Atlantic Electronic
Journal of Mathematics
Volume **6**, Number **1**, Summer **2014**

http://aejm.ca
http://aejm.ca/rema
pp. **26-37**

GENERATING SETS AND DECOMPOSITIONS FOR IDEMPOTENT TREE LANGUAGES

MARK THOM AND SHELLY WISMATH

Abstract. A tree language of a given type is any set of terms of that type. We consider here a binary operation + on the set of all arity n terms of the type, which produces a semigroup on the set. Using the characterization by Denecke, Sarasit and Wismath of languages which are idempotent with respect to this binary operation, we give a number of examples of idempotent languages, define generating sets for idempotent languages, and show how any idempotent language may be decomposed into a union of disjoint subsets. This decomposition allows us to assign to every term in an idempotent language a natural number called its idempotency level.

Keywords: Tree language, recognizable language, idempotent tree language, tree language decomposition.
AMS Subject Classification: 08C99, 20M17.
Acknowledgment: Research supported by NSERC of Canada.

1. **Introduction and Background.** Universal or general algebra involves the study of algebraic structures of different types. Such structures have in common a set of objects, one or more operations on the objects, and some identities or axioms which the objects in the set all satisfy. The *type* of such an algebraic structure is information about how many operations there are on the objects and what arities those operations have. For a finite type, this information is usually provided in a list: for example, groups can be viewed as structures of type $(2, 1)$, meaning that they have one binary operation and one unary operation.

Terms of a given type are formal expressions using the operation symbols of the type and a fixed set of variable or alphabet symbols. For example, using a binary operation f and two variables x_1 and x_2, we can form terms such as x_1, $f(x_2, x_1)$, $f(x_2, f(x_1, x_1))$, and so on. Terms themselves then become an object of study: the set of all such terms (of a fixed type and variable set) forms an algebraic structure of the same type, since we can always combine terms t_1 and t_2 into a new compound term $f(t_1, t_2)$.

In this paper we are interested in sets of terms of a fixed type. Such sets of terms are called *term languages* or also *tree languages*, since terms can be represented by tree diagrams. There is a rich literature on term languages and their connection with automata theory; see for instance [4], [5], [7] and [11]. This connection grew out of the work of Eilenberg ([5]) in the 1970's on languages recognizable by finite automata. The languages in question were sets of "words", more precisely terms of type (2) assuming the law of associativity. The classic result here is Eilenberg's Theorem ([4], [5]), that certain collections of formal languages (of words) correspond in a precise way to certain collections of semigroups. The semigroup collections are pseudovarieties, which are collections of *finite* algebras of a fixed type which are closed under the formation of

subalgebras, homomorphic images and finite direct products. The classes corresponding to finite pseudovarieties of semigroups, under the Eilenberg correspondence, were therefore called varieties of languages, by analogy. The languages determined by finite deterministic automata are precisely the regular languages.

Tree languages are used to generalize this finite automata-semigroup connection to arbitrary type: instead of words as type (2) semigroup terms, one can consider as languages any sets of terms of any fixed type. The analogue of a finite automaton is a tree automaton, which accepts or rejects terms and thus determines a language of (accepted or recognized) terms. Generalizing the Eilenberg correspondence, Rutten ([8]) has shown that deterministic tree automata suffice to accept regular tree languages.

Our goal here is to study tree languages which are idempotent with respect to a certain binary operation on sets of terms, defined using a superposition operation. This operation makes the family of all tree languages of a fixed type into a semigroup, and idempotent elements of a semigroup generally carry much information about the structure of the semigroup. In this paper we characterize those languages which are idempotent, define generating sets for such languages, and show how any such language may be uniquely decomposed into disjoint subsets.

With this context in mind, we now introduce the formal terminology needed to discuss sets of terms of a fixed arbitrary type. Let $\tau = (n_i)$ be any type of algebras, with operation symbols $(f_i)_{i\in I}$ indexed by set I. For convenience we assume that no constant ($n_i = 0$) terms are allowed. A (term) language of type τ is any set of terms of that type. Our terms will be defined over finite sets $X_n = \{x_1, \ldots, x_n\}$ of variables, for $n \geq 1$, and over the infinite variable set $X = \{x_1, x_2, x_3, \ldots\}$. Terms of type τ are defined inductively: each variable symbol x_1, ..., x_n is a term of arity n, and if $t_1, \ldots, t_{n_i}$ are n-ary terms and f_i is an n_i-ary operation symbol, then $f_i(t_1, \ldots, t_{n_i})$ is an n-ary term. Because of this, many definitions and proofs involving properties of terms are done inductively, using induction on the depth of a term. Formally, the depth of a term t is the number $depth(t)$, equal to 0 if t is a variable, and equal to $1 + max\{depth(t_1), \ldots, depth(t_{n_i})\}$, for $t = f_i(t_1, \ldots, t_{n_i})$. We denote by $W_\tau(X_n)$ the set of all terms of type τ on alphabet X_n.

Although terms are purely formal expressions on a given type and variable set, they can be combined much as functions of different arities on a fixed set can be combined. For example, if functions α and β have arities 3 and 2 respectively, the classical substitution of functions gives a 3-ary function $\beta(\alpha, \alpha)$. The analogous operation on terms is called *superposition*. It can be defined for any arities, but in this paper we need only the special case where all terms involved have the same arity n, for a natural number $n \geq 1$. This is the basis for the following definition.

Definition 1. For any natural number $n \geq 1$, the function

$$S_n \ : \ W_\tau(X_n)^{n+1} \ \rightarrow \ W_\tau(X_n),$$

is defined inductively by:

(i) $S_n(x_j, t_1, \ldots, t_n) := t_j$ for any variable $x_j \in X_n$, and

(ii) $S_n(f_i(s_1, \ldots, s_{n_i}), t_1, \ldots, t_n) \ := \ f_i(S_n(s_1, t_1, \ldots, t_n), \ldots, S_n(s_{n_i}, t_1, \ldots, t_n))$.

The next definition extends the superposition operation S_n to apply to sets of terms.

Definition 2. Let $n \geq 1$ be a natural number. We define

$$\hat{S}_n : \mathcal{P}(W_\tau(X_m))^{n+1} \rightarrow \mathcal{P}(W_\tau(X_n))$$

inductively as follows. Let B and $B_1, \ldots, B_n$ be in $\mathcal{P}(W_\tau(X_n))$.

(i) If $B = \{x_j\}$ for $1 \leq j \leq n$, then $\hat{S}_n(\{x_j\}, B_1, \ldots, B_n) := B_j$.

(ii) If $B = \{f_i(t_1, \ldots, t_{n_i})\}$ and the sets $\hat{S}_n(\{t_j\}, B_1, \ldots, B_n)$ for $1 \leq j \leq n_i$ have been defined, then $\hat{S}_n(\{f_i(t_1, \ldots, t_{n_i})\}, B_1, \ldots, B_n)$
$:= \{f_i(r_1, \ldots, r_{n_i}) \mid r_j \in \hat{S}_n(\{t_j\}, B_1, \ldots, B_n), 1 \leq j \leq n_i\}$.

(iii) If B is an arbitrary subset of $W_\tau(X_n)$, then
$\hat{S}_n(B, B_1, \ldots, B_n) := \bigcup_{b \in B} \hat{S}_n(\{b\}, B_1, \ldots, B_n)$.

(iv) If one of the sets $B, B_1, \ldots, B_n$ is empty, then $\hat{S}_n(B, B_1, \ldots, B_n) = \emptyset$.

Now we can define the binary operation $+_n$ on languages to be studied here, as it was introduced by Denecke and Sarasit in [1] and [2].

Definition 3. For any languages B_1 and B_2 on X_n, let

$$B_1 +_n B_2 := \hat{S}_n(B_1, B_2, \ldots, B_2).$$

This operation has been shown to be associative. This means that for any $n \geq 1$ we have a semigroup $(\mathcal{P}(W_\tau(X_n)); +_n)$. We are interested in languages L which are idempotent with respect to this operation $+_n$, that is, in languages which satisfy $L +_n L = L$.

2. **Idempotent Tree Languages.** Idempotence with respect to $+_n$ was characterized in [3], using the concept of *random replacement of variables* over a language. We begin by defining this concept.

Definition 4. Let L be a language of type τ. A term s is said to be obtained from a term t by *random replacement of variables* over L if s can be formed from t by replacing every occurrence of a variable in t by some term from L, randomly in the sense that different occurrences of the same variable in t can be replaced by different terms.

This is also referred to in the literature as OI-substitution (see [6]).

To formalize this concept of random replacement of variables over L, we define for any term t a set $RRV_L(\{t\})$ of terms, inductively as follows:

(i) If $t = x_j$ is a variable, then $RRV_L(\{t\}) = L$;

(ii) If $t = f_i(t_1, \ldots, t_{n_i})$ where all the terms $t_1, \ldots, t_{n_i}$ are variables, then $RRV_L(\{t\}) = \{f_i(s_1, \ldots, s_{n_i}) \mid s_1, \ldots, s_{n_i} \in L\}$;

(iii) If $t = f_i(t_1, \ldots, t_{n_i})$ for some terms $t_1, \ldots, t_{n_i}$, then
$RRV_L(\{t\}) = \{f_i(s_1, \ldots, s_{n_i}) \mid s_j \in RRV_L(\{t_j\}),$ for $1 \leq j \leq n_i\}$.

Then the set $RRV_L(\{t\})$ consists exactly of those terms which may be formed from t by replacing each occurrence of each variable in t, randomly in the sense that different occurrences of the same variable may be replaced by different terms, by a term from L. We denote by $RRV_L(L)$ the union of all sets of the form $RRV_L(\{t\})$ for $t \in L$. Then we say that a set L is closed under random replacement of variables over L if $RRV_L(L) \subseteq L$.

These definitions can now be used to characterize idempotence of languages L. First, it is easy to see that for any non-empty language L, we have $L \subseteq L +_n L$ iff $L \cap X_n \neq \emptyset$; that is, iff L contains at least one variable. For the inclusion $L +_n L \subseteq L$ we need the next lemma.

Lemma 1. *For any language L of terms of type τ, $L +_n L = RRV_L(L)$.*

Proof. Since both $L +_n L$ and $RRV_L(L)$ are defined as unions, it will suffice to show that $RRV_L(\{t\}) = \hat{S}_n(\{t\}, L, \ldots, L)$ for any term t of type τ. We do this by induction on the complexity of the term t. If t is a variable, then both sets $RRV_L(\{t\})$ and

$\hat{S}_n(\{t\}, L, \ldots, L)$ reduce to L itself. In the case that t has the form $t = f_i(t_1, \ldots, t_{n_i})$ for some terms $t_1, \ldots, t_{n_i}$ which are all variables, both sets reduce to $\{f_i(s_i, \ldots, s_{n_i}) \mid s_j \in L \text{ for } 1 \leq j \leq n_i\}$. Inductively, suppose that $t = f_i(t_1, \ldots, t_{n_i})$ for some terms $t_1, \ldots, t_{n_i}$, and that $RRV_L(\{t_j\}) = \hat{S}_n(\{t_j\}, L, \ldots, L)$, for $1 \leq j \leq n_i$. Then we have

$$\begin{aligned}
&\hat{S}_n(\{t\}, L, \ldots, L) \\
&= \{f_i(s_1, \ldots, s_{n_i}) \mid s_j \in \hat{S}_n(\{t_j\}, L, \ldots, L) \text{ for } 1 \leq j \leq n_i\} \\
&= \{f_i(s_1, \ldots, s_{n_i}) \mid s_j \in RRV_L(\{t_j\}) \text{ for } 1 \leq j \leq n_i\} \\
&= RRV_L(\{t\}).
\end{aligned}$$

□

Corollary 1. *Let L be any language of terms of type τ. Then $L +_n L \subseteq L$ iff $RRV_L(L) \subseteq L$; that is, iff L is closed under random replacement of variables over L.*

This result then gives us the following theorem characterizing languages idempotent with respect to the operation $+_n$.

Theorem 1. ([3]) Let L be a non-empty language of type τ over X_n. Then L is idempotent with respect to $+_n$ iff L contains at least one variable and is closed under random replacement of variables over L.

This characterization allows us to present some examples of idempotent languages. Clearly both $\emptyset$ and $W_\tau(X_n)$ are idempotent. Any finite set of variables is idempotent, and such sets are the only finite idempotent languages. An obvious question to consider is whether the intersection of idempotent sets is again idempotent. This will be true as long as the sets contain a common variable; but in general it is possible for two idempotent sets to have a non-empty intersection which does not contain any variables, making the intersection non-idempotent.

Our next examples of idempotent languages make use of structural properties of terms.

Example 2. The content of a term t is the set of all variables which occur in t. For any non-empty set A of variables from X_n, let $Cont_A$ be the language consisting of all terms t whose content is a subset of A. Then $Cont_A$ is idempotent, by Theorem 2.4. The case $A = X_n$ of course gives the language $W_\tau(X_n)$. Another special case occurs for $A = \{x_j\}$ for some variable x_j, in which case we get the language of all terms using only the variable x_j.

Example 3. For any term t, let $l(t)$ be the variable symbol that occurs on the leftmost of t. For any fixed non-empty subset A of X_n, let $Left_A$ be the language consisting of all terms t for which $l(t)$ is in A. Then $Left_A$ contains at least one variable, and is clearly closed under the random replacement of variables over $Left_A$, so it is again an idempotent language. A dual example can be made, using $r(t)$ as the rightmost variable symbol in t.

Example 4. For any natural number $k \geq 1$, let D_k be the set of all terms of type τ which have depth at least k. We define $Depth_k$ to be the language $X_n \cup D_k$. Again this gives an idempotent language, by Theorem 2.4. The same is true for $A \cup D_k$, for any non-empty subset A of X_n.

3. **Generating Sets for Idempotent Languages.** In order to define a generating set for a language, we first consider what it means for a set to generate a language, and what an idempotent language generated by a given set B of terms would be. For any such $B \subseteq W_\tau(X_n)$, the usual method to construct the smallest idempotent language containing B is to form the intersection of all the idempotent languages

containing B. There is at least one such idempotent language, namely $W_\tau(X_n)$ itself. However, in general the intersection of idempotent languages containing B need not be an idempotent language, since it need not contain any variables. We shall discuss briefly below what happens in the case that B contains no variables, but for the most part we consider in this section only sets B of terms which contain at least one variable.

Definition 5. Let $B \subseteq W_\tau(X_n)$ be a set of terms containing at least one variable term. We define $\langle B\rangle_{id}$ to be the intersection of all idempotent languages on X_n which contain B. Clearly $\langle B\rangle_{id}$ is the smallest idempotent language to contain B, and we shall call it the idempotent language generated by set B. We shall also say that a subset B of an idempotent language L generates L if B contains at least one variable and $\langle B\rangle_{id} = L$.

By Theorem 2.4, we could also define $\langle B\rangle_{id}$ as the intersection of all languages containing B which are closed under random replacement of variables over themselves. Moreover, $\langle B\rangle_{id}$ is closed under random replacement of variables over itself.

Another useful observation is that if B contains at least one variable, then there is an idempotent language M containing B with exactly the same set of variables as B; an example is the language M consisting of all terms of $W_\tau(X_n)$ except for those variables from X_n which are not in B. This means that $\langle B\rangle_{id}$ will contain exactly the same variables as B. This allows us to clarify what happens if our base set B does not contain any variables. In this situation, for any non-empty set A of variables from X_n, the set $\langle B \cup A\rangle_{id}$ is an idempotent language containing B and having exactly the variables from set A. Thus we cannot find a unique smallest idempotent language containing B, when $B \cap X_n$ is empty, but rather one such language for each choice of a variable to use in the idempotent language.

There is another natural way to produce the smallest idempotent language generated by a set B containing at least one variable. That is to add to B any terms in $L +_n L$, and then terms formed from those by random replacement of variables, and so on. The next definition formalizes this idea.

Definition 6. Let B be a set of terms containing at least one variable. We set $B^0 = B \cap X_n$, $\quad B^1 = B$ and $B^2 = B^1 +_n B^1$. Then inductively we set $B^{m+1} = B^1 +_n B$, for $m \geq 2$. Finally, let $\overline{B}$ be the union of the sets B^m for $m \geq 0$.

It was shown in [1] that the operation $+_n$ is an associative one, so we can think of $B^3 = B^1 +_n B^2 = B^2 +_n B^1$, and so on for B^m. It follows that for any natural numbers a and b, we have $B^a +_n B^b = B^{a+b}$.

Theorem 5. *Let B be a subset of $W_\tau(X_n)$ containing at least one variable term. Then $\langle B\rangle_{id} = \overline{B}$.*

Proof. By definition any language idempotent with respect to $+_n$ which contains B must also contain all of $\overline{B}$. Therefore we have $\overline{B} \subseteq \langle B\rangle_{id}$. For the opposite inclusion, we shall show that $\overline{B}$ is an idempotent language, clearly containing B, and so must contain $\langle B\rangle_{id}$. To show that $\overline{B}$ is idempotent, we note that since it contains a variable term we have $\overline{B} \subseteq \overline{B} +_n \overline{B}$, and must now show that $\overline{B} +_n \overline{B} \subseteq \overline{B}$.

We let t be any term in $\overline{B} +_n \overline{B}$. By Lemma 2.2, this means that t is in $RRV_{\overline{B}}(\overline{B})$. That is, t can be obtained from some term p in $\overline{B}$ by random replacement of the variables in p by terms from $\overline{B}$. This means that there is a finite list $(x_{j_1}, x_{j_2}, \ldots, x_{j_k})$ of all the variables in p, including all multiplicities, for some $k \geq 1$. There is also a corresponding list $(t_{j_1}, t_{j_2}, \ldots, t_{j_k})$ of terms from $\overline{B}$ such that term t_{j_r} is used to replace variable occurrence x_{j_r} when t is formed from p, for $1 \leq r \leq k$.

Each of the terms $p, t_{j_1}, \ldots, t_{j_k}$ is in $\overline{B}$. By construction of $\overline{B}$ therefore, there exist natural number indices m and $m_{j_1}, \ldots, m_{j_k}$ such that $p \in B^m$ and $t_{j_r} \in B^{m_{j_r}}$ for each $1 \leq r \leq k$. Now we take M to be the maximum of the finite set $\{m, m_{j_1}, \ldots, m_{j_k}\}$. Since B contains at least one variable we have $B^a \subseteq B^b$ for any natural numbers $a \leq b$, and hence the sets B^m, $B^{m_{j_1}}$, ..., $B^{m_{j_k}}$ are all subsets of B^M. That is, all of the terms $p, t_{j_1}, \ldots, t_{j_k}$ are in B^M. Our term t then can be formed from term p, which is in B^M, by random replacement of variables over B^M. This shows that $t \in RRV_{B^M}(B^M)$, which by Lemma 2.2 equals $B^M +_n B^M$. Therefore t is in B^{2M} and so is in $\overline{B}$. □

We now give some examples of generating sets, for the idempotent languages described in Section 2. In several cases we shall illustrate our examples using $n = 2$ and binary languages of type (2), where we have two variables x and y and one binary operation symbol, which we shall denote simply by juxtaposition.

Example 6. Let $W = W_\tau(X_n)$ be the language of all n-ary terms of type τ. This language can be generated by the set B which contains all the variables in X_n, along with for every $i \in I$ one term of the form $f_i(x_{j_1}, \ldots, x_{j_k})$, where $x_{j_1}, \ldots, x_{j_k}$ are variables. For example, in type (2), we can use any of the following sets to generate W:

$B_1 = \{x_1, x_2, x_1x_2\}$; $B_2 = \{x_1, x_2, x_2x_1\}$; $B_3 = \{x_1, x_2, x_1x_1\}$; $B_4 = \{x_1, x_2, x_2x_2\}$.

Notice here that given the two variable terms x_1 and x_2, we can generate any three of the four terms x_1x_2, x_2x_1, x_1x_1 or x_2x_2 from the fourth, using random replacement of variables.

This example motivates an observation regarding the depth of terms obtainable by random replacement of variables. Each of the four depth one terms in the example is obtainable from any of the others by a random replacement in which only variables are substituted, rather than more complex terms. We shall say that term s is obtained from term t by a *simple random replacement of variables* over a set L if s is obtained from t by replacing (randomly) every variable in t by a variable in L. In this case, the depth of term t is equal to the depth of the term s obtained from it.

Example 7. Consider the language $L = Cont_A$ from Example 2.5. To illustrate, we use type (2), with $n = 2$, and $A = \{x\}$. In this case our language consists of all binary terms which contain only the variable symbol x. It is clear that any generating set for this language must contain at least the terms x and xx, and that these are the only depth 0 or 1 terms in L. We take $B = \{x, xx\}$. There are three terms in L of depth 2, the terms $x(xx)$, $(xx)x$ and $(xx)(xx)$, and these can be obtained from B by random replacement of variables over B, so are in $B +_n B$. Thus far we see that for $m = 0, 1, 2$ we have that any term t in L has $depth(t) = m$ iff $t \in L^m \setminus L^{m-1}$. This can be extended by a straightforward induction to all natural numbers $m \geq 0$. This shows that B is a basis for L, and that the sets B^m correspond to the terms in L at depth of m or lower.

Example 8. Next we consider the language $Left_A$ from Example 2.6, again with type (2) and $n = 2$, and $A = \{x\}$ only. That is, our language consists of all binary type (2) terms in which the leftmost variable symbol is x. We claim that any generating set B for this language must be infinite. To prove this we shall show that any term which is in $\langle B \rangle_{id}$ but not in B must contain at least two occurrences of the variable x. But there are terms in $Left_A$, of arbitrarily high depth, which contain exactly one occurrence of x: for example, terms of the form $x(y(y(\ldots(yy)\ldots)))$. This means that no finite generating set is possible.

Since by Theorem 3.3 $\langle B\rangle_{id} = \bigcup_{m\geq 1} B^m$, it will suffice to prove by induction on m that any term t in $B^m \setminus B$, for $m \geq 2$, must have at least two occurrences of the variable x in it. For $m = 2$, any term in $B^2 = B +_n B$ can be obtained by random replacement of variables over B from some term p in B. If p is a variable, the result is a term in B. If p is not a variable, then the result of the random replacement on p involves at least one instance of a juxtaposition of terms from B, all of which start with x on their left, and so the result of the random replacement has at least two occurrences of x. Now inductively let $t \in B^{m+1} = B^1 +_n B^m$. Again we can obtain t by random replacement of variables from a term $p \in B$, where the terms used to replace variables in p come from B^m. If p is a variable term, then the result of the replacement is in B^m, and by induction if it is not in B it contains at least two occurrences of x. And if p is not a variable, then it involves at least one binary juxtaposition operation on terms from B^m, and so has at least two occurrences of x.

Example 9. For the language $L = Depth_k$ from Example 2.7, again in type (2) and with $k = 2$, we shall show that a finite generating set exists. We let B be the set of all terms of depth 2 or 3 with content x only, along with the two variables x and y.

First, we observe that any term s in the language L can be generated, by a simple random replacement over $\{x, y\}$, from a term t of the same depth as s but using only the variable x; the term t is simply the term s with every variable replaced by x. Thus it is enough to consider in our generating set B only terms of depth two or three which have content x.

Moreover, to prove that B does generate L it suffices to verify that we can produce any content x term t of depth four or more, from the set B. We do this by induction on the depth of t. Since t has depth at least four, there exists one or more instances in t of a variable x occurring at the end of a path in t of length at least four. Any such occurrence is a leaf-node on a depth two subterm w_t, again using only x, and such terms w_t are in our set B. Now let $\hat{t}$ be the term obtained by removing all such subterms w_t from t, and replacing each one with a variable x. By construction, t can be obtained by random replacement of variables from $\hat{t}$, with each newly occurring x being replaced by the appropriate w_t. When t has depth four, the term $\hat{t}$ has depth two, and so t is obtainable from B by random replacement of variables over B. Inductively, if t has depth $k \geq 5$, the term $\hat{t}$ has depth at least three, and is obtainable from B, so that t is also obtainable from B.

4. **Decomposition of Idempotent Languages.** Let L be an idempotent language, and let B be a generating set for L, containing at least one variable. In Theorem 3.3 we proved that L can be expressed as $\overline{B} = \bigcup_{m\geq 0} B^m$. This gives us a way to decompose the idempotent language generated by B into disjoint sets: we can take $C^0 = B^0 = B \cap X_n$, and then $C^m = B^m - B^{m-1}$, for $m \geq 1$. Since the sets B^m are nested, the sets C^m give us disjoint sets whose union is all of L. We call this union $L = \bigcup_{m\geq 0} C^m$ an *idempotent decomposition* of language L. Moreover, this allows us to assign to each term in L an *idempotent level*: t has idempotent level m if $t \in C^m$. We shall use the notation $idval_B$ for the function from L to $\mathbb{N}_0$ which assigns to each term its idempotent level in the decomposition determined by the generating set B.

There are many ways to decompose a language into such nested sets, the most obvious one here being by the depth of the terms. We shall show in some examples that the idempotent decomposition sometimes coincides with the depth decomposition, but does not always do so. First, we note from Example 3.4 that our decomposition of an idempotent language is not unique. In type (2), the language W of all binary terms

on the two variables x_1 and x_2 has four different generating sets of size three, depending on which of the four depth 1 terms we put in the generating set B. In fact the depth 1 term selected for B will have idempotent level 1, while the other three terms, which can be obtained in $B +_n B$, will have level 2. In this case it is thus a matter of choice which of the four depth one terms is given idempotent level 1, while the other three are given level 2. But these terms are in some sense equal in complexity, since each one is obtainable from any of the others by a simple random replacement, using only variables as replacement. To avoid such simple random replacements having different levels, we define the following procedure for decomposing an idempotent language L into disjoint subsets.

Definition 7. Let L be a non-empty language which is idempotent under the operation $+_n$. Let

$$\begin{aligned}\mathcal{G}_L^{(1)} \quad = \quad & \{t \in L \mid t = S_n(s_1, t_1, \ldots, t_n),\ s_1 \in L \setminus X_n,\ t_i \in L \text{ for } 1 \leq i \leq n, \\ & \exists\ j \ \text{ such that } t_j \in L \setminus X_n,\ x_j \in var(s_1)\}.\end{aligned}$$

We let $L^{(0)} = X_n \cap L$, and $L^{(1)} = L \setminus \mathcal{G}_L^{(1)}$. Inductively, let $L^{(m)} = L^{(1)} +_n L^{(m-1)}$ for $m \geq 2$. To decompose L into disjoint sets, we form $L^{(0)*} = X_n \cap L$ and $L^{(m)*} = L^{(m)} \setminus L^{(m-1)}$ for $m \geq 1$.

Lemma 2. *Let L be an idempotent language. Then any terms in $\mathcal{G}_L^{(1)}$ have depth at least 2; and as a consequence, any term in L of depth 0 or 1 must be in $L^{(1)}$.*

Proof. Let $t \in \mathcal{G}_L^{(1)}$. Then $t = S_n(s, t_1, \ldots, t_n)$ for some terms $s, t_1, \ldots, t_n$ with the following properties. First, s is not a variable, and there is at least one variable x_j occurring in s for which the corresponding term t_j is not a variable either. This means that both s and t_j have depth at least one, and hence the composition $t = S_n(s, t_1, \ldots, t_n)$ must have depth at least two. □

Proposition 1. *Suppose $L \subseteq W_\tau(X_n)$ is idempotent under $+_n$. Then*

$$L = \bigcup_{m=0}^{\infty} L^{(m)*}$$

Proof. L is idempotent, and every element of $L^{(m)*}$ is obtained by combining terms of L using $+_n$. Therefore, we immediately have

$$\bigcup_{m=0}^{\infty} L^{(m)*} \subseteq L.$$

Conversely, we must show that any term t in L is in some $L^{(m)*}$, which we do by induction on the depth of t. By the previous lemma, any terms of depth 0 or 1 in L must be in $L^{(1)}$. Now let t be a term from L of depth k. If $t \notin \mathcal{G}_L^{(1)}$, then t is by definition in $L^{(1)}$. So we suppose that $t \in \mathcal{G}_L^{(1)}$, and we can write $t = S_n(s_1, t_1, \ldots, t_n)$ for some non-variable term s_1 and some terms $t_1, \ldots, t_n$, all in L, with the additional property that there is a variable x_j occurring in s_1 for which the term t_j is not a variable.

Now each of the terms $s_1, t_1, \ldots, t_n$ is in L and has depth $< k$. Therefore by induction there are indices m_{s_1} and m_{t_r} such that $s_1 \in L^{(m_{s_1})}$ and $t_r \in L^{(m_{t_r})}$ for $1 \leq r \leq n$. By taking M to be sufficiently large, for example the sum of all these indices, we can make t an element of $L^{(M)}$. □

Corollary 2. *Let $L \subseteq W_\tau(X_n)$ be a non-empty idempotent language. Then there exists a sequence $I(L) = (L^{(m)} \mid m \in \mathbb{N}_0)$ of languages whose union is equal to L, with the properties that $L^{(m)} +_n L^{(k)} = L^{(m+k)}$ for every $m, k \geq 1$ and $L^{(m)} \subseteq L^{(k)}$ for every $k \geq m \geq 0$. There exists a valuation function $v : L \to \mathbb{N}_0$ defined on terms t in L by $v(t) = m$ if and only if $t \in L^{(m)*}$.*

At this point we introduce some new terminology.

Definition 8. Given any idempotent language $L \subseteq W_\tau(X_n)$, we refer to $L^{(1)}$ as the *core* of L, and $\bigcup_{m=0}^{\infty} L^{(m)*}$ as the *core entanglement* of L. We refer to the mapping $idval_{L^{(1)}}$ from Corollary 4.4 as the *idempotency valuation* induced by L, and for ease of notation shall denote it simply as $idval_L$.

We observe that terms which can be formed from each other by simple random replacement of variables are given the same idempotency level here. If L is a language closed under random replacement of variables, we can consider $L^{(1)}$ as the minimal language contained within L that generates L as its core entanglement. Then L is closed under random replacement if and only if L is equal to its core entanglement, and idempotent if and only if it is closed under random replacement and $L^{(0)} \neq \emptyset$.

Example 10. Let $W = W_\tau(X_n)$ be the language of all terms of type $\tau = (2)$. It follows from Lemma 4.2 that all depth 0 and depth 1 terms must be in the core of the language W, and it is clear that all terms of higher depth can be obtained from such terms by means of $+_n$. Thus the set of depth 0 and 1 terms forms the core of W. Moreover, it can be shown by straightforward induction that in this example, the idempotency valuation function $idval_L$ coincides with the depth function on $W_\tau(X_n)$.

Example 11. Consider the language $L = Cont_A$ from Examples 2.5 and 3.5. We have a core $B = L^{(1)}$ for L containing two terms, one in B^0 at depth 0 and one in B^1 at depth 1. We see from the argument in Example 3.5 that in this case the idempotent valuation function coincides with the depth function on the set of all terms.

Example 12. The idempotent languages of Examples 3.6 and 3.7 show situations in which the idempotent valuation mapping does not agree with the depth mapping on the set of all terms.

5. **Extensions of Idempotency-Valuation Maps.** We have seen in the previous section that for any idempotent term language L, we have associated with the core and core entanglement of L a mapping $idval_L$ from L to $\mathbb{N}_0$. We now consider whether it is possible to extend this mapping to a mapping on all of $W_\tau(X_n)$, in a consistent way.

Definition 9. Let L be an idempotent language with idempotency valuation $idval_L$. We shall say that this valuation is *extendable to* $W_\tau(X_n)$ if there exists a set B containing $L^{(1)}$ such that

$$W_\tau(X_n) = \bigcup_{m=0}^{\infty} B^m \qquad \text{and} \qquad B^m \cap L = L^{(m)} \qquad \text{for every } m \in \mathbb{N}_0.$$

Given the definition of the idempotency valuation mappings, the second condition in this definition is equivalent to the requirement that for all terms t in the language L, we must have $idval_B(t) = idval_L(t)$. That is, the idempotent valuation determined by B must agree with that for L on the set L.

The first condition of this definition means that the set B must generate the full language $W_\tau(X_n)$. This is equivalent to the condition that all terms of depth 0 or 1

must be in B. If we take B to be precisely this set, then the decomposition of $W = W_\tau(X_n)$ into the sets $W^{(m)}$ corresponds to the depth valuation on W, as in Example 4.6. In this case then, an idempotency valuation $idval_L$ on a language L can only be extended to $W_\tau(X_n)$ if v agrees with the depth function on L.

Example 13. The valuation on the language $Cont_A$ from Example 4.7 is extendable to $W_\tau(X_n)$, in the case $A = \{x\}$, $n = 2$ and $\tau = (2)$.

Example 14. Let L be any language containing some but not all of the variables from X_n, along with all non-variable terms of type τ. Then the valuation $idval_L$ respects the depth function, and so can be extended to the depth valuation on $W_\tau(X_n)$.

It is also possible to take the set B to be larger than the set of all terms of depth 0 or 1. In particular, some of the languages L in the examples of the previous section had some terms of depth ≥ 2 in $L^{(1)}$. In order to satisfy the second condition of Definition 5.1, we need to have $L^{(1)}$ contained in B. This condition also means that no other terms from L are contained in B; but it is possible that some other terms not in L could be contained in B.

We can summarize this by saying that if the valuation of the language L is going to be extendable to $W_\tau(X_n)$, then the generating set B to be used must contain at least all of the terms of type τ of depth 0 or 1, along with all terms from $L^{(1)}$.

6. **Varieties of Recognizable Languages.** An interesting class of languages is that of the *recognizable* languages; see for example [7], [9], [10], and [11]. By a straightforward extension of the finite automaton case, a term or tree language is recognizable if it is the set of terms accepted by some tree automaton. The family $Rec(\tau, X_n)$ of all recognizable languages of type τ on alphabet X_n forms a Boolean algebra. That is, the empty set and $W_\tau(X_n)$ itself are both recognizable, every finite tree language is recognizable, and the union, intersection and set difference of any two recognizable languages is recognizable. Moreover, $Rec(\tau, X_n)$ has two other important properties. The first is closure under inverse homomorphisms, in the following sense: if $L \subseteq W_\tau(X_n)$ is recognizable, and $\varphi : \mathcal{F}_\tau(X_m) \to \mathcal{F}_\tau(X_n)$ is a homomorphism, then $\varphi^{-1}(L)$ is also a recognizable language on X_m. The second property is closure under *inverse translation*. Let $L \subseteq W_\tau(X_n)$ and let p be any unary polynomial operation symbol of type τ on X_n. The inverse translation or cancellation of L under p is defined by

$$p^{-1}(L) := \{t \mid t \in W_\tau(X_n) \text{ and } p(t) \in L\}.$$

A class of languages is closed under inverse translation if for every language L in the class and every unary polynomial symbol p, we have $p^{-1}(L)$ also in the class.

These three properties are used to define a variety of tree languages.

Definition 10. A *variety* of tree languages of type τ is a sequence $VL := (VL_n)_{n\in\mathbb{N}^+}$ such that
(i) each VL_n forms a Boolean subalgebra of $Rec(\tau, X_n)$;
(ii) each VL_n is closed under inverse translation;
(iii) VL is closed under inverse homomorphisms.

Example 15. For each $n \geq 1$, let $VL_n = \{\emptyset, W_\tau(X_n)\}$, and let $VL = (VL_n)_{n\geq 1}$. Each of the sets in VL_n is recognizable, and it can easily be shown that this is a variety of tree languages. Moreover, it is the smallest of all such varieties, in the sense that it must be contained in every variety. For if VL is a variety of tree languages, and for each $n \geq 1$ there is a language L in VL_n, then $L - L = \emptyset$ must be in VL_n and so also its complement $W_\tau(X_n)$ must be in VL_n.

It can be shown that the intersection of varieties of tree languages is also a variety. Steinby showed (see [10] and [11]) that there is a one-to-one correspondence between varieties of tree languages of type τ and pseudovarieties of finite algebras of type τ. A variation on the definition of a variety of languages is that of a positive variety ([11]). A sequence of sets of recognizable languages is called a *positive* variety of tree languages, if all the necessary properties of a variety hold except possibly complementation. Salehi ([9]) has shown that positive varieties of tree languages correspond to ordered pseudovarieties of finite ordered algebras.

Imitating the proof from [1] that the sum $L +_n M$ is recognizable when L and M are recognizable languages, it can be shown that $\mathcal{G}_L^{(1)}$ is a recognizable language if L is recognizable. Thus, the set difference $L \setminus \mathcal{G}_L^{(1)}$ is also recognizable. It follows that if L is a recognizable idempotent language, then each element $L^{(m)}$ of its decomposition sequence is also recognizable.

The languages from Example 5.3 give us an example of a positive variety of languages. For each $n \geq 1$, let VL_n be the class of all languages of the form $A \cup D_1$, where D_1 is the set of all terms from $W_\tau(X_n)$ of depth one or more, and A is any subset of the set X_n of variables. The languages in VL_n all have finite complement, and since any finite set is recognizable, the languages are also recognizable. We note that as long as A is a non-empty set of variables, the language $A \cup D_1$ is idempotent, by Theorem 2.4. In the case that A is the empty set, the language $A \cup D_1$ is not idempotent; but this language must be included in VL_n in order to give us closure under intersection.

Proposition 2. *For each $n \geq 1$, let VL_n consist of the languages on $W_\tau(X_n)$ of the form $A \cup D_1$, where A is any subset of X_n and D_1 is the set of all terms of type τ of depth one or higher. Then the family $VL = (VL_N)_{n \geq 1}$ is a positive variety of languages, with the property that for all $n \geq 1$, all but one of the languages in VL_n is idempotent.*

Proof. We have shown that the languages in VL_n are recognizable, and are all idempotent except for D_1. It is clear that VL_n is closed under union and intersection, since the union of two sets $A_1 \cup D_1$ and $A_2 \cup D_1$ is $(A_1 \cup A_2) \cup D_1$ while their intersection is $(A_1 \cap A_2) \cup D_1$.

For each $n \geq 1$, the set $W_\tau(X_n)$ forms the universe of the free algebra $\mathcal{F}_\tau(X_n)$. Closure under inverse homomorphisms requires that if $\varphi : \mathcal{F}_\tau(X_m) \to \mathcal{F}_\tau(X_n)$ is any homomorphism of free algebras, then for any $L = A \cup D_1$ in VL_n, the language $\varphi^{-1}(L)$ must be in VL_m. We also need each VL_n to have closure under *inverse translation* or *cancellation*. Let $L \subseteq W_\tau(X_n)$ and let p be any unary polynomial operation of type τ on X_n. The inverse translation or cancellation of L under p is defined by

$$p^{-1}(L) := \{t \mid t \in W_\tau(X_n) \text{ and } p(t) \in L\}.$$

It follows easily from the definition of the languages in VL_n that both of these closure properties hold. □

REFERENCES

[1] K. Denecke and N. Sarasit, Semigroups of Tree Languages, *Asian-European J. Mathematics* **1,4** (2008) 489–507.

[2] K. Denecke and N. Sarasit, Products of Tree Languages, preprint, 2009.

[3] K. Denecke, N. Sarasit and S. L. Wismath, Idempotent Tree Lanuages, to appear in Demonstratio Mathematicae.

[4] S. Eilenberger, *Automata, Languages, and Machines, Volume A* Academic Press, New York, 1974.

[5] S. Eilenberg, *Automata, Languages, and Machines, Vol. B.*, Pure and Applied Mathematics, Vol. 59, Academic Press, New York & London, 1976.
[6] Joost Engelfriet and Erik Meineche Schmidt, IO and OI, I. J. Comput. Syst. Sci. Vol **15(3)** 328 - 353.
[7] F. Gécseg and M. Steinby, Tree Languages, *Handbook of Formal Languages, Vol.* **3**, Springer, Berlin, 1997, 1–68.
[8] J. Rutten, Universal Co-algebra: A Theory of Systems, *Theoretical Computer Science* **249** (2000) no. 1, 3 - 80.
[9] S. Salehi, Varieties of Tree Languages, TUCS Dissertations, N0. 64, July 2005.
[10] M. Steinby, Syntactic algebras and varieties of recognizable sets, in: M. Bidoit and M. Dauchet,(eds.), (eds.)*Proc. CAAP'79*, (University of Lille 1979), 226–240.
[11] M. Steinby, A theory of tree language varieties, in: Nivat, M. & Podolski, A. (eds.) *Tree Automata and Languages* Elsevier, Amsterdam (1992).

Authors' Addresses:
Mark Thom, Dept. of Mathematics, University of British Columbia, B.C., Canada; email markth@math.ubc.ca
Shelly Wismath, Dept. of Mathematics and Computer Science, University of Lethbridge, Lethbridge AB Canada; email: wismaths@uleth.ca