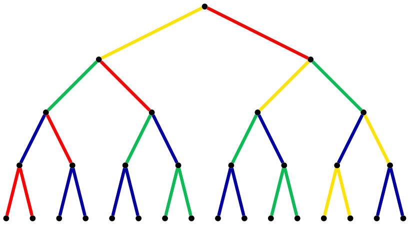
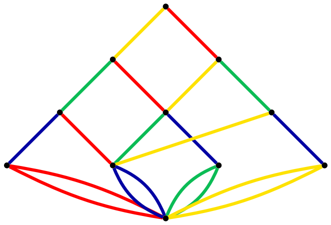
Computing in Free Bands

Reinis Cirpons

School of Mathematics and Statistics, University of St Andrews





A detour into universal algebra

Definition

An *algebra* is a set S together with a collection $\{f_i : i \in I\}$ of operations $f_i : S^{n_i} \to S$. The sequence $(n_i : i \in I)$ is called the *type* of the algebra.

Example

- Any semigroup is an algebra of type (2) with multiplication $\circ : S^2 \to S$ as its operation.
- Any monoid is an algebra of type $(2, 0)$ with the null-ary operation $1 : S^0 \to S$ returning the identity.
- Any inverse semigroup is an algebra of type $(2, 1)$ with the unary operation ${}^{-1} : S^1 \to S$ inverting an element.
- Any group is an algebra of type $(2, 0, 1)$ with operations $\{\circ, 1, {}^{-1}\}$.

Varieties

Definition

An *variety* of a given type is the collection of all algebras of the given type satisfying a set of universally quantified equations.

Example

- The variety of semigroups **Sg** has type (2) and is defined by the single equation $\mathbf{Sg} = [x \circ (y \circ z) = (x \circ y) \circ z]$,
- The variety of monoids **Mon** has type $(2, 0)$ and is defined by the equations

$$\mathbf{Mon} = [x \circ (y \circ z) = (x \circ y) \circ z, 1 \circ x = x, x \circ 1 = x],$$

- The variety of inverse semigroups **Inv** has type $(2, 1)$ and is defined by

$$\mathbf{Inv} = \Big[x(yz) = (xy)z, \left(x^{-1}\right)^{-1} = x, (xy)^{-1} = x^{-1}y^{-1},$$
$$xx^{-1}x = x, xx^{-1}yy^{-1} = yy^{-1}xx^{-1}\Big],$$

- The variety of groups **Gp** has type $(2, 0, 1)$.

Example

- The variety of bands **B** has type (2) and is defined by the equations

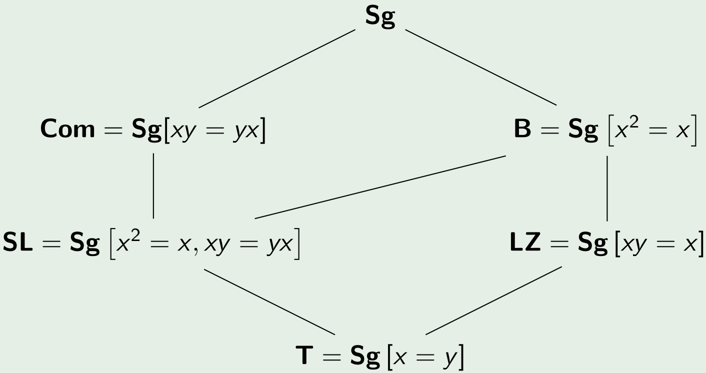
$$\mathbf{B} = \left[x(yz) = (xy)z, x^2 = x\right] = \mathbf{Sg}\left[x^2 = x\right],$$

- The variety of commutative semigroups $\mathbf{Com} = \mathbf{Sg}[xy = yx]$,
- The variety of semilattices $\mathbf{Sl} = \mathbf{Sg}[xy = yx, x^2 = x]$,
- Left zero semigroups $\mathbf{LZ} = \mathbf{Sg}[xy = x]$,
- The trivial semigroup variety $\mathbf{T} = \mathbf{Sg}[x = y]$,
- We will also be interested in the variety of band monoids

$$\mathbf{BM} = \mathbf{Mon}[x^2 = x].$$

A subclass of a variety that is itself a variety is called a *subvariety*. A variety $\mathcal{V}_1$ is a subvariety of $\mathcal{V}_2$ if and only if it satisfies all the equations defining $\mathcal{V}_2$. It can be shown that subvarieties form a lattice.

Example



Note that $\mathbf{LZ} = \mathbf{Sg}[xy = x] \neq \mathbf{Sg}[x = y]$ since the semigroup $S = \{0, 1\}$ with multiplication

$\circ$	0	1
0	0	0
1	1	1

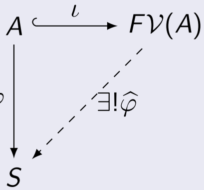
is a nontrivial semigroup in $\mathbf{LZ}$.
On the other hand, $\mathbf{Mon}[xy = x] = \mathbf{Mon}[x = y]$. Indeed if $S \in \mathbf{Mon}[xy = x]$, then $xy = x$ for all $x, y \in S$. Hence setting $x = 1$ we get that $y = 1$ for all $y \in S$. But then S is trivial and $x = y$ holds for all elements $x, y \in S$.

$\mathcal{V}$-free algebras

Definition

Let $\mathcal{V}$ be a variety and A be a set. The *$\mathcal{V}$-free algebra generated by A* is the algebra $F\mathcal{V}(A)$ such that

- there exists an injection $\iota : A \hookrightarrow F\mathcal{V}(A)$,
- for every algebra S in $\mathcal{V}$ and function $\phi : A \to S$ there exists a unique morphism $\widehat{\varphi} : F\mathcal{V}(A) \to S$ which agrees with φ on A. In other words the following diagram commutes:



Just as with free objects, there is an equivalent notion for presentations of algebras within a variety.

Project (with J. D. Mitchell)

Can we generalize algorithms for computing with finitely presented semigroups and groups, such as the Todd-Coxeter and Knuth-Bendix algorithm, to work within other varieties such as **Inv** or **B**?

In order to answer this question for **B** we need to first have a good computational representation of the Free Band.

The Free Band

We will write $FB(A)$ for the free band generated by A.

Theorem

Let β be the least congruence on A^+ generated by the set

$$\left\{(w, w^2) : w \in A^+\right\},$$

then $FB(A) \cong A^+/\beta$.

So we can represent elements of $FB(A)$ as words. We write $x \sim y$ if the words $x, y \in A^*$ represent the same element in $FB(A)$. And we write $x/\sim \in FB(A)$ for the element represented by $x \in A^+$.

The word problem

Given words $x, y \in A^+$ determine if $x \sim y$ in $FB(A)$.

The minimal word representative problem

Given a word $x \in A^+$ determine $\min(x) \in A^+$ the shortest word such that $x \sim \min(x)$.

Five important operations on words

The following functions are crucial to the study of $FB(A)$:

- The content $\mathrm{cont} : A^+ \to \mathcal{P}(A)$, where $\mathrm{cont}(x)$ is the set of letters occurring in x.
- The prefix $\mathrm{pref} : A^+ \to A^*$, where $\mathrm{pref}(x)$ is the longest prefix of x such that $|\mathrm{cont}(\mathrm{pref}(x))| = |\mathrm{cont}(x)| - 1$.
- The last letter to occur first $\mathrm{ltof} : A^+ \to A$, where $\mathrm{ltof}(x)$ is the letter immediately after $\mathrm{pref}(x)$ in x.
- The suffix $\mathrm{suff} : A^+ \to A^*$ and $\mathrm{ftol} : A^+ \to A$ are defined dually by taking the longest suffix and letter immediately before the suffix in x.

Example

Consider $x = ababbbcbcbc$, then

- $\mathrm{cont}(x) = \{a, b, c\}$,
- $\mathrm{pref}(x) = ababbb$,
- $\mathrm{ltof}(x) = c$,
- $\mathrm{suff}(x) = bbbcbcbc$,
- $\mathrm{ftol}(x) = a$

The Greens relations of $FB(A)$

Since every element of $FB(A)$ is idempotent, it follows that Greens $\mathcal{H}$-relation is trivial.

Theorem (Green and Rees 1952)

Let $x, y \in A^+$ and let $s = x/\sim, t = y/\sim \in FB(A)$, then

- $s\mathcal{D}t \iff \mathrm{cont}(x) = \mathrm{cont}(y)$,
- $s\mathcal{R}t \iff \mathrm{pref}(x) \sim \mathrm{pref}(y)$ *and* $\mathrm{ltof}(x) = \mathrm{ltof}(y)$,
- $s\mathcal{L}t \iff \mathrm{suff}(x) \sim \mathrm{suff}(y)$ *and* $\mathrm{ftol}(x) = \mathrm{ftol}(y)$.

It follows that $|D_s| = |R_s| \cdot |L_s|$. Let c_i be the size of the $\mathcal{D}$-class consisting of all elements $s \in FB(A)$ with $|\mathrm{cont}(s)| = i$. Then $c_1 = 1$ and

$$c_i = |R_s| \cdot |L_s| = c_{i-1} \cdot i \cdot c_{i-1} \cdot i = i^2 c_{i-1}^2.$$

If $|A| = k$, then

$$|FB(A)| = \sum_{i=1}^{k} \binom{k}{i} c_i.$$

In particular, every finitely generated free band is finite. So, by constructing a Cayley graph for $FB(A)$, we can solve the word problem and find minimal word reps in linear time.

The end?

Does this mean we are done?

The catch

- How do we construct the Cayley graph?
- Even if we do construct it, we need to store it.

$\lvert A \rvert$	$\lvert FB(A) \rvert$
1	1
2	6
3	159
4	$332380 = 3.32 \cdot 10^5$
5	$2751884514765 = 2.75 \cdot 10^{12}$
6	$272622932796281408879065986 = 2.76 \cdot 10^{26}$
7	$3.64 \cdot 10^{54}$
k	$\geq \frac{1}{4} 2^{2^k}$

For $\lvert A \rvert = 5$, even assuming every element can be stored using 1 bit, we would need at least 343 GB of memory to store the Cayley graph!

Comparison table

Let $|A| = k$ and $n = |x| + |y|$ be the total length of words $x, y \in A^+$. The below table will keep track of the time and space complexity of various algorithms for solving the word problem:

Algorithm	Time	Space	Minimal word reps?
Cayley graph	$\mathcal{O}(n)$	$\mathcal{O}(\lvert A\rvert \cdot \lvert FB(A)\rvert)$	Yes

Comparison table

Let $|A| = k$ and $n = |x| + |y|$ be the total length of words $x, y \in A^+$. The below table will keep track of the time and space complexity of various algorithms for solving the word problem:

Algorithm	Time	Space	Minimal word reps?
Cayley graph	$\mathcal{O}(n)$	$\mathcal{O}\left(k \cdot 2^{2^k}\right)$	Yes

Green and Rees method

Theorem (Green and Rees 1952)

Let $x, y \in A^+$ and let $s = x/\sim, t = y/\sim \in FB(A)$, then

- $s\mathcal{D}t \iff \mathrm{cont}(x) = \mathrm{cont}(y)$,
- $s\mathcal{R}t \iff \mathrm{pref}(x) \sim \mathrm{pref}(y)$ *and* $\mathrm{ltof}(x) = \mathrm{ltof}(y)$,
- $s\mathcal{L}t \iff \mathrm{suff}(x) \sim \mathrm{suff}(y)$ *and* $\mathrm{ftol}(x) = \mathrm{ftol}(y)$.

To check if $x \sim y$, compute $\mathrm{pref}, \mathrm{suff}, \mathrm{ltof}, \mathrm{ftol}$ for x and y. If $\mathrm{ltof}(x) \neq \mathrm{ltof}(y)$ or $\mathrm{ftol}(x) \neq \mathrm{ftol}(y)$ then $x \not\sim y$ and we are done. Otherwise recursively check if both $\mathrm{pref}(x) \sim \mathrm{pref}(y)$ and $\mathrm{suff}(x) \sim \mathrm{suff}(y)$.

$$t(n,k) = n + 2 \cdot t(n-1, k-1) \Rightarrow t(n,k) \in \mathcal{O}\left(n \cdot 2^k\right)$$

Algorithm	Time	Space	Minimal word reps?
Cayley graph	$\mathcal{O}(n)$	$\mathcal{O}\left(k \cdot 2^{2^k}\right)$	Yes
Green and Rees	$\mathcal{O}\left(n \cdot 2^k\right)$	$\mathcal{O}(n)$	No

Infinite complete rewriting system

Theorem (Siekmann and Szabó 1982)

Let A be a set and R be the infinite rewriting system on A^+ consisting of rules

$$x^2 \to x, \forall x \in A^+$$
$$xyz \to xz, \forall x, y, z \in A^+, s.t. \operatorname{cont}(y) \subseteq \operatorname{cont}(x) = \operatorname{cont}(z)$$

then R is a complete rewriting system for $FB(A)$.

Note R is strictly length reducing, so we will need to apply it at most n times to fully reduce a word of length n. So the time complexity of a solution to the word problem based on the rewriting system takes $\mathcal{O}(n \cdot t(n, k))$ time where $t(n, k)$ is the time necessary to recognize the application of a rule in R to the input word.

To apply R, need to be able to

- find occurrences of subwords of the form x^2. This is well studied and can be done in $\mathcal{O}(n)$ time, see e.g. Crochemore 1981,
- find occurrences of subwords xyz where $\mathrm{cont}(y) \subseteq \mathrm{cont}(x) = \mathrm{cont}(z)$. It is not clear how to do this efficiently, and literature has come up empty.

While the efficiency of the second step is not clear, certainly $t(n, k) \in \mathcal{O}(n)$. So we can reduce a word using R in $\mathcal{O}(n^2)$ time. Note furthermore that the reduced form of a word is the minimal word representative.

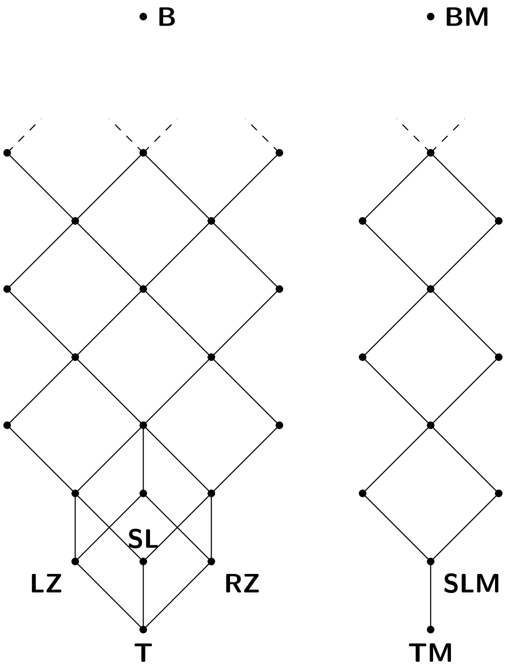
Algorithm	Time	Space	Minimal word reps?
Cayley graph	$\mathcal{O}(n)$	$\mathcal{O}\left(k \cdot 2^{2^k}\right)$	Yes
Green and Rees	$\mathcal{O}\left(n \cdot 2^k\right)$	$\mathcal{O}(n)$	No
Siekmann and Szabó	$\mathcal{O}\left(n^2\right)$	$\mathcal{O}(n)$	Yes

The lattice of subvarieties of bands

Theorem (Gerhard 1970)

Every subvariety of bands is determined by exactly one extra equation.

The lattice of subvarieties of bands was first determined by Gerhard 1970, Biryukov 1970 and Fennemore 1970. The lattice of subvarieties of band monoids was determined by Wismath 1986.


B
BM
SL
LZ
RZ
SLM
T
TM

Inertial trees

Definition

A *rooted binary tree* T is a directed graph such that

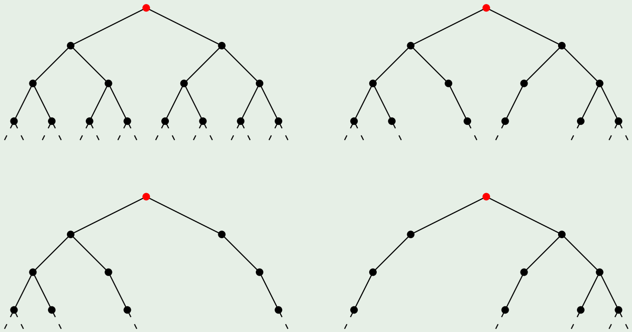
- there is a vertex called the root, that is not the target vertex of any edge,
- each vertex except the root is the target vertex of a unique edge,
- there is a unique path from the root to each vertex,
- every edge is labelled as either a right edge or a left edge,
- each vertex $v \in V(T)$ has at most one outgoing right and one outgoing left edge. The targets of these edges, if they exist, are labelled $r(v)$ and $l(v)$ respectively.

Definition

An *inertial tree* is a binary tree such that

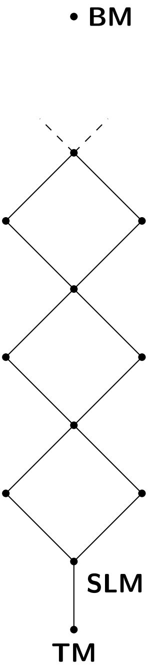
- if $r(v)$ exists, then so does $r(r(v))$ and furthermore the induced subtrees rooted at $r(v)$ and $r(r(v))$ are isomorphic.
- the above condition holds for l as well.

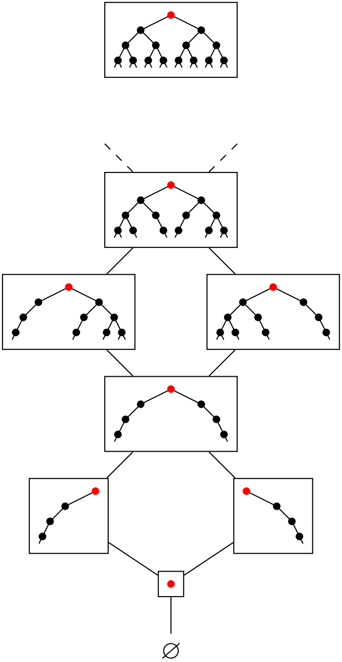
Example



Theorem (Neto and Sezinando 1996)

The lattice of inertial binary trees is isomorphic to the lattice of varieties of band monoids.


• BM
SLM
TM

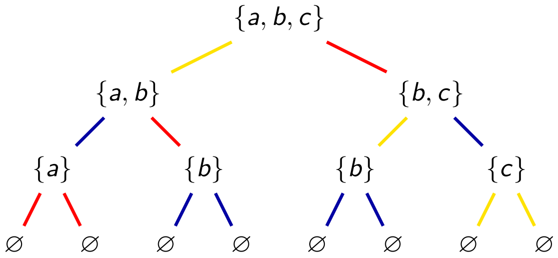

∅

Admissible maps

Definition

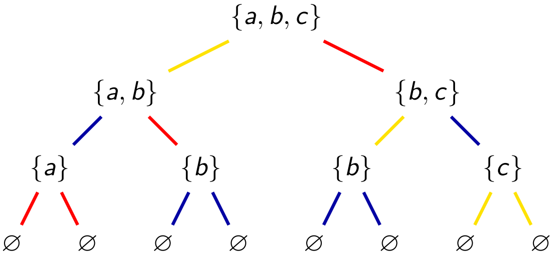
Let A be a set and T be an inertial tree. A map $\varphi : V(T) \to \mathcal{P}(A)$ is *admissible* if

- if $u, v \in V(T)$ and v is reachable from u, then $\varphi(v) \subseteq \varphi(u)$,
- if $\varphi(v) \neq \varnothing$ and $r(v)$ is defined, then $|\varphi(r(v))| = |\varphi(v)| - 1$,
- the above conditions holds for I as well.

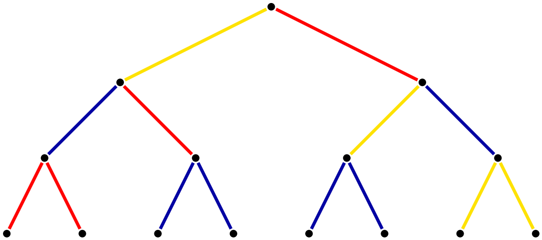


Theorem (Neto and Sezinando 1996)

Let A be a set, $\mathcal{V}$ be a subvariety of bands and $T_{\mathcal{V}}$ be the corresponding inertial tree. Then there is a bijective correspondence between admissible maps $\varphi : T_{\mathcal{V}} \to \mathcal{P}(A)$ and elements of $F\mathcal{V}(A)$.



$ababbbcbcbc$



$ababbbcbcbc$

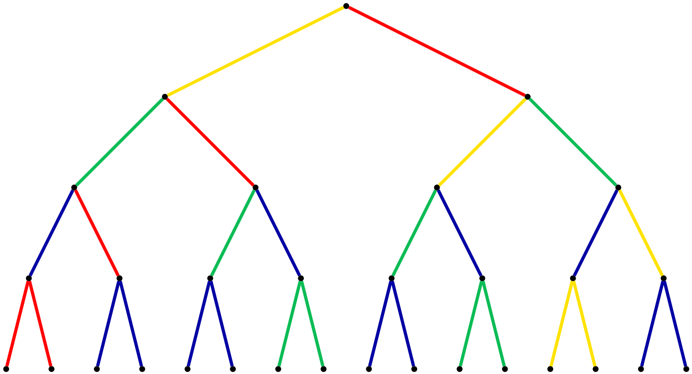
Algorithm	Time	Space	Minimal word reps?
Cayley graph	$\mathcal{O}(n)$	$\mathcal{O}\left(k \cdot 2^{2^k}\right)$	Yes
Green and Rees	$\mathcal{O}\left(n \cdot 2^k\right)$	$\mathcal{O}(n)$	No
Siekmann and Szabó	$\mathcal{O}\left(n^2\right)$	$\mathcal{O}(n)$	Yes
Neto and Sezinando	$\mathcal{O}\left(n \cdot 2^k\right)$	$\mathcal{O}\left(n + 2^k\right)$	No

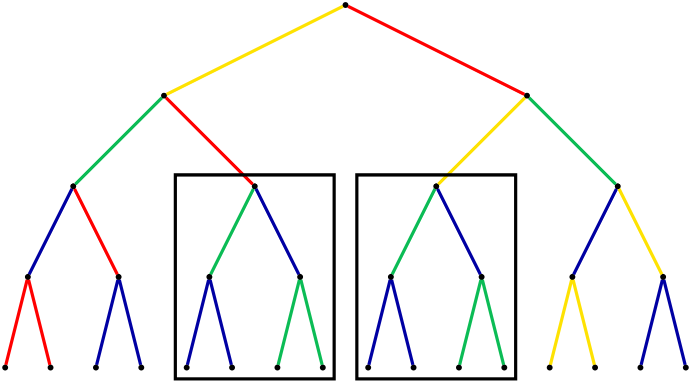
Compression

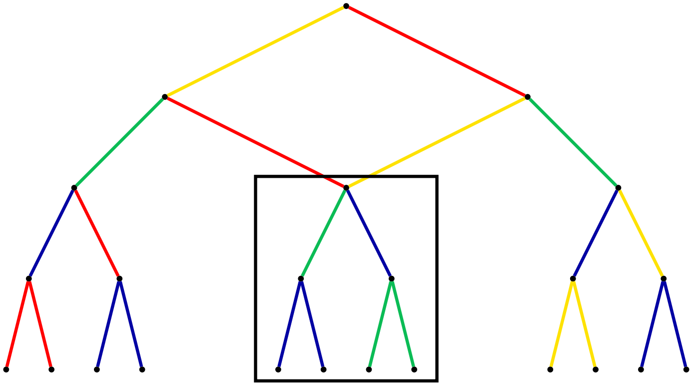
Theorem (Radoszewski and Rytter 2010)

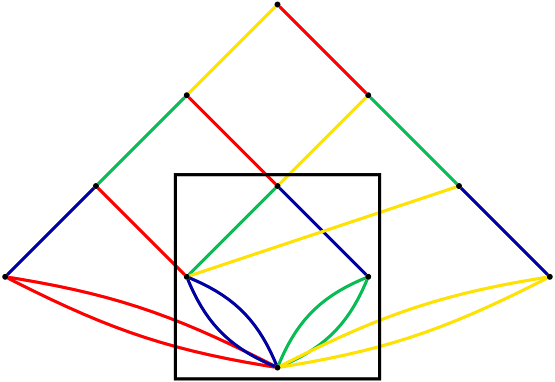
There is a $\mathcal{O}(n \cdot k)$ time and $\mathcal{O}(n)$ space algorithm for solving the word problem in the free band.

Algorithm	Time	Space	Minimal word reps?
Cayley graph	$\mathcal{O}(n)$	$\mathcal{O}\left(k \cdot 2^{2^k}\right)$	Yes
Green and Rees	$\mathcal{O}\left(n \cdot 2^k\right)$	$\mathcal{O}(n)$	No
Siekmann and Szabó	$\mathcal{O}\left(n^2\right)$	$\mathcal{O}(n)$	Yes
Neto and Sezinando	$\mathcal{O}\left(n \cdot 2^k\right)$	$\mathcal{O}\left(n + 2^k\right)$	No
Radoszewski and Rytter	$\mathcal{O}\left(n \cdot k\right)$	$\mathcal{O}(n)$	No









Our contributions

As before let $|A| = k$.

Theorem (C and Mitchell 2023)

The minimized binary tree representing $s \in FB(A)$ *has at most* $k \cdot |\min(s)|$ *vertices, where* $\min(s)$ *is the shortest word representative of* s*.*

Theorem (C and Mitchell 2023)

Given two binary trees τ_s, τ_t *representing elements* $s, t \in FB(A)$ *we can compute a binary tree representing* st *in* $\mathcal{O}(|\tau_s| + |\tau_t| + k^2)$ *time and space.*

Theorem (C and Mitchell 2023)

There is a $\mathcal{O}(n \cdot k)$ *time and space algorithm for finding the minimum word representative of a given word in the free band.*

Algorithm	Time	Space	Minimal word reps?
Cayley graph	$\mathcal{O}(n)$	$\mathcal{O}\left(k \cdot 2^{2^k}\right)$	Yes
Green and Rees	$\mathcal{O}\left(n \cdot 2^k\right)$	$\mathcal{O}(n)$	No
Siekmann and Szabó	$\mathcal{O}\left(n^2\right)$	$\mathcal{O}(n)$	Yes
Neto and Sezinando	$\mathcal{O}\left(n \cdot 2^k\right)$	$\mathcal{O}\left(n + 2^k\right)$	No
Radoszewski and Rytter	$\mathcal{O}\left(n \cdot k\right)$	$\mathcal{O}(n)$	No
C and Mitchell	$\mathcal{O}\left(n \cdot k\right)$	$\mathcal{O}(n \cdot k)$	Yes

Reference Python implementations of the method from our paper are available as part of the `freebandlib` library:

https://github.com/reiniscirpons/freebandlib

Future work

Problems

- Find a Neto-Sezinando like tree correspondence for all varieties of bands.
- Apply the tree and compression approach to efficiently solve the word problem and find minimal word representatives in all varieties of band monoids.
- Devise an efficient algorithm for solving the word problem in finitely presented bands.
- Can we generalize our tree based methods to solve in linear time the word problem in the free completely regular semigroups, which has a similar description of elements in terms of trees?
- What about the free Clifford semigroup (the free completely regular inverse semigroup)?

References I

Biryukov, A. P. (May 1970). "Varieties of idempotent semigroups". en. In: *Algebra and Logic* 9.3, pp. 153–164. ISSN: 1573-8302. DOI: 10.1007/BF02218673. URL: https://doi.org/10.1007/BF02218673 (visited on 11/07/2023).

C, R. and J.D. Mitchell (2023). "Polynomial time multiplication and normal forms in free bands". In: *Theoretical Computer Science* 953, p. 113783. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2023.113783. URL: https://www.sciencedirect.com/science/article/pii/S0304397523000968.

Crochemore, Max (1981). "An optimal algorithm for computing the repetitions in a word". In: *Information Processing Letters* 12.5, pp. 244–250. ISSN: 0020-0190. DOI: https://doi.org/10.1016/0020-0190(81)90024-7. URL: https://www.sciencedirect.com/science/article/pii/0020019081900247.

Fennemore, Charles (Dec. 1970). "All varieties of bands". en. In: *Semigroup Forum* 1.1, pp. 172–179. ISSN: 1432-2137. DOI: 10.1007/BF02573031. URL: https://doi.org/10.1007/BF02573031 (visited on 11/07/2023).

References II

Gerhard, J.A (1970). “The lattice of equational classes of idempotent semigroups”. In: *Journal of Algebra* 15.2, pp. 195–224. ISSN: 0021-8693. DOI: https://doi.org/10.1016/0021-8693(70)90073-6. URL: https://www.sciencedirect.com/science/article/pii/0021869370900736.

Green, J. A. and D. Rees (1952). “On semi-groups in which $x^r = x$”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 48.1, pp. 35–40. DOI: 10.1017/S0305004100027341.

Neto, O. and H. Sezinando (Dec. 1996). “Trees, band monoids and formal languages”. en. In: *Semigroup Forum* 52.1, pp. 141–155. ISSN: 0037-1912, 1432-2137. DOI: 10.1007/BF02574091. URL: http://link.springer.com/10.1007/BF02574091 (visited on 08/03/2023).

Radoszewski, J. and W. Rytter (Jan. 2010). “Efficient Testing of Equivalence of Words in a Free Idempotent Semigroup”. In: *SOFSEM 2010: Theory and Practice of Computer Science*, pp. 663–671. DOI: 10.1007/978-3-642-11266-9_55.

References III

Siekmann, J. and P. Szabó (Dec. 1982). "A Noetherian and confluent rewrite system for idempotent semigroups". In: *Semigroup Forum* 25.1, pp. 83–110. DOI: 10.1007/bf02573590. URL: https://doi.org/10.1007/bf02573590.

Wismath, Shelly L. (Dec. 1986). "The lattices of varieties and pseudovarieties of band monoids". en. In: *Semigroup Forum* 33.1, pp. 187–198. ISSN: 1432-2137. DOI: 10.1007/BF02573192. URL: https://doi.org/10.1007/BF02573192 (visited on 10/10/2023).