Identities and bases in the sylvester and Baxter monoids

Alan J. Cain¹ · António Malheiro¹,² · Duarte Ribeiro¹,²

Received: 28 June 2021 / Accepted: 6 November 2022 / Published online: 21 January 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
This paper presents new results on the identities satisfied by the sylvester and Baxter monoids. We show how to embed these monoids, of any rank strictly greater than 2, into a direct product of copies of the corresponding monoid of rank 2. This confirms that all monoids of the same family, of rank greater than or equal to 2, satisfy exactly the same identities. We then give a complete characterization of those identities, and prove that the varieties generated by the sylvester and the Baxter monoids have finite axiomatic rank, by giving a finite basis for them.

Keywords Sylvester monoid · Baxter monoid · Variety · Identities · Equational basis · Axiomatic rank

Mathematics Subject Classification 08B05 · 05E99 · 20M05 · 20M07 · 20M32

This work is funded by National Funds through the FCT – Fundação para a Ciência e a Tecnologia, I.P., under the scope of the projects UIDB/00297/2020 and UIDP/00297/2020 (Center for Mathematics and Applications) and the project PTDC/MAT-PUR/31174/2017. The third author is funded by National Funds through the FCT – Fundação para a Ciência e a Tecnologia, I.P., under the scope of the studentship SFRH/BD/138949/2018.

Duarte Ribeiro
dc.ribeiro@campus.fct.unl.pt

Alan J. Cain
a.cain@fct.unl.pt

António Malheiro
ajm@fct.unl.pt

1 Centro de Matemática e Aplicações, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829–513 Caparica, Portugal

2 Departamento de Matemática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-513 Caparica, Portugal
1 Introduction

When studying the identities satisfied by a given semigroup $S$, if $S$ indeed satisfies a non-trivial identity, two natural questions arise: The first is the finite basis problem, that is, are the identities satisfied by $S$ consequences of those in some finite subset (see [50, 54]). There exist several powerful methods with which to approach the problem for finite semigroups, however, such is not the case for infinite semigroups. The second question is the computational complexity of the identity checking problem $\text{CHECK-Id}(S)$ [29], that is, the decision problem whose instance is an arbitrary identity $u \approx v$, and the answer to such an instance is ‘YES’ if $S$ satisfies $u \approx v$, and ‘NO’ if it does not. It is well-known that, for any finite semigroup $S$, the problem $\text{CHECK-Id}(S)$ is decidable, since there are only finitely many substitutions of the variables occurring in the identity by elements of $S$. Furthermore, $\text{CHECK-Id}(S)$ belongs in the complexity class $\text{co-NP}$. However, in the case of infinite semigroups, the brute-force approach used in the finite case does not work, and only recently there have been results on the computational complexity of identity checking for infinite semigroups, beyond undecidability and trivial or easy decidability in linear time (see [15, 16, 30]).

The plactic monoid $\text{plac}$ [35], whose elements can be identified with Young tableaux, has long been considered an important monoid, due to its numerous applications in different areas of mathematics, such as algebraic combinatorics [37], representation theory [18, 21], symmetric functions [38, 52], Kostka–Foulkes polynomials [34], and crystal bases [6]. By its definition via Schensted’s insertion algorithm [51], the plactic monoid has decidable word problem. The question of identities satisfied by the plactic monoid is actively studied [26, 33], since it is an infinite monoid with a powerful combinatorial structure. Cain et al. [8] have shown that the plactic monoid of finite rank $n$ does not satisfy any non-trivial identity of length less than or equal to $n$, thus showing that there is no single “global” identity satisfied by every plactic monoid of finite rank. On the other hand, Johnson and Kambites [27] gave a tropical representation of the plactic monoid of every finite rank, thus showing that they all satisfy non-trivial identities. By results given in [16] and [28], it is known that the identity checking problem in monoids of upper triangular tropical matrices (see, for example, [39]) is in the complexity class $P$. Daviaud et al. [16] also have shown that the monoid of $2 \times 2$ upper triangular tropical matrices, the bicyclic monoid, and the plactic monoid of rank 2 satisfy exactly the same identities. Since the bicyclic monoid is not finitely based [53], none of these monoids are.

In the context of combinatorial Hopf algebras, whose bases are indexed by combinatorial objects, the plactic monoid is used to construct the Hopf algebra of free symmetric functions $\text{FSym}$ [17, 46], whose bases are indexed by standard Young tableaux. In this context, other monoids arise with similar combinatorial properties to the plactic monoid: the Hopf algebra $\text{Sym}$ of non-commutative symmetric functions [19], whose bases are indexed by integer compositions, is obtained from the hypoplactic monoid $\text{hypo}$ [32, 43], the monoid of quasi-ribbon tableaux; the Loday–Ronco Hopf algebra $\text{PBT}$ [24, 36], whose bases are indexed by planar binary trees, is obtained from the sylvester monoid $\text{sylv}$ [24], the monoid of right strict binary search trees; the Baxter Hopf algebra $\text{Baxter}$ [20, 48], whose bases are indexed by Baxter permutations [4], is obtained from the Baxter monoid $\text{baxt}$ [20], the monoid of pairs of
twin binary search trees. These monoids satisfy identities, and the shortest identities have been characterized [9]. Unlike in the case of the plactic monoid, these identities are independent of rank, except for the case of rank 1.

The identities satisfied by the hypoplactic monoid have been studied in depth by the present authors in [13]. It was shown that the hypoplactic monoids of rank greater than or equal to 2 all satisfy exactly the same identities, which have been fully characterized. Furthermore, a finite basis was given for the variety generated by hypo, thus proving that it has finite axiomatic rank. Although not stated in the article, the characterization of the identities implies that CHECK-Id(hypo) is in the complexity class P. These results were obtained by extensively using an alternate characterization of the hypoplactic monoid using inversions, which arises as a consequence of [43, Subsection 4.2].

This paper focuses on the sylvester and Baxter monoids, as well as the \#-sylvester monoid, whose elements are identified with left strict binary search trees and whose properties can be derived from those of the sylvester monoid by parallel reasoning. These monoids are closely related to each other (see [20, Proposition 3.7]), as well as to the hypoplactic monoid (see [11, 47]). The main goal of the paper is to present a systematic study of the identities satisfied by these monoids, in the same way as the one given for the hypoplactic monoid in [13]. This paper also gives an alternate characterization of these monoids, by introducing the concepts of right and left precedences, which serve the same purpose as inversions for the hypoplactic monoid. The authors of this paper remark that Theorems 3.9, 3.11, 3.12, 4.16, 4.17 and 4.18 have been proven independently, using different methods, in [14, Propositions 6.6 and 6.10 and Theorems 6.7 and 6.11] and in [22, Theorems 4.6, 4.7 and 4.10]. Corollary 4.15 was also proved independently, using different methods, in [14, Propositions 6.6 and 6.10].

2 Preliminaries and notation

This section gives the necessary background on universal algebra (see [5, 7, 40, 42]), in the context of monoids, followed by the definition and essential facts about the sylvester and Baxter monoids.

For the necessary background on semigroups and monoids, see [25]; for presentations, see [23]; for computational complexity, see [44]; for a general background on the plactic monoid, see [37, Chapter 5], and on the hypoplactic monoid, see [43] and [10].

2.1 Varieties, identities and bases

The background given in this subsection is mostly identical to the one given in [13, Subsection 2.1], of which this paper is a sequel. As such, we omit most of the subsection, with the following exceptions:

We define the content and support of a balanced identity as the content and support of both sides of the identity, respectively.
An equational theory $\Sigma$ is left 1-hereditary if, for every identity $u \approx v$ of $\Sigma$ and any variable $x \in \text{supp}(u \approx v)$, the identity $u' \approx v'$ is in $\Sigma$, where $u_1$ (respectively, $v_1$) is the longest prefix of $u$ (respectively, $v$) where $x$ does not occur (see [41, 45, 54]). We define right 1-hereditary equational theories in a dual way. The equational theory of the variety generated by the bicyclic monoid, which coincides with that of the variety generated by the plactic monoid of rank 2 (see [27]), is both left and right 1-hereditary (see [45, 53]).

For a given semigroup $S$, its identity checking problem $\text{Check- Id}(S)$ is the following combinatorial decision problem: the instance is an arbitrary identity $u \approx v$; the answer to such an instance is ‘YES’, if $S$ satisfies the identity, and ‘NO’ otherwise. Notice that the semigroup itself is fixed, as such, it is only the identity $u \approx v$ that serves as the input. Therefore, the time/space complexity of $\text{Check- Id}(S)$ should be measured only in terms of the size of the identity. Naturally, the problem can also be considered for monoids.

### 2.2 The sylvester and #-sylvester monoids

This subsection gives a brief overview of the sylvester and #-sylvester monoids and their related combinatorial objects and insertion algorithms, as well as results from [9]. We introduce a new characterization of these monoids, analogous to the one given in [43] for the hypoplactic monoid, as well as some new notation. For more information on the sylvester monoid, see [24] and [11]; for more information on binary search trees, see [31] and [2].

Let $\mathcal{A} = \{1 < 2 < 3 < \cdots\}$ denote the set of positive integers, viewed as an infinite ordered alphabet, and let $\mathcal{A}_n = \{1 < \cdots < n\}$ denote the set of the first $n$ positive integers, viewed as a finite ordered alphabet. For brevity, we will write ‘the node $a'$ instead of ‘the node labelled with $a$’.

A right strict binary search tree is a labelled rooted binary tree where the label of each node is greater than or equal to the label of every node in its left subtree, and strictly less than every node in its right subtree. A left strict binary search tree is a labelled rooted binary tree where the label of each node is strictly greater than the label of every node in its left subtree, and less than or equal to every node in its right subtree. The following are examples of, respectively, right and left strict binary search trees:

\begin{align*}
\text{(2.1)} & & \begin{array}{c}
\includegraphics[scale=0.5]{tree1.png}
\end{array} \\
\text{(2.2)} & & \begin{array}{c}
\includegraphics[scale=0.5]{tree2.png}
\end{array}
\end{align*}

The (left-to-right) postfix (or postorder) traversal of a labelled rooted binary search tree $T$ is the sequence of nodes obtained by recursively performing the postfix traversal.
of the left subtree of the root of \( T \), then recursively performing the postfix traversal of the right subtree of the root of \( T \), and then adding the root of \( T \) to the sequence. The (left-to-right) postfix reading of a labelled rooted binary search tree \( T \) is the word \( \text{Post}(T) \) obtained by listing the labels of the nodes in the order visited during the postfix traversal. For example, the postfix reading of the tree 2.1 is 1142557654.

The (left-to-right) prefix (or preorder) traversal of a labelled rooted binary search tree \( T \) is the sequence of nodes obtained by first adding the root of \( T \) to the sequence, then recursively performing the prefix traversal of the left subtree of the root of \( T \), and then recursively performing the prefix traversal of the right subtree of the root of \( T \). The (left-to-right) prefix reading of a labelled rooted binary search tree \( T \) is the word \( \text{Pre}(T) \) obtained by listing the labels of the nodes in the order visited during the prefix traversal. For example, the prefix reading of the tree 2.2 is 5411245765.

The infix (or inorder) traversal of a labelled rooted binary search tree \( T \) is the sequence of nodes obtained by recursively performing the infix traversal of the left subtree of the root of \( T \), then adding the root of \( T \) to the sequence, and then recursively performing the infix traversal of the right subtree of the root of \( T \). The following result is immediate from the definitions of right and left strict binary search trees:

**Proposition 2.1** For any right or left strict binary search tree \( T \), if a node \( a \) is encountered before a node \( b \) in an infix traversal, then \( a \leq b \).

In other words, the infix traversal visits nodes in weakly increasing order, in right or left strict binary search trees.

Let \( T \) be a right or left strict binary search tree, and let \( a \in S \) be a symbol which labels a node of \( T \). We say that a node \( a \) is topmost if all other nodes \( a \) are its descendants. The following lemma, and its consequences, will be used thoroughly in this section:

**Lemma 2.2** ([12, Lemma 6.6]) Every node \( a \) appears on a single path descending from the root to a leaf; thus, there is a unique topmost node \( a \).

Let \( T \) be a right or left strict binary search tree, and consider two nodes of \( T \), respectively labelled with \( a \) and \( b \), for some \( a, b \in S \). Consider the lowest common ancestor of nodes \( a \) and \( b \), labelled with \( c \), for some \( c \in S \). If the node \( a \) is in the left subtree of the node \( c \) or coincides with it, and the node \( b \) is in the right subtree of the node \( c \) or coincides with it, and the nodes \( a \) and \( b \) do not both coincide with the node \( c \), then we say that the node \( a \) is to the left of the node \( b \), and the node \( b \) is to the right of the node \( a \). It is immediate to see that a node \( a \) is to the left of a node \( b \) if and only if the infix traversal visits the node \( a \) before the node \( b \), hence \( a \) is less than or equal to \( b \). Furthermore, if we consider a subset of nodes of \( T \), the definition of leftmost and rightmost nodes follows naturally.

The following algorithm allows us to insert a letter from \( S \) into an existing right strict binary search tree, as a leaf node in the unique place that maintains the property of being a right strict binary search tree:

Let \( u \in S^n \). Using the insertion algorithm above, we can compute a unique right strict binary search tree \( P_{\text{slyl}}(u) \) from \( u \): we start with the empty tree and insert the letters of \( u \), one-by-one from right-to-left. The arrow in the notation indicates the direction
Algorithm 1: Right strict leaf insertion.

Input: A right strict binary search tree $T$ and a symbol $a \in \mathcal{A}$.
Output: A right strict binary search tree $T \leftarrow a$.

1. if $T$ is empty, then
2. create a node and label it $a$;
3. else
4. examine the label $x$ of the root node; if $a > x$, recursively insert $a$ into the right subtree of the root node; otherwise recursively insert $a$ into the left subtree of the root node;
5. return the resulting tree.

in which the word should be read, when computing the right strict binary search tree. Notice that, for any right strict binary search tree $T$, we have that $P^\leftarrow_{\text{sylv}} (\text{Post}(T)) = T$, that is, the right strict insertion algorithm, with the postfix reading of $T$ as input, gives back $T$. As such, any right strict binary search tree can be seen as an output of the right strict insertion algorithm.

We define the relation $\equiv_{\text{sylv}}$ on $\mathcal{A}^*$ as follows: For $u, v \in \mathcal{A}^*$,

$$u \equiv_{\text{sylv}} v \iff P^\leftarrow_{\text{sylv}} (u) = P^\leftarrow_{\text{sylv}} (v).$$

This relation is a congruence on $\mathcal{A}^*$, called the sylvester congruence. The factor monoid $\mathcal{A}^*/\equiv_{\text{sylv}}$ is the infinite-rank sylvester monoid, denoted by sylv. The congruence $\equiv_{\text{sylv}}$ naturally restricts to a congruence on $\mathcal{A}^*_n$, and the factor monoid $\mathcal{A}^*_n/\equiv_{\text{sylv}}$ is the sylvester monoid of rank $n$, denoted by sylv$_n$.

It follows from the definition of $\equiv_{\text{sylv}}$ that each element $[u]_{\text{sylv}}$ of sylv can be identified with the combinatorial object $P^\leftarrow_{\text{sylv}} (u)$. As such, for each right strict binary search tree $T$, the set of words $u \in \mathcal{A}^*$ such that $P^\leftarrow_{\text{sylv}} (u) = T$ is called the sylvester class of $T$, and the postfix reading of $T$ is called the canonical word of the sylvester class of $T$.

Recall that the content of $u$ describes the number of occurrences of each letter of $\mathcal{A}$ in $u$. It is immediate from the definition of the sylvester monoid that if $u \equiv_{\text{sylv}} v$, then $\text{cont}(u) = \text{cont}(v)$. Thus, we can define the content of an element of sylv as the content of any word which represents it. Furthermore, since $\text{cont}(u) = \text{cont}(v)$ implies that $\text{supp}(u) = \text{supp}(v)$, we can also define the support of an element of sylv as the support of any word which represents it. We define the content and support of a right strict binary search tree as the content and support of its corresponding sylvester class.

Notice that sylv$_n$ is a submonoid of sylv, for each $n \in \mathbb{N}$, and, for $n, m \in \mathbb{N}$, if $n \leq m$, then sylv$_n$ is a submonoid of sylv$_m$.

The sylvester monoid can also be defined by the presentation $\langle A \mid \mathcal{R}_{\text{sylv}} \rangle$, where

$$\mathcal{R}_{\text{sylv}} = \{(caub, acub) : a \leq b < c, u \in \mathcal{A}^*\}.$$ 

These defining relations are known as the sylvester relations. A presentation for the sylvester monoid of rank $n$, for some $n \in \mathbb{N}$, can be obtained by restricting generators and relations of the above presentation to generators in $\mathcal{A}_n$. 

Springer
We now give a new alternative characterization of the sylvester monoid, inspired by the characterization of the hypoplactic monoid using inversions (see [13, 43]). Let \( u \in \mathcal{A}^* \) and let \( a, b \in \text{supp}(u) \) be such that \( a < b \). We say \( u \) has a \( b-a \) right precedence if, when reading \( u \) from right to left, \( b \) occurs before the first occurrence of \( a \) and, for any \( c \in \text{supp}(u) \) such that \( a < c < b \), the symbol \( c \) does not occur before the first occurrence of \( a \). The number of occurrences of \( b \) before the first occurrence of \( a \) is the index of the right precedence.

Notice that, by the definition of a right precedence, for any given \( a \in \text{supp}(u) \), there is at most one \( b \in \text{supp}(u) \) such that \( u \) has a \( b-a \) right precedence (of index \( k \), for some \( k \in \mathbb{N} \)). On the other hand, \( u \) can have several right precedences of the form \( b-x \), for a fixed \( b \).

**Example 2.3** The word 3123 has a 2-1 and a 3-2 right precedence, both of index 1, however, it does not have a 3-1 right precedence, since 2 occurs before the first occurrence of 1; the word 2313 has a 3-1 right precedence of index 1 and a 3-2 right precedence of index 2; and the word 3132 has a 2-1 right precedence of index 1, and does not have a 3-1 right precedence.

In order to prove that the sylvester monoid can be characterized using only the content and right precedences of words, we require some lemmata:

**Lemma 2.4** Let \( u \in \mathcal{A}^* \), and let \( a, b \in \text{supp}(u) \) be such that \( a < b \) and \( b \) occurs at least \( k \) times in \( u \), for some \( k \in \mathbb{N} \). Then, \( u \) has a \( b-a \) right precedence of index \( k \) if and only if the topmost node \( a \) in \( P_{\text{sylv}}(u) \) has exactly \( k \) ancestor nodes labelled with \( b \), and no ancestor nodes labelled with \( c \), for any \( c \in \text{supp}(P_{\text{sylv}}(u)) \) such that \( a < c < b \).

**Proof** Suppose that \( u \) has a \( b-a \) right precedence of index \( k \). It is clear that, as a consequence of the insertion algorithm 1, the topmost node \( a \) in \( P_{\text{sylv}}(u) \) corresponds to the rightmost occurrence of \( a \) in \( u \). Thus, there are exactly \( k \) symbols \( b \) inserted before \( a \), when computing \( P_{\text{sylv}}(u) \). Notice that, by Lemma 2.2, the corresponding nodes must be in a single path from the root to any leaf of \( P_{\text{sylv}}(u) \). Furthermore, since no symbol \( c \) occurs before the rightmost symbol \( a \), when reading \( u \) from right-to-left, for any \( a < c < b \), we have that the rightmost symbol \( a \) must be inserted as a left child of a node \( b \), in particular, the node corresponding to the \( k \)-th inserted symbol \( b \). This is due to the fact that, during the “searching” step of the insertion algorithm, the symbol \( a \) will satisfy exactly the same criteria as the last inserted symbol \( b \), except when checking the \( k \)-th node \( b \). Thus, the topmost node \( a \) in \( P_{\text{sylv}}(u) \) must have exactly \( k \) ancestor nodes labelled with \( b \), and no ancestor nodes labelled with \( c \).

On the other hand, suppose \( u' \) is such that \( u \equiv_{\text{sylv}} u' \). Suppose, in order to obtain a contradiction, that \( u' \) does not have a \( b-a \) right precedence of index \( k \). Then, by the previous part of this proof, we have that \( P_{\text{sylv}}(u') \neq P_{\text{sylv}}(u) \), which contradicts our hypothesis. \( \Box \)

It is clear, from the previous lemma, that all words in the same sylvester class must share exactly the same right precedences. However, it is not immediate that two words which share the same content and right precedences will produce exactly the same output, when considered as inputs for the insertion algorithm.
Lemma 2.5 Let \( T \) be a right strict binary search tree, and let \( a, b \in \text{supp}(T) \), with \( a \) strictly less than \( b \), be such that the topmost node \( a \) is a left child of a node \( b \). Then, all words in the sylvester class of \( T \) have a \( b-a \) right precedence of index \( k \), where \( k \) is the number of ancestor nodes of the topmost node \( a \) labelled with \( b \).

Proof If the topmost node \( a \) had an ancestor node labelled with \( c \), for some \( a < c < b \), then the node \( a \) would be in the left subtree of a node \( c \), but even more so, since the node \( a \) is a left child of the node \( b \), then the node \( b \) would also be in the left subtree of a node \( c \), which contradicts the hypothesis that \( c \) is strictly less than \( b \). The result follows from Lemma 2.4. \( \square \)

More generally, we can also see that if a topmost node \( a \) is in the left subtree of some node \( b \), then the words in the sylvester class of \( T \) will have a \( c-a \) right precedence, for some \( c \in \text{supp}(T) \) such that \( a < c \leq b \).

Suppose \( u, v \in \Sigma^* \) are words which share the same content, but \( P_{\text{sylv}}^< (u) \neq P_{\text{sylv}}^< (v) \). It is immediate that at least more than one different symbol must occur in \( u \) and \( v \). Furthermore, if the roots are labelled differently, it is clear that the words in the sylvester class of the tree whose root label is higher will have a right precedence concerning the root of the other tree, while the words in the sylvester class of the tree whose root label is lower will not. Thus, we will consider that, up to a certain depth, the two trees \( P_{\text{sylv}}^< (u) \) and \( P_{\text{sylv}}^> (v) \) are identical.

Let \( T_{u, v} \) be the right strict binary search tree obtained by removing all nodes of decreasing depth in \( P_{\text{sylv}}^< (u) \) and \( P_{\text{sylv}}^> (v) \) until we obtain the same tree.

Lemma 2.6 Any node of \( P_{\text{sylv}}^< (u) \) and \( P_{\text{sylv}}^> (v) \), corresponding to a leaf of highest depth in \( T_{u, v} \), has a left (respectively, right) child in \( P_{\text{sylv}}^< (u) \) if and only if it has a left (respectively, right) child in \( P_{\text{sylv}}^> (v) \).

Proof Suppose, in order to obtain a contradiction, that there is a node in \( P_{\text{sylv}}^< (u) \) and \( P_{\text{sylv}}^> (v) \), corresponding to a leaf of highest depth in \( T_{u, v} \) and labelled with \( a \), for some \( a \in \text{supp}(T_{u, v}) \), which has a child node in \( P_{\text{sylv}}^< (u) \) that does not occur in \( P_{\text{sylv}}^> (v) \). Assume, without loss of generality, that it is a left child. Notice that the node \( a \) cannot be the only leaf of highest depth in \( T_{u, v} \). Furthermore, this node cannot correspond to the leftmost leaf of highest depth in \( T_{u, v} \), otherwise, this would imply that \( P_{\text{sylv}}^< (u) \) has at least one node labelled with a symbol which either cannot occur in \( P_{\text{sylv}}^> (v) \), or occurs more times in \( P_{\text{sylv}}^< (u) \) than in \( P_{\text{sylv}}^> (v) \), which contradicts our hypothesis that \( u \) and \( v \) share the same content.

Let \( b \) be the label of the left child of the node \( a \), in \( P_{\text{sylv}}^< (u) \); let \( c \) be the label of the node corresponding to the rightmost leaf of highest depth in \( T_{u, v} \) to the left of the node \( a \); let \( d \) be the label of the lowest common ancestor of the nodes \( a \) and \( c \). Notice that the nodes \( a \) and \( b \) are in the right subtree of the node \( d \), hence \( c \leq d < b \leq a \). On the other hand, the infix traversal, in \( P_{\text{sylv}}^< (u) \), first visits the node \( c \), then the node \( d \), then the node \( b \), and immediately afterwards the node \( a \). Since \( u \) and \( v \) have the same content, the infix traversal, in \( P_{\text{sylv}}^> (v) \), must also first visit the node \( c \), then all nodes labelled with \( b \), and then the node \( a \). As such, since in \( P_{\text{sylv}}^> (v) \), the node \( a \) does not have a left child, and the node \( c \) corresponds to the rightmost leaf of highest depth in \( T_{u, v} \) to the left of the node \( a \), all nodes labelled with \( b \) must be descendants of the node.
c. But this implies that, in $P_{\text{sylv}}^\leftarrow (v)$, all nodes labelled with $b$ are in the left subtree of the node $d$, hence $b \leq d$, which contradicts our hypothesis.

This lemma shows that two different right strict binary search trees are identical up to a certain depth, and the first difference between the trees, when searching in depth, is a difference between the labels of some nodes. Now, we are ready to prove the right precedence characterization of the sylvester monoid:

**Proposition 2.7** For $u, v \in \mathcal{A}_n^*$, we have that $u \equiv_{\text{sylv}} v$ if and only if $u$ and $v$ share exactly the same content and right precedences.

**Proof** We already know that if two words $u$ and $v$ are in the same sylvester class, then they share the same content and right precedences, by Lemma 2.4. Suppose now that $u$ and $v$ share the same content, but $P_{\text{sylv}}^\leftarrow (u) \neq P_{\text{sylv}}^\leftarrow (v)$. Once again, consider the right strict binary search tree $T_{u,v}$, obtained by removing all nodes of decreasing depth in $P_{\text{sylv}}^\leftarrow (u)$ and $P_{\text{sylv}}^\leftarrow (v)$ until we obtain the same tree. By the previous lemma and our hypothesis, there must exist a node in $P_{\text{sylv}}^\leftarrow (u)$ and $P_{\text{sylv}}^\leftarrow (v)$, corresponding to a leaf of highest depth in $T_{u,v}$, which has a child node labelled differently in $P_{\text{sylv}}^\leftarrow (u)$ than in $P_{\text{sylv}}^\leftarrow (v)$.

Let $a$ be the label of the child node in $P_{\text{sylv}}^\leftarrow (u)$ and $b$ be the label in $P_{\text{sylv}}^\leftarrow (v)$. Assume, without loss of generality, that $a < b$. Notice that the node $a$ in $P_{\text{sylv}}^\leftarrow (u)$ must be topmost, otherwise, the node $b$ in $P_{\text{sylv}}^\leftarrow (v)$ would be in the left subtree of a node labelled with $a$. Let $c$ be the label of the parent node of node $a$ in $P_{\text{sylv}}^\leftarrow (u)$ and node $b$ in $P_{\text{sylv}}^\leftarrow (v)$. Using an argument similar to the one used in the proof of Lemma 2.6, we can see that the topmost node $a$ is also a descendant of the node $c$, in $P_{\text{sylv}}^\leftarrow (v)$. Even more so, the topmost node $a$ must be in the left subtree of the node $b$, in $P_{\text{sylv}}^\leftarrow (v)$.

Suppose nodes $a$ and $b$ are left children. First of all, notice that $a < b \leq c$. Furthermore, by Lemma 2.5, we have that $u$ has a $c$-$a$ right precedence of index $k$, where $k$ is the number of ancestor nodes of the node $a$ labelled with $c$. If $b$ is equal to $c$, then $v$ either has a $c$-$a$ right precedence of index strictly greater than $k$, or it does not have a $c$-$a$ right precedence at all; otherwise, if $b$ is strictly less than $c$, then $v$ does not have a $c$-$a$ right precedence.

Suppose nodes $a$ and $b$ are right children. If $u$ has a $d$-$a$ right precedence of index $k$, for some $d \in \text{supp}(T_{u,v})$ and some $k \in \mathbb{N}$, then the node $a$ is in the left subtree of a node $d$, in $P_{\text{sylv}}^\leftarrow (u)$ as well as in $P_{\text{sylv}}^\leftarrow (v)$, and the node $b$ is in the left subtree of that node $d$ in $P_{\text{sylv}}^\leftarrow (v)$. If $b$ is equal to $d$, then $v$ either has a $d$-$a$ right precedence of index greater than $k$, or it does not have a $d$-$a$ right precedence at all; otherwise, if $b$ is strictly less than $d$, then $v$ does not have a $d$-$a$ right precedence. On the other hand, $u$ does not have a $d$-$a$ right precedence, for any $d \in \text{supp}(T_{u,v})$, then $v$ has a $d'$-$a$ right precedence, for some symbol $d'$ which is either equal to $b$ or labels an ancestor node of the topmost node $a$ in the left subtree of the node $b$, in $P_{\text{sylv}}^\leftarrow (v)$.

Thus, we can conclude that if $u$ and $v$ share the same content, but $P_{\text{sylv}}^\leftarrow (u) \neq P_{\text{sylv}}^\leftarrow (v)$, then $u$ and $v$ do not share the same right precedences.

The authors would like to remark that there is a slightly different proof of Proposition 2.7, and its required lemmata, in the third author’s Ph.D. thesis [49, Section 4.3, Proposition 4.3.9].
In light of the previous result, we say that an element \([u]_{\text{sylv}}\) of \(\text{sylv}\) has a \(b\)-\(a\) right precedence (of index \(k\)) if the word \(u\) itself, and hence any other word in \([u]_{\text{sylv}}\), has a \(b\)-\(a\) right precedence (of index \(k\)).

The sylvester and hypoplactic monoids are closely related (see [11, 47]). In fact, the hypoplactic monoid is a homomorphic image of the sylvester monoid. Notice that, for any right strict binary search tree with support \(\{a_1 < \cdots < a_k\}\), all words in its sylvester class have a \(a_{i+1} - a_i\) inversion if and only if a node \(a_{i+1}\) appears in a right subtree of a node \(a_i\). Hence, the map defined by

\[
[u]_{\text{sylv}} \mapsto [u]_{\text{hypo}},
\]

for \(u \in \mathcal{A}\), is a natural homomorphism of \(\text{sylv}\) onto \(\text{hypo}\). Therefore, \(\text{hypo}\) is in the variety generated by \(\text{sylv}\), which we will denote by \(V_{\text{sylv}}\), and thus must satisfy all identities satisfied by \(\text{sylv}\).

The insertion algorithm for left strict binary search trees is dual to Algorithm 1:

**Algorithm 2:**

- **Input:** A left strict binary search tree \(T\) and a symbol \(a \in \mathcal{A}\).
- **Output:** A left strict binary search tree \(T' \leftarrow a\).

1. if \(T\) is empty, then
2. create a node and label it \(a\);
3. else
4. examine the label \(x\) of the root node; if \(a < x\), recursively insert \(a\) into the left subtree of the root node; otherwise recursively insert \(a\) into the right subtree of the root node;
5. return the resulting tree.

Let \(u \in \mathcal{A}^*\). Using the insertion algorithm above, we can compute a unique left strict binary search tree \(P_{\text{#-sylv}}(u)\) from \(u\): we start with the empty tree and insert the symbols of \(u\), one-by-one from left-to-right. Notice that, for any left strict binary search tree \(T\), we have that \(P_{\text{#-sylv}}(\text{Pre}(T)) = T\), that is, the left strict insertion algorithm, with the prefix reading of \(T\) as input, gives back \(T\). As such, any left strict binary search tree can be seen as an output of the left strict insertion algorithm.

We define the \(#\)-sylvester congruence \(\equiv_{\text{#-sylv}}\), the infinite-rank \(#\)-sylvester monoid \(\text{sylv}\), the \(#\)-sylvester monoid of rank \(n\) \(\text{sylv}_n\), and the content and support of a \(#\)-sylvester class in a similar fashion as before.

The \(#\)-sylvester monoid can also be defined by the presentation \((\ast | A)R_{\text{sylv}}\), where

\[
R_{\text{sylv}} = \{(buac, buca) : a < b \leq c, u \in \mathcal{A}^*\}.
\]These defining relations are known as the \(#\)-sylvester relations.

The sylvester and \(#\)-sylvester monoids of finite rank \(n\) are anti-isomorphic: a natural anti-isomorphism of \(\text{sylv}_n\) into \(\text{sylv}_n\) arises by taking a right strict binary search tree, reflecting it about a vertical axis, and renumbering the label \(i\) of each node to \(n - i + 1\),
thus obtaining a left strict binary search tree. Similarly, we can define a natural anti-
isomorphism of $\text{sylv}_n^#$ into $\text{sylv}_n$. These anti-isomorphisms allow us to easily deduce
results for finite-rank $\#$-sylvester monoids from results for the finite-rank sylvester
monoids. However, notice that these natural anti-isomorphisms do not arise in the
infinite rank case, as there is no natural way to renumber the labels of the nodes. In
fact, we have the following:

**Proposition 2.8** There is no anti-isomorphism of $\text{sylv}$ into $\text{sylv}^#$.

**Proof** Suppose, in order to obtain a contradiction, that there exists such an anti-
isomorphism $\sigma : \text{sylv} \rightarrow \text{sylv}^#$. Notice that, for any element of the form $[a]_{\text{sylv}}$, where $a \in \mathcal{A}$, we have that $|\sigma([a]_{\text{sylv}})| = 1$, since $|\sigma([a]_{\text{sylv}})| > 1$ implies that

$$|\sigma([a]_{\text{sylv}})| = |\sigma^{-1}(\sigma([a]_{\text{sylv}}))| > 1,$$

along with the fact that the identity of sylv is mapped to the identity of sylv#. Thus, $\sigma$
maps the sylv-classes of generators into sylv#-classes of generators.

Thus, we have that $\sigma([1]_{\text{sylv}}) = [x]_{\text{sylv}^#}$, for some $x \in \mathcal{A}$. Since we are considering
the infinite-rank case, there must exist $b, y \in \mathcal{A}$ such that $y > x$ and $\sigma([b]_{\text{sylv}}) = [y]_{\text{sylv}^#}$. Notice that, since $\sigma$ maps 1 to $x$, then $b > 1$. As such, we have that $b11 \equiv_{\text{sylv}} 1b1$, but on the other hand,

$$\sigma([b11]_{\text{sylv}}) = [xxy]_{\text{sylv}^#} \neq [xyx]_{\text{sylv}^#} = \sigma([1b1]_{\text{sylv}}),$$

which contradicts the hypothesis that $\sigma$ is an anti-isomorphism. \qed

We now give a new alternative characterization of the $\#$-sylvester monoid, parallel
to the one given for the sylvester monoid. Let $u \in \mathcal{A}^*$ and let $a, b \in \text{supp}(u)$ be such
that $a < b$. We say $u$ has a $a$-$b$ left precedence of index $k$ if, when reading $u$ from left
to right, $a$ occurs before the first occurrence of $b$ and, for any $c \in \text{supp}(u)$ such that
$a < c < b$, the symbol $c$ does not occur before the first occurrence of $b$. The number
of occurrences of $a$ before the first occurrence of $b$ is the index of the left precedence.

Notice that, by the definition of a left precedence, for any given $b \in \text{supp}(u)$, there
is at most one $a \in \text{supp}(u)$ such that $u$ has a $a$-$b$ left precedence (of index $k$, for some
$k \in \mathbb{N}$). On the other hand, $u$ can have several left precedences of the form $a$-$x$, for a
fixed $a$.

**Example 2.9** The word 1231 has a 1-2 and a 2-3 left precedence, both of index 1, while
1312 has a 1-2 left precedence of index 2 and a 1-3 left precedence of index 1, and
3121 has a 1-2 left precedence of index 1.

The following proposition mirrors Proposition 2.7:

**Proposition 2.10** For $u, v \in \mathcal{A}_n^*$, we have that $u \equiv_{\text{sylv}} v$ if and only if $u$ and $v$ share
exactly the same content and left precedences.
In light of the previous result, we say that an element $[u]_{\text{sylv}^\#}$ of $\text{sylv}^\#$ has a $a\cdot b$ left precedence (of index $k$) if the word $u$ itself, and hence any other word in $[u]_{\text{sylv}^\#}$, has a $a\cdot b$ left precedence (of index $k$).

The hypoplactic monoid is also a homomorphic image of the $\#$-sylvester monoid. Hence, $\text{hypo}$ is in the variety generated by $\text{sylv}^\#$, which we will denote by $\mathbb{V}_{\text{sylv}^\#}$, and thus must satisfy all identities satisfied by $\text{sylv}^\#$.

### 2.3 The Baxter monoid

This subsection gives a brief overview of the Baxter monoid and its related combinatorial object and insertion algorithm, as well as results from [9]. Due to its connection with the sylvester and $\#$-sylvester monoids, we also give a new characterization of this monoid, derived from the characterizations of sylv and $\text{sylv}^\#$ given in the previous subsection. For more information, see [20] and [11].

The canopy of a rooted binary tree $T$ is the word over $\{0, 1\}$ obtained by doing an infix traversal of $T$, outputting 1 when an empty left subtree is encountered and 0 when an empty right subtree is encountered, then omitting the first and last symbols of the resulting word (which correspond, respectively, to the empty left subtree of the leftmost node and the empty right subtree of the rightmost node).

A pair of twin binary search trees consists of a left strict binary search tree $T_L$ and a right strict binary search tree $T_R$, such that $T_L$ and $T_R$ have the same content, and the canopies of $T_L$ and $T_R$ are complementary, in the sense that the $i$-th symbol of the canopy of $T_L$ is 0 (respectively 1) if and only if the $i$-th symbol of the canopy of $T_R$ is 1 (respectively 0). The following is an example of a pair of twin binary search trees:

Let $u \in A^*$. Due to [20, Proposition 4.5], the pair of binary search trees $(P_{\text{sylv}^\#}(u), P_{\text{sylv}}(u))$ is a pair of twin binary search trees. As such, by defining $P_{\text{baxt}}(u)$ as this pair, we can use Algorithms 1 and 2 to compute a unique pair of twin binary search trees $P_{\text{baxt}}(u)$ from $u$.

We define the Baxter congruence $\equiv_{\text{baxt}}$, the infinite-rank Baxter monoid $\text{baxt}$, the Baxter monoid of rank $n$ $\text{baxt}_n$, and the content and support of a Baxter class in a similar fashion as before.

The Baxter monoid can also be defined by the presentation $\langle A \mid \mathcal{R}_{\text{baxt}} \rangle$, where

$$\mathcal{R}_{\text{baxt}} = \{(\text{cuda}uvb, \text{cuad}uvb) : a \leq b < c \leq d, u, v \in A^*\} \cup \{(\text{budavc}, \text{buadvc}) : a < b \leq c < d, u, v \in A^*\}.$$

These defining relations are known as the Baxter relations.
The Baxter, sylvester and #-sylvester monoids are closely related, due to [20, Proposition 3.7]: For \( u, v \in A^* \), we have that \( u \equiv_{\text{baxt}} v \) if and only if \( u \equiv_{\text{sylv}} v \) and \( u \equiv_{\text{sylv}#} v \). As a consequence of this, and Propositions 2.7 and 2.10, we have the following:

**Corollary 2.11** For \( u, v \in A^*_n \), \( u \equiv_{\text{baxt}} v \) if and only if \( u \) and \( v \) share exactly the same content and left and right precedences.

In light of the previous result, we say that an element \([u]_{\text{baxt}}\) of \( \text{baxt} \) has a \( b \)-\( a \) right (respectively, left) precedence of index \( k \) if the word \( u \) itself, and hence any other word in \([u]_{\text{baxt}}\), has a \( b \)-\( a \) right (respectively, left) precedence of index \( k \).

It is also easy to see that the maps defined by

\[
[u]_{\text{baxt}} \mapsto [u]_{\text{sylv}} \quad \text{and} \quad [u]_{\text{baxt}} \mapsto [u]_{\text{sylv}#},
\]

for \( u \in A^* \), are natural homomorphisms of \( \text{baxt} \) onto \( \text{sylv} \) and \( \text{sylv}# \), respectively. As such, \( \text{baxt} \) is a subdirect product of \( \text{sylv} \) and \( \text{sylv}# \). Therefore, both \( \text{sylv} \) and \( \text{sylv}# \) are monoids in the variety generated by \( \text{baxt} \), which we will denote by \( V_{\text{baxt}} \), and thus must satisfy all identities satisfied by \( \text{baxt} \).

On the other hand, the map defined by

\[
[u]_{\text{baxt}} \mapsto ([u]_{\text{sylv}#}, [u]_{\text{sylv}}),
\]

for \( u \in A^* \), is an embedding of \( \text{baxt} \) into \( \text{sylv}# \times \text{sylv} \), due to [20, Proposition 3.7]. Therefore, \( \text{baxt} \) is a monoid in the varietal join \( V_{\text{sylv}} \vee V_{\text{sylv}#} \), and thus must satisfy all identities which are satisfied by both \( \text{sylv} \) and \( \text{sylv}# \), by Birkhoff’s Theorem. Furthermore, we have that \( V_{\text{baxt}} = V_{\text{sylv}} \vee V_{\text{sylv}#} \). This last observation was suggested by an anonymous reviewer.

The previously mentioned maps can be restricted to the finite rank case, and we obtain analogous results regarding the varieties generated by \( \text{sylv}_n \), \( \text{sylv}^+_n \) and \( \text{baxt}_n \).

### 3 Embeddings

In this section, we prove that the sylvester monoids of ranks greater than or equal to 2 satisfy exactly the same identities. We do this by constructing embeddings of sylvester monoids of any rank greater than 2 into direct products of copies of the sylvester monoid of rank 2, as it is not possible to embed one into another. Thus, they generate the same variety and, by Birkhoff’s Theorem, satisfy exactly the same identities. We also show that the basis rank of the variety generated by \( \text{sylv} \) is 2.

By parallel reasoning, we prove the same results for the \( \# \)-sylvester monoids of ranks greater than or equal to 2. Furthermore, as a consequence of [20, Proposition 3.7], we also obtain the same results for the Baxter monoids of ranks greater than or equal to 2.
3.1 Non-existence of embedding into a monoid of lesser rank

It is not possible to embed a sylvester monoid of finite rank into a sylvester monoid of lesser rank:

**Proposition 3.1** For all \( n > m \geq 1 \), there is no embedding of sylv\(_n\) into sylv\(_m\).

**Proof** First of all, notice that sylv\(_1\) is isomorphic to the free monogenic monoid and sylv\(_n\) is non-commutative, for any \( n \geq 2 \). Thus, there is no embedding of sylv\(_n\) into sylv\(_1\).

On the other hand, if there exists an embedding of sylv\(_n\) into sylv\(_m\), for some \( n > m \geq 2 \), then, since sylv\(_m\) is a submonoid of sylv\(_{n-1}\), there must also exist an embedding of sylv\(_n\) into sylv\(_{n-1}\). As such, we just need to prove that this second embedding cannot exist.

Suppose, in order to obtain a contradiction, that there exists \( n \geq 3 \) such that we have an embedding \( \phi : \text{sylv}_n \mapsto \text{sylv}_{n-1} \). Without loss of generality, suppose \( n \) is the smallest positive integer in such conditions.

Observe that \( \text{supp} (\phi ([2 \cdots n]_{\text{sylv}_n})) = \mathcal{A}_{n-1} \), that is, the image of the product of all generators of sylv\(_n\), except for 1, has all the possible letters of \( \mathcal{A}_{n-1} \). Indeed, if \( \text{supp} (\phi ([2 \cdots n]_{\text{sylv}_n})) \subsetneq \mathcal{A}_{n-1} \), we would be able to construct an embedding from the submonoid isomorphic to sylv\(_{n-1}\) of sylv\(_n\), generated by all generators of sylv\(_n\) except for 1, into a submonoid of sylv\(_{n-1}\) isomorphic to sylv\(_{n-2}\). This contradicts the minimality of \( n \).

Hence, since all letters of \( \mathcal{A}_{n-1} \) already occur in \( \phi ([2 \cdots n]_{\text{sylv}_n}) \), if we multiply this element by any other element of sylv\(_{n-1}\) to the left, we obtain an element with the same right precedences as \( \phi ([2 \cdots n]_{\text{sylv}_n}) \). Thus, by Proposition 2.7, since \( \phi ([12]_{\text{sylv}}) \) and \( \phi ([21]_{\text{sylv}}) \) have the same content, we have that

\[
\phi ([12]_{\text{sylv}_n} \cdot [2 \cdots n]_{\text{sylv}_n}) = \phi ([21]_{\text{sylv}_n} \cdot [2 \cdots n]_{\text{sylv}_n}).
\]

On the other hand, we have that

\[
[12]_{\text{sylv}_n} \cdot [2 \cdots n]_{\text{sylv}_n} \neq [21]_{\text{sylv}_n} \cdot [2 \cdots n]_{\text{sylv}_n},
\]

since the left-hand side has a 2-1 right precedence of index 2, and the right-hand side has a 2-1 right precedence of index 1.

This contradicts our hypothesis that \( \phi \) is injective. Hence, for all \( n \geq 2 \), there is no embedding of sylv\(_n\) into sylv\(_{n-1}\). As such, there is no embedding of sylv\(_n\) into sylv\(_m\), for \( n > m \geq 2 \).

**Corollary 3.2** There is no embedding of sylv into sylv\(_n\), for any \( n \in \mathbb{N} \).

**Proof** If such an embedding existed, for some \( n \in \mathbb{N} \), then, by restricting the embedding to the first \( n + 1 \) generators of sylv, we would obtain an embedding of sylv\(_{n+1}\) into sylv\(_n\), which contradicts the previous proposition.

If there existed an embedding of a #-sylvester monoid of finite rank into a #-sylvester monoid of lesser rank, then we would be able to compose it with the natural
anti-isomorphisms, thus obtaining an embedding for the sylvester case. Therefore, no such embedding exists, as well as no embedding of the infinite-rank #-sylvester monoid into a #-sylvester monoid of finite rank. We can also prove the corresponding result for the Baxter monoid:

**Proposition 3.3** For all \( n > m \geq 1 \), there is no embedding of \( \text{baxt}_n \) into \( \text{baxt}_m \).

**Proof** The proof follows the same reasoning as given in the proof of Proposition 3.1: Instead of considering the elements 

\[
[12]_{\text{sylv}} \cdot [2 \cdots n]_{\text{sylv}} \quad \text{and} \quad [21]_{\text{sylv}} \cdot [2 \cdots n]_{\text{sylv}},
\]

we consider, respectively, the elements 

\[
[2 \cdots n]_{\text{baxt}} \cdot [12]_{\text{baxt}} \cdot [2 \cdots n]_{\text{baxt}} \quad \text{and} \quad [2 \cdots n]_{\text{baxt}} \cdot [21]_{\text{baxt}} \cdot [2 \cdots n]_{\text{baxt}},
\]

which are different in \( \text{baxt}_n \) but whose images under an embedding would have the same content and left and right precedences. \( \square \)

**Corollary 3.4** There is no embedding of \( \text{baxt} \) into \( \text{baxt}_n \), for any \( n \in \mathbb{N} \).

### 3.2 Embedding into a direct product of copies of monoids of rank 2

Although it is not possible to embed the sylvester, #-sylvester and Baxter monoids, of rank greater than 2, into their corresponding monoids of rank 2, we now show how to embed each of them into a direct product of copies of their corresponding monoid of rank 2, starting with the sylvester monoid.

For any \( i, j \in \mathcal{A} \), with \( i < j \), define a map from \( \mathcal{A} \) to \( \text{sylv}_2 \) in the following way: For any \( a \in \mathcal{A} \),

\[
a \mapsto \begin{cases} 
[1]_{\text{sylv}} & \text{if } a = i; \\
[2]_{\text{sylv}} & \text{if } a = j; \\
[21]_{\text{sylv}} & \text{if } i < a < j; \\
[\varepsilon]_{\text{sylv}} & \text{otherwise}; 
\end{cases}
\]

and extend it to a homomorphism \( \varphi_{ij} : \mathcal{A}^* \rightarrow \text{sylv}_2 \), in the usual way. This homomorphism is analogous to the one given in [13, Subsection 3.2]. The proofs of the following lemmata and propositions make use of the new characterization using right precedences for the sylvester monoid.

Notice that \( \varphi_{ij}(w) \) is the sylvester class of the word obtained from \( w \) by replacing any occurrence of \( i \) by 1; any occurrence of \( j \) by 2; any occurrence of an \( a \), with \( i < a < j \), by 21; and erasing any occurrence of any other element.

**Lemma 3.5** \( \varphi_{ij} \) factors to give a homomorphism \( \varphi_{ij} : \text{sylv} \rightarrow \text{sylv}_2 \).
Proof Since sylv is given by the presentation \( \langle A \mid \mathcal{R}_{\text{sylv}} \rangle \), we just need to verify that both sides of the sylvester relations have the same image under \( \varphi_{ij} \).

Let \( a, b, c \in \mathcal{A} \) and \( u \in \mathcal{A}^* \) be such that \( a \leq b < c \). If \( \varphi_{ij} \) maps either \( a \) or \( c \) to \([\varepsilon]\)\text{sylv}_2\), then the images of \( \text{caub} \) and \( \text{acub} \) under \( \varphi_{ij} \) coincide. Assume, without loss of generality, that \( \varphi_{ij} \) does not map any letter to \([\varepsilon]\)\text{sylv}_2\). Then, \( \varphi_{ij} \) maps \( a \) to \([1]\)\text{sylv}_2\), \( b \) to either \([2]\)\text{sylv}_2\) or \([1]\)\text{sylv}_2\), and \( c \) to \([2]\)\text{sylv}_2\). Notice that \( b \) is mapped to an element with no right precedences. As such, we have that

\[
\varphi_{ij}(\text{caub}) = \varphi_{ij}(\text{acub}),
\]

since both sides of the equality have no right precedences. Hence, \( \mathcal{R}_{\text{sylv}} \subseteq \ker \varphi_{ij}. \)

Let \( w \in \mathcal{A}_n^* \), for some \( n \geq 3 \). Suppose \( \text{supp}(w) = \{a_1 < \cdots < a_m\} \), for some \( m \in \mathbb{N} \). Observe that, ranging \( 1 \leq i < m \), we can get the number of occurrences of \( a_i \) and \( a_{i+1} \) in \( w \) from the images of \([w]_{\text{sylv}}\) under the maps \( \varphi_{a_ia_{i+1}} \), since, when reading any word in \( \varphi_{a_ia_{i+1}}([w]_{\text{sylv}}) \), every occurrence of 1 corresponds exactly to an occurrence of \( a_i \) in \( w \), and every occurrence of 2 corresponds exactly to an occurrence of \( a_{i+1} \).

Recall that there is at most one index \( j, \) with \( i < j \leq m \), such that \( w \) has a \( a_j-a_i \) right precedence. Ranging \( 1 \leq i < j \leq m \), we can also check if \( w \) has a \( a_j-a_i \) right precedence: If \( w \) does not have any \( b-a_i \) right precedence, for \( b < a_j \), then no \( b \) occurs before \( a_i \), when reading \( w \) from right-to-left. As such, we have that, when reading any word in \( \varphi_{a_ia_j}([w]_{\text{sylv}}) \) from right-to-left, the first occurrence of 1 corresponds to the first occurrence of \( a_i \), when reading \( w \) from right-to-left, and all occurrences of 2 before the first occurrence of 1 correspond to all occurrences of \( a_j \) before the first occurrence of \( a_i \). Thus, \( \varphi_{a_ia_j}([w]_{\text{sylv}}) \) has a 2-1 right precedence if and only if \( w \) has a \( a_j-a_i \) right precedence, and the indexes must coincide. Hence, we get the following lemma:

Lemma 3.6 Let \( u, v \in \mathcal{A}_n^* \). Then, \( u \equiv_{\text{sylv}} v \) if and only if \( \varphi_{ij}([u]_{\text{sylv}}) = \varphi_{ij}([v]_{\text{sylv}}) \), for all \( 1 \leq i < j \leq n \).

Proof The proof of the forward implication is trivial, since \( \varphi_{ij} \) is well-defined as a map, for all \( 1 \leq i < j \leq n \). The proof of the converse follows from the previous observations, as well as Proposition 2.7. \( \square \)

For each \( n \in \mathbb{N} \), with \( n \geq 3 \), let \( I_n \) be the index set

\[
\{(i, j) : 1 \leq i < j \leq n\},
\]

and let \( I = \bigcup_{n \in \mathbb{N}} I_n \). Now, consider the map

\[
\phi_n : \text{sylv}_n \longrightarrow \prod_{I_n} \text{sylv}_2,
\]

whose \( (i, j) \)-th component is given by \( \varphi_{ij}([w]_{\text{sylv}}) \), for \( w \in \mathcal{A}_n^* \) and \( (i, j) \in I_n \).

Proposition 3.7 The map \( \phi_n \) is an embedding.
**Proof** It is clear that $\phi_n$ is a homomorphism. It follows from the definition of $\phi_n$ and Lemma 3.6 that, for any $u, v \in \mathcal{A}^*_n$, we have $u \equiv_{\text{sylv}_n} v$ if and only if $\phi_n([u]_{\text{sylv}}) = \phi_n([v]_{\text{sylv}})$, hence $\phi_n$ is an embedding. □

Thus, for each $n \in \mathbb{N}$, we can embed $\text{sylv}_n$ into a direct product of copies of $\text{sylv}_2$. Similarly, we can embed $\text{sylv}$ into a direct product of infinitely many copies of $\text{sylv}_2$. Consider the map 

$$\phi : \text{sylv} \longrightarrow \prod_{I} \text{sylv}_2,$$

whose $(i, j)$-th component is given by $\phi_{ij}([w]_{\text{sylv}})$, for $w \in \mathcal{A}^*$ and $(i, j) \in I$.

**Proposition 3.8** The map $\phi$ is an embedding.

**Proof** It is clear that $\phi$ is a homomorphism. Notice that, for any word $w \in \mathcal{A}^*$, there must exist $n \in \mathbb{N}$ such that $w \in \mathcal{A}^*_n$. Furthermore, for $(i, j) \in I_n$, we have that the $(i, j)$-th component of $\phi([w]_{\text{sylv}})$ is equal to the $(i, j)$-th component of $\phi_n([w]_{\text{sylv}})$. Thus, for $u, v \in \mathcal{A}^*$, if $\phi([u]_{\text{sylv}}) = \phi([v]_{\text{sylv}})$, then $\phi_n([u]_{\text{sylv}}) = \phi_n([v]_{\text{sylv}})$, for some $n \in \mathbb{N}$ such that $u, v \in \mathcal{A}^*_n$.

It follows from Lemma 3.6 that, for any $u, v \in \mathcal{A}^*$, we have $u \equiv_{\text{sylv}} v$ if and only if $\phi([u]_{\text{sylv}}) = \phi([v]_{\text{sylv}})$, hence $\phi$ is an embedding. □

As such, all sylvester monoids of rank higher than 2 are in the variety generated by $\text{sylv}_2$. Since $\text{sylv}_2$ is a submonoid of $\text{sylv}$ and $\text{sylv}_n$, for any $n \geq 3$, they all generate the same variety $\mathcal{V}_{\text{sylv}}$. Thus, by Birkhoff’s Theorem, we have the following result:

**Theorem 3.9** For any $n \geq 2$, $\text{sylv}$ and $\text{sylv}_n$ satisfy exactly the same identities.

Another consequence of $\mathcal{V}_{\text{sylv}}$ being generated by $\text{sylv}_2$ is the following:

**Proposition 3.10** The basis rank of $\mathcal{V}_{\text{sylv}}$ is 2.

**Proof** Since $\mathcal{V}_{\text{sylv}}$ is generated by $\text{sylv}_2$, and $\text{sylv}_2$ is defined by a presentation where the alphabet has two generators, then $r_b(\mathcal{V}_{\text{sylv}})$ is less than or equal to 2.

On the other hand, notice that any monoid generated by a single element is commutative. Since $\text{sylv}$ is not commutative, $\mathcal{V}_{\text{sylv}}$ cannot be generated by any single monoid which is itself generated by a single element. As such, $r_b(\mathcal{V}_{\text{sylv}})$ is strictly greater than 1.

Hence, the basis rank of $\mathcal{V}_{\text{sylv}}$ is 2. □

By parallel reasoning, we can also prove that the $\#$-sylvester monoids of rank greater than or equal to 2 embed into a direct product of copies of $\text{sylv}_2#$. For any $i, j \in \mathcal{A}$, with $i < j$, define the homomorphism $\phi_{ij}^# : \mathcal{A}^* \longrightarrow \text{sylv}_2#$ in an identical fashion to $\phi_{ij}$, mapping a word to a $\#$-sylvester class instead of a sylvester class.

As with the case of $\phi_{ij}$ (see Lemma 3.5), we have that $\phi_{ij}^#$ factors to give a homomorphism $\phi_{ij}^# : \text{sylv}^# \longrightarrow \text{sylv}_2#$. Furthermore, we can also deduce the number of occurrences of each symbol in a word $w \in \mathcal{A}^*$, for some $n \geq 3$, and its left precedences, by looking at the images of $[w]_{\text{sylv}^#}$ under $\phi_{ij}^#$, ranging $1 \leq i < j \leq n$. Thus,
for \( u, v \in \mathcal{A}_n^* \), we have that \( u \equiv_{\text{sylv}_n^#} v \) if and only if \( \varphi_{ij}^#([u]_{\text{sylv}_n^#}) = \varphi_{ij}^#([v]_{\text{sylv}_n^#}) \), for all \( 1 \leq i < j \leq n \).

For each \( n \in \mathbb{N} \), we can embed \( \text{sylv}_n^# \) into a direct product of copies of \( \text{sylv}_2^# \), using the embedding

\[
\phi_n^# : \text{sylv}_n^# \longrightarrow \prod_{I_n} \text{sylv}_2^#,
\]

whose \((i, j)\)-th component is given by \( \varphi_{ij}^#([w]_{\text{sylv}_n^#}) \), for \( w \in \mathcal{A}_n^* \) and \((i, j) \in I_n\). Similarly, we can embed \( \text{sylv}^# \) into a direct product of infinitely many copies of \( \text{sylv}_2^# \), using the embedding

\[
\phi^# : \text{sylv}^# \longrightarrow \prod_{I} \text{sylv}_2^#,
\]

whose \((i, j)\)-th component is given by \( \varphi_{ij}^#([w]_{\text{sylv}^#}) \), for \( w \in \mathcal{A}_n^* \) and \((i, j) \in I\).

As such, all \( \#\)-sylvester monoids of rank higher than 2 are in the variety generated by \( \text{sylv}_2^# \). Since \( \text{sylv}_2^# \) is a submonoid of \( \text{sylv}^# \) and \( \text{sylv}_n^# \), for any \( n \geq 3 \), they all generate the same variety \( V_{\text{sylv}^#} \). Thus, by Birkhoff’s Theorem, we have the following result:

**Theorem 3.11** For any \( n \geq 2 \), \( \text{sylv}^# \) and \( \text{sylv}_n^# \) satisfy exactly the same identities.

The basis rank of \( V_{\text{sylv}^#} \) is also 2.

Notice that the embeddings of the \( \#\)-sylvester monoids of finite rank greater than or equal to 2 into a direct product of copies of \( \text{sylv}_2^# \) can also be obtained using the anti-isomorphisms between sylvester and \( \#\)-sylvester monoids of finite rank, and the previously obtained embeddings. However, we cannot use this argument for the infinite rank case, due to Proposition 2.8. On the other hand, since anti-isomorphisms exist in the finite case, we can conclude that, due to Theorems 3.9 and 3.11, any monoid anti-isomorphic to \( \text{sylv} \) (respectively, \( \text{sylv}^# \)) is in the variety generated by \( \text{sylv}^# \) (respectively, \( \text{sylv} \)).

Once again, we can also prove that the Baxter monoids of rank greater than or equal to 2 embed into a direct product of copies of \( \text{baxt}_2 \). For any \( i, j \in \mathcal{A} \), with \( i < j \), define the homomorphism \( \vartheta_{ij} : \mathcal{A} \longrightarrow \text{baxt}_2 \) in an identical fashion to \( \varphi_{ij} \), mapping a word to a Baxter class instead of a sylvester class.

As with the case of \( \varphi_{ij} \), we have that \( \vartheta_{ij} \) factors to give a homomorphism \( \vartheta_{ij} : \text{baxt} \longrightarrow \text{baxt}_2 \). Furthermore, we can also deduce the number of occurrences of each symbol in a word \( w \in \mathcal{A}_n^* \), for some \( n \geq 3 \), and its left and right precedences, by looking at the images of \([w]_{\text{baxt}} \) under \( \vartheta_{ij} \), ranging \( 1 \leq i < j \leq n \). Thus, for \( u, v \in \mathcal{A}_n^* \), we have that \( u \equiv_{\text{baxt}_n} v \) if and only if \( \vartheta_{ij}([u]_{\text{baxt}}) = \vartheta_{ij}([v]_{\text{baxt}}) \), for all \( 1 \leq i < j \leq n \).

For each \( n \in \mathbb{N} \), we can embed \( \text{baxt}_n \) into a direct product of copies of \( \text{baxt}_2 \), using the embedding

\[
\theta_n : \text{baxt}_n \longrightarrow \prod_{I_n} \text{baxt}_2,
\]
whose \((i, j)\)-th component is given by \(\vartheta_{ij}([w]_{\text{baxt}})\), for \(w \in \mathcal{A}_n^*\) and \((i, j) \in I_n\). Similarly, we can embed \(\text{baxt}\) into a direct product of infinitely many copies of \(\text{baxt}_2\), using the embedding

\[
\theta : \text{baxt} \rightarrow \prod_I \text{baxt}_2,
\]

whose \((i, j)\)-th component is given by \(\vartheta_{ij}([w]_{\text{baxt}})\), for \(w \in \mathcal{A}_n^*\) and \((i, j) \in I\).

As such, all Baxter monoids of rank higher than 2 are in the variety generated by \(\text{baxt}_2\). Since \(\text{baxt}_2\) is a submonoid of \(\text{baxt}\) and \(\text{baxt}_n\), for any \(n \geq 3\), they all generate the same variety \(V_{\text{baxt}}\). Thus, by Birkhoff’s Theorem, we have the following result:

**Theorem 3.12** For any \(n \geq 2\), \(\text{baxt}\) and \(\text{baxt}_n\) satisfy exactly the same identities.

The basis rank of \(V_{\text{baxt}}\) is also 2.

We also have that all Baxter monoids of rank greater than or equal to 2 embed into a direct product of copies of \(\mathcal{S}_2\) and \(\text{sylv}_2\). Since all Baxter monoids of rank greater than or equal to 2 embed into the direct product of the \#-sylvester and sylvester monoids of the same rank, the following diagrams commute:

\[
\begin{align*}
\text{baxt}_n & \hookrightarrow \mathcal{S}_n \times \text{sylv}_n & \text{baxt} & \hookrightarrow \mathcal{S} \times \text{sylv} \\
\prod_I \text{baxt}_2 & \hookrightarrow \left(\prod_I \mathcal{S}_2\right) \times \left(\prod_I \text{sylv}_2\right) & \prod_I \text{baxt}_2 & \hookrightarrow \left(\prod_I \mathcal{S}_2\right) \times \left(\prod_I \text{sylv}_2\right)
\end{align*}
\]

As such, \(V_{\text{baxt}}\) is generated by \(\mathcal{S}_2\) and \(\text{sylv}_2\). This implies, by Birkhoff’s Theorem, the following corollary:

**Corollary 3.13** The identities satisfied by the Baxter monoids of rank greater than or equal to 2 are exactly those identities which are simultaneously satisfied by the \#-sylvester and sylvester monoids of rank greater than or equal to 2.

### 4 Identities and bases

In this section, we obtain a complete characterization of the identities satisfied, respectively, by the sylvester, \#-sylvester and Baxter monoids, finite bases for \(V_{\text{sylv}}, V_{\mathcal{S}_2}\) and \(V_{\text{baxt}}\), and also their axiomatic rank.

#### 4.1 Characterization of the identities satisfied by the sylvester, \#-sylvester and Baxter monoids

The identities satisfied by the sylvester, \#-sylvester and Baxter monoids and those satisfied by their respective monoids of rank 2 are exactly the same. As such, we shall use the monoids of rank 2 to obtain a characterization of those identities.
For a word \( u \) over an alphabet of variables \( X \), and for variables \( x, y \in \text{supp}(u) \), we denote the number of occurrences of \( y \) before the first occurrence of \( x \) in \( u \), when reading \( u \) from right-to-left (respectively, from left-to-right), by \( o_{x \leftarrow y}(u) \) (respectively, \( o_{y \rightarrow x}(u) \)).

**Theorem 4.1** The identities \( u \approx v \) satisfied by sylv are exactly the balanced identities such that, for any variables \( x, y \in \text{supp}(u \approx v) \), \( o_{x \leftarrow y}(u) = o_{x \leftarrow y}(v) \).

**Proof** We first prove by contradiction that an identity satisfied by sylv \(_2\) must satisfy the stated conditions. Suppose \( u \approx v \) is an identity satisfied by sylv \(_2\). Since sylv \(_2\) contains the free monogenic submonoid, we know that any identity satisfied by sylv \(_2\) must be a balanced identity. Thus, we assume \( u \approx v \) is a balanced identity.

Suppose, in order to obtain a contradiction, that there exist variables \( x, y \in \text{supp}(u \approx v) \), such that \( o_{x \leftarrow y}(u) \neq o_{x \leftarrow y}(v) \). Then, if we consider the words \( u|_{x,y} \) and \( v|_{x,y} \), obtained from \( u \) and \( v \), respectively, by eliminating every occurrence of a variable other than \( x \) or \( y \), we have that \( u|_{x,y} \) admits the suffix \( xy^{o_{x \leftarrow y}(u)} \) and \( v|_{x,y} \) admits the suffix \( xy^{o_{x \leftarrow y}(v)} \).

Taking the evaluation \( \psi \) of \( X \) in sylv \(_2\) such that \( \psi(x) = [1]_{\text{sylv}_2}, \psi(y) = [2]_{\text{sylv}_2} \) and \( \psi(z) = [\varepsilon]_{\text{sylv}_2} \), for all other variables \( z \in X \), we have

\[
\psi(u) = \psi(u|_{x,y}) = [u']_{\text{sylv}_2} \cdot [12^{o_{x \leftarrow y}(u)}]_{\text{sylv}_2} \quad \text{and} \\
\psi(v) = \psi(v|_{x,y}) = [v']_{\text{sylv}_2} \cdot [12^{o_{x \leftarrow y}(v)}]_{\text{sylv}_2},
\]

for some words \( u', v' \in \mathcal{A}_2^* \). Since \( o_{x \leftarrow y}(u) \neq o_{x \leftarrow y}(v) \), we have that \( \psi(u) \) and \( \psi(v) \) cannot share a 2-1 right precedence of the same index. Thus, by Proposition 2.7, we have that \( \psi(u) \neq \psi(v) \), which contradicts our hypothesis that \( u \approx v \) is an identity.

We now prove by contradiction that an identity which satisfies the previously mentioned conditions must also be satisfied by sylv \(_2\). Suppose that \( u \approx v \) is a balanced identity, such that \( o_{x \leftarrow y}(u) = o_{x \leftarrow y}(v) \), for any variables \( x, y \in \text{supp}(u \approx v) \). Suppose, in order to obtain a contradiction, that there is some evaluation \( \psi \) of \( X \) in sylv \(_2\) such that \( \psi(u) \neq \psi(v) \).

Notice that, since \( u \approx v \) is a balanced identity, then \( \psi(u) \) and \( \psi(v) \) have the same content. As such, we have that \( \text{supp}(\psi(u)) = \text{supp}(\psi(v)) = \{1, 2\} \), and, by Proposition 2.7, either \( \psi(u) \) and \( \psi(v) \) have 2-1 right precedences of different indexes, or one of them has a 2-1 right precedence and the other does not. Assume, without loss of generality, that words in \( \psi(u) \) admit a suffix of the form \( 12^a \), and words in \( \psi(v) \) admit a suffix of the form \( 12^b \), for some \( a, b \in \mathbb{N}_0 \) such that \( a > b \). Notice that this assumption covers both the case where \( \psi(v) \) has a right precedence and the case where it does not. Furthermore, the assumption implies that \( \psi(u) \) has a 2-1 right precedence of index \( a \).

Observe that \( u \) must be of the form \( u = u_1zu_2 \), with \( z \in X \) and \( u_2 \in X^* \), such that \( \psi(u_2) \) has support \( \{2\} \) and \( \psi(z) \) has either support \( \{1, 2\} \) or support \( \{1\} \). As such, \( \psi(zu_2) \) has a 2-1 right precedence of index \( a \), the same as \( \psi(u) \). Furthermore, notice that \( z \) cannot occur in \( u_2 \). Thus, by our hypothesis, we have that \( o_{x \leftarrow z}(u) = o_{x \leftarrow z}(v) \), for any variable \( x \in \text{supp}(u_2) \). Therefore, \( v \) must also be of the form \( v = v_1zv_2 \), where \( v_2 \) has the same content as \( u_2 \). But this implies that \( \psi(v_2) \) has support \( \{2\} \),...
hence $\psi(zv_2)$ also has a 2-1 right precedence of index $a$. Since $\psi(zv_2)$ must also have the same right precedence as $\psi(v)$, we have reached a contradiction.

Thus, there is no evaluation $\psi$ of $X$ in sylv$_2$ such that $\psi(u) \neq \psi(v)$. Therefore, $u \approx v$ is an identity satisfied by sylv$_2$.

Since all identities satisfied by sylv must also be satisfied by sylv$_2$, we obtain the stated result.

By parallel reasoning, we can obtain the characterization of the identities satisfied by the #-sylvester monoid:

**Theorem 4.2** The identities $u \approx v$ satisfied by sylv$#$ are exactly the balanced identities such that, for any variables $x, y \in \text{supp}(u \approx v)$, $o_{x\rightarrow y}(u) = o_{x\rightarrow y}(v)$.

From the previous two theorems, we can easily obtain the characterization of the identities satisfied by the Baxter monoid:

**Theorem 4.3** The identities $u \approx v$ satisfied by baxt are exactly the balanced identities such that, for any variables $x, y \in \text{supp}(u \approx v)$, $o_{x\rightarrow y}(u) = o_{x\rightarrow y}(v)$ and $o_{x\leftarrow y}(u) = o_{x\leftarrow y}(v)$.

**Proof** Let $u \approx v$ be an identity and let $\psi$ be an evaluation of $X$ in baxt. Let $u', v' \in A^*$ be words such that $\psi(u) = [u']_{\text{baxt}}$ and $\psi(v) = [v']_{\text{baxt}}$.

Notice that, by composing $\psi$ with the natural homomorphisms of baxt into sylv and sylv$#$, we obtain evaluations $\psi_1$ and $\psi_2$ of $X$ into sylv and sylv$#$, respectively. Furthermore, notice that $\psi_1(w) = [w]_{\text{sylv}}$ and $\psi_2(w) = [w]_{\text{sylv}#}$, for $w \in \{u, v\}$.

Hence, by [20, Proposition 3.7], we have that $\psi(u) = \psi(v)$ if and only if $\psi_1(u) = \psi_1(v)$ and $\psi_2(u) = \psi_2(v)$. As such, if $u \approx v$ is satisfied by both sylv and sylv$#$, then it must also be satisfied by baxt.

On the other hand, any identity satisfied by baxt must be satisfied by both sylv and sylv$#$, since both these monoids are in the variety generated by baxt. Thus, the result follows as a consequence of Theorems 4.1 and 4.2.

are equivalent if one can be obtained from the other by renaming variables or swapping both sides of the identities. With these characterizations, we recover the following corollaries:

**Corollary 4.4** ([9, Proposition 2.0]) The sylvester monoid satisfies the non-trivial identity $xyxyxy \approx yxyxy$. Furthermore, up to equivalence, this is the shortest non-trivial identity satisfied by sylv.

**Corollary 4.5** ([9, Proposition 2.4]) The #-sylvester monoid satisfies the non-trivial identity $yxyxyx \approx yxyxy$. Furthermore, up to equivalence, this is the shortest non-trivial identity satisfied by sylv$#$.

**Corollary 4.6** ([9, Proposition 2.6]) The Baxter monoid satisfies the non-trivial identities $xxxyyxy \approx xxxyxy$ and $xxxxyy \approx xyxyxy$. Furthermore, up to equivalence, these are the shortest non-trivial identities satisfied by baxt.
The following corollaries are useful alternative characterizations of the identities satisfied by sylv, sylv# and baxt. They imply that, when reading both sides of an identity satisfied by the sylvester, #-sylvester or Baxter monoid, the first occurrence of a variable is read at the same time in both words:

**Corollary 4.7** The identities $u \approx v$ satisfied by sylv (respectively, sylv#) are balanced identities such that, for any $x \in \text{supp}(u \approx v)$, the longest suffix (respectively, prefix) of $u$ where $x$ does not occur has the same content as the longest suffix (respectively, prefix) of $v$ where $x$ does not occur.

**Proof** We give the proof for the sylv case. The reasoning for the sylv# case is parallel.

Let $u = u_1 xu_2$ and $v = v_1 xv_2$, where $u_2$ and $v_2$ are words where $x$ does not occur. Notice that, due to Theorem 4.1, for any variable $y$ which occurs in $u_2$ or $v_2$, we have that $o_x \leftarrow y(u) = o_x \leftarrow y(v)$. The result follows immediately. \(\square\)

**Corollary 4.8** The identities $u \approx v$ satisfied by baxt are balanced identities such that, for any $x \in \text{supp}(u \approx v)$, the longest prefix of $u$ where $x$ does not occur has the same content as the longest prefix of $v$ where $x$ does not occur, and the longest suffix of $u$ where $x$ does not occur has the same content as the longest suffix of $v$ where $x$ does not occur.

**Proof** The result follows from the previous corollary. \(\square\)

These alternate characterizations allow us to obtain algorithms which check if identities are satisfied by the sylvester, #-sylvester and Baxter monoids in polynomial time. For brevity’s sake, we only show the algorithm for the sylvester case:

**Proposition 4.9** Algorithm 3 is sound and complete, and has time complexity $\mathcal{O}(k^2 \log(k))$, where $k$ is the length of the word $u$, for input $u \approx v$.

**Proof** Algorithm 3 first checks if $u$ and $v$ have the same length, in line 1. If they do not, then $u \approx v$ is not a balanced identity, and as such, is not satisfied by sylv. This is done in $2k + 1$ time, in the worst-case scenario where the length of $v$ is greater than the length of $u$.

The algorithm scans $u$ and $v$, from right-to-left, in the for loop in line 5. The arrays $C$ and $D$ stand for, respectively, the content vectors of the suffixes of $u$ and $v$ read so far, while the word $\leftarrow s$ stands for the support of these suffixes. Notice that, since $u$ is of length $k$, then at most $k$ variables occur in $u$. Hence, $C$ and $D$ have length $k$.

In each iteration of the loop, the algorithm checks if the letter which is being read in $u$ is the same as the one being read in $v$. If they are the same, and they do not occur in $\leftarrow s$, this means that this is the first occurrence of a variable $x$. The algorithm checks if the arrays $C$ and $D$ are equal. If they are not, this implies that the longest suffix of $u$ where $x$ does not occur does not have the same content as the longest suffix of $v$ where $x$ does not occur. Hence, by Corollary 4.7, $u \approx v$ is not satisfied by sylv. If $C$ and $D$ are equal, then the algorithm registers the new variable in $\leftarrow s$ and updates the content vectors $C$ and $D$. On the other hand, if $x$ occurs already in $\leftarrow s$, the algorithm simply updates $C$ and $D$.

If the letters which are being read in $u$ and $v$ are different, then the algorithm checks if they both occur in $\leftarrow s$. If that does not happen, then that means at least one of them...
Algorithm 3: Identity checking algorithm for the sylvester monoid.

**Input:** An identity $u \approx v$.

**Output:** True if sylv satisfies $u \approx v$, False otherwise.

1. if $|u| \neq |v|$ then return False; $k \leftarrow |u|$;
2. $C[1, \ldots, k], D[1, \ldots, k] \leftarrow [0, \ldots, 0]$;
3. $\overrightarrow{s} \leftarrow \emptyset$;
4. for $0 \leq i \leq k - 1$ do
   5. if $u_{k-i} = v_{k-i}$ then
      6. if $u_{k-i} \notin \text{supp}(\overrightarrow{s})$ then
         7. if $C \neq D$ then
            8. return False;
         9. else
            10. append $u_{k-i}$ to $\overrightarrow{s}$;
            11. $j \leftarrow |\overrightarrow{s}|$;
            12. $C[j] \leftarrow C[j] + 1; D[j] \leftarrow D[j] + 1$;
      13. else
         14. $j \leftarrow $ index of $u_{k-i}$ in $\overrightarrow{s}$;
         15. $C[j] \leftarrow C[j] + 1; D[j] \leftarrow D[j] + 1$;
   16. else
      17. if $u_{k-i}, v_{k-i} \in \text{supp}(\overrightarrow{s})$ then
         18. $j \leftarrow $ index of $u_{k-i}$ in $\overrightarrow{s}$;
         19. $l \leftarrow $ index of $v_{k-i}$ in $\overrightarrow{s}$;
         20. $C[j] \leftarrow C[j] + 1; D[l] \leftarrow D[l] + 1$;
      21. else
         22. return False;
23. if $C \neq D$ then return False; return True

is the first occurrence of a variable in one of the words, but not in the other. Hence, by Corollary 4.7, $u \approx v$ is not satisfied by sylv. Otherwise, if they both occur in $\overrightarrow{s}$, the algorithm simply updates $C$ and $D$.

It is clear that the algorithm is sound and complete, since it always detects when a new variable is read, if it is read at the same time in both $u$ and $v$, and if the content of the suffixes is the same.

Taking into consideration that operations of addition and comparing numbers are logarithmic time in a Turing machine model, and that accessing coordinates of vectors is a linear-time operation, we have that comparing the content vectors $C$ and $D$ takes at most $O(k \log(k))$ time, and updating them takes $O(k \log(k))$ time as well. On the other hand, checking if a variable occurs in $\overrightarrow{s}$ takes $O(k)$ time. As such, each iteration of the for loop has time complexity $O(k \log(k))$. Since there are $k$ iterations of the loop, and no other part of the algorithm takes as much time as the loop, we can conclude that Algorithm 3 has time complexity $O(k^2 \log(k))$. □

Corollary 4.10 The decision problem CHECK-ID(sylv) belongs to the complexity class P.
We can also construct an algorithm that checks if identities hold in sylv#, with time complexity $O(k^2 \log(k))$. From that algorithm and Algorithm 3, we can construct another algorithm for the Baxter case. As such, we also have the following corollary:

**Corollary 4.11** The decision problems Check-Id(sylv#) and Check-Id(baxt) belong to the complexity class $P$.

We also easily obtain some important non-trivial identities satisfied by these monoids:

**Example 4.12** Consider the following non-trivial identities:

\[ xyzty \approx yztxy; \]  
\[ xzytx \approx xzytx; \]  
\[ xzyty \approx xzyty. \]

The sylvester monoid satisfies (L), but satisfies neither (M) nor (R), while the #-sylvester monoid satisfies (R), but satisfies neither (L) nor (M).

The Baxter monoid satisfies the following non-trivial identities:

\[ xzytyrxy \approx xzytyxry; \]  
\[ xzytyrxy \approx xzytyxry. \]

These identities form bases for the varieties generated by the sylvester monoid, as stated in Theorem 4.16, the #sylvester monoid, as stated in Theorem 4.17, and the Baxter monoid, as stated in Theorem 4.18. Furthermore, we can conclude that the varietal meet $V_{sylv} \cap V_{sylv}$ strictly contains the variety generated by the hypoplactic monoid, since the variety generated by the hypoplactic monoid is defined by all three identities (L), (M) and (R) (see [13, Example 4.3]), but monoids in $V_{sylv} \cap V_{sylv}$ do not necessarily need to satisfy the identity (M). This last observation was suggested by one of the anonymous reviewers.

The following corollaries will be important in the next subsection:

**Corollary 4.13** The shortest non-trivial identities, with $n$ variables, satisfied by sylv or by sylv#, are of length $n + 2$.

**Proof** Since any identity satisfied by sylv must also be satisfied by hypo, and since we already know that the shortest non-trivial identity, with $n$ variables, satisfied by hypo, is of length $n + 2$ (see [13, Corollary 4.6]), then a non-trivial identity, with $n$ variables, satisfied by sylv, must be of length at least $n + 2$.

On the other hand, by Theorem 4.1, it is immediate that

\[ xya_1 \ldots a_{n-2} yx \approx yxa_1 \ldots a_{n-2} yx \]  

\( Springer \)
is an identity satisfied by sylv, for variables \( x, y, a_1, \ldots, a_{n-2} \).

The reasoning for identities satisfied by sylv\(^\#\) is parallel to the one given previously.

**Corollary 4.14** The shortest non-trivial identity, with \( n \) variables, satisfied by \( \text{baxt} \), is of length \( n + 4 \).

**Proof** It is immediate, by Theorem 4.3, that for variables \( x, y, a_1 \ldots a_{n-2} \),

\[ xy xy a_1 \ldots a_{n-2}yx \approx xy yx a_1 \ldots a_{n-2}yx \]

is an identity satisfied by \( \text{baxt} \).

On the other hand, let \( u \approx v \) be a non-trivial identity, with \( n \) variables, satisfied by \( \text{baxt} \), such that \( u = wxu' \) and \( v = wyv' \), for some words \( w, u', v' \) over the alphabet of variables \( X \).

Observe that \( y \) must occur in \( w \), otherwise, we would have \( o_{x \rightarrow y}(u) > o_{x \rightarrow y}(v) \). Furthermore, it must also occur in \( u' \), since \( u \approx v \) is a balanced identity, and so \( \text{cont}(xu') = \text{cont}(yv') \). Similarly, \( x \) must occur in both \( w \) and \( v' \). On the other hand, by Corollary 4.8, \( x \) must occur in \( u' \) and \( y \) must occur in \( v' \), since \( |u'| = |v'| \). Therefore, \( x \) and \( y \) both occur at least three times each in \( u \) and \( v \). Since \( u \approx v \) is an identity with \( n \) variables, it must be of length at least \( n + 4 \).

Finally, we can also clarify the relation between the Baxter monoids and the plactic monoids:

**Corollary 4.15** The variety generated by \( \text{baxt} \) is strictly contained in the variety generated by \( \text{plac}_2 \).

**Proof** Let \( u \approx v \) be an identity satisfied by \( \text{plac}_2 \). Thus, it must be a balanced identity. Let \( x, y \in \text{supp}(u \approx v) \). Suppose, in order to obtain a contradiction, that \( o_{x \rightarrow y}(u) > o_{x \rightarrow y}(v) \). Let

\[ u = u_1 y u_2 \quad \text{and} \quad v = v_1 y v_2, \]

where \( u_1 \) (respectively, \( v_1 \)) is the longest prefix of \( u \) (respectively, \( v \)) where \( y \) does not occur. Since the equational theory of the variety generated by \( \text{plac}_2 \) is left 1-hereditary, then \( u_1 \approx v_1 \) must be satisfied by \( \text{plac}_2 \). Hence, it must be a balanced identity. But \( |u_1|_x = o_{x \rightarrow y}(u) > o_{x \rightarrow y}(v) = |v_1|_x \). We have reached a contradiction, hence, \( o_{x \rightarrow y}(u) \neq o_{x \rightarrow y}(v) \).

By this reasoning, we prove that \( o_{x \rightarrow y}(u) = o_{x \rightarrow y}(v) \) and \( o_{x \leftarrow y}(u) = o_{x \leftarrow y}(v) \). Hence, by Theorem 4.3, \( u \approx v \) must be satisfied by \( \text{baxt} \).

On the other hand, it is well-known that the shortest non-trivial identity satisfied by the bicyclic monoid is Adian’s identity \( xyyxyxyyx \approx xyxyxyyx \) (see [1]). As such, \( \text{plac}_2 \) does not satisfy any non-trivial identity of length less than 10. But \( \text{baxt} \) satisfies an identity of length 6, as seen in Corollary 4.6. Thus, not all identities satisfied by \( \text{baxt} \) are satisfied by \( \text{plac}_2 \).

Therefore, as a consequence of Birkhoff’s Theorem, the variety generated by \( \text{baxt} \) is strictly contained in the variety generated by \( \text{plac}_2 \).
4.2 The axiomatic rank of the varieties generated by the sylvester, #-sylvester and Baxter monoids

We now prove that that the varieties generated by the sylvester, #-sylvester and Baxter monoids are finitely based, and therefore have finite axiomatic rank. We give bases for $V_{\text{sylv}}$ and $V_{\text{sylv}#}$ with one identity each, of length 6, over a four-letter alphabet. Trivially, these bases are minimal with regards to the number of identities in the basis; they are also minimal with regards to the number of variables occurring in these identities, and the length of these identities. We also give a basis for $V_{\text{baxt}}$, with two identities, of length 10, over a six-letter alphabet. This basis is also minimal with regards to the number of identities in the basis, the number of variables occurring in these identities, and the length of these identities.

Furthermore, we also prove that there exist no bases for $V_{\text{sylv}}$ or $V_{\text{sylv}#}$ with only identities over an alphabet with at most three variables, thus showing that the axiomatic rank of $V_{\text{sylv}}$ and $V_{\text{sylv}#}$ is 4. We also prove that there exists no basis for $V_{\text{baxt}}$ with only identities over an alphabet with at most five variables, thus showing that the axiomatic rank of $V_{\text{baxt}}$ is 6.

**Theorem 4.16** $V_{\text{sylv}}$ admits a finite basis $B_{\text{sylv}}$, consisting of the following identity:

$$xyztxy \approx yztxxy.$$ (L)

**Proof** Let $B_{\text{sylv}}$ be the set comprising the identity (L). Notice that this identity is given in Example 4.12.

The following proof will be by induction, in the following sense: We order identities by the length of the common suffix of both sides of the identity. The induction will be on the length of the prefix, that is, the length of the identity minus the length of the common suffix.

The base cases for the induction, for identities of length $n$ (with $n \geq 4$), are those identities of the form

$$xyw \approx yxw,$$

where $w$ is a word of length $n-2$ and $x$, $y$ are variables. Observe that, since any identity $u \approx v$ satisfied by sylv is a balanced identity, there are no non-trivial identities, of length $n$, with a common prefix of length greater than $n-2$, satisfied by sylv. Furthermore, $x$ and $y$ must both occur in $w$, otherwise, we would have $o_{x \leftarrow y}(xyw) > o_{x \leftarrow y}(yxw)$. Thus, $w$ is of the form

$$w_1xw_2yw_3 \text{ or } w_1yw_2xw_3,$$

for some words $w_1$, $w_2$, $w_3$. Therefore, by replacing $z$ with $w_1$, and $t$ by $w_2$, and, if necessary, renaming $x$ and $y$, we can immediately deduce this identity from the identity (L). Notice that, when $n = 4$, the base cases correspond to the identities given in Corollary 4.4.

$\textcircled{Springer}$
The idea of the proof of the induction step is that, for any identity \( u \approx v \), we can apply the identity (L), finitely many times, to deduce a new identity \( u \approx u^* \) from \( u \approx v \), such that \( u^* \) is “closer” to \( v \) than \( u \), in the sense that \( u^* \) and \( v \) have a common suffix which is strictly longer than the common suffix of \( u \) and \( v \). Notice that \( u^* \approx v \) is a consequence of \( \mathcal{B}_{\text{sy1v}} \), by the induction hypothesis. As such, we can conclude that \( u \approx v \) is a consequence of \( \mathcal{B}_{\text{sy1v}} \).

The technical part of the proof allows us to show that there is always a way to shuffle some variables of \( u \) in such a way that we obtain \( u^* \). We show that these variables must occur several times in \( u \), thus allowing us to apply the identity (L) to shuffle \( u \) and obtain \( u^* \).

By the induction hypothesis, we know that \( u^* \approx v \) is a consequence of \( \mathcal{B}_{\text{sy1v}} \). Thus, we conclude that \( u \approx v \) is also a consequence of \( \mathcal{B}_{\text{sy1v}} \).

Let \( u \approx v \) be a non-trivial identity satisfied by \( \text{sy1v} \) and let \( \mathcal{X} \) denote its support. Since \( u \approx v \) is a non-trivial identity, we must have \( u = u'xw \) and \( v = v'yw \), for some words \( w, u', v' \in \mathcal{X}^* \) and variables \( x, y \) such that \( x \neq y \). Notice that \( u' \) and \( v' \) cannot be the empty word, otherwise, we would have \( u = xw \) and \( v = yw \), which contradicts the fact that \( \text{cont}(u) = \text{cont}(v) \).

On the other hand, since \( \text{cont}(u'x) = \text{cont}(v'y) \), we have that \( y \) occurs in \( u' \). Thus, to distinguish the rightmost \( y \) in \( u' \), we have that

\[ u'x = u_1yau_2, \]

for some variable \( a \) and words \( u_1 \) and \( u_2 \), such that \( y \) does not occur in \( u_2 \) and \( a \neq y \). Once again, since \( \text{cont}(u'x) = \text{cont}(v'y) \), we have that \( a \) occurs in \( v' \). Thus, to distinguish the rightmost \( a \) in \( v' \), we have that

\[ v' = v_1av_2, \]

for some words \( v_1 \) and \( v_2 \), such that \( a \) does not occur in \( v_2 \). To sum up, we have that

\[ u = u_1yau_2w \quad \text{and} \quad v = v_1av_2yw, \]

where \( y \) does not occur in \( u_2 \) and \( a \) does not occur in \( v_2 \).

Notice that, by Theorem 4.1, \( y \) must occur in \( w \), otherwise, we would have \( o_{y \leftrightarrow x}(u) > o_{y \leftrightarrow x}(v) \). Thus, \( a \) must also occur in \( w \), otherwise, we would have \( o_{a \leftrightarrow y}(u) < o_{a \leftrightarrow y}(v) \). As such, we can deduce the word \( u_1ayu_2w \), by applying the identity (L), renaming \( x \) to \( a \) and replacing \( z \) and \( t \) by the appropriate words.

Observe that we can repeatedly apply this reasoning until we obtain a word of the form

\[ u^* = u''yw, \]

for some word \( u'' \), since the only restriction imposed on the variable \( a \) was that \( a \neq y \). Thus, we have proven that, for any non-trivial identity \( u \approx v \) satisfied by \( \text{sy1v} \), we can obtain a new word \( u^* \) from \( u \) such that the common suffix of \( u^* \) and \( v \) is strictly longer than the common suffix of \( u \) and \( v \), by applying the identity (L) finitely many times.
times. By induction, we conclude that \( u \approx v \) is a consequence of \( \mathcal{B}_{\text{sylv}} \), thus proving that \( \mathcal{B}_{\text{sylv}} \) is a basis for \( V_{\text{sylv}} \).

By the same reasoning, we can also prove the following result:

**Theorem 4.17** \( V_{\text{sylv}}^\# \) admits a finite basis \( \mathcal{B}_{\text{sylv}}^\# \), consisting of the following identity:

\[
xyztyx \approx xyztxy.
\]

**(R)**

**Proof** The proof follows the same reasoning as the proof of Theorem 4.16, the main difference being that, within a set of identities of the same length, they are ordered on the length of the common prefix of both sides of the identity. Therefore, the induction is on the length of the suffix after the common prefix of both sides of the identity. The induction step resorts to Theorem 4.2.

The following theorem arises as a consequence of Theorem 4.3 and [3, Theorem 5.7 and Lemma 5.23], as observed by one of the anonymous reviewers. We present here an alternative proof, using the same reasoning as that given in the proof of Theorem 4.16:

**Theorem 4.18** \( V_{\text{baxt}} \) admits a finite basis \( \mathcal{B}_{\text{baxt}} \), consisting of the following identities:

\[
xyztyx rxsy \approx xyztyx rxsy; \quad (O)
\]

\[
xyztyx rysx \approx xyztyx rysx. \quad (E)
\]

**Proof** The proof follows the same reasoning as the proof of Theorem 4.16. As such, we only give the reasoning for the base cases and the induction step.

The base cases for the induction on the length of the prefix before the common suffix, for identities of length \( n \) (with \( n \geq 6 \)), are those identities of the form

\[
xyzxyw \approx xyyxw,
\]

where \( w \) is a word of length \( n - 4 \) and \( x, y \) are variables. Notice that both sides of the identity must have a prefix of the form \( xy \), due to Corollary 4.8. By the same reason, observe that, since any identity \( u \approx v \) satisfied by baxt is a balanced identity, there are no non-trivial identities, of length \( n \), with a common suffix of length greater than \( n - 4 \), satisfied by baxt. Furthermore, \( x \) and \( y \) must both occur in \( w \), otherwise, we would have \( \sigma_{x \leftarrow y} (xyw) > \sigma_{x \leftarrow y} (yxw) \). Thus, \( w \) is of the form

\[
w_1 x w_2 y w_3 \quad \text{or} \quad w_1 y w_2 x w_3,
\]

for some words \( w_1, w_2, w_3 \). Therefore, by replacing \( z \) and \( t \) with the empty word, \( r \) with \( w_1 \), and \( s \) by \( w_2 \), and, if necessary, renaming \( x \) and \( y \), we can immediately deduce this identity from the identity \( (O) \) or the identity \( (E) \), depending on the form of \( w \). Notice that, when \( n = 6 \), the base cases correspond to the identities given in Corollary 4.6.

\( \square \)
Let \( u \approx v \) be a non-trivial identity satisfied by \( \mathcal{B} \) and let \( \mathcal{B}^* \) denote its support. Since \( u \approx v \) is a non-trivial identity, we must have \( u = u'xw \) and \( v = v'yw \), for some words \( w, u', v' \in \mathcal{B}^* \) and variables \( x, y \) such that \( x \neq y \). By Corollary 4.8, we have that \( x \) must occur at least once in \( u' \) and \( w \) and at least twice in \( v' \), and \( y \) must occur at least twice in \( u' \) and at least once in \( v' \) and \( w \). Thus, to distinguish the leftmost \( y \) in \( u' \), we have that

\[
u'x = u_1ya\nu_2
\]

for some variable \( a \) and words \( u_1 \) and \( u_2 \), such that \( y \) does not occur in \( u_2 \) and \( a \neq y \). Notice that \( y \) must occur in \( u_1 \). Since \( \text{cont}(u'x) = \text{cont}(v'y) \), we have that \( a \) occurs in \( v' \). Thus, to distinguish the rightmost \( a \) in \( v' \), we have that

\[
v' = v_1av_2,
\]

for some words \( v_1 \) and \( v_2 \), such that \( a \) does not occur in \( v_2 \). To sum up, we have that

\[
u = u_1ya\nu_2w \quad \text{and} \quad v = v_1av_2yw,
\]

where \( y \) occurs in \( u_1 \) but not in \( u_2 \) and \( a \) does not occur in \( v_2 \).

Suppose, in order to obtain a contradiction, that \( a \) does not occur in \( u_1 \). This implies that \( |u'|_y = o_{y\rightarrow a}(u) \). But \( |u'|_y = |v'|_y + 1 \), hence

\[
o_{y\rightarrow a}(v) \leq |v'|_y < |u'|_y = o_{y\rightarrow a}(u).
\]

Thus, by Theorem 4.3, we obtain a contradiction. As such, \( a \) must occur in \( u_1 \). By the same theorem, \( a \) must occur in \( w \) as well, otherwise, we would have \( o_{a\leftarrow y}(u) < o_{a\leftarrow y}(v) \). Therefore, \( y \) and \( a \) both occur at least once in \( u_1 \) and \( w \). As such, we can deduce the word \( u_1ya\nu_2w \), by applying the identity (O) or the identity (E) to \( u \), depending on where \( y \) and \( a \) occur in \( u_1 \) and \( w \), renaming \( x \) to \( a \) and replacing the remaining variables by the appropriate words. \( \square \)

An immediate consequence of having a finite basis is the following:

**Corollary 4.19** The varieties \( V_{\text{svylv}}, V_{\text{svylv}}^\# \) and \( V_{\text{baxt}} \) have finite axiomatic rank.

By [13, Proposition 4.10] and [13, Proposition 4.11], we know that the identities (L) and (R) are not consequences of the set of non-trivial identities, satisfied by hypo, over an alphabet with four variables, excluding themselves and equivalent identities. Since the identities satisfied by sylv and sylv\(^\#\) must also be satisfied by hypo, we can conclude the following:

**Corollary 4.20** The identity (L) is not a consequence of the set of non-trivial identities, satisfied by sylv, over an alphabet with four variables, excluding (L) itself and equivalent identities. Furthermore, any basis for \( V_{\text{svylv}} \) with only identities over an alphabet with four variables must contain the identity (L), or an equivalent identity.
Corollary 4.21 The identity \((R)\) is not a consequence of the set of non-trivial identities, satisfied by \(\text{sylv}^\#\), over an alphabet with four variables, excluding \((R)\) itself and equivalent identities. Furthermore, any basis for \(V_{\text{sylv}}^\#\) with only identities over an alphabet with four variables must contain the identity \((R)\), or an equivalent identity.

Hence, \(V_{\text{sylv}}\) and \(V_{\text{sylv}}^\#\) do not admit any bases with only identities over an alphabet with two or three variables. In other words, we have that:

Corollary 4.22 The axiomatic rank of \(V_{\text{sylv}}\) and \(V_{\text{sylv}}^\#\) is 4.

We now show that the identities \((O)\) and \((E)\) must be in any basis for \(V_{\text{baxt}}\) which contains only identities over an alphabet with six variables:

Proposition 4.23 Neither of the identities \((O)\) or \((E)\) is a consequence of the set of non-trivial identities, satisfied by \(\text{baxt}\), over an alphabet with six variables, excluding itself (but not the other) and equivalent identities.

Proof We prove the result for the identity \((O)\). Parallel reasoning shows the analogous result for \((E)\).

Let \(X = \{x, y, z, t, r, s\}\) and let \(\mathcal{I}\) be the set of all non-trivial identities, satisfied by \(\text{baxt}\), over an alphabet with six variables, excluding \((O)\) and equivalent identities. Suppose, in order to obtain a contradiction, that \((O)\) is a consequence of \(\mathcal{I}\). As such, there must exist a non-trivial identity \(u \approx v\) in \(\mathcal{I}\), and a substitution \(\sigma\), such that

\[
xzyt x y r x s y = w_1 \sigma(u) w_2,
\]

where \(w_1, w_2\) are words over \(X\), and \(\sigma(u) \neq \sigma(v)\). Notice that \(u \approx v\) must be balanced, and that there must be at least two variables occurring in \(u\) and \(v\), otherwise, \(u \approx v\) would be a trivial identity.

By the same reasoning as in the proof of [13, Proposition 4.10], we can assume, without loss of generality, that \(\sigma\) does not map any variable to the empty word. Due to this, and since only \(x\) and \(y\) occur three times in \(xzyt x y r x s y\), and all other variables each occur one time, we have that each variable occurring in \(u \approx v\) can occur at most three times, and only two variables can occur more than one time. Furthermore, by Corollary 4.14, which gives us a lower bound for the length of the identities, we have that \(u \approx v\) is of length at least 6. Notice that it is exactly of length 6 if only two variables occur in it. Thus, up to renaming of variables, \(x\) and \(y\) occur exactly three times in \(u \approx v\), and \(t, z, r\) and \(s\) can occur at most one time.

Suppose now, in order to obtain a contradiction, that \(w_1 \neq \varepsilon\). Then, since \(u \approx v\) is of length at least 6, we must have \(w_1\) of length at most 4, that is, \(w_1\) is either \(x\), \(xz\), \(xzy\) or \(xzyt\). Therefore, \(x\) can occur only twice in \(\sigma(u)\). But \(x\) and \(y\) occur three times in \(u\), and \(\sigma\) does not map any variable to the empty word, hence, there must be at least two variables which occur three times in \(\sigma(u)\). However, only \(x\) and \(y\) occur three times in \(xzyt x y r x s y\). We have reached a contradiction, hence, \(w_1 = \varepsilon\). Using a similar argument, we can also conclude that \(w_2 = \varepsilon\). Therefore, we have that

\[
xzyt x y r x s y = \sigma(u).
\]
As such, we can immediately conclude that only up to five variables occur in $u \approx v$: If $u \approx v$ were to be a six-variable identity, then it would be of length 10, and $\sigma$ would be simply renaming the variables, thus implying that $u \approx v$ was equivalent to (O), which contradicts our hypothesis.

Notice that, regardless of the number of variables occurring in $u \approx v$, we have that both $\sigma(x)$ and $\sigma(y)$ are a single variable, otherwise, more than two variables would have to occur three times in $xzt \ xy \ rxsy$, or one variable would have to occur six times. Furthermore, $\sigma(x)$ and $\sigma(y)$ can only be $x$ or $y$, since these are the only variables occurring three times in $xzt \ xy \ rxsy$. Hence, if $u \approx v$ is an identity where up to five variables occur, then $u \approx v$ cannot be a two-variable identity, and, furthermore, there is at least one variable $z$ occurring in $u \approx v$ such that $\sigma(z)$ is of length at least 2, and neither $x$ nor $y$ can occur in $\sigma(z)$. This is impossible, since $x$ or $y$ occur in every factor of $xzt \ xy \ rxsy$ of length 2.

As such, we can conclude that (O) is not a consequence of the set of non-trivial identities, satisfied by baxt, over an alphabet with six variables, excluding (O) itself and equivalent identities.

Therefore, we can conclude that $V_{\text{baxt}}$ does not admit any basis with only identities over an alphabet with up to five variables. In other words, we have that:

**Corollary 4.24** The axiomatic rank of $V_{\text{baxt}}$ is 6. Furthermore, any basis for $V_{\text{baxt}}$ with only identities over an alphabet with six variables must contain the identities (O) and (E), or equivalent identities.

**Acknowledgements** The authors thank the anonymous reviewers for their careful reading of the paper and helpful comments, in particular, for the observations on the varietal join and meet of the varieties generated by the sylvester and #-sylvester monoids, and the observation on the finite basis given for the variety generated by the Baxter monoid.

**Data Availability** Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

**References**

1. Adjan, S.I.: Defining relations and algorithmic problems for groups and semigroups. Trudy Mat. Inst. Steklov. **85**, 123 (1966)
2. Aho, A.V., Ullman, J.D.: Foundations of Computer Science. Computer Science Press, New York, Principles of Computer Science Series (1992)
3. Auinger, K., Volkov, M.: Equational theories of endomorphism monoids of categories with a topological flavor (2020). https://doi.org/10.48550/arXiv:2002.01016
4. Baxter, G.: On fixed points of the composite of commuting functions. Proc. Amer. Math. Soc. **15**, 851–855 (1964). https://doi.org/10.2307/2034894
5. Bergman, C.: Universal algebra, Vol. 301 of Pure and Applied Mathematics (Boca Raton), CRC Press, Boca Raton, FL, fundamentals and selected topics (2012)
6. Bump, D., Schilling, A.: Crystal bases, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, representations and combinatorics. (2017). https://doi.org/10.1142/9876
7. Burris, S., Sankappanavar, H.P.: A course in universal algebra. Graduate Texts in Mathematics, vol. 78. Springer-Verlag, New York-Berlin (1981)
8. Cain, A.J., Klein, G., Kubat, Ł, Malheiro, A., Okniński, J.: A note on identities in plactic monoids and monoids of upper-triangular tropical matrices (2017). https://doi.org/10.48550/arXiv:1705.04596
9. Cain, A. J., Malheiro, A.: Identities in plactic, hypoplactic, sylvester, Baxter, and related monoids, Electron. J. Combin. **25**(3), 3.30, 19 (2018). https://doi.org/10.37236/6873
35. Lascoux, A., Schützenberger M.-P.: Le monoïde plaxique, In: Noncommutative Structures in Algebra and Geometric Combinatorics (Naples, 1978), Vol. 109 of Quad. “Ricerca Sci.”, CNR, Rome, pp. 129–156, (1981)
36. Loday, J.-L., Ronco, M.O.: Hopf algebra of the planar binary trees. Adv. Math. 139(2), 293–309 (1998).
https://doi.org/10.1006/aima.1998.1759
37. Loday, J.-L., Ronco, M.O.: Hopf algebra of the planar binary trees. Adv. Math. 139 (2), 293–309 (1998).
https://doi.org/10.1006/aima.1998.1759
38. Lothaire, M.: Algebraic combinatorics on words, Vol. 90 of Encyclopedia of Mathematics and Its Applications, Cambridge University Press, Cambridge; a collective work by Jean Berstel. Dominique Perrin, Patrice Seebold, Julien Cassaigne, Aldo De Luca, Steffano Varricchio, Alain Lascoux, Bernard Leclerc, Jean-Yves Thibon, Veronique Bruyere, Christiane Frougny, Filippo Mignosi, Antonio Restivo, Christophe Reutenauer, Dominique Foata, Guo-Niu Han, Jacques Desarmenien, Volker Diekert, Tero Harju, Juhani Karhumaki and Wojciech Plandowski, With a preface by Berstel and Perrin. (2002).
https://doi.org/10.1017/CBO9781107326019
39. Lothaire, M.: Algebraic combinatorics on words, Vol. 90 of Encyclopedia of Mathematics and Its Applications, Cambridge University Press, Cambridge; a collective work by Jean Berstel. Dominique Perrin, Patrice Seebold, Julien Cassaigne, Aldo De Luca, Steffano Varricchio, Alain Lascoux, Bernard Leclerc, Jean-Yves Thibon, Veronique Bruyere, Christiane Frougny, Filippo Mignosi, Antonio Restivo, Christophe Reutenauer, Dominique Foata, Guo-Niu Han, Jacques Desarmenien, Volker Diekert, Tero Harju, Juhani Karhumaki and Wojciech Plandowski, With a preface by Berstel and Perrin. (2002).
https://doi.org/10.1017/CBO9781107326019
40. Macdonald, I.G.: Symmetric functions and Hall polynomials, 2nd Edition, Oxford Classic Texts in the Physical Sciences, The Clarendon Press, Oxford University Press, New York, with contribution by A. V. Zelevinsky and a foreword by Richard Stanley, Reprint of the 2008 paperback edition [MR1354144] (2015).
41. Macdonald, I.G.: Symmetric functions and Hall polynomials, 2nd Edition, Oxford Classic Texts in the Physical Sciences, The Clarendon Press, Oxford University Press, New York, with contribution by A. V. Zelevinsky and a foreword by Richard Stanley, Reprint of the 2008 paperback edition [MR1354144] (2015).
42. Mal’cev, A.I., Algebraic systems, Die Grundlehren der mathematischen Wissenschaften, Band 192, Springer-Verlag, New York-Heidelberg; posthumous edition. In: Smirnov, D. (ed.) and M. Ta˘ ıclin, Translated from the Russian by B. D. Seckler and A. P. Doohovskoy (1973)
43. Matveev, S.: On the finite basis problem for left hereditary systems of semigroup identities, in: Semigroups, automata and languages (Porto, 1996), World Sci. Publ., River Edge, NJ, pp. 167–181 (1996)
44. McKenzie, R.N., McNulty, G.F., Taylor, W.F.: Algebras, lattices, varieties. Vol. 1, AMS Chelsea Publishing/American Mathematical Society, Providence, RI, 2018, reprint of [MR0883644], (1969).
https://doi.org/10.1090/chel/383.H
45. Novelli, J.-C.: On the hypoplactic monoid, Discrete Math. 217 (1-3) (2000) 315–336, formal power series and algebraic combinatorics (Vienna, 1997). https://doi.org/10.1016/S0012-365X(99)00270-8
46. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley Publishing Company, Reading, MA (1994)
47. Pastijn, F.: Polyhedral convex cones and the equational theory of the bicyclic semigroup. J. Aust. Math. Soc. 81(1), 63–96 (2006).
https://doi.org/10.1017/S1446788700014646
48. Poirier, S., Reutenauer, C.: Algèbres de Hopf de tableaux. Ann. Sci. Math. Québec 19(1), 79–90 (1995)
49. Priez, J.-B.: Lattice of combinatorial Hopf algebras: binary trees with multiplicities, In: Goupil, A., Schaeffer, G. (Eds.), 25th International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2013), Vol. AS, 25th International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2013) of DMTCS Proceedings, Discrete Mathematics and Theoretical Computer Science, Paris, France, 1137–1148 (2013). https://doi.org/10.46298/dmtcs.2372
50. Reading, N.: Lattice congruences, fans and Hopf algebras. J. Combin. Theory Ser. A 110(2), 237–273 (2005).
https://doi.org/10.1016/j.jcta.2004.11.001
51. Ribeiro, D.: Identities and bases in plactic, hypoplactic, sylvester, and related monoids, Ph.D. thesis, NOVA School of Science and Technology, NOVA University Lisbon (Portugal) (2022). http://hdl.handle.net/10362/134505
52. Sapir, M.V.: Combinatorial algebra: syntax and semantics, Springer Monographs in Mathematics, Springer, Cham; with contributions by Victor S. Volkov, Guba and Mikhail V (2014). https://doi.org/10.1007/978-3-319-08031-4
53. Schensted, C.: Longest increasing and decreasing subsequences. Can. J. Math. 13, 179–191 (1961).
https://doi.org/10.4153/CJM-1961-015-3
54. Schützenberger, M.-P.: La correspondance de Robinson, In: Combinatoire et représentation du groupe symétrique (Actes Table Ronde CNRS, Univ. Louis-Pasteur Strasbourg, Strasbourg, 1976), Lecture Notes in Math., Vol. 579, Springer, Berlin, pp. 59–113 (1977)
55. Shneerson, L.M.: On the axiomatic rank of varieties generated by a semigroup or monoid with one defining relation. Semigroup Forum 39(1), 17–38 (1989).
https://doi.org/10.1007/BF02573281
56. Volkov, M.V.: The finite basis problem for finite semigroups. Sci. Math. Jpn. 53(1), 171–199 (2001)
Publisher's Note  Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.