On Structural Properties of Eta-expansions of Identity *

B. Intrigila and M. Nesi

Dipartimento di Informatica
Università degli Studi di L'Aquila
via Vetoio, 67010 L'Aquila, Italy

email: {intrigila,monica}@di.univaq.it

Abstract. We investigate some structural properties of a particular class of λ-terms, the so-called η-expansions of identity $\mathbf{I}$, considered inside the β-calculus. This class of terms can be directly identified with a class of (unlabelled) trees, *plane* (or *ordered*) *trees*, which correspond to the so-called parentheses systems. By means of this identification, the class of plane trees can be equipped with an *application* operation as well as with a *composition* operation. We show that composition corresponds to a *union* operation on trees, while application can be characterized in terms of composition and a natural operation of *shift* over trees. By means of this characterization a number of results can be easily proved about the structure of this class of terms.

1 Introduction

This paper presents an investigation on some structural properties of a particular class of λ-terms, the so-called η-expansions of identity $\mathbf{I}$, considered inside the β-calculus. By the well-known Böhm trees construction (see [1], 10.1.3), this class of terms can be directly identified with a class of unlabelled trees, *plane* (or *ordered*) *trees* [4, 6], since the term reconstruction can be done univocally without any use of labels. Plane trees are a combinatorial structure of great importance, since they correspond to the so-called *parentheses systems*.

By the above mentioned identification, the class of plane trees can be equipped with an *application* operation as well as with a *composition* operation. This is due to the fact that η-expansions of identity are closed under such operations. The following question naturally arises:

Question: Is there a natural structural description of application and composition directly on trees?

In this paper we solve this question in the affirmative by showing that composition corresponds to a *union* operation on trees, while application can be

* Work partially supported by MURST 40% "Architetture Software per Infrastrutture di Rete ad Accesso Eterogeneo (SAHARA)".

characterized in terms of composition and a natural operation of *shift* over trees. By means of this characterization a number of results can be easily proved about the structure of this class of terms. We plan to extend this kind of analysis to wider classes of terms, possibly by making use of more powerful techniques. A natural extension we are working on is addressing infinite η-expansions of identity and infinite trees, defined as the unlabelled versions of, respectively, the infinite η-expansions and the Böhm-like trees.

2 Basic Definitions

In the following, *identity* is as usual the combinator $\lambda x.x$, denoted by $\mathbf{I}$. In this section we briefly recall some well-known facts about λ-calculus and the basic facts concerning η-expansions of $\mathbf{I}$. As a general reference we indicate [1].

Definition 1. *The class $\Lambda_\eta^{\mathbf{I}}$ of the η-*expansions of identity *is the set of λ-terms which $\beta\eta$-reduce to identity.*

Example 1. The following λ-terms are η-expansions of $\mathbf{I}$: $\lambda xyz.xyz$, $\lambda xy.x(\lambda z.yz)$, $\lambda xy.x(\lambda z.y(\lambda w.zw))$.

Proposition 1. *Every λ-term in $\Lambda_\eta^{\mathbf{I}}$ has a β-normal form.*

In the following the η-expansions of $\mathbf{I}$ will be denoted with $M, N, P, \ldots$.

Definition 2. *The equivalence relation $=_\beta$ on $\Lambda_\eta^{\mathbf{I}}$ is defined as follows: for all $M, N \in \Lambda_\eta^{\mathbf{I}}$ $M =_\beta N$ if and only if M and N have the same β-normal form.*

The relation $=_\beta$ partitions the set $\Lambda_\eta^{\mathbf{I}}$ into equivalence classes and, given any η-expansion of $\mathbf{I}$, its β-normal form will be taken as representative of the class. In this paper, we consider η-expansions of $\mathbf{I}$ modulo $=_\beta$, shortened in $=$ when no confusion arises.

Definition 3. *For any λ-terms M, N, the* composition $M \circ N$ *is defined as follows: $M \circ N = \lambda x.M(Nx)$ where x is a fresh variable in M and N.*

Lemma 1. *The equivalence relation $=_\beta$ is a congruence with respect to application and composition.*

It is well known, due to Church, that combinators form a monoid with respect to composition and with $\mathbf{I}$ as the unit element in the $\beta\eta$-calculus [1]. In Section 3 we will show that $\Lambda_\eta^{\mathbf{I}}$ is also a monoid with respect to composition and with $\mathbf{I}$ as the unit element in the β-calculus. Moreover, this monoid turns out to be idempotent and commutative (see Corollary 2).

3 η-Expansions of Identity and Trees

We now introduce the class of *plane* trees (see [6]) and study how they can be related to the η-expansions in $\Lambda_\eta^{\mathbf{I}}$.

Definition 4. *A* plane tree $\mathcal{A}$ *is a non-empty finite set of finite sequences in* $\mathbb{N}$ *such that:*
(i) if s *is an initial segment of* $s_1 \in \mathcal{A}$*, then also* $s \in \mathcal{A}$*;*
(ii) for all $s \in \mathcal{A}$ *there exists* $n \in \mathbb{N}$ *such that* $s \cdot \langle i \rangle \in \mathcal{A}$ *if and only if* $1 \leq i \leq n$.

Note that Definition 4 is one of many equivalent characterizations of this class. Property (i) characterizes *trees*, while property (ii) gives a restriction on them asserting that a plane tree is a tree with no "holes". This amounts to saying that the sons of a node are taken in a fixed order, so that such trees can also be called *ordered trees*. An equivalent definition is as follows (see [4], Section 2.3):

Definition 5. *An* ordered tree $\mathcal{A}$ *is a non-empty finite set of* nodes *such that:*
(a) there is one specially designated node called the root *of* $\mathcal{A}$*;*
(b) the remaining nodes are partitioned into $m \geq 0$ *disjoint sets* $\mathcal{A}_1, \ldots, \mathcal{A}_m$*, in a fixed order, and each of these sets is in turn an ordered tree. The trees* $\mathcal{A}_1, \ldots, \mathcal{A}_m$ *are called the* subtrees *of* $\mathcal{A}$.

It is easy to see that the two definitions specify the same class.
Let $\mathcal{F}_\mathcal{A}$ denote the family of plane trees, usually denoted with $\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$. The tree $m(M)$ associated to a term M is defined as follows.

Definition 6. *Let* $M \in \Lambda_\eta^{\mathbf{I}}$ *and* N *be its* β*-normal form. The* tree $m(M)$ asso*ciated to* M *is recursively defined as follows:*
(i) if $N \equiv \lambda x.x$*, then* $m(M) = \{\epsilon\}$*;*
(ii) if $N \equiv \lambda x x_1 \ldots x_n . x M_1 \ldots M_n$*, then*
$m(M) = \{\epsilon\} \cup \{\langle i \rangle \cdot \alpha \mid 1 \leq i \leq n,\ \alpha \in m(\lambda x_i . M_i)\}$.

Thus, the tree associated to $\mathbf{I}$ is $m(\mathbf{I}) = \{\epsilon\}$, that is a tree consisting of the only root with no subtrees. We introduce the empty set $\emptyset$ among the trees. No λ-term is associated to $\emptyset$, however the empty tree will be useful when proving some results in the following. Note that the trees under consideration are the (unlabelled) Böhm trees of the corresponding terms.

Lemma 2. *The function* m *is a bijection from* $\Lambda_\eta^{\mathbf{I}}$ *to* $\mathcal{F}_\mathcal{A}/\emptyset$.

Proof. A function $m^{-1} : \mathcal{F}_\mathcal{A}/\emptyset \longrightarrow \Lambda_\eta^{\mathbf{I}}$ is defined as follows:

- if $\mathcal{A} \equiv \{\epsilon\}$, then $m^{-1}(\mathcal{A}) = \mathbf{I}$,
- if $\mathcal{A}$ has subtrees $\mathcal{A}_1, \ldots, \mathcal{A}_n$, where $m^{-1}(\mathcal{A}_i) = M_i$ $(1 \leq i \leq n)$, then $m^{-1}(\mathcal{A}) = \lambda x x_1 \ldots x_n . x(M_1 x_1) \ldots (M_n x_n)$.

The proof is by induction on the tree structure and m is proved to be a bijection by showing that m^{-1} is both left and right inverse of m.

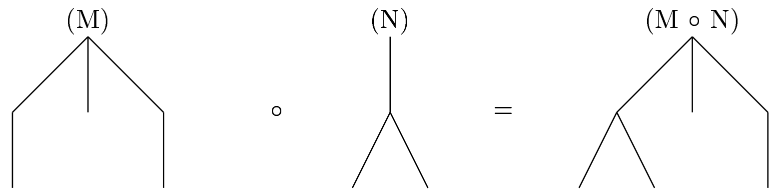
(i) Let $M \in \Lambda_\eta^{\mathbf{I}}$ and N be its β-normal form. We have the following cases:
   1. $N \equiv \lambda x.x$, then $(m^{-1} \circ m)(M) = m^{-1}(m(M)) = m^{-1}(\{\epsilon\}) = \mathbf{I}$.
   2. $N \equiv \lambda x x_1 \ldots x_n.x M_1 \ldots M_n$, then $(m^{-1} \circ m)(M) = m^{-1}(m(M))$ where the tree $m(M)$ has subtrees $m(\lambda x_1.M_1), \ldots, m(\lambda x_n.M_n)$. By inductive hypothesis we have $(m^{-1} \circ m)(\lambda x_i.M_i) = \lambda x_i.M_i$ $(1 \le i \le n)$.
   Thus, $(m^{-1} \circ m)(M) = \lambda x x_1 \ldots x_n.x M_1 \ldots M_n$.

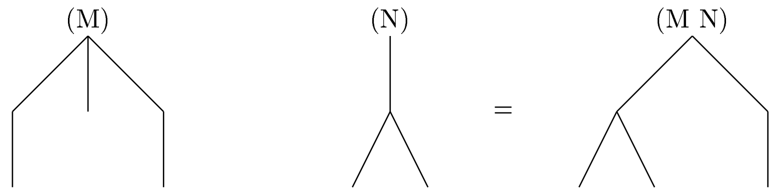
(ii) Let $\mathcal{A}$ be a non-empty tree. We have the following cases:
   1. $\mathcal{A} \equiv \{\epsilon\}$, then $(m \circ m^{-1})(\{\epsilon\}) = m(m^{-1}(\{\epsilon\})) = m(\mathbf{I}) = \{\epsilon\}$.
   2. $\mathcal{A}$ has subtrees $\mathcal{A}_1, \ldots, \mathcal{A}_n$, then
   $(m \circ m^{-1})(\mathcal{A}) = m(\lambda x x_1 \ldots x_n.x(M_1 x_1) \ldots (M_n x_n))$.
   As $M_i x_i \stackrel{*}{\to}_\beta N_i$, where N_i is in β-normal form, then we have
   $m(\lambda x x_1 \ldots x_n.x N_1 \ldots N_n) = \mathcal{A}$. □

In the following we will indifferently refer to terms in $\Lambda_\eta^{\mathbf{I}}$ or plane trees. Moreover, we shorten "plane trees" with "trees". Based on Lemma 2 we can define the application and the composition on trees.

Definition 7. *For any trees $\mathcal{A}, \mathcal{B}$, we define:*
(i) $\mathcal{A}\mathcal{B} = m(m^{-1}(\mathcal{A})m^{-1}(\mathcal{B}))$;
(ii) $\mathcal{A} \circ \mathcal{B} = m(\lambda x.m^{-1}(\mathcal{A})(m^{-1}(\mathcal{B})x))$
where x is a fresh variable in $m^{-1}(\mathcal{A})$ and $m^{-1}(\mathcal{B})$.

Thus, by Definition 7, for terms $M \equiv \lambda xyzw.x(\lambda\overline{x}.y\overline{x})z(\lambda\overline{x}.w\overline{x})$ and $N \equiv \lambda xy.x(\lambda zw.yzw)$ we can compute:





The natural question arises whether one could do better, computing composition and application directly on trees. We shall treat application in Section 4. In order to deal with composition, a *max* operation is defined on trees as a union operation.[1]

Definition 8. *For all trees $\mathcal{A}, \mathcal{B}$, we define $max(\mathcal{A}, \mathcal{B}) = \mathcal{A} \cup \mathcal{B}$.*

[1] We use *max* to denote both the operation on trees and the usual maximum operation on numbers. It will be clear from the context to which operation we refer.

$(\mathcal{F}_{\mathcal{A}}, max)$ is a commutative idempotent monoid with unity $\{\epsilon\}$. Composition on trees simply corresponds to the max operation.

Proposition 2. *For any trees $\mathcal{A}, \mathcal{B}$, we have $\mathcal{A} \circ \mathcal{B} = max(\mathcal{A}, \mathcal{B})$.*

Proof. The proof is by induction on the tree structure. Let $\mathcal{A}$ have subtrees $\mathcal{A}_1, \ldots, \mathcal{A}_n$ and $\mathcal{B}$ have subtrees $\mathcal{B}_1, \ldots, \mathcal{B}_m$. Let $m^{-1}(\mathcal{A}) = \lambda y y_1 \ldots y_n . y(M_1 y_1) \ldots (M_n y_n)$ and $m^{-1}(\mathcal{B}) = \lambda z z_1 \ldots z_m . z(N_1 z_1) \ldots (N_m z_m)$.
We have the following three cases:

1. $n < m$.
   $\mathcal{A} \circ \mathcal{B} = m(\lambda x y_1 \ldots y_n z_{n+1} \ldots z_m .$
   $x(N_1(M_1 y_1)) \ldots (N_n(M_n y_n))(N_{n+1} z_{n+1}) \ldots (N_m z_m))$.
   By inductive hypothesis we have:
   $m(\lambda y_i.(N_i(M_i y_i))) = \mathcal{B}_i \circ \mathcal{A}_i = max(\mathcal{A}_i, \mathcal{B}_i)$ for all i, $1 \leq i \leq n$ and
   $m(\lambda z_j.(N_j z_j)) = \mathcal{B}_j = max(\emptyset, \mathcal{B}_j)$ for all j, $n+1 \leq j \leq m$.
   Thus, $\mathcal{A} \circ \mathcal{B} = max(\mathcal{A}, \mathcal{B})$.
2. $n > m$.
   $\mathcal{A} \circ \mathcal{B} = m(\lambda x y_1 \ldots y_n .$
   $x(N_1(M_1 y_1)) \ldots (N_m(M_m y_m))(M_{m+1} y_{m+1}) \ldots (M_n y_n))$.
   By inductive hypothesis we have:
   $m(\lambda y_i.(N_i(M_i y_i))) = \mathcal{B}_i \circ \mathcal{A}_i = max(\mathcal{A}_i, \mathcal{B}_i)$ for all i, $1 \leq i \leq m$ and
   $m(\lambda y_j.(M_j y_j)) = \mathcal{A}_j = max(\mathcal{A}_j, \emptyset)$ for all j, $m+1 \leq j \leq n$.
   Thus, $\mathcal{A} \circ \mathcal{B} = max(\mathcal{A}, \mathcal{B})$.
3. $n = m$.
   $\mathcal{A} \circ \mathcal{B} = m(\lambda x y_1 \ldots y_n . x(N_1(M_1 y_1)) \ldots (N_n(M_n y_n)))$.
   By inductive hypothesis we have $m(\lambda y_i.(N_i(M_i y_i))) = \mathcal{B}_i \circ \mathcal{A}_i = max(\mathcal{A}_i, \mathcal{B}_i)$ for all i, $1 \leq i \leq n$. Thus, $\mathcal{A} \circ \mathcal{B} = max(\mathcal{A}, \mathcal{B})$.

The cases where either $\mathcal{A}$ or $\mathcal{B}$ (or both) is $\{\epsilon\}$ are derived as special cases with $n = 0$ or (and) $m = 0$. $\square$

As immediate corollaries we get the following results.

Corollary 1. *$(\mathcal{F}_{\mathcal{A}}, max)$ is isomorphic to $(\Lambda_\eta^{\mathbf{I}}, \circ)$.*

Corollary 2. *(i) $\Lambda_\eta^{\mathbf{I}}$ is closed with respect to application and composition.*
(ii) $(\Lambda_\eta^{\mathbf{I}}, \circ)$ is a commutative idempotent monoid with unity $\mathbf{I}$.

In order to prove some of the results of Section 4, we introduce the concept of maximal tree of given depth and outdegree of its nodes. For non-negative n and h, the *maximal* tree $Top_{n,h}$ is the tree in which every leaf is at depth h and every other node has outdegree n. Any tree $\mathcal{A}$ is a subset of $Top_{n,h}$ for some n and h. For any non-negative n and h, let $T_{n,h}$ denote the (finite) class of all those trees such that the maximum outdegree of their nodes is n and the maximum depth of their leaves is h. The following result follows easily.

Proposition 3. *For any non-negative n and h, the class $T_{n,h}$ is closed under the union and subtree operations. Thus, $T_{n,h}$ is closed under composition.*

4 Shift Operations and Some Structural Properties

While composition on η-expansions of $\mathbf{I}$ is isomorphic to the max operation on the associated trees, application can be defined in terms of a suitable shift operation on the corresponding trees.

4.1 The Shift Operation

Given any tree $\mathcal{A}$, a *shift* operation on $\mathcal{A}$ moves each subtree of $\mathcal{A}$ of one position towards left, i.e. the subtree $\mathcal{A}_i$ $(i > 1)$ becomes the subtree in position $i-1$ in the resulting tree, by deleting the leftmost subtree $\mathcal{A}_1$. Formally, if the subtrees of $\mathcal{A}$ are, in the order, $\mathcal{A}_1, \ldots, \mathcal{A}_n$, then $\mathit{shift}(\mathcal{A})$ is the unique tree $\mathcal{A}'$ such that the subtrees of $\mathcal{A}'$ are, in the order, $\mathcal{A}_2, \ldots, \mathcal{A}_n$. Whenever $n = 0$ or $n = 1$, in either cases the resulting tree consists of the root only.
A generalized version of this shift operation is defined below, which consists of "overlapping" the leftmost subtree $\mathcal{A}_1$ with the tree resulting from shifting $\mathcal{A}$ of one position towards left.

Definition 9. *For any tree $\mathcal{A}$, the* generalized shift *$G(\mathcal{A})$ is defined as $G(\mathcal{A}) = max(\mathcal{A}_1, \mathit{shift}(\mathcal{A}))$.*

Let us show some properties of the generalized shift operator.

Proposition 4. *For any trees $\mathcal{A}, \mathcal{B}$, we have $G(\mathcal{A} \circ \mathcal{B}) = G(\mathcal{A}) \circ G(\mathcal{B})$.*

Proof.

$$\begin{array}{ll}
G(\mathcal{A} \circ \mathcal{B}) = & \text{(Proposition 2)} \\
G(max(\mathcal{A}, \mathcal{B})) = & \text{(Definition 9)} \\
max((max(\mathcal{A}, \mathcal{B}))_1, \mathit{shift}(max(\mathcal{A}, \mathcal{B})))) = & \text{(properties of max)} \\
max(max(\mathcal{A}_1, \mathcal{B}_1), max(\mathit{shift}(\mathcal{A}), \mathit{shift}(\mathcal{B}))). &
\end{array}$$

$$\begin{array}{ll}
G(\mathcal{A}) \circ G(\mathcal{B}) = & \text{(Proposition 2)} \\
max(G(\mathcal{A}), G(\mathcal{B})) = & \text{(Definition 9)} \\
max(max(\mathcal{A}_1, \mathit{shift}(\mathcal{A})), max(\mathcal{B}_1, \mathit{shift}(\mathcal{B}))) = & \text{(properties of max)} \\
max(max(\mathcal{A}_1, \mathcal{B}_1), max(\mathit{shift}(\mathcal{A}), \mathit{shift}(\mathcal{B}))). \quad \square &
\end{array}$$

Proposition 5. *The identity $\mathbf{I}$ is a fixed point of shift and G.*

Proof. We have $\mathit{shift}(\mathbf{I}) = \mathbf{I}$ and $G(\mathbf{I}) = max(\emptyset, \mathit{shift}(\mathbf{I})) = max(\emptyset, \mathbf{I}) = \mathbf{I}$. $\square$

Moreover, G is *nilpotent* in the sense that, given any $M \in \Lambda_\eta^{\mathbf{I}}$, its $\beta\eta$-normal form $\mathbf{I}$ is obtained by finitely iterating the application of G starting from the tree associated to M.

Proposition 6. *For any (finite) tree $\mathcal{A}$, there exists $k \geq 0$ such that $G^k(\mathcal{A}) = \mathbf{I}$.*

Proof. If $\mathcal{A} \neq \mathbf{I}$, then at each application of G at least one node is lost. $\square$

4.2 Application on Trees

It turns out that the application between trees can be defined in terms of generalized shift and composition as follows.

Proposition 7. *For any trees $\mathcal{A}, \mathcal{B}$, we have $\mathcal{AB} = G(\mathcal{A}) \circ \mathcal{B}$.*

Proof. The proof is by induction on the tree structure. Let $\mathcal{A}$ have subtrees $\mathcal{A}_1, \ldots, \mathcal{A}_n$ and $\mathcal{B}$ have subtrees $\mathcal{B}_1, \ldots, \mathcal{B}_m$. Let us consider the case $n > m$ (the other cases are similar). Given $m^{-1}(\mathcal{A}) = \lambda y y_1 \ldots y_n . y(M_1 y_1) \ldots (M_n y_n)$ and $m^{-1}(\mathcal{B}) = \lambda z z_1 \ldots z_m . z(N_1 z_1) \ldots (N_m z_m)$, we have:

$$\begin{aligned}
\mathcal{AB} &= m(m^{-1}(\mathcal{A}) m^{-1}(\mathcal{B})) \\
&= m(\lambda y_1 \ldots y_n . (\lambda z z_1 \ldots z_m . z(N_1 z_1) \ldots (N_m z_m))(M_1 y_1) \ldots (M_n y_n)) \\
&= m(\lambda y_1 \ldots y_n . \qquad (1) \\
&\qquad (M_1 y_1)(N_1(M_2 y_2)) \ldots (N_m(M_{m+1} y_{m+1}))(M_{m+2} y_{m+2}) \ldots (M_n y_n)).
\end{aligned}$$

As M_1 is an η-expansion of $\mathbf{I}$, M_1 has the form: $M_1 \equiv \lambda w w_1 \ldots w_k . w(M_{1_1} w_1) \ldots (M_{1_k} w_k)$, where M_{1_i} are subterms of M_1 for all i, $1 \leq i \leq k$. Let us suppose $k > n$ (the other cases are similar). By reducing the term (1), we get:

$$\begin{aligned}
&m(\lambda\, y_1 \ldots y_n w_n \ldots w_k . \\
&\qquad y_1(M_{1_1}(N_1(M_2 y_2))) \ldots (M_{1_m}(N_m(M_{m+1} y_{m+1}))) \\
&\qquad (M_{1_{m+1}}(M_{m+2} y_{m+2})) \ldots (M_{1_{n-1}}(M_n y_n))(M_{1_n} w_n) \ldots (M_{1_k} w_k)).
\end{aligned}$$

As for all i, $1 \leq i \leq m$, we have $\lambda y_{i+1} . M_{1_i}(N_i(M_{i+1} y_{i+1})) = M_{1_i} \circ (N_i \circ M_{i+1})$, by m and by inductive hypothesis we obtain:

$$\mathcal{A}_{1_i} \circ (\mathcal{B}_i \circ \mathcal{A}_{i+1}) = max(\mathcal{A}_{1_i}, max(\mathcal{B}_i, \mathcal{A}_{i+1})) = max(max(\mathcal{A}_{1_i}, \mathcal{A}_{i+1}), \mathcal{B}_i).$$

For all i, $m+1 \leq i \leq n-1$, we have $\lambda y_{i+1} . M_{1_i}(M_{i+1} y_{i+1}) = M_{1_i} \circ M_{i+1}$, thus by m and by inductive hypothesis we get:

$$\begin{aligned}
\mathcal{A}_{1_i} \circ \mathcal{A}_{i+1} &= max(\mathcal{A}_{1_i}, \mathcal{A}_{i+1}) \\
&= max(max(\mathcal{A}_{1_i}, \mathcal{A}_{i+1}), \emptyset) = max(max(\mathcal{A}_{1_i}, \mathcal{A}_{i+1}), \mathcal{B}_i).
\end{aligned}$$

For all i, $n \leq i \leq k$, we have:

$$\lambda w_i . M_{1_i} w_i = \mathcal{A}_{1_i} = max(max(\mathcal{A}_{1_i}, \emptyset), \emptyset) = max(max(\mathcal{A}_{1_i}, \mathcal{A}_{i+1}), \mathcal{B}_i).$$

Thus, $\mathcal{AB} = max(max(\mathcal{A}_1, shift(\mathcal{A})), \mathcal{B}) = G(\mathcal{A}) \circ \mathcal{B}$. □

An application of $\mathbf{I}$ yields a one-position shift, i.e. for any tree $\mathcal{A}$, $\mathcal{A}\mathbf{I} = G(\mathcal{A})$.

By the previous characterization, one gets simple proofs for the following propositions, which illustrate some structural aspects of the class $\Lambda_\eta^{\mathbf{I}}$.

Proposition 8. *Application on trees is distributive with respect to composition.*

Proof. Let $\mathcal{A},\mathcal{B},\mathcal{C}$ be trees.
(i) We prove that $\mathcal{A}(\mathcal{B}\circ\mathcal{C}) = \mathcal{AB}\circ\mathcal{AC}$.

$$\begin{array}{ll}
\mathcal{A}(\mathcal{B}\circ\mathcal{C}) = & \text{(Proposition 7)} \\
G(\mathcal{A})\circ(\mathcal{B}\circ\mathcal{C}) = & \text{(Proposition 2)} \\
max(G(\mathcal{A}), max(\mathcal{B},\mathcal{C})) = & \text{(properties of max)} \\
max(max(G(\mathcal{A}),\mathcal{B}), max(G(\mathcal{A}),\mathcal{C})) = & \text{(Proposition 2)} \\
(G(\mathcal{A})\circ\mathcal{B})\circ(G(\mathcal{A})\circ\mathcal{C}) = & \text{(Proposition 7)} \\
\mathcal{AB}\circ\mathcal{AC}. &
\end{array}$$

(ii) We prove that $(\mathcal{A}\circ\mathcal{B})\mathcal{C} = \mathcal{AC}\circ\mathcal{BC}$.

$$\begin{array}{ll}
(\mathcal{A}\circ\mathcal{B})\mathcal{C} = & \text{(Proposition 7)} \\
G(\mathcal{A}\circ\mathcal{B})\circ\mathcal{C} = & \text{(Proposition 4)} \\
(G(\mathcal{A})\circ G(\mathcal{B}))\circ\mathcal{C} = & \text{(Proposition 2)} \\
max(max(G(\mathcal{A}), G(\mathcal{B})),\mathcal{C}) = & \text{(properties of max)} \\
max(max(G(\mathcal{A}),\mathcal{C}), max(G(\mathcal{B}),\mathcal{C})) = & \text{(Proposition 2)} \\
(G(\mathcal{A})\circ\mathcal{C})\circ(G(\mathcal{B})\circ\mathcal{C}) = & \text{(Proposition 7)} \\
\mathcal{AC}\circ\mathcal{BC}. \quad \square &
\end{array}$$

Proposition 9 follows easily by definition of $T_{n,h}$ and Proposition 10 extends the closure under application to any finite set of terms.

Proposition 9. *For any non-negative n and h, the class $T_{n,h}$ is closed under the shift and the generalized shift operations. Thus, $T_{n,h}$ is closed under application.*

Proposition 10. *Let $X = \{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ be any finite set of trees. Let Y be the set of trees resulting from applying the trees in X among themselves in any way. Then, Y is finite.*

Proof. Since any finite set of terms belongs to some $T_{n,h}$, every finite set has a finite closure under application. $\square$

Proposition 12 below asserts that for any tree $\mathcal{A}$, if we iteratively apply $\mathcal{A}$ to itself, we eventually reach a tree $\mathcal{B}$ such that $\mathcal{BA} = \mathcal{B}$. Let A^k denote the k-times application of $\mathcal{A}$ to itself.

Proposition 11. *For any tree $\mathcal{A}$ and $k \geq 0$, we have $A^k = G^k(A)\circ G^{k-1}(A)\circ \ldots\circ G(A)\circ A$.*

Proof. By mathematical induction on k.
Base case $A^0 = A = G^0(A)$.
Induction case Using the induction hypothesis and Proposition 8, we get:

$$\begin{array}{l}
A^{k+1} = A^k A = (G^k(A)\circ G^{k-1}(A)\circ\ldots\circ G(A)\circ A)A = \\
G^k(A)A\circ G^{k-1}(A)A\circ\ldots\circ G(A)A\circ AA
\end{array}$$

By applying Proposition 7, Proposition 8 and Corollary 2, we obtain:

$$\begin{array}{l}
G^k(A)A\circ G^{k-1}(A)A\circ\ldots\circ G(A)A\circ AA = \\
(G^{k+1}(A)\circ A)\circ(G^k(A)\circ A)\circ\ldots\circ(G^2(A)\circ A)\circ(G(A)\circ A) = \\
(G^{k+1}(A)\circ G^k(A)\circ\ldots\circ G^2(A)\circ G(A))\circ A = \\
G^{k+1}(A)\circ G^k(A)\circ\ldots\circ G^2(A)\circ G(A)\circ A. \quad \square
\end{array}$$

Proposition 12. *For any tree $\mathcal{A}$ there exists a number $k \geq 1$ such that if we let $\mathcal{B} = A^k$, then $\mathcal{BA} = \mathcal{B}$.*

Proof. By Proposition 11 we have $A^k \subseteq A^{k+1}$. Hence, $A, AA, AAA, \ldots$ is an increasing sequence contained in $T_{n,h}$ for some non-negative n and h, where A has depth h and maximum node outdegree n. Since $T_{n,h}$ is a finite partial order with respect to set inclusion, this sequence must be eventually constant. □

5 Further Research

Current research is addressing the extension to infinite η-expansions of **I** and the corresponding infinite trees. The infinite η-expansions of **I** are all the terms whose infinite η-normal form is **I**. Using the relationships between infinite η-normal forms and Scott's model D_∞, most of the results given for the finite case can be extended to the infinite case. Whenever infinite trees are restricted to (infinite) rational trees [3], a structural description of η-expansions of **I** can be given as detailed as in the finite case.

As the shift operations are central to the general theory of formal series, a line of research that suggests itself is to investigate the possible connections between our work and that by Girard [2] on power series and the λ-calculus. Moreover, shift operations are also considered from a "dynamic" point of view [5]. It is natural to study trees and the generalized shift from the same perspective. Proposition 12 is an example of this approach and other more general results can be given along this line.

Acknowledgements

We would like to thank the anonymous referee for correcting a few errors in the previous version of the paper and for suggesting various improvements, in particular the short and elegant proof of Proposition 12.

References

1. Barendregt H. P., '*The lambda calculus. Its Syntax and Semantics*', Revised Edition, North-Holland, Amsterdam, 1984.
2. Girard J.-Y., 'Normal Functors, Power Series and λ-Calculus', Annals of Pure and Applied Logic, 37 (1988), pp. 129–177.
3. Huet G., 'Regular Böhm Trees', Mathematical Structures in Computer Science, 8 (1998), pp. 671–680.
4. Knuth D., *The Art of Computer Programming - Vol. 1 Fundamental Algorithms*', Addison-Wesley, 1968.
5. Lind D. and B. Marcus, '*An Introduction to Symbolic Dynamics and Coding*', Cambridge University Press, 1995.
6. Lothaire M., '*Combinatorics on words*', Cambridge Mathematical Library, Cambridge University Press, 1997.