Contemporary Mathematics

A Language Theoretic Analysis of Combings

Sarah Rees

ABSTRACT. A group is combable if it can be represented by a language of words satisfying a fellow traveller property; an automatic group has a synchronous combing which is a regular language. This paper gives a systematic analysis of the properties of groups with combings in various formal language families, and of the closure properties of the associated families of groups; it explores the relationship between properties of the formal languages and those of the groups. It builds on previous work, in particular of Epstein et al. and Bridson and Gilman.

1. Introduction

The concept of a combing for a finitely generated group has grown out of the definition of an automatic group (as introduced in [**12**]); in the terminology of this paper, a group is automatic precisely when it possesses a regular synchronous combing. Roughly speaking, a combing is an orderly set of strands through a Cayley graph of the group, and is regular if it is defined by a finite state automaton; a formal definition is given in Section 2.

The family of automatic groups has valuable properties (particularly from a computational point of view) and contains a wealth of examples. However the absence from this family of particular groups, such as the fundamental groups of compact manifolds modelled on the Nil or Sol geometries (see [**6, 7, 12**]; the groups of all other compact geometrisable 3-manifolds are automatic) and any nilpotent group which is not virtually abelian, seems sufficient reason to examine generalisations of the definition.

The study of 'combings' associated with formal languages other than the regular languages associated with automatic groups is motivated by the proof by Bridson and Gilman ([**10**]) that the fundamental group of any compact geometrisable 3-manifold possesses an asynchronous combing which is an indexed language (as defined in [**1**]). The class of groups with combings of this type is defined by weakening both of the two restrictions on the 'language' associated with an automatic group; the geometric 'fellow traveller condition' is relaxed from a synchronous to

1991 *Mathematics Subject Classification.* Primary 20F10, 20-04, 68Q40; Secondary 03D40.

The author would like to thank the Fakultät für Mathematik of the Universität Bielefeld for its warm hospitality while this work was carried out, the Deutscher Akademischer Austauschdienst and EPSRC (grant no. GR/K80150) for financial support.

an asynchronous condition, and the language theoretic requirement of regularity is replaced by the requirement that the language be indexed (that is, recognised by a one-way nested stack automaton, a type of machine, defined in [**2**], which is a little more general than a pushdown automaton). Bridson and Gilman's techniques can also be used to construct combings for many further nilpotent and soluble groups (see [**15**]). These combings are sometimes also indexed languages, and are always real-time languages (that is, are recognised by Turing machines which halt as they finish reading their input; see [**26**] for a formal definition).

Naturally some properties of the family of automatic groups are lost by the move to a more general definition; however, widening the definition admits more constructions within a family of combable groups, and so allows more examples. Our aim in this paper is to give a systematic analysis of the properties of families of synchronously and asynchronously combable groups defined by various types of formal languages, to see to what extent properties of the family of automatic groups still hold in these other families, and what further properties can be deduced. We show the connection between properties of the language classes and properties of the related families of groups. We build on results in [**12**], [**10**], [**8**] and [**22**], in particular the first two of those. Often the proofs of these papers remain valid with a small amount of adjustment in the more general setup we need; details from the original proofs are sometimes repeated for greater clarity. Our goal is to prove these results in the most general context possible, the reason for this being that we want to develop the machinery for a formal language based family of combings without the limitations of a preconception that a particular formal language family is the right one to work with.

Most of the recent more general results on combings have concentrated on those which, as in Bridson and Gilman's case, are asynchronous; these are certainly easier to find than synchronous combings. However, from a practical point of view, synchronous combings, when they exist, are more useful. The value of the study of these is increased by a very recent, and as yet unpublished, result of Bestvina and N. Brady, which constructs a synchronous combing for a non-automatic group. Hence in this article we take care to observe when the construction of synchronous combings is possible for the groups under consideration.

Section 2 defines the terminology of this article; the first part is devoted to the definitions of various types of combings and fellow traveller properties while the second introduces the formal language theory which is used. Section 3 discusses some of the losses in moving beyond automatic groups and away from regular languages. Section 4 looks at the closure of families of combings under change of generators and various group operations (such as passage to finite index subgroups, and certain kinds of products and extensions), showing how existing proofs of the analogous results for (usually) automatic groups can be modified for use in a more general language theoretic scenario.

The recent article [**5**] also studies families of groups obtained by weakening the language theoretic requirements of automatic groups, and so should certainly be mentioned here, for comparison. There, the family of parallel polypushdown groups is defined; such a group is associated with a set of context-free languages (and hence a set of pushdown automata); the significant difference between this definition and that of the combable groups considered here is a further weakening of the fellow traveller condition; the fellow traveller condition satisfied by combable groups can in fact be stated in terms of finite state automata (although a geometric statement often seems more natural), while the analogous condition satisfied by parallel polypushdown groups relates to the more general concept of pushdown automata. Parallel polypushdown groups, like combable groups, have soluble word problem, but, unlike combable groups, need not be finitely presented. We shall not consider this concept further in this article.

2. Definitions and notation

2.1. Fellow traveller properties, languages and combings. Let G be a group, with identity element e, and finite generating set X. Without loss of generality, we may assume that X is inverse closed, that is, contains the inverse of each of its elements; we shall make this assumption thoughout this paper. We call a product of elements in X a word over X, and denote by X^* the set of all such words. We use the symbol ϵ to denote the empty word (of length 0). Let $\Gamma = \Gamma_{G,X}$ be the Cayley graph for G over X, with vertices corresponding to the elements of G, and, for each $x \in X$, a directed edge from the vertex g to the vertex gx, labelled by x. Let $d_{G,X}$ measure (graph theoretical) distance between vertices of $\Gamma_{G,X}$. For words $w, v \in X^*$, we write $w = v$ if w and v are identical as words, $w =_G v$ if w and v represent the same element of G. We define $l(w)$ to be the length of w as a string, and $l_G(w)$ to be the length of the shortest word v with $v =_G w$. It is straightforward to extend $d_{G,X}$ to a differentiable metric on the 1-skeleton of Γ. Then each word w can be associated with a differentiable path from e labelled by w and parameterised by the non-negative real numbers in such a way that, for $t < l(w)$, the path from e to $w(t)$ has length t, and for $t \geq l(w)$, $w(t) = w(l(w))$.

Suppose that v, w are words in X^*, and that $K \in \mathbf{N}$. Various *fellow traveller* properties can describe the relationship between v and w. We say that v and w *synchronously* K-fellow travel if for all t, $d_{G,X}(v(t), w(t)) \leq K$. More generally, we say that v and w *asynchronously* K-fellow travel if there is a strictly increasing, differentiable function $h : \mathbf{R} \to \mathbf{R}$, mapping $[0, l(v) + 1]$ onto $[0, l(w) + 1]$, with the property that, for all $t > 0$, $d_{G,X}(v(t), w(h(t))) \leq K$. We call h a *relative speed* function of v and w. We say that v and w asynchronously K-fellow travel with *asynchronicity bound* M, if for all $t > 0$ an associated relative speed function h satisfies $1/M \leq h'(t) \leq M$. Note that v and w synchronously fellow travel if and only if they asynchronously fellow travel with relative speed function h defined by $h(t) = t$.

We define a *language* for G over X to be a set of words in X which contains at least one representative for each element of G. A language is *bijective* if it contains exactly one representative of each group element, *prefix closed* if it contains every prefix of each of its elements. It is *geodesic* if for each $w \in L$, $l(w) = l_G(w)$, and *near geodesic* if, for some constant δ, for all $w \in L$, $l(w) - l_G(w) < \delta$ (this is stronger than being quasigeodesic, see [**12**]).

As an example, consider the free abelian group $G = \langle a, b \mid ab = ba \rangle$, isomorphic to $\mathbf{Z}^2$. The Cayley graph for G is simply the lattice $\mathbf{Z}^2$, which embeds naturally in $\mathbf{R}^2$. The set $L_1 = \{a^m b^n : m, n \in \mathbf{Z}\}$ is a geodesic, prefix closed, bijective language for G over $X = \{a, a^{-1}, b, b^{-1}\}$. The larger set L_2 of all words over X in which all powers of a have the same sign, and similarly all powers of b, is a geodesic, prefix closed language for G, but is not bijective. A geodesic, but not prefix closed, language L_3 (see [**9**]) is constructed using the embedding of $\mathbf{Z}^2$ in $\mathbf{R}^2$; the group element g is represented by the words labelling paths in $\mathbf{Z}^2$ which run as close as possible to the straight line path in $\mathbf{R}^2$ from the origin to g. The language L_3 is not bijective, but bijective sublanguages can easily be described.

Suppose that L is a language for G. We call L a *synchronous combing* if for some constant K (the *fellow travelling constant* for L), the K-fellow traveller condition is satisfied by all pairs of words $v, w \in L$ for which $w =_G vx$ for some $x \in X \cup \{e\}$. If the K-fellow traveller condition is also satisfied by all pairs of words xv, w with $v, w \in L$ and $w =_G xv$, for $x \in X$, then we call L a *synchronous bicombing*. Alternatively, we call L an *asynchronous combing* if for some K the asynchronous K-fellow traveller condition is satisfied by all pairs of words $v, w \in L$ for which $w =_G vx$ for some $x \in X \cup \{e\}$, and an *asynchronous bicombing* if the same is also true for words xv, w with $v, w \in L$ and $w =_G xv$, for $x \in X$. We say that an asynchronous combing (or bicombing) L is *boundedly asynchronous* if for some constants K, M, relevant pairs of words asynchronously K-fellow travel with asynchronicity bound M. (Boundedly asynchronous combings are important for the study of asynchronously automatic groups in [**12**].) The language L_1 described above is a synchronous bicombing for $\mathbf{Z}^2$, since words of the form $a^m b^n$ clearly fellow travel (at distance at most 2) with words of the form $a^{m+1}b^n$ and $a^m b^{n+1}$. The language L_2 is neither a synchronous nor an asynchronous combing, since for any m the words $a^m b^m$ and $b^m a^m$, which represent the same group element, are distance $2m$ apart at their midpoints. The language L_3 is a synchronous bicombing.

When we state results in this article, we need to be very clear about the various fellow traveller conditions which are currently under discussion. Hence we shall consider the words *combing* and *bicombing* to have a rather general meaning, when used without further qualification. We introduce the notion of *type* of a combing for situations where we need to be more precise.

More specifically, we shall use the term *combing* for a language which is either a synchronous, asynchronous, or boundedly asynchronous combing or bicombing, and call a group *combable* if it has a combing. Similarly, we shall use the general term *bicombing* for a language which is either a synchronous, asynchronous, or boundedly asynchronous bicombing, and call a group *bicombable* if it has a bicombing. A *type*, τ, of combing will be one of the six families of combings (asynchronous, boundedly asynchronous or synchronous combings or bicombings) we have defined above; for any such τ, we shall say that a combing in the family τ *has type* τ. (Clearly a combing may therefore have more than one type.) The following inclusions between these types are fairly clear:-

$$\begin{array}{ccccc} \text{sync. combings} & \subset & \text{bdd. async. combings} & \subset & \text{async. combings} \\ \cup & & \cup & & \cup \\ \text{sync. bicombings} & \subset & \text{bdd. async. bicombings} & \subset & \text{async. bicombings} \end{array}$$

We call fellow traveller properties holding between pairs of words v, w with $w =_G v$ or $w =_G vx$ *right* fellow traveller properties, and fellow traveller properties holding between pairs of words v, w with $w =_G xv$ *left* fellow traveller properties.

We call a combing *bijective*, *prefix closed*, *geodesic* or *near geodesic* if the underlying language has those properties. Note that the literature contains various uses of the term combing more restrictive than ours; [**12**], for instance, requires that the language consist of quasigeodesics, [**8**], [**27**] and [**10**] require that the language be bijective, and [**13**] and [**27**] that it be synchronous. Although we use a more general definition, when we construct combings out of other combings, it will usually be clear from the construction that properties such as bijectivity, prefix closure, geodesity or near geodesity possessed by the original combings would be inherited by the new ones. Moreover, in general the constructions are not dependent on those properties.

From [**8**], we observe that any asynchronously combable group is finitely presented, has an exponential isoperimetric inequality, and hence (by, for instance, [**12**], Theorem 2.2.5) soluble word problem. From [**13**], we observe that a group with a prefix closed synchronous combing has a quadratic isoperimetric inequality, and from [**27**], where isoperimetric and isodiametric inequalities for synchronously combable groups are also examined, that a group with a synchronous bicombing has soluble conjugacy problem. (Although bijectivity of the combing is assumed in all of [**8**], [**13**] and [**27**], the results are not dependent on that assumption.) Note that, since finitely presented soluble groups of derived length 3 are known with insoluble word problem ([**21, 4**]), it is clear that finitely presented soluble groups of derived length 3 exist which are not asynchronously combable.

A natural restriction to put on a combing is to require that it be a language in one of the families of formal languages, that is, a language recognised by some theoretical model of computation (or equivalently, defined by a formal grammar). The combing of an automatic group is required to be a regular language. Bridson and Gilman in [**10, 11**] studied bijective, asynchronous combings in various formal language families, in particular the families of bounded languages, regular languages, context-free languages and indexed languages, and more generally any full abstract family of languages (of which the last three are examples, see [**20**]). We aim to give a more general analysis, covering both a range of fellow traveller conditions (synchronous and even boundedly asynchronous conditions are in general harder to obtain than asynchronous ones) and a range of language families.

2.2. Families of languages. In general, for any finite set X, whether or not X is associated with a group, we define a *language* with *alphabet* X to be a subset of X^* (that is, a set of *words* over X). Suppose that X' is a second finite set of the same size as X and that $f : X \to X'$ is the monoid isomorphism induced by a bijection from X to X', and mapping ϵ to ϵ. Then the language $f(L)$ is said to be *naturally equivalent* to L. We define a *family* $\mathcal{F}$ of languages to be any finite set of languages over finite alphabets with the property that if L is in $\mathcal{F}$ then any language L' which is naturally equivalent to L is also in $\mathcal{F}$. (This condition does not actually form a part of the definition in [**20**].)

Primarily we are interested in combings associated with families of *formal languages*. By a formal language with alphabet X, we mean a language which is recognised by some form of theoretical computer. Such a computer takes words over X as input, and for each such word uses some kind of decision process to decide whether to accept it; the set of accepted words forms the language of the computer. Different types of computer define different families of formal languages. The simplest kind of theoretical computer is a finite state automaton, and a language is called regular if it is accepted by such a computer; the most complex kind of theoretical computer is a Turing machine, for which the associated language is a recursively enumerable set.

Our results will look at the closure properties of the families of languages being considered, and deduce from these closure properties of the associated families of combable groups. Note that our definition of language family includes the set of all languages with finite alphabet. It will be clear that this family of languages (which is not a formal language family) is closed under all the relevant operations, and hence that our results hold also for combings for which there is no language theoretic restriction.

It is not necessary for the purposes of understanding this paper to have a detailed knowledge of formal languages, or of the computers which define them. It suffices to have a basic understanding of finite state automata. A finite state automaton reads each input word, one symbol at a time, from left to right, and, as it does so, fluctuates between the states in a finite set S. At the beginning, the automaton is in a particular *start* state; it accepts a word if after reading the whole word it is in one of a finite number of *accept* states. At any stage during the reading of a word, the decision to move from one state to another depends only on the current state of the automaton and the symbol which has just been read. Such an automaton can easily be represented by a finite graph, of which the states are the vertices, and for which transitions between vertices are represented by directed edges, labelled by alphabet symbols. The language of the automaton is then defined to contain any word which labels a path through the graph from the vertex representing the start state to a vertex representing an accept state. We allow the possibility that some words label more than one path (in which case the automaton is *non-deterministic*). We choose to exclude from the graphical representation any non-accept state from which no sequence of transitions leads to an accept state, and also to exclude any transition to or from such a state. Figure 1 represents an automaton which accepts the language L_1 described in Section 2.1. (Following standard notation, a free arrow points to the start state and accept states are ringed. In this example all vertices represent accept states.) Hence L_1 is a regular combing for $\mathbf{Z}^2$, and $\mathbf{Z}^2$ is automatic (the language L_2 of Section 2.1 is also regular, but L_3 is not).

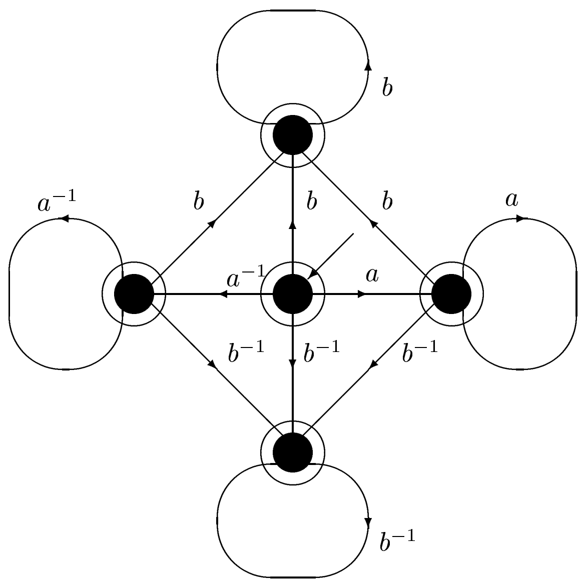

FIGURE 1. A finite state automaton recognising L_1

For more details of regular languages and the other families of formal languages mentioned in this paper, we refer the reader to the literature. A very useful introduction to formal language theory from a group theoretical standpoint is provided by [**14**]. A thorough treatment, which covers many families of formal languages, can be found in [**20**]; this covers the four families of the Chomsky hierarchy (regular sets, context-free languages, context-sensitive languages and recursively enumerable sets), as well as the recursive sets, and (briefly) the indexed languages. [**16**] provides a more detailed mathematical treatment of context-free languages, which are defined by pushdown automata. The indexed languages were originally defined and are more fully described in [**1**] and [**2**]. These languages are probably best understood through the grammatical rules which define them (described in [**1**] and [**10**]), rather than through the machines (one-way nested stack automata) which recognise them. As a family of languages they lie between the context-free and context-sensitive languages. Bounded languages are described in [**18**], and real-time languages in [**26**].

From now on, we shall use the symbol $\mathcal{F}$ to denote a family of languages, and in particular Rg to denote the family of regular sets, Cf the context-free languages, DCf the deterministic subfamily of Cf, Ind the indexed languages, Cs the context-sensitive languages, Rt the real-time languages, Rc the recursive sets, RcE the recursively enumerable sets, and $\mathcal{U}$ the family of all languages over finite alphabets. We call a combing for a group G an *$\mathcal{F}$-combing* if it is a combing in the family $\mathcal{F}$ of languages. Of course, every combing is a $\mathcal{U}$-combing.

The usefulness of an $\mathcal{F}$-combing for a group may well depend on the decidability of certain problems for languages in $\mathcal{F}$; the decidability properties of languages in various formal language families are shown in Table 1. For the purposes of this article we are interested also in the closure properties of the various families of formal languages, that is, questions such as whether or not the intersection of two languages in a family $\mathcal{F}$, or the concatenation of two such languages, is also in $\mathcal{F}$; Table 2 lists the relevant closure properties of various formal language families.

The definitions of most of the closure properties referred to in Table 2 are straightforward, and are given within that table. The properties related to homomorphisms and GSM mappings may need some clarification. Suppose that X and Y are finite alphabets and that L is a language over X. We define a *homomorphism* from L to Y^* to be the restriction to L of a semigroup homomorphism θ from X^* to Y^* which maps ϵ to ϵ. Such a homomorphism is said to be *ϵ-free* if θ maps no string in X^* to ϵ, apart from ϵ itself, or *ϵ-bounded* (or to give *limited erasing*) if, for some constant K, and for any string $a \in L$, at most K consecutive symbols of a are mapped to ϵ by θ. A *generalised sequential machine (GSM) mapping* from L to Y^* is defined (as in [**20**]) in terms of a generalised sequential machine, and generalises the idea of a homormorphism. A *generalised sequential machine* is basically a (possibly non-deterministic) finite state automaton with attached output string. Corresponding to each transition between states of the underlying automaton is a (possibly trivial) word which is appended to the output string as the transition is made. If the word is accepted, the word currently on the output string is output as the image of that word under the GSM mapping; otherwise no image is defined. The image of a language under a GSM mapping is defined to be the set of all defined images of its elements. A GSM mapping is called ϵ-free if non-trivial output is associated with every transition of the underlying automaton. An elementary example of a GSM mapping which is not a homomorphism is given by the mapping from $\{a,b\}^*$ to $\{a,b\}^*$ which replaces every second occurrence of the symbol a in any word by the word bab, but otherwise leaves the word unchanged. Of course, in this example, every input string is accepted, and so has an image under the mapping. Figure 2 gives a graphical representation of the associated generalised sequential machine; each transition is labelled by a pair of symbols of which the first is the input symbol and the second the output symbol.

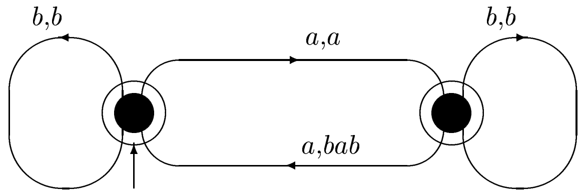

FIGURE 2. A GSM mapping

Most of the information in Tables 1 and 2 comes from [**20**]; some results for indexed languages come from [**1, 2, 19**], and the properties of real-time languages from [**26**]. Following [**17**], a detailed study of formal languages through their closure properties is given in [**20**]. In recognition of the fact that some closure properties are consequences of others, the concepts of abstract and full abstract families of languages, and also of trios and full trios, for which various closure properties are satisfied, are introduced there. Where the properties $P_{\cap \mathsf{Rg}}$ etc. are as defined in Table 2, a family of languages is called a *trio* if it satisfies $P_{\cap \mathsf{Rg}}$, $P_{\epsilon-free\,h}$ and $P_{h^{-1}}$, a *full trio* if it is a trio which also satisfies P_h, an *abstract family of languages* (AFL) if it is a trio which also satisfies $P_\cup$, $P_\circ$ and P_+, and a *full* abstract family of languages if it is a full trio which also satisfies $P_\cup$, $P_\circ$ and P_*. The families Rg, Cf, Ind and RcE are full AFL's; Rc and Cs are AFL's ([**20, 10**]). DCf and Rt are not even trios. In particular, Rt has quite different closure properties from most of the other language families mentioned in this article, and so many of the results of this paper do not obviously apply. Despite this, in general, the proofs can be modified to deal with Rt. We give no details of that here, since real-time languages are separately considered in [**15**].

3. Using regularity to find better combings and solve the word problem fast

If G is an automatic group with appropriate language L (over X), then it is well known (and described in [**12**]) that we can always find a regular synchronous combing L_0 for G which is bijective; we simply select the subset of L consisting of the minimal representatives of the group elements under the shortlex word order, $\preceq$ (for which $w \preceq v$ if w is shorter than v, or the same length as v and preceding it lexicographically). Further, it can be proved that L_0 is a quasigeodesic language. Hence, using L_0, we can construct a quadratic time algorithm to solve the word problem in G. An outline of the algorithm appears below; the details are described in [**12**].

Property	Definition	True in
$P_\emptyset$	Emptiness of a language in $\mathcal{F}$ is decidable.	Rg, DCf, Cf, Ind, not Cs.
$P_\in$	Membership of a language in $\mathcal{F}$ is decidable. Implied by combination of $P_\emptyset$ and $P_{\cap \mathsf{Rg}}$ (defined in Table 2).	Rg, Cf, DCf, Ind, Rc.
P_∞	Finiteness/infiniteness of a language in $\mathcal{F}$ is decidable.	Rg, Cf, Ind.
$P_=$	Equality of two languages in $\mathcal{F}$ is decidable.	Rg, not Cf, Ind.

TABLE 1. Decidability properties of formal languages

Property	Definition	True in
$P_\cap$	The intersection of two languages in $\mathcal{F}$ is in $\mathcal{F}$.	Rg, Rt, Cs, Rc, RcE, not DCf, Cf, Ind.
$P_\cup$	The union of two languages in $\mathcal{F}$ is in $\mathcal{F}$.	Any AFL, Rt, not DCf.
$P_\circ$	The concatenation of two languages in $\mathcal{F}$ is in $\mathcal{F}$.	Any AFL, not DCf, Rt.
P_+	The positive closure $L^+ = \cup_{n>0} L^n$ of a language L in $\mathcal{F}$ is in $\mathcal{F}$.	Any AFL, not DCf, Rt.
P_*	The Kleene closure $L^* = \cup_{n\geq 0} L^n$ of a language L in $\mathcal{F}$ is in $\mathcal{F}$.	Any full AFL, not DCf, Rt, Cs.
P_c	For any $L \subset X^* \in \mathcal{F}$, $X^* \setminus L$ is in $\mathcal{F}$.	Rg, DCf, Rt, Rc, not Cf, Ind, RcE.
$P_{\cap \mathsf{Rg}}$	The intersection of a language in $\mathcal{F}$ with a regular set is in $\mathcal{F}$.	Any trio, DCf, Rt.
P_h	The image of a language in $\mathcal{F}$ under a homomorphism is in $\mathcal{F}$.	Any full trio, not DCf, Rt.
$P_{\epsilon-free\,h}$	The image of a language in $\mathcal{F}$ under an ϵ-free homomorphism is in $\mathcal{F}$.	Any trio, not DCf.
$P_{\epsilon-bdd\,h}$	(limited erasing) The image of a language in $\mathcal{F}$ under an ϵ-bounded homomorphism is in $\mathcal{F}$.	Any trio.
$P_{h^{-1}}$	If θ is a homomorphism from X^* to Y^*, and $L \in \mathcal{F}$ is a language over Y, then $\theta^{-1}(L) \in \mathcal{F}$.	Any trio, DCf, Rt.
P_{GSM}	The image of a language in $\mathcal{F}$ under the operation of a generalised sequential machine (GSM) mapping is in $\mathcal{F}$. Implies in particular P_h, $P_{\cap \mathsf{Rg}}$.	Any full trio.
$P_{\epsilon-free\,GSM}$	The image of a language in $\mathcal{F}$ under the operation of an ϵ-free GSM mapping is in $\mathcal{F}$. Implies in particular $P_{\epsilon-free\,h}$, $P_{\cap \mathsf{Rg}}$.	Any trio.

TABLE 2. Closure properties of formal languages

What happens when we generalise from automatic groups to other types of combings? Given any type of combing L for a group G, we can still construct a bijective combing L_0 of the same type as a sublanguage, just as above, but there is in general no reason to suppose that L_0 should lie in the same formal language class as L_0, or that it should be quasigeodesic. In fact, we should not in general expect a quasigeodesic combing for G to exist. For, by [**8**], the existence of such a combing implies a quadratic upper bound on the Dehn function for G, and it is not hard to find non-automatic combable groups with above quadratic Dehn function (such as the Baumslag-Solitar groups, which are asynchronously automatic, with exponential Dehn function, and the free class c nilpotent groups, which are real-time combable, with polynomial Dehn functions of degree $c+1$). The algorithm to solve the word problem described in [**12**] can certainly be applied using any recursively enumerable combing for G; but we cannot expect it to run in quadratic time in general.

The regularity of L_0 and the fact that it is quasigeodesic in the case where G is automatic are basically consequences of particular properties of the regular languages. We give some details of this below. The construction of L_0, the fact that it is quasigeodesic and the algorithm to solve the word problem depend on the use of the *difference machine* D_K, which can be used to recognise whether or not two words synchronously K-fellow travel, so we shall start by describing this.

The difference machine D_K is a finite state automaton, with alphabet the set of ordered pairs from $X \cup \{e\}$. Its accept states correspond to *word differences*, defined as the words of length at most K which connect the same length prefixes of a pair of fellow travelling words (that is, which label minimal length paths between those prefixes in the Cayley graph). Suppose that d is a word difference and that $x, x' \in X \cup \{e\}$. If some word difference d' is equal in G to $x^{-1}dx'$, then there is a transition from d to d' in D_K on the pair (x, x'). If there is no such word difference d', then transition from d on (x, x') is to the sole non-accept state of D_K. The non-accept state is a fail state, that is, there are no transitions away from it. Any pair (w, v) of words over X can (by first padding out the shorter or the two words on the right with copies of e to get two words of the same length) be interpreted as a sequence of elements from the alphabet of D_K, and hence as valid input to that machine. D_K accepts a pair of words (w, v) if w and v synchronously K-fellow travel. Further, the target state of (w, v) corresponds to the word difference $w^{-1}v$, and hence D_K can also be used to recognise equality or simple relationships between pairs of fellow travelling words.

A machine can also be built to recognise pairs of words which fellow travel asynchronously. However this is no longer a finite state automaton.

Using the difference machine D_K of any (synchronous or otherwise) combing L, we can define a bijective combing L_0 by

$$L_0 = \{v \in L : \nexists w \in L, ((w, v) \in L(D_K), w \preceq v)\}$$

Since the set of pairs of words (w, v) with $w \preceq v$ can be recognised by a finite state automaton, the fact that L_0 is regular if L is a regular synchronous combing follows from the closure of the family of regular languages under boolean operations and the application of the existential operator. Most formal language families are not closed under all these operations.

When L is a regular synchronous combing, then the language

$$L' = \{(w, v) : w \in L, v \in L, (w, v) \in L(D_K)\}$$

(viewed as a language over the alphabet of ordered pairs from $X \cup \{e\}$) is also regular. In fact, from this, finite state automata usually known as multipliers can be constructed, which recognise multiplication by the elements of $X \cup \{e\}$; the existence of mulipliers is often taken instead of the fellow traveller property as the second defining property of an automatic group. The pumping lemma for regular languages (see [**20**]), applied to L', implies the existence of a bound M such that, given any $x \in X$, and any $w \in L$, some representative in L of wx is at most M symbols longer than w; hence L contains a quasigeodesic sublanguage (and L_0, since it is bijective, must itself be a quasigeodesic language). The proof of this fact depends heavily on the fact that L' is a regular language, and on the very strong pumping lemma for those languages; in the case where G is combable but not automatic there is not even any particular reason why L' should be in the same language family as L_0.

Whenever G has a recursively enumerable combing L, the practical algorithm to solve the word problem, described in [**12**] for automatic groups, can be applied. Given a word v in the generators of G, the algorithm finds first $w_e \in L$, with $w_e =_G e$, then $w_v \in L$, with $w_v =_G v$, and finally tests to see if $w_v =_G w_e$. Each of these phases breaks up into basic steps, each of which depends on the ability, given $u \in L$ and $x \in X$, to find $u' \in L$, with $u' =_G ux$; this can be done with the help of the difference machine. Where G is automatic, the fact that L can be chosen to be quasigeodesic ensures that the length of the elements of L encountered during the algorithm is bounded by a linear function of $l_G(v)$, and hence that the number of the steps in the algorithm is quadratically bounded.

4. The closure properties of families of combable groups

4.1. Finite variations and change of generators. In this section we examine how moving to a group related by some kind of finite variation to a given group G or changing the generating set for G affects combability.

The following technical lemma will be useful.

LEMMA 4.1. *Suppose that G is a group with finite generating sets X and Y, and that L is a language for G over X. Suppose that ϕ is a map from L into Y^*, and that λ and C are positive constants such that, for any $v \in L$, and any t,*

$$d_{G,Y}(\phi(v)(t), v(\lambda t)) < C.$$

Then any fellow traveller properties satisfied by L over X are satisfied by $\phi(L)$ over Y.

Note: it is not required that $\phi(v) =_G v$.

PROOF. We start by fixing a set of words $\{w_x : x \in X\}$ over Y, such that $w_x =_G x$, and a set of words $\{w_y : y \in Y\}$ over X, such that $w_y =_G y$. Let M be the maximum length of any such w_x or w_y. Suppose that v, w are two words in L, and $\phi(v), \phi(w)$ their images in $\phi(L)$.

Our first step is to verify that whenever v and w fellow travel in some way, then $\phi(v)$ and $\phi(w)$ do too. So suppose that v and w asynchronously fellow travel over X, with relative speed function h, and fellow traveller constant K. Define

$H : \mathbf{R} \to \mathbf{R}$ by $\lambda H(t) = h(\lambda t)$. Then

$$\begin{aligned} d_{G,Y}(\phi(v)(t), \phi(w)(H(t))) &\leq d_{G,Y}(\phi(v)(t), v(\lambda t)) + d_{G,Y}(\phi(w)(H(t)), w(\lambda H(t))) \\ &\quad + d_{G,Y}(v(\lambda t), w(\lambda H(t))) \\ &\leq 2C + M d_{G,X}(v(\lambda t), w(h(\lambda t)) \leq 2C + MK. \end{aligned}$$

So $\phi(v)$ and $\phi(w)$ asynchronously fellow travel over Y with relative speed function H and fellow traveller constant $2C + MK$. Since $h' = H'$, we see that $\phi(v)$ and $\phi(w)$ boundedly asynchronously fellow travel over Y if and only if v and w do so over X. If $h(t) = t$ for all t, then $H(t) = t$ for all t, and so further, when v and w synchronously fellow travel, so do $\phi(v)$ and $\phi(w)$.

We now deduce from the above that $\phi(L)$ inherits the fellow traveller properties of L. We deal with right and left fellow traveller properties separately.

To see that $\phi(L)$ inherits right fellow traveller conditions from L, suppose that $v, w \in L$ and that $\phi(w) =_G \phi(v)$ or $\phi(w) =_G \phi(v)y$, for some $y \in Y$. Then $\phi(v) =_G vg$, for some $g \in G$ which has length at most C as a word over Y, and hence length at most MC as a word over X; similarly, for some g' of length at most C as a word over X, $\phi(w) =_G wg'$. So $d_{G,X}(v, w) < 2MC + 1$, and hence if L satisfies a right fellow traveller condition with fellow traveller constant K, v and w must fellow travel in the appropriate way at distance at most $K(2MC+1)$. The above then implies that $\phi(v)$ and $\phi(w)$ similarly fellow travel (with an appropriate constant).

Now suppose that L satisfies both right and left fellow traveller conditions. We want to show that $\phi(L)$ then satisfies left fellow traveller conditions. Suppose that $v, w \in L$ and that $\phi(w) =_G y\phi(v)$ for some $y \in Y$. Then $\phi(v) =_G vg$ and $\phi(w) =_G wg'$, for g, g' as above, so $yvg =_G wg'$. Then the right and left fellow traveller properties of L together imply that v and w fellow travel in the appropriate way at a distance of at most $K(2MC+1)$. Hence $\phi(v)$ and $\phi(w)$ similarly fellow travel. □

The results in the following proposition are mostly generalisations of results for automatic groups in [**12**]. Most of these results are also proved in a more general language theoretic setup (but not as general as ours) in [**10**] for asynchronous (but not synchronous) combings.

Proposition 4.2 (Finite variations and change of generators). *Let $\mathcal{F}$ be a family of languages, and τ one of the 6 families of combings of Section 2.*

- **(a):** *Let N be a finite normal subgroup of G. If G has an $\mathcal{F}$-combing of type τ, then so does G/N. Conversely, if $\mathcal{F}$ is closed under concatenation with a finite set, and G/N has an $\mathcal{F}$-combing of type τ, then so does G.*
- **(b):** *Suppose that $\mathcal{F}$ is closed under concatenation with a finite set. Then if G has an asynchronous, boundedly asynchronous or synchronous $\mathcal{F}$-combing, so does any group J of which G is a subgroup of finite index.*
- **(c):** *Suppose that $\mathcal{F}$ is closed under ϵ-free GSM mappings, and limited erasing. Then, if G has an $\mathcal{F}$-combing L of type τ,*
  - **(i):** *G has an $\mathcal{F}$-combing of type τ over any finite generating set,*
  - **(ii):** *any subgroup H of finite index in G has an $\mathcal{F}$-combing of type τ. If L is bijective then so is the combing for H.*

Note that all the language theoretic conditions of this result are satisfied by Rg, Cf, Ind, Cs, Rc, RcE, and of course the family $\mathcal{U}$ of all languages with finite alphabet.

Proof. To prove part (a), let ν represent the natural homomorphism from G to its quotient G/N.

Let $X = \{x_1, \ldots, x_n\}$ be a generating set for G, and L an $\mathcal{F}$-combing of type τ for G over X. Then the set $X' = \{x_1^\nu, \ldots x_n^\nu\}$ (in which no identifications are made between images of distinct elements of X which are equal in the quotient group) generates G/N. Define L' to be the set of images of elements of L under ν. As a language L' is naturally equivalent to L. It is straightforward to verify, using the finiteness of N, that L' inherits right and left fellow traveller properties from those of L (the fellow traveller bound for L' is at most $M+1$ times that of L, where M is the maximal length of an element of N expressed as a word over X).

Now to prove the converse statement, suppose that Y is a generating set for G/N, and that now L' is an $\mathcal{F}$-combing of type τ for G/N. Let Z be a set of elements of G which maps bijectively to Y under ν. Then $Z \cup N$ generates G. Choose L'' to be the natural set of words over Z which maps bijectively under ν to L'. L'' is naturally equivalent to L' as a language. It is clear that the concatenation of L'' with N is a language in $\mathcal{F}$ for G over $Z \cup N$, and right and left fellow traveller properties are easily seen to be inherited by $L''N$ from L'.

Now to prove part (b), let L be an $\mathcal{F}$-combing for G over an alphabet X. Let T be a set of symbols representing a left transversal for G in J. Certainly the concatenation L' of L with T is a language in $\mathcal{F}$ for J over $X \cup T$. Now suppose that $v = ut$ and $w = u't'$ are two words in that language, with $u, u' \in L$, $t, t' \in T$. If $w =_J v$, then $u =_G u'$. If $w =_J v\xi$, for some $\xi \in X \cup T$, then $u' =_G uy_{t,\xi}$, where $y_{t,\xi} \in L$ is a word over X representing a Schreier generator of G, that is, satisfying the rule $t\xi =_G y_{t,\xi}t_{t,\xi}$, for $t_{t,\xi} \in T$. Since the set of such generators is finite, $y_{t,\xi}$ can be chosen to have bounded length as a word over X. Hence right fellow traveller properties of L are inherited by L'. However, left multiplication by elements of T is not controlled, and so left traveller properties of L are not clearly seen to be inherited by L'.

An asynchronous version of (c)(i) is proved in [**10**]; achieving a synchronous or boundedly asynchronous fellow traveller condition is harder. The argument is basically that of [**12**], but stated here in more general language theoretic terms.

First suppose that X is a generating set which does not contain the identity element. Then G has an $\mathcal{F}$-combing over X if and only if G has an $\mathcal{F}$-combing over $X \cup \{e\}$. One implication is obvious; a combing over X is also a combing over $X \cup \{e\}$. On the other hand, given a combing L in $\mathcal{F}$ over $X \cup \{e\}$, we construct (as in [**12**]) a language L' over X from L, by replacing (for some $m > 0$) every m-th occurrence of e in a word by a particular length m word w_e, and deleting all other occurrences of e. L' is easily seen to be the image of L under a GSM mapping; the associated generalised sequential machine has m states and is constructed in much the same way as the example shown in Figure 2. Although this GSM mapping is not ϵ-free, we can find it as the composite of an ϵ-free GSM mapping (which replaces all but the m-th occurrences of e by some dummy symbol), and a homomorphism with limited erasing (which maps the dummy symbol to ϵ, and leaves other symbols unchanged). Hence L' is in $\mathcal{F}$. The conditions of lemma 4.1 are satisfied with $\lambda = 1$ and $C = m$; hence L' inherits the fellow traveller conditions of L.

Now let X, Y be finite generating sets, and assume that G has an $\mathcal{F}$-combing of type τ over X. By the above, we may without loss of generality assume that $e \in Y$. For each $x \in X$, we can find some word $w_x \in Y^*$ which is equal to x in G. By appending copies of e where necessary, we can ensure that all the words w_x have the same length m. A language L'' over Y is now defined as the image of L under the homomorphism which maps x to w_x;the condition $P_{\epsilon-free\,GSM}$ implies $P_{\epsilon-free\,h}$, and so ensures that L'' is in $\mathcal{F}$. The homomorphism expands the lengths both of words in L and their subwords in L by a constant factor m; hence lemma 4.1 applies.

If only asynchronous fellow traveller properties are required, it is not essential for $\mathcal{F}$ to satisfy $P_{\epsilon-bdd\,h}$, since padding with copies of the symbol e is no longer necessary; hence in this case it is sufficient for $\mathcal{F}$ to satisfy $P_{\epsilon-free\,h}$.

To prove (c)(ii), suppose that G has an $\mathcal{F}$-combing L over a generating set X. Suppose that T is a set of symbols representing the elements of a left transversal for H in G, which includes the identity. As above, the Schreier generators for H are defined to be all the products $y_{t,x} = txt_{t,x}^{-1}$, for $x \in X$, and $t, t_{t,x} \in T$, where $t_{t,x}$ is the representative in T of the coset containing tx. Let Y be a set of symbols corresponding to those generators. An ϵ-free generalised sequential machine M can be defined, which first rewrites any word $x_1 \dots x_k$ in G as a product $y_1 \dots y_k t$, for $y_i \in Y$ and $t \in T$, by reading from left to right and applying a succession of rewrite rules of the form $tx \to y_{t,x}t_{t,x}$, and then accepts $x_1 \dots x_k$ and outputs $y_1 \dots y_k$ in the case where $t =_G e$, but otherwise rejects $x_1 \dots x_k$. (The associated generalised sequential machine has states corresponding to the elements of T; e is the start state and only accept state. For each $t \in T$, there is a transition on the input symbol x from t to $t_{t,x}$, with associated output $y_{t,x}$.) Let ϕ be the associated GSM mapping. Then where $v = x_1 \dots x_k$ and $\phi(v) = y_1 \dots y_k$, we see that $d_{G,Y\cup T}(\phi(v)(i), v(i)) \leq 1$. Hence we can apply lemma 4.1 to see that $\phi(L)$ satisfies the same fellow traveller conditions over $Y \cup T$ that L satisfies over X. Since any word in $\phi(L)$ is in fact a word over Y, and word differences between such words are therefore elements of H, a little thought shows that $\phi(L)$ satisfies the same fellow traveller conditions over the subset Y of $Y \cup T$. It is straightforward to see that if L is bijective then so is $\phi(L)$. □

4.2. Products and extensions of combable groups. In this section we look at free and direct products, (nearly) split extensions and central extensions of $\mathcal{F}$-combable groups, for $\mathcal{F}$ a class of formal languages. All the language theoretic conditions of these results are satisfied by Rg, Cf, Ind, Rc, RcE, and of course by the family $\mathcal{U}$ of all languages with finite alphabets.

The situation for free and direct products is quite straightforward. For (nearly) split extensions we describe a result of Bridson from [**9**], which deals with split extensions of combable groups over groups, such as abelian or word hyperbolic groups, which possess rather stable $\mathcal{F}$-combings. For central extensions we show that a result of Neumann and Reeves ([**22**]) for biautomatic groups can also be applied to combings associated with other language families.

We refer the reader to [**27**] for proofs of the synchronous combability of retracts and free products (possibly with amalgamation over finite subgroups) of synchronously combable groups, and of the synchronous bicombability of retracts and free and direct products of groups with synchronous bicombings satisfying additional length restrictions; language theoretic issues are not considered in [**27**].

PROPOSITION 4.3 (Free products). *Let $\mathcal{F}$ be a family of languages, which is closed under concatenation, positive closure, and intersection with regular sets, and suppose that G_1 and G_2 have $\mathcal{F}$-combings L_1 and L_2, over generator sets X_1 and X_2. Define L_1' and L_2' to be the subsets of L_1 and L_2 formed by deleting all representatives of the identity. Then*

- **(a):** *$L = (L_1L_2)^+$ is an $\mathcal{F}$-combing for $G_1 * G_2$ (and is a bicombing if both L_1 and L_2 are),*
- **(b):** *if $\mathcal{F}$ is also closed under Kleene closure and L_1 and L_2 are boundedly asynchronous, then $L' = (L_1'L_2')^*$ is a boundedly asynchronous $\mathcal{F}$-combing for G_1*G_2 (and is a boundedly asynchronous bicombing if L_1 and L_2 are). If, in addition, L_1 and L_2 are synchronous and bijective, then L' is synchronous and bijective.*

PROOF. Clearly the fact that L is in $\mathcal{F}$ is immediate. To see that the appropriate asynchronous fellow traveller poperties are inherited by L from L_1 and L_2, note that if two words w, w' in $(L_1L_2)^+$ are related by an equation of the form $w =_G w'$, $w =_G w'x$ or $w =_G xw'$, for $x \in X_1 \cup X_2$, then, for some k, we can write $w = v_1v_2 \dots v_k$ and $w' = v_1'v_2' \dots v_k'$, where for each i, and $j = 1$ or 2, $v_i, v_i' \in L_j \cup \{\epsilon\}$, $v_i =_G v_i', v_i'x$ or xv_i', and, for all but one i, $v_i =_G v_i'$.

Now suppose that L_1 and L_2 are boundedly asynchronous or synchronous. Let ϵ denote the empty word. Then for some K, every representative in L_1 or L_2 of the identity must boundedly asynchronously K-fellow travel with the shortest such representatives, and hence with ϵ; hence each of L_1 and L_2 can contain only finitely many representatives of the identity. In this case the languages L_1' and L_2' can be described as the intersections of L_1 and L_2 with regular sets, and so lie in $\mathcal{F}$; hence so does L'. The same argument as for L above shows that L' inherits boundedly asynchronous fellow traveller properties from L_1 and L_2. Where L_1 and L_2 are synchronous and bijective, bijectivity ensures that for all but one v_i in that argument, $v_i = v_i'$, and hence synchronous fellow traveller properties are also inherited by L' from L_1 and L_2.

Essentially this proof is given in [**3**] for automatic groups; there bijectivity can be assumed. □

PROPOSITION 4.4 (Direct products). *Suppose that $\mathcal{F}$ is closed under concatenation, and that G_1 and G_2 have $\mathcal{F}$-combings L_1 and L_2. Then L_1L_2 is an asynchronous $\mathcal{F}$-combing for $G_1 \times G_2$.*

If L_1 and L_2 are both bicombings, both boundedly asynchronous, or both bijective, then the same is true of L_1L_2. If L_1 and L_2 are both synchronous and L_1 is a near geodesic language, then L_1L_2 is also synchronous.

PROOF. The proof follows the argument of [**12**] for automatic groups, and uses the obvious decomposition of the Cayley graph for $G_1 \times G_2$ (with respect to a sensible generating set) into Cayley graphs for G_1 and G_2. The near geodesic condition ensures that, for $v, v' \in L_1, w, w' \in L_2$ with $vw =_G v'w'$, $vwx =_G v'w'$ or $yvw =_G v'w'$ (for generators x, y), the lengths of v and v' differ by a bounded amount; for automatic groups, a language can always be found in which this condition holds. □

A result of Bridson in [**9**] builds an asynchronous combing for a split extension (semidirect product) of two groups out of asynchronous combings for the two groups. Bridson worked with bijective combings, but that condition can easily be dropped. Further, the construction often works when the extension is only nearly split. So we state the result in a slightly more general form, and give a proof for the sake of completeness.

We need to establish some notation before we can state the result. Given a group G with normal subgroup N, we define a group G to be a *split extension* of N by G/N if $G = NH$ for some subgroup H of G intersecting N trivially, or more generally (following [**25**]) a *nearly split extension* if a finite index subgroup of G is of the form NH for a subgroup H of G which intersects N in a finite subgroup. In either case, since N is normal in G, H acts as a group of automorphisms of N (by conjugation).

Now suppose that groups N and H as above have asynchronous combings L_N, L_H over generating sets X, Y; for $n \in N$, denote by n^h the image of n under h. For $n \in N$, let $v_n = x_1x_2\ldots x_m$ be a word in L_N which represents n. We call a word w an *image* of v_n under h if w is of the form $v_{x_1^h}v_{x_2^h}\ldots v_{x_m^h}$ such that, for each i, $v_{x_i^h}$ is a word in L_N representing x_i^h.

PROPOSITION 4.5 ((Nearly) split extensions, [**9**]). *Let G be a nearly split extension of groups $N \lhd G$ and H. Suppose that N and H have asynchronous $\mathcal{F}$-combings L_N and L_H over generating sets X and Y, for some language family $\mathcal{F}$ which is closed under concatenation. If $H \cap N$ is non-trivial, suppose further that L_N is an asynchronous bicombing.*

Suppose also that for all $n \in N$, all $y \in Y$, and all v_n and v_{n^y} in L representing n and n^y respectively, v_{n^y} asynchronously fellow travels with each of the images of v_n under y (Condition $(*)$).

Then G has an asynchronous combing, L_G. If L_H and L_N are bijective and the extension is split, then L_G is also bijective.

Note that to verify $(*)$ it is enough to check that for each n, h, some v_{n^h} asynchronously fellow travels with some image of v_n under h; the rest of that condition follows from the fellow traveller properties of L_N.

PROOF. By proposition 4.2 (b), we lose no generality by assuming that G is equal to the product NH. In this case, we define $L_G = L_HL_N$. This is certainly a language for G, and our conditions ensure that it is in $\mathcal{F}$. Throughout the proof, we use the notation v_n, v'_n for elements of the language L_N which represent n, and w_h, w'_h for elements of L_H representing h.

Let $\Gamma_H, \Gamma_N, \Gamma_G$ be the Cayley graphs of the three groups. Of course Γ_H and Γ_N embed in Γ_G; in fact each embeds many times (for any vertex of Γ_G an embedding may be chosen which places the basepoint of Γ_N or Γ_H at that vertex). If u and u' are paths of Γ_G which lie in the same embedding of Γ_H or Γ_N, then any fellow traveller condition satisfied by them in that graph clearly also holds between them in Γ_G.

Suppose that $n \in N$, $h \in H$, and let $x \in X \cup \{e\}$. Consider words $u_1 = w_hv_n$ and $u_2 = w'_hv'_{nx}$ in L_G (representing hn and hnx). The portions of u_1 and u_2 along w_h and w'_h asynchronously fellow travel within an embedded Γ_H, and the portions from h to hn and hnx along v_n and v_{nx} asynchronously fellow travel within an embedded Γ_G (based at h). Hence u_1 and u_2 asynchronously fellow travel.

Now choose $y \in Y$. Since $hny = hyn^y$, hny is represented in L_G by words of the form $u_3 = w_{hy}v_{n^y}$. The portions of u_1 and u_3 along w_h and w_{hy} asynchronously fellow travel in Γ_G because they do in Γ_H. Now the portion of u_3 from hy to hny

along v_{n^y} asynchronously fellow travels with any path u' running from h to hn and labelled by an image of v_{n^y} under the action of y^{-1}; edges labelled by y run from any such path to the second part of u_3. The condition $(*)$ implies that u' also asynchronously fellow travels (within an embedded copy of Γ_N) with the path from h to hn labelled by v_n, that is, with the second part of u_1; hence we have verified that u_1 and u_3 asynchronously fellow travel.

In the case where $N \cap H$ is trivial, any element of G has a unique representation as a product of the form hn, for $h \in H$, $n \in N$, and so the above analysis covers all representatives in L_G of elements g, g' for which $g' =_G gx$ for $x \in X \cup \{e\}$ or $g' =_G gy$ for $y \in Y$, and the proof is complete. If, on the other hand, $N \cap H$ is a non-trivial finite group F, in order to finish the proof we need to verify that, for $h, h' \in H$ and $n, n' \in N$, with $hn =_G h'n'$, words of the form $v_h w_n$ and $v_{h'} w_{n'}$ asynchronously fellow travel. In this situation, the equation $hn =_G h'n'$ implies that $h' = hf$, $n' = f^{-1}n$ for some $f \in F$. Since F is finite, f has bounded length as a product of generators, and hence the fact that v_h and $v_{h'}$ asynchronously fellow travel follows from the fellow traveller condition on H; similarly, v_n and $v_{n'}$ asynchronously fellow travel, provided that L_N is an asynchronous bicombing. □

Proposition 4.5 is very much a result about asynchronous combings. A little thought shows that for the construction to give a synchronous combing for G, the action of H on N would need to be so restricted that G would have a direct product as a finite index subgroup (in addition to all the fellow traveller conditions on L_H, L_N and in $(*)$ being synchronous). Furthermore, to get conditions for any type of bicombing seems to be more complicated. For notice that, for $x \in X$, $xhg = hx^h g$. Left fellow traveller conditions are more easily satisfied by the language $L_N L_H$ (rather than $L_H L_N$), but this does not as easily have good right fellow traveller properties.

Bridson observes in [**9**] that the condition $(*)$ is satisfied by any geodesic combing of a word hyperbolic group; these combings are regular. In section 4 of [**24**], Neumann and Shapiro observe that pure braid groups (as repeated split extensions of free groups) and the fundamental groups of closed hyperbolic 3-manifolds which fibre over the circle (as split extensions of $\mathbf{Z}$ by a hyperbolic surface group) are hence asynchronously automatic. Further the asynchronous structures for pure braid groups easily extend to full braid groups.

Bridson also shows that any finitely generated abelian group has a synchronous combing which satisfies $(*)$. For the torsion free abelian group $\mathbf{Z}^n$ such a language is found by embedding the Cayley graph in the obvious way in $\mathbf{R}^n$, and selecting for each element g the lexicographically earliest of the words which remain closest to the strange line path in $\mathbf{R}^m$ from e to g. In the case $n = 2$, this language is proved (in [**10**]) to be an indexed language (and hence the fundamental groups of compact, geometrisable 3-manifolds are proved indexed combable); in general it is a real-time language ([**15**]). (Of course, this language is a bijective sublanguage of the language L_3 described in Section 2.) In both the above situations, the combings are in fact bicombings, and so the result can be applied to nearly split extensions. It is observed in [**15**] that many nilpotent groups and also many non-nilpotent soluble groups can be related to split extensions over $\mathbf{Z}^n$ and so have asynchronous combings, which are real-time languages.

We finish by looking at Neumann and Reeves' construction of combings for (virtually) central extensions of biautomatic groups ([**22, 23**]). The result can be generalised to (virtually) central extensions of $\mathcal{F}$-combable groups of various types without much difficulty. However it is not clear that the restrictive conditions will often be satisfied, for non-regular combings. Before we can state the result, we need some notation.

Let H be a group with generating set X, and A a finitely generated abelian group, on which H acts as a group of automorphisms. An extension G of A by H is then specified by a section $s : H \to G$ for H in G, with $s(e) = e$, together with a cocycle $\sigma : H \times H \to A$, which satisfies the rule

$$s(h)s(h') =_G s(hh')\sigma(h,h').$$

Group elements can be written as products of the form $s(h)a$, and the above rule, together with the action of H on A, defines multiplication between those products. The extension of A by H is *central* if H acts trivially on A, and *virtually central* if the natural image of H in $Aut(A)$ is finite.

Let L be a language for H. Then σ is defined in [**22**] to be an *L-regular cocycle* if both (a) the sets $\sigma(X,H)$ and $\sigma(H,X)$ are finite, and (b) for each $x \in X$ and $a \in A$, the set $\{v \in L : \sigma(v,x) = a\}$ is regular.

Neumann and Reeves prove in [**22**] that, where G is a virtually central extension of A by H corresponding to a cocycle σ, G is biautomatic if and only if H has a regular bicombing L for which σ is L-regular. Using this, they deduce in [**23**] that every central extension of a finitely generated abelian group by a word hyperbolic group is biautomatic. The following, slightly weaker but more general result is straightforward to prove using their methods.

PROPOSITION 4.6 (Central extensions, extending [**22**], Theorem A). *Let H be a group with an $\mathcal{F}$-combing L over a generating set X, and let A be finitely generated abelian. Let G be a central extension of A by H, defined by a section $s : H \to G$ and a cocycle $\sigma : H \times H \to A$. Suppose that $\mathcal{F}$ is closed under GSM mappings and concatenation with a regular set. Suppose also that σ is L-regular. Then G has an asynchronous $\mathcal{F}$-combing L'. If L is a bicombing, then so is L'.*

If L is synchronous or boundedly asynchronous, then so is L', provided that L is a near geodesic language.

PROOF. The proof is basically that of [**22**], translated into the appropriate language.

Define Y to be the set of all elements of G of the form $y_{x,h} = s(x)\sigma(h,x)^{-1}$ for $h \in H, x \in X$. The finiteness of $\sigma(X,H)$ ensures that Y is finite. Let Z be a generating set for A, and L_Z a regular synchronous bicombing for A over Z. For each $w = x_1x_2\dots x_n \in L$, $s(w)$ is equal in G to

$$s(x_1)s(x_2)\sigma(x_1,x_2)^{-1}s(x_3)\sigma(x_1x_2,x_3)^{-1}\dots s(x_n)\sigma(x_1x_2\dots x_{n-1},x_n)^{-1}$$

and hence $s(w)$ is represented by the word

$$w_0 = y_{x_1,e}y_{x_2,x_1}y_{x_3,x_1x_2}\cdots y_{x_n,x_1x_2\dots x_{n-1}}$$

over Y. Let L_Y be the set of all words of that form, and define L' to be the language L_YL_Z. Then L' is clearly a language for G.

The fellow traveller properties of L' can be seen to be inherited from L and the synchronous bicombing L_Z, using arguments similar to those of [**22**], as follows.

Suppose that w_0v and $w_0'v'$ are words in L' satisfying $w_0v =_G w_0'v'\xi$ for some $\xi \in \{e\} \cup Y \cup Z$. Then $w_0v =_G w_0'\xi v'$, and w_0 and w_0' represent $s(w)$ and $s(w')$ for some $w, w' \in L$, for which either $w =_H w'$ or $w =_H w'x$, for some $x \in X$.

Further, for any t, $w_0(t)$ and $w_0'(t)$ represent $s(w(t))$ and $s(w_0(t))$; so if w and w' fellow travel, then so do w_0 and w_0', in the same way. Hence asynchronous right fellow traveller properties of L, together with the fellow traveller properties of L_Z imply at least asynchronous right fellow traveller properties of L'. Similarly asynchronous left fellow traveller properties, if they exist, are carried over. Provided that the difference in lengths of w_0 and w_0' is bounded, boundedly asynchronous or synchronous fellow traveller properties of L are transferred to L'; such a bound is implied by the near geodesity of L (we note that this condition is ensured if H is automatic).

It remains to show that L' is in $\mathcal{F}$. We show this by effectively translating the construction of [**22**] into language theoretic terms. Since $\mathcal{F}$ is closed under concatenation with a regular set, it is enough to show that L_Y is in $\mathcal{F}$. We construct a GSM mapping which maps L to L_Y. For each $x \in X$ and each $a \in \sigma(G,X)$, let $W_{x,a}$ be the finite state automaton accepting $\{w \in L : \sigma(w,x) = a\}$, and let $S_{x,a}$ be its set of states; there are finitely many such sets. We form a (non-deterministic) generalised sequential machine M with input alphabet X and output alphabet Y as follows. The state set of M consists of the elements of the Cartesian product $\prod S_{x,a}$, together with a failure state F. For any input symbol x, any $a \in \sigma(G,X)$, and any state s for which the x,a component is an accept state, there is an arrow in M taking s to the state of which each component is the target under x of the corresponding component of s, and such that the symbol $y_{x,a}$ is appended to the output. The GSM mapping associated with M maps L to L_Y. □

References

[1] Alfred V. Aho, Indexed grammars - an extension of context-free grammars, J. Assoc. Comp. Math 15 (1968), 647–671.

[2] Alfred V. Aho, Nested stack automata, J. Assoc. Comp. Mach. 16 (1969) 383–406.

[3] G. Baumslag, S.M. Gersten, M. Shapiro, H. Short, Automatic groups and amalgams, Journal of Pure and Applied Algebra 76 (1991) 229–316.

[4] G. Baumslag, D. Gildenhuys, R. Strebel, Algorithmically insoluble problems about finitely presented solvable groups, Lie and associative algebras. I, Journal of Pure and Applied Algebra 39 (1986) 53–94.

[5] Gilbert Baumslag, Michael Shapiro and Hamish Short, Parallel poly pushdown groups, to appear in Journal of Pure and Applied Algebra.

[6] Noel Brady, The geometry of asynchronous automatic structures on groups, Ph. D. thesis, U. C. Berkeley, 1993.

[7] Noel Brady, Sol geometry groups are not asynchronously automatic, preprint.

[8] Martin R. Bridson, On the geometry of normal forms in discrete groups, Proc. London Math. Soc. (3) 67 (1993) 596–616.

[9] Martin R. Bridson, Combings of semidirect products and 3-manifold groups, Geometric and Functional Analysis 3 (1993) 263–278.

[10] M.R. Bridson and R.H. Gilman, Formal language theory and the geometry of 3-manifolds, Commentarii Math. Helv. 71 (1996) 525–555.

[11] M.R. Bridson and R.H. Gilman, On bounded languages and the geometry of nilpotent groups, Combinatorial and geometric group theory, LMS Lecture Notes Ser. 204, ed. Duncan, Gilbert and Howie, 1–15.

[12] David B.A. Epstein J.W. Cannon, D. F. Holt, S. Levy, M. S. Patterson and W. Thurston,Word processing in groups, Jones and Bartlett, 1992.

[13] S.M. Gersten, Bounded cocycles and combings of groups, Internat. J. Algebra Comput. 2 (1992) 307–326.

[14] Robert H. Gilman, Formal languages and infinite groups, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 25, AMS 1996, 27–51.

[15] Robert H. Gilman, Derek F. Holt and Sarah Rees, Combings for nilpotent and polycyclic groups, to appear in Internat. J. Algebra Comput.
[16] Seymour Ginsburg, The mathematical theory of context-free languages, McGraw-Hill, New York etc., 1966.
[17] S Ginsburg and S Greibach, Abstract families of languages, IEEE Conf. Record of Eighth Ann. Symp. on Switching and Automata Theory (1967), 128–139.
[18] Seymour Ginsburg and Edwin H. Spanier, Bounded algol-like languages, Trans. A.M.S. 113 (1964) 333–368.
[19] T. Hayashi, On derivation trees of indexed grammars, Publ. RIMS Kyota Univ. 9 (1973) 61–92.
[20] John E. Hopcroft and Jeffrey D. Ullman, Introduction to automata theory, languages and computation, Addison-Wesley, 1979.
[21] O.G. Kharlampovich, A finitely presented group with unsolvable word problem (Russian), Izv. Akad. Nauk SSSR Ser. Mat. 45 (1981) 852–873.
[22] Walter D. Neumann and Lawrence Reeves, Regular cocycles and biautomatic structures, Internat. J. Algebra Comput. 6 (1996) 313–324.
[23] Walter D. Neumann and Lawrence Reeves, Central extensions of word hyperbolic groups, Annals of Math. 145 (1997) 183–192.
[24] Walter D. Neumann and Michael Shapiro, Equivalent automatic structures and their boundaries, Internat. J. Algebra Comput. 2 (1992) 443–469.
[25] Derek J.S. Robinson, Splitting theorems for infinite groups, Symposia Mathematica 17, Convegni del Novembre e del Dicembre 1973, Academic Press 1976, 441–470.
[26] Arnold L. Rosenberg, Real-time definable languages, J. Assoc. Comput. Mach. 14 (1967) 645–662.
[27] Hamish Short, Groups and combings, preprint.

Department of Mathematics, University of Newcastle, Newcastle NE1 7RU, UK
E-mail address: `Sarah.Rees@ncl.ac.uk`