Lemma 3.6 *The invariant tree I of a lower 1-transitive linear order $(X, \leqslant)$ is Dedekind-MacNeille complete.*

Proof We need to show that

(i) the supremum of any two vertices in I is also a vertex in I,
(ii) every descending chain of vertices in the tree which is bounded below has an infimum in the tree, and
(iii) every ascending chain of vertices in the tree which is bounded above has a supremum in the tree.

To show (i), consider two vertices $p_1, p_2 \in I$ that are parts of two partitions π_1, π_2, respectively. Without loss of generality, assume that π_1 refines π_2. Then either $p_1 \subseteq p_2$ (and p_2 is the supremum of p_1 and p_2) or $p_1 \subseteq p_2'$, where p_2' is an element of π_2 different from p_2. So this problem reduces to showing that the supremum of any two vertices on the same level is in I.

We know that $p_2, p_2' \subseteq p$ with $p \in \pi$, for some $\pi \in I$ which coarsens π_2 – for instance $\{X\}$ itself. Let $\sim_\pi$ be the equivalence relation corresponding to π. Then $a \sim_\pi b$ for $a, b \in p_2, p_2'$ respectively.

Consider the partitions π' that refine π for which $a \sim_{\pi'} b$, where $\sim_{\pi'}$ is the corresponding equivalence relation, and then consider the set P of parts in this set that contain both a and b. By Lemma 3.4, the set P is a descending chain in I. Let q be such that $p_2, p_2' \leqslant q \leqslant p$. Then $q \in P$, so if P has an infimum, then p_2, p_2' have a supremum. So the verification of (i) reduces to that of (ii).

For (ii), consider a descending chain of vertices p_γ that are parts of a descending chain of partitions π_γ bounded below by some $p \in I$. Let $\sim_\gamma$ be the equivalence relation corresponding to π_γ. Then define $x \sim y$ if $x \sim_\gamma y$ for all γ. Let f be a lower isomorphism of $(X, \leqslant)$. Then $x \sim y$ implies $f(x) \sim_\gamma f(y)$ for all γ because each of the $\sim_\gamma$ is an invariant relation. Hence $f(x) \sim f(y)$ and so $\sim$ is an invariant relation. If π is the corresponding partition, then π is a partition into lower 1-transitive, lower isomorphic convex subsets of

X, and so its parts are vertices in I. Then p is contained in some member of p' of π, and π' is the infimum of the p_γ.

The proof that an ascending chain that is bounded above has a supremum is similar, except that we take $x \sim y$ if $x \sim_\gamma y$ for some γ. □

Theorem 3.7 *The invariant tree I of a lower 1-transitive linear order $(X, \leqslant)$ is an expanded coding tree whose leaves are order-isomorphic to $(X, \leqslant)$.*

Proof Firstly, the leaves of I are singletons containing the elements of X, and so they are isomorphic to X.

Definition 3.5 ensures that I is a levelled tree whose root is X. The tree ordering is containment, the ordering of the levels is the one induced by $\subseteq$ on the set of invariant partitions of X, and the ordering of the children of a parent vertex is the one induced by the linear order on X. Since X is countable, I has countably many leaves. It is clear that every vertex of I is a leaf or is above a leaf. So conditions 1 to 4 of Definition 2.5 are satisfied. Moreover, I is Dedekind-MacNeille complete by Lemma 3.6.

In order to verify condition 6 of Definition 2.5, we need to show that the order type of the children of a parent vertex in I is one of $\mathbb{Z}$, ω^*, $\mathbb{Q}$, $\dot{\mathbb{Q}}$, $\mathbb{Q}_n$ or $\dot{\mathbb{Q}}_n$ (for $2 \leqslant n \leqslant \aleph_0$). Consider a successor level π_{i+1} of I, so π_i is the predecessor. Let $p \in \pi_{i+1}$. Then p is lower 1-transitive, and the children of p are those elements of π_i which are convex subsets of p. These children are lower 1-transitive linear orders and are lower isomorphic to each other. Let $\sim_{\pi_i}$ be the equivalence relation that defines π_i. Then, by Lemma 3.2, $p/\sim_{\pi_i}$ is also lower 1-transitive, and the order type of $p/\sim_{\pi_i}$ tells us how the children of p are ordered. In order to describe the possible order types, we look at the structure forced by a specific invariant equivalence relation, namely, the relation $\sim_{\text{fin}}$ that identifies points that are finitely far apart, defined by

$$x \sim_{\text{fin}} y \ \text{ iff } x \leqslant y \ \text{ and}[x, y] \ \text{ is finite, or } y \leqslant x \ \text{ and } [y, x] \ \text{ is finite.}$$

For any linear order, the equivalence classes of $\sim_{\text{fin}}$ must be either finite, ω, ω^* or $\mathbb{Z}$. If $(X, <)$ is lower 1-transitive, the equivalence classes of this form are either singletons, ω^*, or $\mathbb{Z}$. If one equivalence class is a singleton, then they all are, and then the ordering is dense with no least endpoint. Hence it is isomorphic to $\mathbb{Q}$ or $\dot{\mathbb{Q}}$.

Since $p/\sim_{\pi_i}$ is a lower 1-transitive linear order, we can take its quotient by $\sim_{\text{fin}}$. There are two cases.

Case 1: the equivalence classes of $(p/\sim_{\pi_i})/\sim_{\text{fin}}$ are non-trivial. Then, since every invariant partition is contained in I and we have assumed that π_i is a successor level, there can be only one equivalence class, that is, $p/\sim_{\pi_i}$ itself. If there is no last child, then p is equal to $\mathbb{Z}$ copies of its children; otherwise, the order type of $p/\sim_{\pi_i}$ is ω^*.

Case 2: the equivalence classes of $(p/\sim_{\pi_i})/\sim_{\text{fin}}$ are trivial. Then the parts of π_i are dense within p. We aim to show that $p/\sim_{\pi_i}$ is a $\mathbb{Q}$, $\dot{\mathbb{Q}}$, $\mathbb{Q}_n$ or $\dot{\mathbb{Q}}_n$ combination of its children.

If all the subtrees rooted at a left child of p are isomorphic, then child(p) is isomorphic to $\mathbb{Q}$, or $\dot{\mathbb{Q}}$ if the right child exists.

If not all the left children of p are isomorphic, then we show that child(p) is isomorphic to $\mathbb{Q}_n$, or $\dot{\mathbb{Q}}_n$ if p has a right child, where the set Γ of (colour, order-)isomorphism types of the left children of p has size n. Suppose, for a contradiction, that p is not the $\mathbb{Q}_n$-combination of its children. Then there are two elements of Γ such that not all other elements of Γ occur between them in p. Let γ be a member of Γ which does not occur between all pairs, and let us define $\sim$ on π by $y \sim z$ if $y = z$, or if no point of $[y, z]$ (or $[z, y]$ if

$z < y$) has the same isomorphism type as γ. This is an invariant partition of π into convex pieces, and is proper and non-trivial, which contradicts π_i and π_{i+1} being on consecutive levels.

This verifies condition 6 of Definition 2.5 for a parent vertex on a successor level of the invariant tree I.

Now consider the levels which are not successor levels. Firstly, this includes the trivial partition, π_0, given by the relation $x \sim_{\pi_0} y \iff x = y$. These vertices are leaves.

There remains the case of vertices which do not have children in I. If one part of an invariant partition does not have a child then, since parts in a same partition are lower-isomorphic to each other, none of them do. Since I contains all the invariant partitions, these vertices have one cone below them: let p be a vertex with no children, and let $p_1, p_2 < p$. We claim there is $c < p$ such that $p_1, p_2 \leq c$. Suppose for a contradiction that for all $c \geqslant p_1$, if $c \geqslant p_2$ then $c = p$. Let $\{p_i^1\}$ be the set of elements of I strictly below p and containing p_1. Then $\cup\{p_i^1\}$ is a part of an invariant equivalent relation, so, by maximality of I, $\cup\{p_i^1\} \in I$. Similarly, there is $\cup\{p_i^2\} \in I$ which is above p_2 and strictly below p. If $\cup\{p_i^1\} \neq \cup\{p_i^2\}$, then $\cup\{p_i^1\}$ and $\cup\{p_i^2\}$ are two distinct children of p, contradicting that p has no children.

For condition 7, let x and x' be two vertices of I on the same level. Then x, x' are parts of an invariant partition, so either they are both parents, or they are both leaves, or both are neither of these, in which case they have a single cone below them. Moreover, if x, x' are both parent vertices, then $(\mathrm{child}(x), \lhd)$ is lower-isomorphic to $(\mathrm{child}(x'), \lhd)$, since $(X, \leqslant)$ is lower 1-transitive and x and x' are parts of an invariant partition.

For condition 8, let $x \in I$ be a parent vertex. Suppose that $(\mathrm{child}(x), \lhd) \cong \mathbb{Q}, \dot{\mathbb{Q}}, \mathbb{Q}_n$ or $\dot{\mathbb{Q}}_n$. Here two children vertices a, b have the same colour when they are isomorphic. This isomorphism induces an isomorphism on the trees rooted at a, b. If $(\mathrm{child}(x), \lhd) \cong \mathbb{Z}$, we wish to show the children of x are all isomorphic, and hence the trees below the children are isomorphic. Now, the children of x are all a finite distance apart. In particular, each child has a successor and a predecessor. If a and b are children of x, the existence of an isomorphism from the successor of a to the successor of b implies that a and b are isomorphic. The argument in the case $(\mathrm{child}(x), \lhd) \cong \omega^*$ is similar.

Finally we show I satisfies condition 9 of Definition 2.5.

Let x and y be distinct vertices on the same level. If x and y have no children, the condition holds trivially. So suppose that x and y are parent vertices and let $a \in x$ and $b \in y$. By lower 1-transitivity, there is an isomorphism $\varphi : (-\infty, a] \to (-\infty, b]$ which induces an isomorphism between $(-\infty, a] \cap x$ and $(-\infty, b] \cap y$. Let x_a, y_b be the children of x, y containing a, b respectively. Then $(-\infty, a] \cap x_a$ and $(-\infty, b] \cap y_b$ are isomorphic. Consider the sets Γ_a, Γ_b of children of x, y to the left of x_a, y_b respectively. Since $\varphi(\Gamma_a) = \Gamma_b$, the left forests of x and y are isomorphic. Since a, b are arbitrary, Γ_a and Γ_b can contain any particular left children of x and y. □

We now show how to construct a coding tree to which the invariant tree of the linear order $(X, \leqslant)$ is associated, and we give an inverse association map. Informally, the coding tree is obtained from I by identifying left children who are siblings and whose trees of descendants are isomorphic. The parent vertex is then labelled according to the order type of its children in I.

For each level s of I we define a relation $\simeq_s$ on I that tells us which vertices to identify: $x \simeq_s y$ if there are $x' \supseteq x,\ y' \supseteq y$ such that

(i) there exists a $\{\leqslant, \lhd\}$-isomorphism $\theta : I^{\leqslant x'} \to I^{\leqslant y'}$

(ii) x', y' are left children of a vertex z and lie on level s, or $x' = y'$
(iii) $\theta(x) = y$.

The isomorphism in (i) will be made explicit in the proof of Lemma 3.8 below. Note that these clauses guarantee that x, y are level. Now we define a relation $\simeq$ on the whole of E as follows:

$$x \simeq y \iff \exists x = x_0, \ldots, x_n = y, \text{ where for each } i = 0, \ldots, n-1 \text{ there is } s_i \text{ with } x_i \simeq_{s_i} x_{i+1}.$$

The relation $\simeq$ is an equivalence relation on I, and T is then the set of equivalence classes on I, labelled as described above. We denote an element of T by $[x]$, where $x \in I$. The next lemma ensures that the ordering on I induces one on T.

Lemma 3.8 *Let* $[x], [y] \in T$ *be such that* $x \leqslant y$ *(in* I*), and let* $x' \in [x]$*. Then there is* $y' \in [y]$ *such that* $x' \leqslant y'$.

Proof Let x, y and x' be as in the statement. Since $x' \in [x]$, there are u, v and w in I such that u, v are left children of w and $x \leq u$, $x' \leqslant v$. Moreover, the tree of descendants of u is isomorphic to the tree of descendants of v by an isomorphism θ such that $\theta(x) = x'$. Now, either $y \geqslant w$ or $y < w$. If $y \geqslant w$ then $x' < w \leqslant y$, so y is the required y'. If $y < w$, then $y \leqslant u$. Then $x' = \theta(x) \leqslant \theta(y)$ and, since $\theta(y) \leqslant v$, this implies that $x' \leqslant v$. But $\theta(y) \in [y]$ because of the way $\simeq$ is defined, so $\theta(y)$ is the required y'. □

Theorem 3.9 *The set of* $\simeq$*-classes on the invariant tree of* $(X, \leqslant)$ *is a coding tree.*

Proof Let T be the family of $\simeq$-equivalence classes on I. Let $[x], [y] \in T$ and define

$$[x] \leqslant [y] \iff (\exists x' \in [x])(\exists y' \in [y])(x' \leqslant y') \text{ (in } I).$$

Lemma 3.8 ensures that $\leqslant$ is well defined and transitive, so $\leqslant$ is an order.

Since $\leqslant$ is the order induced by that on I, T is a tree with root $[r]$ and, since $\simeq$ is level preserving, T is a levelled tree. Moreover, T is countable, and every vertex of T is a leaf or is above a leaf. We verify Dedekind-MacNeille completeness. Firstly note that all leaf-branches of T are isomorphic to some leaf-branch of I and so the leaf-branches of T are Dedekind complete. We must now show that the least upper bound of any two vertices $[x], [y] \in T$ is in T. Since I is Dedekind-MacNeille complete, any $x' \in [x]$ and $y' \in [y]$ have a least upper bound in I. Let

$$\Gamma = \{z \in I : z \text{ is the least upper bound of } x' \text{ and } y' \text{ for some } x' \in [x], y' \in [y]\}.$$

If $z \in \Gamma$, then $[x'] = [x] \leqslant [z]$ and $[y'] = [y] \leqslant [z]$ for some x', y', and so $[z]$ is an upper bound for $[x]$ and $[y]$. Now let $\Gamma' = \{[z] : z \in \Gamma\}$. Since Γ' contains the upper bounds of $[x]$ and $[y]$, it is linearly ordered. Moreover, it is bounded above by $[r]$ and below by $[x]$. Let Γ be the chain

$$\ldots [z_{-n}] \geqslant \ldots \geqslant [z_0] \geqslant [z_1] \geqslant \ldots \; [z_n] \geqslant \ldots$$

By Lemma 3.8, for any $u \in [z_{i+1}]$ there is $v \in [z_i]$ such that $u \leqslant v$. Hence we can construct a corresponding chain of vertices in I. If the $[z_i]$ do not have an infimum, then there is a chain of vertices in I bounded below by $x \in [x]$ and without an infimum. This contradicts the Dedekind-MacNeille completeness of I. Then the infimum of Γ' is the least upper bound of $[x]$ and $[y]$.

Next we examine the labelling. Suppose $x \in I$ is a parent vertex. Then $[x] \in T$ is also a parent vertex and we let $\varsigma([x]) = (\text{child}(x), \lhd)$, the order type of the children of x in I.

This is well defined, as $x \simeq y$ implies that x and y are isomorphic and hence the sets of their children have the same order type. Now, since $(\mathrm{child}(x), \lhd)$ is one of $\mathbb{Z}$, ω^*, $\mathbb{Q}$, $\dot{\mathbb{Q}}$, $\mathbb{Q}_n$, $\dot{\mathbb{Q}}_n$ (for $2 \leqslant n \leqslant \aleph_0$), it follows that $\varsigma([x])$ is also one of the above.

If x is neither a parent nor a leaf, then neither is $[x]$. Hence we label $[x]$ by lim. The leaves are labelled $\{1\}$.

Let $[x], [y] \in T$ be level parent vertices and let $x, y \in I$ be representatives. Then $\varsigma([x]) \cong_l \varsigma([y])$ follows from the fact that $(\mathrm{child}(x), \lhd) \cong_l (\mathrm{child}(y), \lhd)$.

When $[x], [y] \in T$ are level but neither parent vertices nor leaves (if $[x]$ is not a parent vertex and $[x]$ are $[y]$ level, then $[y]$ is not a parent vertex), both are labelled lim, as remarked earlier. Hence $\varsigma([x]) = \varsigma([y])$ as required. The case when $[x], [y] \in T$ are leaves is similar.

We now show that T fulfils condition 7 of Definiton 2.4. The number of children of $[x] \in T$ is the number of equivalence classes of the children of vertices $x' \in [x]$ in I. We consider various cases.

Case 1: $(\mathrm{child}(x), \lhd) \cong \mathbb{Z}, \mathbb{Q}$

All the children of x are left children. We have also seen that they are all isomorphic and hence they are all $\simeq$-equivalent. Therefore there is one equivalence class below $[x]$.

Case 2: $(\mathrm{child}(x), \lhd) \cong \omega^*, \dot{\mathbb{Q}}$

Again all the left children of x are isomorphic and hence they are all $\simeq$-equivalent. A right child of x forms its own equivalence class under $\simeq$. In these cases $[x]$ has two children.

Case 3. $(\mathrm{child}(x), \lhd) \cong \mathbb{Q}_n, \dot{\mathbb{Q}}_n$.

The 'colours' are the isomorphism types of the children of x in I. There are n isomorphism types amongst the left children. The left children which are isomorphic are also $\simeq$-equivalent. Hence there are n ($n + 1$ in the case of $\dot{\mathbb{Q}}_n$) $\simeq$-classes below $[x]$.

Clause 8 of Definition 2.4 follows from the corresponding fact about the expanded coding tree. Given two order isomorphic forests in the expanded coding tree, clearly the $\simeq$-classes on two such forests are also isomorphic.

Finally, since $\simeq$ identifies isomorphic trees of descendants of sibling left vertices, the tree of descendants of two sibling vertices in the resulting T will not be isomorphic. □

We now need to show that the coding tree that we have obtained in Theorem 3.9 does encode $(X, \leqslant)$.

Theorem 3.10 *The coding tree* $(T, \leqslant, \lhd, \varsigma, \ll)$ *obtained from the invariant tree of* $(X, \leqslant)$ *encodes* $(X, \leqslant)$ *in the sense of Definition 2.7.*

Proof Firstly we show that the expanded coding tree I of invariant partitions of X is associated with T in the sense of Definition 2.6. The association function $\phi \to T$ is defined by $\phi(x) = [x]$, and the labelling function on T is defined as follows:

(i) if x is a parent vertex, the label of $\phi(x)$ is equal to $(\mathrm{child}(x), \lhd)$, the (coloured) order type of the children of x in I,
(ii) if x is neither a parent nor a leaf, the label of $\phi(x)$ is lim,
(iii) if x is a leaf, the label of $\phi(x)$ is $\{1\}$.

As remarked in the proof of Theorem 3.9, this labelling is well defined. Moreover, the labels satisfy condition (iv) of Definition 2.6. By the way T is constructed, it is clear that ϕ preserves levels. Moreover, the ordering on T is such that $x \leqslant y$ in I implies that $\phi(x) \leqslant \phi(y)$ in T. This ensures that conditions (i), (ii) and (iii) of Definition 2.6 are satisfied.

The construction of I ensures that $(X, \leqslant)$ is order-isomorphic to the set of leaves of I. Therefore T encodes the linear order X in the sense of Definition 2.7, as required. □

Theorem 3.10 concludes our construction of a coding tree from a lower 1-transitive linear order. The next section describes the converse construction of a lower 1-transitive linear order from a coding tree, which will give our classification.

4 Construction of a Linear Order from a Coding Tree

In Theorem 4.2 below, we show how to recover a linear order from a coding tree, and in Theorem 4.3 we show that a linear order obtained in this way is in fact lower 1-transitive.

In order to do this, we need to define certain functions called *decoding functions*, whose domains are the leaf-branches of a given coding tree and which take a vertex x to an element of the ordered set $\varsigma(x)$. To cut down to a countable set of functions, even when the coding tree is not well founded or conversely well founded, we choose arbitrary default values for each of the labels.

Definition 4.1 Given a coding tree T, we choose default values for its labels as follows: for each of $\mathbb{Z}$ and $\mathbb{Q}$, we pick one default value. For ω^* and $\dot{\mathbb{Q}}$, we pick two default values, one for the end point and one other. For $\mathbb{Q}_n$, we pick n default values, one of each 'colour', and for $\dot{\mathbb{Q}}_n$ we pick the same default values as for $\mathbb{Q}_n$, plus an additional one for the endpoint.

Then a **decoding function** is a function f from a leaf-branch B of T to $\omega^* \cup \mathbb{Z} \cup \mathbb{Q} \cup \dot{\mathbb{Q}} \cup \mathbb{Q}_n \cup \dot{\mathbb{Q}}_n \cup \{\lim\}$ and such that

1. the set of non-default values taken by f is finite;
2. for each $x \in B$ with $\varsigma(x) \neq \lim$, $f(x) \in \varsigma(x)$;
3. if x is a parent vertex and a left child of x is in dom f, then $f(x) \neq d_e$, where d_e is the default value for the endpoint;
4. if x is a parent vertex and the right child of x is in dom f, then $f(x) = d_e$;
5. if $\varsigma(x) = \mathbb{Q}_n$ or $\dot{\mathbb{Q}}_n$ and dom f contains a left child of x with 'colour' m, then $f(x)$ has the colour m;
6. if $\varsigma(x) = \lim$, then $f(x) = \lim$.

Theorem 4.2 *Every coding tree encodes a linear order that is unique up to isomorphism.*

Proof We proceed as in [2]. Given a coding tree T, we construct an expanded coding tree which is associated with T as in Definition 2.6.

Let Σ_T be the set of decoding functions on T. For $f, g \in \Sigma_T$, define

$$f < g \text{ if } f(x_0) < g(x_0), \text{ where } x_0 \text{ is the greatest point for which } f(x) \neq g(x).$$

We show that $<$ is a linear ordering. Let f and g be decoding functions such that $f \neq g$. We want to show that if $f \neq g$ then there is a greatest $x_0 \in T$ such that $f(x_0) \neq g(x_0)$. Suppose that $\text{dom}(f) \neq \text{dom}(g)$. Then, since all coding trees are Dedekind-MacNeill complete, the supremum of the symmetric difference of dom(f) and dom(g) exists. We call this supremum s. Since $\text{dom}(f) \setminus \text{dom}(g)$ and $\text{dom}(g) \setminus \text{dom}(f)$ must lie in different cones of s, the label of s cannot be lim, and s has children. This means that dom(f) and dom(g) contain different children of s and so $f(s) \neq g(s)$. Therefore if $f \neq g$ then f and g disagree on $\text{dom}(f) \cap \text{dom}(g)$. They can only disagree finitely often, so there is a greatest x_0 such that $f(x_0) \neq g(x_0)$.

Then $f(x_0) < g(x_0) \Rightarrow f < g$ and $f(x_0) > g(x_0) \Rightarrow f > g$. It is clear this relation is irreflexive and transitive, hence $(\Sigma_T, <)$ is a linear order.

In order for T to encode $(\Sigma_T, <)$ according to Definition 2.7, we must produce an expanded coding tree associated with T. Such a tree is given by

$$E = \{(x, f \restriction (x, r]) : f \in \Sigma_T, x \in \operatorname{dom} f\}.$$

The tree ordering is given by letting $(x, f \restriction (x, r]) \leqslant (y, f \restriction (y, r])$ if $x \leqslant y \in \operatorname{dom} f$. In addition $(v_1, f \restriction (v_1, r])$ is level with $(v_2, g \restriction (v_2, r])$ if and only if v_1 is level with v_2. It is now clear that E is a levelled tree. Its root is $(r, \varnothing)$. Also, any $(x, f \restriction (x, r])$ lies above a leaf $(l, f \restriction (l, r])$ where l is a leaf in $\operatorname{dom} f$.

Each leaf-branch of E is isomorphic to a leaf-branch of T, and so it is Dedekind complete. Furthermore, since T contains all its ramification points, so does E, and therefore E is Dedekind-MacNeille complete.

We define an order on the children of a parent vertex $(x, f \restriction (x, r])$ in E that depends on the label of x in T. The children of $(x, f \restriction (x, r])$ have the form $(y, g \restriction (y, r])$, where $y \in \operatorname{child}(x) \subseteq T$ and g is any decoding function such that $x \in \operatorname{dom}(g)$ and $f \restriction (x, r] = g \restriction (x, r]$. Since for such a g we have that $g \restriction (y, r] = g \restriction [x, r]$, the order type of the children of $(x, f \restriction (x, r])$ is determined by

$$\{g(x) : g \in \Sigma_T,\ x \in \operatorname{dom} g,\ f \restriction (x, r] = g \restriction (x, r]\}\,.$$

Since x is a parent vertex, $g(x) \in \varsigma(x)$. Hence the label of x in T induces an order on the children of $(x, f \restriction (x, r])$. In the case of $\varsigma(x) = \mathbb{Q}_n$ or $\dot{\mathbb{Q}}_n$ we may say that the 'coloured' order type of the children in E is $\mathbb{Q}_n$ or $\dot{\mathbb{Q}}_n$.

If $(x, f \restriction (x, r])$ is neither a parent vertex nor a leaf, then x is neither a parent nor a leaf, and so x is labelled lim.

The association mapping ϕ is given by $\phi((x, f \restriction (x, r])) = x$. This preserves root, leaves and, as we have just seen, it preserves the relation between labels of vertices in T and the (coloured) order type of the children of those vertices in E. Also $x \leqslant y \Rightarrow \phi(x) \leqslant \phi(y)$, and it is clear that for each vertex x of E, ϕ maps $\{u \in E : u \leqslant x\}$ onto $\{u \in T : u \leqslant \phi(x)\}$ and for any leaf l of E, ϕ maps $[l, r]$ onto $[\phi(l), \phi(r)]$. Therefore E is associated with T, and Σ_T is order isomorphic to the set of leaves of E. Hence T encodes Σ_T.

A back-and-forth argument shows that any two countable linear orders encoded by the same coding tree $(T, \leqslant, \varsigma, \ll, \lhd)$ are isomorphic. □

Theorem 4.3 *The ordering Σ_T encoded by the coding tree $(T, \leqslant, \varsigma, \ll, \lhd)$ is countable and lower 1-transitive.*

Proof The way in which Σ_T has been defined ensures that it is countable.

We now show that Σ_T is lower 1-transitive. Take any $f, g \in \Sigma_T$ and consider the initial segments $(-\infty, f]$ and $(-\infty, g]$. We need to show that $(-\infty, f] \cong (-\infty, g]$.

Now, Σ_T is defined to be the set of all functions on the leaf-branches of T which take a default value at all but finitely many points. By definition of the ordering on Σ_T, an initial segment of Σ_T at f can be written as

$$(-\infty, f] = \{f\} \cup \{p \in \Sigma_T : (\exists x \in \operatorname{dom} f)(p(x) < f(x)) \wedge (\forall y > x)(p(y) = f(y))\}.$$

Let L_i be the ith level of the tree and let x_i^f denote the element of $\operatorname{dom} f$ on the level L_i. Then define

$$\Gamma_i^f = \{p \in (-\infty, f] : p(x_i^f) < f(x_i^f) \wedge (\forall y > x_i^f)(p(y) = f(y))\},$$

that is, the set of elements of Σ_T such that the greatest point on which they differ with f is on level i.

Then, by definition of the $\ll$ - ordering of the levels, it is clear that $(-\infty, f]$ is the disjoint union of all the Γ_i^f, and furthermore that $i \ll j \Rightarrow \Gamma_i^f > \Gamma_j^f$ (where this means that every element of Γ_i^f is greater than every element of Γ_j^f). Since the same holds for $(-\infty, g]$ and the Γ_i^g, to show that $(-\infty, f] \cong (-\infty, g]$, it suffices to show that there are $(\leqslant, \varsigma, \ll)$-isomorphisms $\Phi_i : \Gamma_i^f \cong \Gamma_i^g$ for each i. We then define the desired isomorphism from $(-\infty, f]$ to $(-\infty, g]$ to be $\bigcup_i \Phi_i$.

If $\varsigma(x_i^f) =$ lim, we have $\Gamma_i^f = \varnothing$. The label lim is not a linear order, so by condition 3 of Definition 2.4, if a vertex on level i is labelled lim, then all vertices on level i are labelled lim. This shows that when i is a level with vertices labelled lim, we have $\Gamma_i^f \cong \Gamma_i^g$.

We now consider the cases where the vertices on level i are not labelled lim. Since x_i^f and x_i^g are level, we have $\varsigma(x_i^f) \cong_l \varsigma(x_i^g)$, and so there is an isomorphism φ from $(-\infty, f(x_i^f)] \cap \varsigma(x_i^f)$ to $(-\infty, g(x_i^g)] \cap \varsigma(x_i^g)$. Moreover, there is an isomorphism ψ between the left forests at the points x_i^f and x_i^g. For $p \in \Gamma_i^f$, let x_j be the member of dom p at the level L_j. We now define

$$\Phi_i(p)(\psi(x_j)) = \begin{cases} g(x_j^g) & \text{if } j > i \\ \varphi(p(x_j)) & \text{if } j = i \\ p(x_j) & \text{if } j < i\,. \end{cases}$$

We must now show that Γ_i^f is mapped 1-1 into Γ_i^g by Φ_i. This gives our result. We have that $\Phi_i(p) \in \Sigma_T$, because all such $\Phi_i(p)$ are defined on leaf-branches of T and they take a default value at all but finitely many points, since both $g(x_j^g)$ and $p(x_j)$ take the default value at all but finitely many points (possibly with $\varphi(p(x_j))$ in addition).

It is easy to see that Φ_i is surjective. For injectivity, suppose $\Phi_i(p_1) = \Phi_i(p_2)$. Then, since $p_1, p_2 \in \Gamma_i^f$, we have $p_1(x_j) = p_2(x_j) = f(x_j^f)$ for all $j > i$ and $p_1(x_j) = p_2(x_j)$ for $j < i$ by the definition of Φ_i. Since φ is an isomorphism, $\varphi(p_1(x_i)) = \varphi(p_2(x_i))$ implies that $p_1(x_j) = p_2(x_j)$. Hence $p_1(x_j) = p_2(x_j)$ for all j. □

Theorems 4.2 and 4.3 conclude our classification of countable lower 1-transitive linear orders. The classification is involved and coding trees are a complicated classifier. However, the class of lower 1-transitive linear orders is wild, and the relationship between the classifier and the classified is descriptive and robust. The classification in this paper is crucial for that of countable 1-transitive trees [3]. The classification of colour lower 1-transitive linear orders is a natural extension of this work and a partial result in this direction also appears in [3].

Acknowledgments The authors wish to thank John Truss and Dugald Macpherson for their extensive help. We are also grateful to the anonymous referees for thorough and thoughtful comments.

References

1. Campero-Arena, G.: Transitivity properties of countable coloured orderings. Ph.D. thesis, Department of Pure Mathematics, University of Leeds (2002)

2. Campero-Arena, G., Truss, J.: Countable 1-transitive coloured linear orderings II. Fund. Math. **183**, 185–213 (2004)
3. Chicot, K., Truss, J.: Countable 1-transitive trees. In: Droste, M., Fuchs, L., Goldsmith, B., Strüngmann, L. (eds.) Groups, Modules and Model Theory - Surveys and Recent Developments, in Memory of Rüdiger Göbel. Springer (2009). To appear
4. Davey, B., Priestley, H.: Introduction to lattices and order. Cambridge University Press, Cambridge (2002)
5. Droste, M.: Structure of partially ordered sets with transitive automorphism groups. Mem. Amer. Math. Soc. **57**(334), (1985)
6. Droste, M., Holland, W., Macpherson, H.: Automorphism groups of infinite semilinear orders (I and II). Proc. Lond. Math. Soc. **58**, 454–494 (1989)
7. Morel, A.: A class of relation types isomorphic to the ordinals. The Michigan Mathematical Journal **12**, 203–215 (1965)