Finitely presented bands

JDM

June 2, 2021

This document contains some questions relating to finitely presented bands.

1 Varieties of bands

A *variety of semigroups* is a class of semigroups that are closed under homomorphic images, subsemigroups, and direct products. For example, the class $\mathscr{B}$ of bands is a variety.

The lattice of varieties was described by Biryukov [1], Fennemore [2], and Gerhard [3]. Fennemore [2] showed that every variety of bands is defined by a single identity.

If A is a finite alphabet, then we denote by $F(A)$ the free band over A. If $R \subseteq F(A) \times F(A)$, then we denote by $R^\sharp$ the least congruence on $F(A)$ containing the set R. We denote the quotient semigroup $F(A)/R^\sharp$ by $\mathscr{B}(A, R)$, and refer to $\mathscr{B}(A, R)$ as a *band presentation*. For example, $F(A)$ is isomorphic to the band defined by the presentation $\mathscr{B}(A, \varnothing)$. Note that since $F(A)$ is finite if A is finite, it follows that every finite band is finitely presented.

Question 1.1. *Is it possible to efficiently compute the least variety of bands that contains a given finite band?*

Suppose that A is an arbitrary finite set and that $\operatorname{End}(F(A))$ denotes the endomorphism monoid of the free band $F(A)$ over A.

We say that a congruence ρ on $F(A)$ is *invariant* if $((x)f, (y)f) \in \rho$ for every $(x, y) \in \rho$ and every $f \in \operatorname{End}(F(A))$. If $\mathscr{B}(A, R)$ is any finitely presented band over A, then we denote by $R^\dagger$ the greatest congruence on $F(A)$ that is invariant and such that $R^\dagger \subseteq R$.

This lemma will be helpful:

Lemma 1.2. *Let ρ be a congruence on $F(A)$. Then $F(A)/\rho$ is the free element $F_{\mathcal{V}}(A)$ of some variety $\mathcal{V}$ if and only if ρ is invariant.*

Proof. This is a question of using the correspondence between endomorphisms and the "free" extension of maps to morphisms.

($\Longrightarrow$) Let (u, v) be any relation in ρ. The words $u = u_1u_2 \dots u_k, v = v_1v_2 \dots v_l$ are free band elements and $u_i, v_j \in A$ for each i, j. Suppose WLOG that $A = \{1, 2, \dots, n\}$. Let $f \in \operatorname{End}(F(A))$ be any endomorphism; f is uniquely determined by where it maps the generators of $F(A)$. Write $(i)f = b_i$ for each $i = 1, 2, \dots, n$. We wish to show $((u)f, (v)f) \in \rho$, i.e.

$$(b_{u_1}b_{u_2} \dots b_{u_k}, b_{v_1}b_{v_2} \dots b_{v_l}) \in \rho.$$

With the aim of using the 'free' property of $F_{\mathcal{V}}(A)$, we define a function $\phi : A \to F_{\mathcal{V}}(A)$ which agrees with f on A, i.e.

$$(i)\phi = b_i \quad \forall i \in A.$$

Since $F_{\mathcal{V}}(A)$ is free in $\mathcal{V}$, and certainly $F_{\mathcal{V}}(A) \in \mathcal{V}$, we know ϕ can be extended to a unique homomorphism $\hat{\phi} : F_{\mathcal{V}}(A) \to F_{\mathcal{V}}(A)$. By assumption, u and v are ρ-related so $u = v$ in $F_{\mathcal{V}}(A)$. Therefore, in $F_{\mathcal{V}}(A)$,

$$b_{u_1}b_{u_2} \dots b_{u_k} = (u_1)\phi(u_2)\phi \dots (u_k)\hat{\phi} = (u)\hat{\phi} = (v)\hat{\phi} = (v_1)\phi(v_2)\phi \dots (v_l)\hat{\phi} = b_{v_1}b_{v_2} \dots b_{v_l},$$

and this proves $((u)f, (v)f) \in \rho$.

($\Longleftarrow$) The proof for this direction follows the same pattern as Proposition 4.6.2 of [4]. □

With this, we can prove that for a finitely presented band $\mathscr{B}(A, R)$, the free object in the smallest variety containing $\mathscr{B}(A, R)$ is in fact $F(A)/R^\dagger$.

Let ρ be the congruence on $F(A)$ giving the free object $F_\mathcal{V}(A)$ of the smallest variety $\mathcal{V}$. Then since $R^\dagger$ is by definition invariant, $F(A)/R^\dagger$ is a free object. Also, since $R^\dagger \subseteq R^\sharp$, we know that $\mathscr{B}(A, R)$ is a homomorphic image of $F(A)/R^\dagger$. So certainly, $F(A)/R^\dagger$ and $\mathscr{B}(A, R)$ lie in the same variety. If $F_\mathcal{V}(A)$ was in a smaller variety than that of $F(A)/R^\dagger$, then we would have $R^\sharp \supseteq R \supseteq \rho \supseteq R^\dagger$, but also ρ invariant, impossible.
Question 1.3. *Given a band B, is it possible to check efficiently whether or not B is a free object is some variety of bands?*

2 Algorithms for finitely presented bands

Given a finitely presented band $\mathscr{B}(A, R)$ how can we compute things about this object? Such as its size, Green's structure, and so on. The main algorithms for computing such things for finitely presented semigroups are the Knuth-Bendix algorithm and the Todd-Coxeter algorithm. Are there versions of these main algorithms for finitely presented semigroups specifically for bands?
Question 2.1. *Suppose that F is a free object in a variety of bands. Is it possible to generalise the algorithm of the other paper to check equality of words in F?*

2.1 Broadening the general Todd-Coxeter Algorithm

We may wonder whether the results proved in [5] hold also if we replace the original **TC1** process with a "smart" version which defines new nodes labelled exclusively by canonical elements in some (relatively) free semigroup. We also wish to give this improved subprocess the ability to define an edge $\mathbf{e}(w, a)$ to point towards an already existing node, if it so happens that the canonical form of wa already exists somewhere in the node set.

Suppose A is a finite alphabet and $\mathcal{V}$ is some variety of semigroups. Call $F_\mathcal{V}(A)$ the free object on A in the variety $\mathcal{V}$. We suppose we are interested in performing a variant of the Todd-Coxeter algorithm where the presentation implicitly includes the fact that the output should be an element of $\mathcal{V}$. Wishing to avoid the presence of two nodes in Γ labelled by $F_\mathcal{V}(A)$-equivalent words, we define a canonical form of $w \in A^*$ to be $\bar{w}$, the shortlex least element of the equivalence class of w in $F_\mathcal{V}(A)$. We now define a smart **TC1** which reduces words to canonical form before adding them to the node set.

TC1$_\mathcal{V}$. If w is a node in $\Gamma(i)$ and $\mathbf{e}_i(w, a)$ is undefined for some $a \in A$:

(a) If $\mathbf{e}_i(\varepsilon, \overline{wa})$ is defined, define $\Gamma(i+1)$ to be the digraph obtained from $\Gamma(i)$ by extending $\mathbf{e}_i$ so that $\mathbf{e}_{i+1}(w, a) = \mathbf{e}_i(\varepsilon, \overline{wa})$.

(b) Otherwise, noting that all prefixes of $\overline{wa}$ must be canonical, let w' be the longest prefix of $\overline{wa}$ such that $\mathbf{e}_i(\varepsilon, w')$ is defined. Express the canonical word as $\overline{wa} = w'a_1a_2\ldots a_n$ where each $a_i \in A$. Add the following nodes to $\mathbf{n}(\Gamma(i+1))$:

$$\begin{gathered} w'a_1, \\ w'a_1a_2, \\ \vdots \\ w'a_1a_2\ldots a_n (= \overline{wa}). \end{gathered}$$

Create new edges:

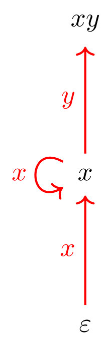
$$\begin{aligned} \mathbf{e}_{i+1}(\mathbf{e}_i(\varepsilon, w'), a_1) &:= w'a_1 \\ \mathbf{e}_{i+1}(w'a_1, a_2) &:= w'a_1a_2 \\ \vdots \quad & \quad \vdots \\ \mathbf{e}_{i+1}(w'a_1a_2\ldots a_{n-1}, a_n) &:= w'a_1a_2\ldots a_n. \end{aligned}$$

Finally, define $\mathbf{e}_{i+1}(w,a) := \overline{wa}$.

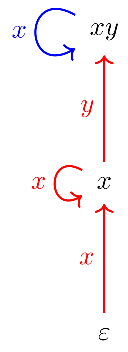
It might take some thought to convince oneself that the nodes created in case (b) do not already exist. Suppose some node $\nu = w'a_1a_2\dots a_i$ already existed. Then since $\mathbf{TC1}_{\mathcal{V}}$ takes a ρ-digraph as input, following the path in $\Gamma(i)$ from ε labelled $w'a_1a_2\dots a_i$ should lead to ν. However, we assumed when constructing the maximal prefix w' that $\mathbf{e}_i(\varepsilon, w')$ was defined but $\mathbf{e}_i(\varepsilon, w'a_1)$ was not. So this path doesn't lead anywhere and ν cannot exist before it is created in $\mathbf{TC1}_{\mathcal{V}}$.

Note that the node $\mathbf{e}_i(\varepsilon, w')$ may not actually be labelled w', even though w' is canonical: coincidences could have been processed that merged a previous node w' with some shortlex-smaller node. In the following example, this is the case.

Example 2.2 (Case (a))**.** *Suppose we are in the variety of commutative semigroups. Orders of generators can be swapped. Our alphabet is $A = \{x, y, z\}$. Also, one of the explicit relations is $xx = x$. At step i, we have the following digraph $\Gamma(i)$:*



We now apply $\mathbf{TC1}_{\mathcal{V}}$ to node $w = xy$ and edge $a = x$. So $wa = xyx$, which in canonical form is $\overline{wa} = xxy$. The node xxy is not in the node set, but if we follow the path labelled xxy from ε, we end up at node xy. So case (a) is triggered, since the path is defined, and we set $\mathbf{e}_{i+1}(w,a) := xy$:



This next example illustrates the node-creating capabilities of $\mathbf{TC1}_{\mathcal{V}}$.

Example 2.3 (Case (b))**.** *Suppose once again that we are in the variety of commutative semigroups and $A = x, y, z$. One of the explicit relations is $xy = x$. At step i, suppose we have the following digraph $\Gamma(i)$:*

$$\begin{tikzcd}
yzz & & \\
yz \arrow[u, "z"] & & \\
y \arrow[u, "z"] & & x \arrow[loop right, "y"] \\
& \varepsilon \arrow[ul, "y"] \arrow[ur, "x"'] &
\end{tikzcd}$$

We now apply ***TC1***$_{\mathcal{V}}$ *to node* $w = yzz$ *and edge* $a = x$*. So* $wa = yzzx$*, which in canonical form is* $\overline{wa} = xyzz$*. Following the path labelled* $xyzz$ *starting at* ε*, edges are defined for* xy *at which point we end up at the node labelled* x*. Then,* x *has no out-edge labelled* z*, so the next two steps are undefined. This triggers case (b) where* $w' = xy$*,* $a_1 = z$ *and* $a_2 = z$*. So we create nodes* $w'a_1 = xyz$ *and* $w'a_1a_2 = xyzz$ *and complete the path we started at* ε*:*

$$\begin{tikzcd}
yzz \arrow[rr, "x"] & & xyzz \\
yz \arrow[u, "z"] & & xyz \arrow[u, "z"'] \\
y \arrow[u, "z"] & & x \arrow[u, "z"'] \arrow[loop right, "y"] \\
& \varepsilon \arrow[ul, "y"] \arrow[ur, "x"'] &
\end{tikzcd}$$

The blue edges and nodes are added through the list provided in case (b) of the definition, while the green edge is the final addition which actually defines the edge we wanted: $\boldsymbol{e}_{i+1}(w, a)$*.*

Notice that our choice of labels means that the new edge leading out of the node x *leads to a node labelled* xyz*. While it would perhaps make more sense to label this new node* xz*, in the general case we have no guarantee that labelling by appending letters to* $\boldsymbol{e}(\varepsilon, w')$ *will produce canonical words. On the other hand, we know that all prefixes of* $\overline{wa}$ *– including, in this case,* xyz *– are necessarily canonical.*

Another thing to keep in mind with ***TC1***$_{\mathcal{V}}$ *node creation is that, as in the example above, nodes are not always labelled by their shortlex minimal representative. For instance,* xyz *is accessible via the path* xz*, a shorter representative for the same element – but no separate node* xz *can be created, since following the path labelled* xz *from* ε *leads to the well-defined node* xyz*.*

We claim now that replacing **TC1** by **TC1**$_{\mathcal{V}}$ in [5] has no effect on the results about the Todd-Coxeter Algorithm in the rest of the paper. The definitions of σ-digraphs and proper congruence enumeration processes are still relevant with this new setup. We will only consider two-sided c.e.p.s.

The important thing to note about our new interpretation of ρ is that we will assume ρ contains all the relations that generate $F_{\mathcal{V}}(A)$. So for example, if $u, v \in A^*$ represent the same element of the relatively free semigroup $F_{\mathcal{V}}(A)$, then $\{u, v\} \in \rho$. On top of this, ρ will contain the additional explicit relations that are passed as input to the congruence enumeration process.

Let's go through every result which needs adapting to fit with **TC1**$_{\mathcal{V}}$.

Lemma 4.3. *If* $\mathbf{TC1}_{\mathcal{V}}$ *is applied to a* ρ*-digraph* $\Gamma(i)$*, then the output digraph* $\Gamma(i+1)$ *is also a* ρ*-digraph.*

Proof. Suppose $\Gamma(i)$ is a ρ-digraph to which we apply $\mathbf{TC1}_{\mathcal{V}}$ to get $\Gamma(i+1)$. The product $\mathbf{e}_{i+1}(w,a)$ is now defined. Check the conditions of the definition.

(a). If any nodes were added when applying $\mathbf{TC1}_{\mathcal{V}}$, then $\overline{wa}$ was added along with any prefixes that were missing. Note that all prefixes of a word in canonical form (i.e. shortlex minimal) are also in canonical form for their congruence classes. The process $\mathbf{TC1}_{\mathcal{V}}$ ensures that all new edges and nodes on the path leading to $\overline{wa}$ are sensibly created in such a way that $\mathbf{e}_{i+1}(\varepsilon,\overline{wa}) = \overline{wa}$, and this holds for prefixes too.

(b). We wish to show the equivalence relation π_{i+1} is contained in ρ. Since $\pi_i \subseteq \rho$, let $\{u,v\} \in \pi_{i+1} \setminus \pi_i$.

If the first scenario of $\mathbf{TC1}_{\mathcal{V}}$ was applied, then no new nodes were created and one edge labelled a from w to $\mathbf{e}_i(\varepsilon,\overline{wa})$ was created. Call this node (that the new edge points to) ν. Since both the paths $\mathbf{e}_i(\varepsilon,\nu)$ and $\mathbf{e}_i(\varepsilon,\overline{wa})$ are assumed to already exist, $\Gamma(i)$ being a ρ-digraph implies that $\overline{wa}\ \rho\ \nu$.

At least one of the paths u,v must have passed through this new edge from w to ν. Assume first that both paths did: we have $u = u_1au_2$ and $v = v_1av_2$, where

$$\mathbf{e}_i(\varepsilon,u_1) = \mathbf{e}_i(\varepsilon,v_1) = w \tag{$*$}$$

and

$$\mathbf{e}_i(\nu,u_2) = \mathbf{e}_i(\nu,v_2) \implies \mathbf{e}_i(\varepsilon,\nu u_2) = \mathbf{e}_i(\varepsilon,\nu v_2). \tag{$**$}$$

We can now deduce a number of relations that are helpful:

- $u_1\ \rho\ w$ by $(*)$, so $u_1a\ \rho\ wa\ \rho\ \overline{wa}$, the last step holding because the two words represent the same element of $F_{\mathcal{V}}(A)$. Since $\overline{wa}\ \rho\ \nu$, We keep $\boxed{u_1a\ \rho\ \nu}$ in mind.
- Similarly to the previous point, $\boxed{v_1a\ \rho\ \nu}$.
- By assumption on π_i and by $(**)$, we have $\boxed{\nu u_2\ \rho\ \nu v_2}$.

Combining the above, we get $u = u_1au_2\ \rho\ \nu u_2$ and $v = v_1av_2\ \rho\ \nu v_2$, and so $u\ \rho\ v$ as required.

This reasoning does not cover the possibility that one or both of the paths pass over the new edge more than once. In the most general case, this could look like $u = u_1au_2a\dots au_n$ and $v = v_1av_2a\dots av_m$ where $\mathbf{e}_i(\varepsilon,u_1) = \mathbf{e}_i(\varepsilon,v_1) = w$, and $\mathbf{e}_i(\nu,u_k) = w$ for all $2 \le k < n$ (same for v). It can be built up by induction that $u_1a\dots au_{n-1}a\ \rho\ \nu$ (same for v) and the final step of the proof follows as above. TODO.

Suppose now that only one path, say u, passed over the new edge (only once - the inductive argument extending this is TODO and proceeds as in the previous remark). Then we can write $u = u_1au_2$ where $\mathbf{e}_i(\varepsilon,u_1) = w$. As in the previous case, we can prove that $u_1a\ \rho\ \nu$. Since we assumed the path v does not pass over any new nodes or edges, we may write

$$\mathbf{e}_i(\varepsilon,v) = \mathbf{e}_i(\nu,u_2).$$

This implies $v\ \rho\ \nu u_2$, and so we get $u = u_1au_2\ \rho\ \nu u_2\ \rho\ v$, as required.

If the second scenario of $\mathbf{TC1}_{\mathcal{V}}$ was applied, we may treat it as a sequence of new node creations (blue in 2.3) followed by a first-scenario application of $\mathbf{TC1}_{\mathcal{V}}$ (green in 2.3) which we considered above. So it suffices to show that adding one node $w'a_1$ and one edge $\mathbf{e}_{i+1}(\mathbf{e}_i(\varepsilon,w'),a_1) = w'a_1$ preserves ρ-digraphs. Any additional nodes and edges (e.g. $w'a_1a_2, w'a_1a_2a_3,\dots$) have analogous proofs.

The approach for one node and one edge addition is in the same vein as the original proof of Lemma 4.3. If $\{u,v\} \in \pi_{i+1} \setminus \pi_i$, then $u = u_1a_1$ and $v = v_1a_1$ for some $u_1,v_1 \in A^*$, and $\mathbf{e}_i(\varepsilon,u_1) = \mathbf{e}_i(\varepsilon,v_1) = \mathbf{e}_i(\varepsilon,w')$. Hence, by assumption, $(u_1,v_1) \in \rho$ and since ρ is a right congruence, $(u,v) = (u_1a_1,v_1a_1) \in \rho$, also.

So we have proved that $\pi_{i+1} \subseteq \rho$.

(c). Applying $\mathbf{TC1}_{\mathcal{V}}$ does not change K, so this condition holds.

□

When thinking about the above proof, it may be helpful to consider a "messiest-case" scenario with lots of paths and loops, etc. The following graph could be part of a ρ-digraph for the variety of bands. We are applying $\mathbf{TC1}_{\mathcal{V}}$ to create an edge labelled c from the node $abcab$ to the already existing node abc. (this falls into the first scenario). One of the explicit relations added is $abac = abcb$.

$$\begin{tikzcd}
 & abca \arrow[dl, "b"'] & & abcb & \\
abcab \arrow[rr, red, "c"] & & abc \arrow[ul, "a"'] \arrow[ur, "b"] & & aba \arrow[ul, "c"'] \\
 & & & ab \arrow[ul, "c"] \arrow[ur, "a"'] & \\
 & & & a \arrow[u, "b"] &
\end{tikzcd}$$

No additional tweaks to proofs are necessary until section 4.2, where we consider:

Lemma 4.13. *If w is a node in $\Gamma(i)$, and there exists $j > i$ such that w is not a node in $\Gamma(j)$, then w is not a node in $\Gamma(k)$ for any $k \geq j$.*

Proof. If w is a node in $\Gamma(i)$ then the path $\mathbf{e}_i(\varepsilon, w)$ certainly exists. Since edges that are defined cannot become undefined, and definition of edges is robust under coincidence processing, this means the path $\mathbf{e}_k(\varepsilon, w)$ is defined for all $k \geq i$. If $w \notin \mathbf{n}(\Gamma(j))$, then the only way w can reappear as a node at a later step is via node creation in $\mathbf{TC1}_{\mathcal{V}}$. However, $\mathbf{TC1}_{\mathcal{V}}$ only creates a node labelled w at step k if $\mathbf{e}_k(\varepsilon, w)$ is undefined, which is not the case for any $k \geq i$. So w is not a node in $\Gamma(k)$ for any $k \geq j$. □

For Lemma 4.14, very little adaptation is needed given we have shown that Lemma 4.13 still holds.

Lemma 4.14. *Every congruence enumeration process stabilises.*

Proof. The proof claims that the definitions of $\mathbf{TC1}_{\mathcal{V}}$, $\mathbf{TC2}$, $\mathbf{TC3}$ are such that $\varepsilon \leq \mathbf{e}_k(w, a) \leq \mathbf{e}_j(w, a)$ for all $k \geq j$, provided both terms are defined. If $\mathbf{e}_j(w, a)$ is defined, then $\mathbf{TC1}_{\mathcal{V}}$ cannot change the value of the edge function on (w, a), so changing the value is only possible via $\mathbf{TC2}$ or $\mathbf{TC3}$ which remain unchanged from their original descriptions. □

It is also still true that $\mathbf{TC1}_{\mathcal{V}}$ cannot be applied to Γ if Γ is complete. Hence, Corollary 4.15 still holds: completeness at some step implies that the algorithm will terminate.

No proofs in Section 4.3 need changing as they all rely mostly on previous results and the definition of $\mathbf{TC2}$, the same goes for Section 4.4 and Section 4.5 builds on previous results. So, we have proved that replacing $\mathbf{TC1}$ with $\mathbf{TC1}_{\mathcal{V}}$ in the general description of a congruence enumeration process has no effect on the validity of that process, provided ρ is defined to contain all explicit relations R along with the generating relations of the free element $F_{\mathcal{V}}(A)$.

2.2 The Todd-Coxeter Algorithm for Bands

Using our newly defined $\mathbf{TC1}_{\mathcal{V}}$ step above, we can see how we may apply this to a band presentation. We will assume throughout that we are in the variety of bands, i.e. $\mathcal{V} = \mathcal{B}$.

Suppose we wish to find the elements of a semigroup given as a band presentation $\mathscr{B}(A, R)$, where the fact that each element is idempotent is treated as implicit. We assume that there is an efficient way of expressing any word $w \in A^*$ in a canonical free band form, which we denote by $\bar{w}$ and which we assume to be lex-least in the set of all words in the free band equivalent to w.

The presentation $\mathscr{B}(A, R)$ is equal to $F(A)/R^\sharp \cong A^*/(\phi(A) \cup R)^\sharp$, where $\phi(A)$ denotes the set of "band relations":

$$\phi(A) = \{\{ww, w\} : w \in F(A) \text{ canonical}\}.$$

We will denote the full set of relations $R \cup \phi(A)$ by $\bar{R}$.

If the generators of some $\phi(A)$ are known, then computing $\mathscr{B}(A,R)$ via Todd-Coxeter can be done by simply appending these generators to the given explicit relations R, and running the Todd-Coxeter algorithm as usual on the given input. However, the number of generators for $\phi(A)$ increases quickly with $|A|$. The minimum size for $|A|=2$ is 6, for $|A|=3$ it is around 40, and anything beyond $|A|=4$ is computationally expensive, if not impossible, to compute.

Instead of relying on expensively making the contents of $\phi(A)$ explicit, we propose to adapt common implementations of the usual Todd-Coxeter algorithm so that the band relations are treated implicitly and internally, while the explicit relations R are treated "as usual".

The *Band Todd-Coxeter* (BTC) algorithm has all its steps built off $\mathbf{TC1}_{\boldsymbol{\mathcal{V}}}$, as well as sub-processes from [5]. We will use:

$\mathbf{TC1}_{\boldsymbol{\mathcal{V}}}$. Define the product of a node w and a node a, either by creating one or several new nodes if the path $\overline{wa}$ doesn't exist, or by adding an edge pointing to a pre-existing node if it does.

TC2. Push a node w through a relation $\{u,v\} \in \bar{R}$.

TC3. Process coincidences.

We will treat the BTC algorithm as an $\bar{R}^\sharp$-process with steps $\mathbf{TC1}_{\boldsymbol{\mathcal{V}}}$, **TC2**, **TC3** in a specific order. A set S, initially equal to $\{\varepsilon\}$, must be maintained in parallel when this process is carried out. We add words to S whenever we add words to the list of nodes via $\mathbf{TC1}_{\boldsymbol{\mathcal{V}}}$, but we do not remove words from S when processing coincidences, so that S contains every word that the algorithm has seen.

Just like there are different strategies for implementing the Todd-Coxeter Algorithm, there are several similar strategies for BTC.

2.2.1 The HLT Strategy

The HLT strategy ensures every coset is pushed through every implicit relation only once by ensuring, when **TC2** is applied to $(w, \{u,v\})$, that $\mathbf{e}_i(w,u)$ and $\mathbf{e}_i(w,v)$ are both defined. For BTC, we adapt this as follows.

1. Initialise the first $\bar{R}^\sharp$-graph, $\Gamma(0)$, as the trivial digraph.
2. Set w, the node the algorithm is currently *considering*, to be the lex-least node in $\mathbf{n}(\Gamma(i))$ that has not already been considered.
3. For each $a \in A$, apply $\mathbf{TC1}_{\boldsymbol{\mathcal{V}}}$ to w and a.
4. For each relation $\{u,v\} \in R$ (note this does not include the implicit relations, only the explicit ones), apply $\mathbf{TC1}_{\boldsymbol{\mathcal{V}}}$ as needed until $\mathbf{e}_i(w,u)$ and $\mathbf{e}_i(w,v)$ are defined. Then push w through $\{u,v\}$ using **TC2**.
5. For each word $u \in S$, push w through $\{uu,u\}$ (note $\{uu,u\} \in \phi(A)$ necessarily since we only ever add canonical words to the nodes), ensuring as above via applications of $\mathbf{TC1}_{\boldsymbol{\mathcal{V}}}$ that both paths are defined.
6. For each word w' in S which was considered in a previous iteration, push w' through $\{ww,w\}$ using **TC2**, once again using $\mathbf{TC1}_{\boldsymbol{\mathcal{V}}}$ to define $\mathbf{e}_i(w',w)$ and $\mathbf{e}_i(w',ww)$ if needed.
7. Process coincidences using **TC3**.
8. Return to step 2 – except if all nodes have been considered, in which case stop.

Note that if all nodes have been considered and coincidences have just been processed, $\mathbf{TC1}_{\boldsymbol{\mathcal{V}}}$ and **TC3** can no longer be applied. We will later prove that applying **TC2** will not change the graph further.

Let's begin by affirming a few basic properties of this algorithm.

Lemma 2.4. *BTC terminates on any input.*

Proof. Notice that each application of **BTC1**, **BTC2** and **TC3** runs in finite time. This is obvious for **BTC2**, and for **TC3** we note that only finitely many pairs can possibly appear in the list of coincidences in one application of the process. **BTC1** is a combination of one application of **TC2**, one of **TC3** and at most $|\overline{wa}|+1$ of **TC1**, so is finite too. Then, we note that since the set of relations and the alphabet are finite, each iteration of the loop for fixed $w \in \mathbf{n}(\Gamma(i))$ runs in finite time. Each such w is a word in the free band on A, which only has finitely many elements, so the algorithm will terminate at the latest when every w has been considered. □

Lemma 2.5. *The sequence of words w in the order they are considered forms a lenlex-increasing sequence.*

Proof. TODO. This will just help clear some messy things up later on. I think we just need to prove no words lenlex-less than the current word being considered can be discovered later on. This can be done by induction maybe? It follows from the fact that words are defined at each layer. □

Lemma 2.6. *Every word that is considered by BTC is pushed through the exact same set of relations.*

Proof. TODO make more rigorous - trivial for R, and steps 5 and 6 cover all cases for implicit relations.
TODO okay there's a problem here. Relations can also be pushed as part of BTC1. I think there must be a way we can argue that they're harmless enough that they won't affect the part of the properness proof I'm using the lemma for. □

To prove validity of BTC, it suffices to show that the described $\bar{R}^\sharp$-process is proper. We run through the conditions (a), (b) and (c) from Definition 3.4 of [5]. We may assume that i is chosen at the end of a loop to avoid an intermediate scenario. Assume the current word that is being processed by the algorithm is t.

(a) The first condition requires that for all $w \in \mathbf{n}(\Gamma(i))$ and for all $a \in A$, if $\mathbf{e}_i(w, a)$ is undefined, then there is $j > i$ such that either w is no longer a node, or the edge is defined. This holds for the same reasons as it does for other Todd-Coxeter algorithm implementations. At step i, for a fixed word w, we begin by defining $\mathbf{e}_i(w, a)$ for each $a \in A$. So if the edge is undefined but the node w exists, it must be that $t \leq w$, and the edge will be defined at the latest when we consider the corresponding w.

(b) Suppose here that at step i, a node w in $\Gamma(i)$ exists such that for some $\{u, v\} \in \bar{R}$, $\mathbf{e}_i(w, u)$ and $\mathbf{e}_i(w, v)$ are defined but not equal. We distinguish between three cases, depending on which subset of $\bar{R}$ the relation lies in.

  1. If $\{u, v\} \in R$, then $\{u, v\}$ is an explicit relation which we expect to push w through at some point. It must be that w has not yet been processed, since otherwise **TC2** and **TC3** would have made the two nodes equal. If w eventually disappears from the list of nodes, it cannot appear again, so we are done. If it doesn't, then it will eventually be considered, and for all j past the point where the corresponding coincidences are merged, the two nodes will be equal.

  2. If $\{u, v\} \in \phi(A)$, then we may assume u is canonical and $v = uu$. Let's first suppose u will be considered (or was considered) as a node at some point. We may also assume w will be considered (or was considered) since otherwise, we know w will eventually disappear from the node set and we are done.

     2.1. If $t \leq u \leq w$, then when considering w, step 5 ensures w is pushed through u.

     2.2. If $t \leq w \leq u$, then when considering u, either w is pushed through u in step 6 or w has disappeared from the node set. In both cases we are done.

     2.3. If $u \leq t \leq w$, then when considering w, step 5 ensures w is pushed through u.

     2.4. If $u \leq w \leq t$, then we have a contradiction: when considering w, w would have been pushed through u in step 5.

     2.5. If $w \leq t \leq u$, then when considering u, either w is pushed through u in step 6 or w has disappeared from the node set. In both cases we are done.

     2.6. If $w \leq u \leq t$, then either w had disappeared from the node set before u was considered, or w would have been pushed through u in step 6 when considering u. The latter case gives a contradiction.

     With these cases proved we now assume u was never considered. We prove that this scenario works by first assuming $w = \varepsilon$. The word u is never considered so it must eventually disappear from the node set, and following the path labelled u from the empty word must lead to some other node label. Since BTC terminates by 2.4, say at step N, we know that we can follow the path $\mathbf{e}_N(\varepsilon, u)$ and end up at a canonical node u' which has been considered. Since this node has been considered, we are guaranteed that

$$\mathbf{e}_N(\varepsilon, u'u') = \mathbf{e}_N(\varepsilon, u') = \mathbf{e}_N(\varepsilon, u) = u'.$$

We now note that for the paths u and u', corresponding to distinct words in the free band, to be merged, it must be that relations were pushed via **TC2** and created coincidences. Let R_u be the set of relations which resulted in the two words merging. Then there must be an elementary sequence from u to u' with respect to R_u. By 2.6, it must be that every relation in R_u was also pushed through u' and all subsequent nodes, so that

$$\mathbf{e}_N(u', u) = \mathbf{e}_N(u', u').$$

Then, we get that

$$\mathbf{e}_N(\varepsilon, uu) = \mathbf{e}_N(\mathbf{e}_N(\varepsilon, u), u) = \mathbf{e}_N(u', u) = \mathbf{e}_N(u', u') = \mathbf{e}_N(\varepsilon, u'u') = \mathbf{e}_N(\varepsilon, u).$$

If w is a node other than ε, I believe it's a question of "shifting the origin". (TODO)

(c) Coincidences are regularly processed so this condition easily holds.

References

[1] A. P. Biryukov. Varieties of idempotent semigroups. *Algebra and Logic*, 9(3):153–164, May 1970.

[2] Charles Fennemore. All varieties of bands. *Semigroup Forum*, 1(1):172–179, December 1970.

[3] J.A Gerhard. The lattice of equational classes of idempotent semigroups. *Journal of Algebra*, 15(2):195–224, June 1970.

[4] John Mackintosh Howie. *Fundamentals of semigroup theory*, volume 12. Clarendon Oxford, 1995.

[5] J. D. Mitchell, F. L. Smith, M. Tsalakou, and T. D. H. Coleman. The todd-coxeter algorithm for semigroups and monoids. *Preprint*, 2020.