On Tree Belts and Belt-Selectors

Eero Lassila*

Abstract

Such finite trees are considered that are rooted and ordered: every tree node is a descendant of a unique root node; and the direct descendant nodes of each node are linearly ordered. A rather general mechanism is presented for the specification of such two-argument functions that take any tree and any node in the tree and return such a cross-section-type subset of the nodes of the argument tree that contains the argument node itself.

1 Introduction

Computerized information processing often involves manipulation of finite strings of symbols. For example, computer programs themselves, when interpreted as data, are finite instruction sequences, and their compile-time generation and optimization can be seen as string manipulation. (In addition to atomic symbols, like the characters in a character string, structured symbols are allowed to occur in the strings considered.) We are especially interested in the case in which the lowest-level string manipulation operations available are *elementary refinements*: one symbol occurrence is replaced with an appropriate new substring, as depicted in Figure 1.

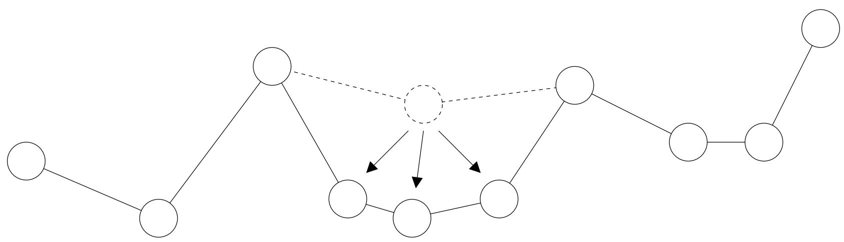

Figure 1: An elementary refinement.

By introducing an auxiliary root node, we are able to represent an arbitrary series of successive elementary refinements as a tree, as suggested in Figure 2. The particular tree in the figure is seen to record five elementary refinements. Obviously, the tree representation partially hides the actual order in which the elementary refinements have been performed.

*Helsinki University of Technology, Laboratory for Theoretical Computer Science, P. O. Box 5400, FIN-02015 HUT, Finland. *E-mail address:* eero.lassila@hut.fi

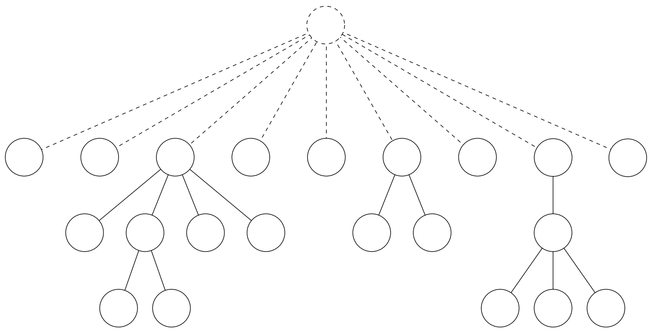

Figure 2: A series of elementary refinements represented as a tree.

For the purpose of tasks like optimizing code generation, the elementary refinements should be unboundedly context-sensitive. Nevertheless, even if we normally want to use a refinement context that is maximally *wide*, we may often be satisfied with a context that is not particularly *greedy*: it may well be appropriate to use some other cross section of the tree than the maximally deep cross section consisting of the current leaf sequence. In the following, we give three code-generation-related examples of a context selection scheme.

- *Macro processors* [2, 4, 3, 5] use a cross section whose left-hand side is maximally deep. Moreover, the left-hand side must be constituted by terminal symbols rather than by other macro calls. (Each elementary refinement, that is, the expansion of each macro call, may be sensitive to the current values of any global macro-time variables, and these values customarily propagate from left to right. In contrast, the right-hand context is usually ignored.) Therefore, the leaf processing order is strictly depth-first and left-to-right, which means that the cross section even as a whole is necessarily maximally deep. Figure 3 shows a sample tree at the unique moment when the leaf marked with a black ring is processable; the refinement context is indicated by white rings, and the checkered nodes in the left-hand context correspond to terminal symbols.
- *Parametric Lindenmayer systems* [20, 19, 14, 13, 18, 17] output sequences of drawing commands and thus indirectly produce high-quality graphics. They are perhaps the best-known example of application-oriented extensions to the basic *Lindenmayer system* model [7, 21, 8]. With Lindenmayer systems (whether parametric or not), the tree nodes are processed in a generation-by-generation fashion, and the “horizontal” cross section constituted by all the nodes in the current generation serves as the refinement context. In practice, the nodes within each single generation may well be processed sequentially, rather than simultaneously, but it should be noticed that the desired horizontal cross section then differs from the maximally deep cross section. The two trees of Figure 4 depict only the two extremes among the possible processing moments for the leaf with the black ring.
- Figure 5 illustrates a context selection scheme that we have earlier em-

ployed in a simplistic prototype [9, 10, 11], called *ReFlEx*, of a still nonexistent tool proposed by us for optimizing machine-level code generation [1, 12, 16, 15]. Now the refinement context is constituted by the least deep cross section possible. Such ungreediness is rewarded as the leaf processing order becomes completely free. The two trees of Figure 5 again depict only two of the possible processing moments for the leaf with the black ring. (The moment depicted on the left-hand side of the figure is of course the earliest possible.)

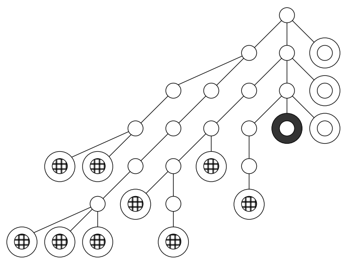

Figure 3: The refinement context used by macro processors.

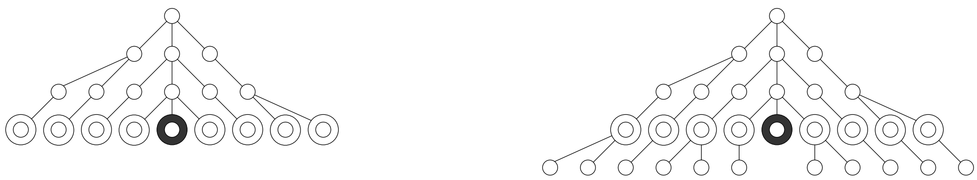

Figure 4: The refinement context used by Lindenmayer systems.

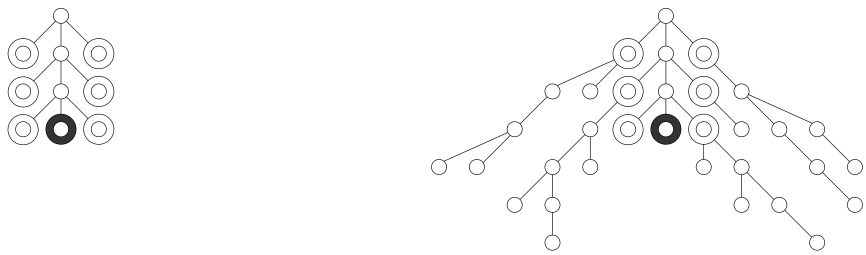

Figure 5: The refinement context used by the ReFlEx prototype.

Our present goal is to find a general mechanism with which one can conveniently specify the particular cross section to be used as the refinement context. On one hand, the mechanism should be expressive by imposing only few constraints on the choice of the cross section; on the other hand, well-designed constraints would probably be helpful by making the consequences of the choice more easily tractable. In the following Section 2, we formulate a simple constraint on cross section selection, and in the final Section 3, we then describe such a cross section specification mechanism that exactly matches the formulated constraint. We suggest that the single constraint is not only simple but also a practical one, even if we do not, as yet, try to provide any concrete evidence for this claim.

2 Definition of a belt-selector

2.1 Trees

A *tree* consists of a finite and non-zero number of *nodes*. Figure 6 depicts a sample tree, which we call A. (By convention, 'node $\mathtt{a}_3$', for instance, refers to the unique node in tree A labeled as '3'. The reason why node $\mathtt{a}_9$ is distinguished in Figure 6 is that we have, more or less arbitrarily, chosen it to have an important role in some examples below.) Each tree is *rooted* and *ordered*, as will be explained next. (This denotation of the term 'tree' adopted here is a standard one within the formal language community; see [6], for instance.)

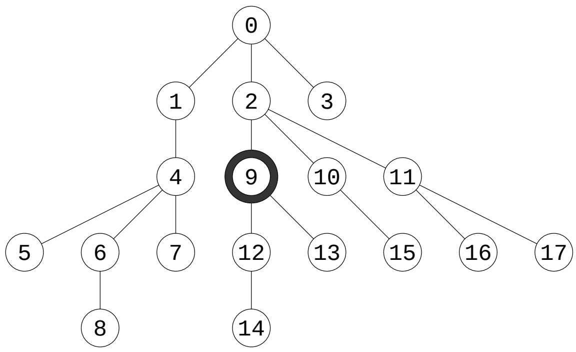

Figure 6: Tree A.

The rootedness means that each tree has exactly one *root*: the root of A is $\mathtt{a}_0$. Every tree node different from the root has exactly one *father* in the tree: the father of $\mathtt{a}_9$ is $\mathtt{a}_2$, and so $\mathtt{a}_9$ (like $\mathtt{a}_{10}$ and $\mathtt{a}_{11}$) is a *son* of $\mathtt{a}_2$. Such tree nodes that have no sons are *leaves* of the tree: A has a total of nine leaves.

We say that a given node n' is an *ancestor* of a given node n if the pair $\langle n', n\rangle$ belongs to the reflexive-transitive closure of the binary 'is a father of' relation. (By 'a pair' we always mean an ordered pair.) Hence the ancestors of $\mathtt{a}_9$ are $\mathtt{a}_9$, $\mathtt{a}_2$, and $\mathtt{a}_0$. If n' is an ancestor of n, then n is a *descendant* of n'. Moreover, n' is a *proper ancestor* of n, and n is correspondingly a *proper descendant* of n', if n' is an ancestor of n and $n' \neq n$.

The orderedness means that there is a total "left-to-right" order among the sons of any given tree node. If two distinct nodes have the same father, then one of them is a *left-brother* of the other, and the latter is a *right-brother* of the former. For instance, $\mathtt{a}_9$ has right-brothers $\mathtt{a}_{10}$ and $\mathtt{a}_{11}$, and $\mathtt{a}_{10}$ has $\mathtt{a}_9$ as a left-brother and $\mathtt{a}_{11}$ as a right-brother.

We say that a given node n' is a *left-relative* of a given node n if there are such nodes n'_0 and n_0 in the tree that n'_0 is a left-brother of n_0, n' is a descendant of n'_0, and n is a descendant of n_0. If n' is a left-relative of n, then n is a *right-relative* of n'. For instance, $\mathtt{a}_8$ is a left-relative of $\mathtt{a}_9$ (since $\mathtt{a}_1$ is a left-brother of $\mathtt{a}_2$), and $\mathtt{a}_9$ is a right-relative of $\mathtt{a}_8$.

Note that for each two distinct nodes n^* and n^{**} in any given tree, exactly one of the following statements holds: n^* is a proper ancestor of n^{**}; n^* is a proper descendant of n^{**}; n^* is a left-relative of n^{**}; or n^* is a right-relative of n^{**}.

2.2 Angles between tree nodes

Each tree node has a unique *degree*, and each pair of tree nodes has a unique *angle*.

Definition 1. The ***degree*** of a given tree node is the number of its proper ancestors.

Definition 2. The ***angle*** of a given tree node pair $\langle n, n' \rangle$ is denoted as $\sphericalangle(n, n')$ and defined as the unique integer triple $\langle i, d, j \rangle$ that meets the following conditions.

1. i [respectively, j] is the difference of the degrees of n [respectively, n'] and the one of the common ancestors of n and n' that has the greatest degree.
2. $d = 0$ if one of n and n' is an ancestor of the other, $d = -1$ if n' is a left-relative of n, and $d = 1$ if n' is a right-relative of n.

Note that $\sphericalangle(n, n') = \langle i, d, j \rangle$ always implies $\sphericalangle(n', n) = \langle j, -d, i \rangle$. Table 1 lists the angles from node a_9 to the other nodes of our sample tree $\mathtt{A}$.

n	$\sphericalangle(\mathsf{a}_9, n)$	n	$\sphericalangle(\mathsf{a}_9, n)$	n	$\sphericalangle(\mathsf{a}_9, n)$
a_0	$\langle 2, 0, 0 \rangle$	a_6	$\langle 2, -1, 3 \rangle$	a_{12}	$\langle 0, 0, 1 \rangle$
a_1	$\langle 2, -1, 1 \rangle$	a_7	$\langle 2, -1, 3 \rangle$	a_{13}	$\langle 0, 0, 1 \rangle$
a_2	$\langle 1, 0, 0 \rangle$	a_8	$\langle 2, -1, 4 \rangle$	a_{14}	$\langle 0, 0, 2 \rangle$
a_3	$\langle 2, 1, 1 \rangle$	a_9	$\langle 0, 0, 0 \rangle$	a_{15}	$\langle 1, 1, 2 \rangle$
a_4	$\langle 2, -1, 2 \rangle$	a_{10}	$\langle 1, 1, 1 \rangle$	a_{16}	$\langle 1, 1, 2 \rangle$
a_5	$\langle 2, -1, 3 \rangle$	a_{11}	$\langle 1, 1, 1 \rangle$	a_{17}	$\langle 1, 1, 2 \rangle$

Table 1: The angles from node a_9 to the other nodes of tree $\mathtt{A}$.

Definition 3. A given integer triple $\langle i, d, j \rangle$ is a ***link*** if there is such a tree node pair $\langle n, n' \rangle$ that $\sphericalangle(n, n') = \langle i, d, j \rangle$.

Note that $\langle i, d, j \rangle$ is a link if and only if all the following conditions are met: $i \geq 0$, $d \in \{-1, 0, 1\}$, $j \geq 0$, and $d = 0 \Leftrightarrow i \times j = 0$.

2.3 Belts and belt-selectors

Definition 4. A ***belt*** of a tree is any such subset of the tree nodes that each leaf of the tree has exactly one ancestor in the subset.

In any tree, both the set consisting of the sole root and the set consisting of all the leaves are belts. For more specific examples, Table 2 lists all such belts of our sample tree $\mathtt{A}$ that contain node a_9.

Definition 5. A ***belt-provider*** is any such two-argument function that takes any tree and any node in the tree and returns one such belt of the tree that contains the node.

$\{\mathtt{a}_1\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{10}, \mathtt{a}_{11}, \mathtt{a}_3\}$	$\{\mathtt{a}_5, \mathtt{a}_6, \mathtt{a}_7\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{10}, \mathtt{a}_{11}, \mathtt{a}_3\}$
$\{\mathtt{a}_1\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{15}, \mathtt{a}_{11}, \mathtt{a}_3\}$	$\{\mathtt{a}_5, \mathtt{a}_6, \mathtt{a}_7\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{15}, \mathtt{a}_{11}, \mathtt{a}_3\}$
$\{\mathtt{a}_1\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{10}, \mathtt{a}_{16}, \mathtt{a}_{17}, \mathtt{a}_3\}$	$\{\mathtt{a}_5, \mathtt{a}_6, \mathtt{a}_7\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{10}, \mathtt{a}_{16}, \mathtt{a}_{17}, \mathtt{a}_3\}$
$\{\mathtt{a}_1\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{15}, \mathtt{a}_{16}, \mathtt{a}_{17}, \mathtt{a}_3\}$	$\{\mathtt{a}_5, \mathtt{a}_6, \mathtt{a}_7\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{15}, \mathtt{a}_{16}, \mathtt{a}_{17}, \mathtt{a}_3\}$
$\{\mathtt{a}_4\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{10}, \mathtt{a}_{11}, \mathtt{a}_3\}$	$\{\mathtt{a}_5, \mathtt{a}_8, \mathtt{a}_7\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{10}, \mathtt{a}_{11}, \mathtt{a}_3\}$
$\{\mathtt{a}_4\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{15}, \mathtt{a}_{11}, \mathtt{a}_3\}$	$\{\mathtt{a}_5, \mathtt{a}_8, \mathtt{a}_7\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{15}, \mathtt{a}_{11}, \mathtt{a}_3\}$
$\{\mathtt{a}_4\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{10}, \mathtt{a}_{16}, \mathtt{a}_{17}, \mathtt{a}_3\}$	$\{\mathtt{a}_5, \mathtt{a}_8, \mathtt{a}_7\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{10}, \mathtt{a}_{16}, \mathtt{a}_{17}, \mathtt{a}_3\}$
$\{\mathtt{a}_4\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{15}, \mathtt{a}_{16}, \mathtt{a}_{17}, \mathtt{a}_3\}$	$\{\mathtt{a}_5, \mathtt{a}_8, \mathtt{a}_7\} \cup \{\mathtt{a}_9\} \cup \{\mathtt{a}_{15}, \mathtt{a}_{16}, \mathtt{a}_{17}, \mathtt{a}_3\}$

Table 2: The sixteen belts of tree $\mathtt{A}$ that contain node $\mathtt{a}_9$.

Definition 6. A given belt-provider s is ***uniangular***, and hence called a ***belt-selector***, if it meets the following condition.

- Let X_1 and X_2 be two trees containing nodes n_1 and n_2, respectively. Suppose that X_1 has a leaf n_1', and let the unique ancestor of n_1' that belongs to $s(X_1, n_1)$ be denoted as n_1''. Similarly, suppose that X_2 has a leaf n_2', and let the unique ancestor of n_2' that belongs to $s(X_2, n_2)$ be denoted as n_2''. Then $\triangleleft(n_1, n_1') = \triangleleft(n_2, n_2')$ implies $\triangleleft(n_1, n_1'') = \triangleleft(n_2, n_2'')$.

2.4 An example

Before a more thorough analysis in Section 3, let us briefly look at some consequences of the uniangularity requirement. Specifically, we will consider some belts of tree $\mathtt{B}$, on the left-hand side of Figure 7, and ask whether there is such a belt-selector that is able to select the particular belt for node $\mathtt{b}_4$. Of course, any belt selected must contain $\mathtt{b}_4$ itself, and we restrict ourselves to only three such belts.

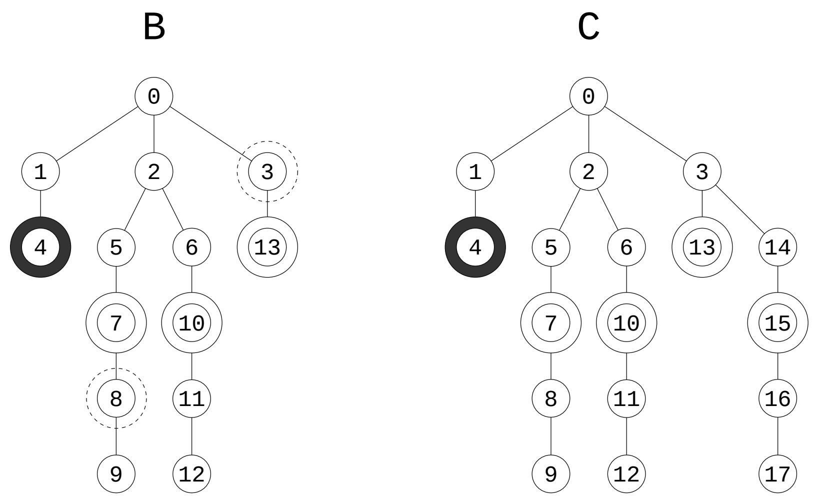

Figure 7: Trees $\mathtt{B}$ and $\mathtt{C}$.

Case 1: $\{\mathtt{b}_4, \mathtt{b}_7, \mathtt{b}_{10}, \mathtt{b}_{13}\}$. In Section 3 below, we will prove that there exists such a belt-selector s for which $s(\mathtt{B}, \mathtt{b}_4)$ equals this belt.

Case 2: $\{\mathtt{b}_4, \mathtt{b}_8, \mathtt{b}_{10}, \mathtt{b}_{13}\}$. (This is the belt of case (1) with $\mathtt{b}_7$ replaced by its single son $\mathtt{b}_8$.) It is readily seen that there exists no such belt-selector s^* for which $s^*(\mathtt{B}, \mathtt{b}_4)$ equals this belt: uniangularity would otherwise be violated, since $\lhd(\mathtt{b}_4, \mathtt{b}_9) = \lhd(\mathtt{b}_4, \mathtt{b}_{12})$ but $\lhd(\mathtt{b}_4, \mathtt{b}_8) \neq \lhd(\mathtt{b}_4, \mathtt{b}_{10})$.

Case 3: $\{\mathtt{b}_4, \mathtt{b}_7, \mathtt{b}_{10}, \mathtt{b}_3\}$. (This is the belt of case (1) with the brotherless $\mathtt{b}_{13}$ replaced by its father $\mathtt{b}_3$.) Again, there exists no such belt-selector s^* for which $s^*(\mathtt{B}, \mathtt{b}_4)$ equals this belt. Our following simple proof is by contradiction; suppose for a moment that such s^* exists. Consider tree $\mathtt{C}$, on the right-hand side of Figure 7, which is otherwise fully isomorphic to tree $\mathtt{B}$ but has a single additional branch consisting of nodes $\mathtt{c}_{14}$, $\mathtt{c}_{15}$, $\mathtt{c}_{16}$, and $\mathtt{c}_{17}$. Because $\lhd(\mathtt{b}_4, \mathtt{b}_{12}) = \lhd(\mathtt{c}_4, \mathtt{c}_{17})$ and $\lhd(\mathtt{b}_4, \mathtt{b}_{10}) = \lhd(\mathtt{c}_4, \mathtt{c}_{15})$, uniangularity requires that $\mathtt{c}_{15}$ belongs to $s^*(\mathtt{C}, \mathtt{c}_4)$. This forces us to include even $\mathtt{c}_{13}$ in $s^*(\mathtt{C}, \mathtt{c}_4)$. The contradiction desired is now that $\lhd(\mathtt{b}_4, \mathtt{b}_{13}) = \lhd(\mathtt{c}_4, \mathtt{c}_{13})$ but obviously $\lhd(\mathtt{b}_4, \mathtt{b}_3) \neq \lhd(\mathtt{c}_4, \mathtt{c}_{13})$, and so uniangularity is violated.

3 More explicit characterization of belt-selectors

We let $\mathbb{N}^+$ denote the set $\{1, 2, \ldots\}$ of all positive integers. The 'less-than' relation is extended from $\mathbb{N}^+$ to $\mathbb{N}^+ \cup \{\infty\}$ simply by stating that $k < \infty$ for every $k \in \mathbb{N}^+$ and requiring that the relation remains irreflexive and transitive.

Definition 7. A ***comb*** is any function from $\mathbb{N}^+ \times \{-1, 1\}$ to $\mathbb{N}^+ \cup \{\infty\}$.

Definition 8. A given comb f is a ***characteristic comb*** of a given belt-provider s if for every tree X, for every node n of X, and for every leaf n' of X, the following conditions are met when $\lhd(n, n')$ is denoted as $\langle i, d, j\rangle$ and the unique ancestor of n' that belongs to $s(X, n)$ is denoted as n''.

1. Suppose $d \neq 0$ and $j \leq f(i, d)$. Then $n'' = n'$.
2. Suppose $d \neq 0$ and $j > f(i, d)$. Then n'' is the unique proper ancestor of n' for which $\lhd(n, n'') = \langle i, d, f(i, d)\rangle$.

Let us tentatively try to associate each one of the three belt selection schemes depicted in Figures 3, 4, and 5 with a characteristic comb. Consider any $i \in \mathbb{N}^+$. Macro processors seem to require that $f(i, -1) = \infty$ but $f(i, 1) = 1$; Lindenmayer systems and the ReFlEx prototype seem to require that $f(i, -1) = f(i, 1) = i$ and $f(i, -1) = f(i, 1) = 1$, respectively.

Notation 9. The set of belt-providers [respectively, of belt-selectors, of combs] is denoted as $\mathcal{P}$ [respectively, $\mathcal{S}$, $\mathcal{F}$].

Our following main result indicates that the 'is a characteristic comb of' relation is actually a one-to-one correspondence between combs and belt-selectors. In particular, the theorem implies that the set of belt-selectors is non-empty, since the set of combs is obviously non-empty. Notice also that it now becomes evident that there does exist a belt-selector realizing case (1) of the example in Section 2.4: we may choose any belt-selector whose characteristic comb f has the property that $f(2, 1) = 3$.

Theorem 10. Let R denote the set of all such members $\langle s, f\rangle$ of $\mathcal{P} \times \mathcal{F}$ that f is a characteristic comb of s. Then $R \subseteq \mathcal{S} \times \mathcal{F}$, and moreover, R is a bijective function from $\mathcal{S}$ to $\mathcal{F}$.

We will be able to prove Theorem 10 after first obtaining some auxiliary results.

Lemma 11. Each comb is a characteristic comb of at least one belt-provider.

Proof. Let f, X, and n be a given comb, a given tree, and a given node of the tree, respectively. We define two subsets N_1 and N_2 of the nodes of X in the following incremental fashion.

1. $N_1 = \{n\} \cup N'$ when N' consists of all such nodes n' of X that $\triangleleft(n, n') = \langle i, d, f(i, d)\rangle$ for some $\langle i, d\rangle \in \mathbb{N}^+ \times \{-1, 1\}$.
2. $N_2 = N_1 \cup N^*$ when N^* consists of all such leaves of X that have no ancestor in N_1.

By the two definitions above, neither set N_1 nor set N_2 contains such a node that is a proper ancestor of some other node in the same set. Consequently, N_2 is easily seen to be such a belt that contains n. Hence, the above two-stage node set construction procedure serves as a belt-provider, and it is straightforward to verify from Definition 8 that f is indeed a characteristic comb of that belt-provider. $\square$

Lemma 12. Each comb is a characteristic comb of at most one belt-provider.

Proof. (By contradiction.)

Assume that a comb f is a characteristic comb of two distinct belt-providers s_1 and s_2. Because $s_1 \neq s_2$, there must be a tree X with such a node n that $s_1(X, n) \neq s_2(X, n)$.

However, Definition 8 picks for each leaf a unique ancestor that must belong to the belt returned by any such belt-provider whose characteristic comb is f. (For each such leaf of X that is a descendant of n, the unique ancestor is obviously n itself, already by the definition of a belt-provider.) Hence, we must have $s_1(X, n) = s_2(X, n)$, which is a contradiction. $\square$

Lemma 13. Each belt-provider has at most one characteristic comb.

Proof. (By contradiction.)

Assume that a belt-provider s has two distinct characteristic combs f_1 and f_2. Because $f_1 \neq f_2$, there must be such $\langle i, d\rangle \in \mathbb{N}^+ \times \{-1, 1\}$ that $f_1(i, d) \neq f_2(i, d)$. Without loss of generality, we may further assume $f_1(i, d) < f_2(i, d)$.

We clearly have $f_1(i, d) < \infty$. Consider then any tree X with such nodes n and n'' that $\triangleleft(n, n'') = \langle i, d, f_1(i, d)\rangle$ and n'' is a father of some leaf n' of X.

First, since f_1 is a characteristic comb of s, condition (2) of Definition 8 requires that $n'' \in s(X, n)$. Second, since f_2 is a characteristic comb of s, condition (1) of Definition 8 requires that $n' \in s(X, n)$. This is obviously a contradiction. $\square$

Lemma 14. Let R^* denote the set of all such members $\langle f, s\rangle$ of $\mathcal{F} \times \mathcal{P}$ that f is a characteristic comb of s. Then R^* is an injective function from $\mathcal{F}$ to $\mathcal{P}$.

Proof. Lemmas 11 and 12 together imply that the specified R^* is a function from $\mathcal{F}$ to $\mathcal{P}$, and Lemma 13 moreover implies that the function is injective. $\square$

Lemma 15. If a belt-provider has a characteristic comb, then the belt-provider is a belt-selector.

Proof. Suppose that a belt-provider s has a characteristic comb f. Let X_1 and X_2 be given trees, let n_1 and n_2 be given nodes of X_1 and X_2, respectively, and let n_1' and n_2' be given leaves of X_1 and X_2, respectively. Let n_1'' denote the unique ancestor of n_1' that belongs to $s(X_1, n_1)$, and let n_2'' similarly denote the unique ancestor of n_2' that belongs to $s(X_2, n_2)$. By Definition 6, it is now sufficient to demonstrate that $\sphericalangle(n_1, n_1') = \sphericalangle(n_2, n_2')$ implies $\sphericalangle(n_1, n_1'') = \sphericalangle(n_2, n_2'')$.

So we assume that $\sphericalangle(n_1, n_1')$ and $\sphericalangle(n_2, n_2')$ are both equal to some link $\langle i, d, j\rangle$, and try to show that $\sphericalangle(n_1, n_1'') = \sphericalangle(n_2, n_2'')$. We divide the task into three cases.

- Suppose $d = 0$. By the definition of a belt-provider, we now have $n_1'' = n_1$ and $n_2'' = n_2$, and so indeed $\sphericalangle(n_1, n_1'') = \langle 0, 0, 0\rangle = \sphericalangle(n_2, n_2'')$.
- Suppose $d \neq 0$ and $j \leq f(i, d)$. By condition (1) of Definition 8, we now have $n_1'' = n_1'$ and $n_2'' = n_2'$, and so indeed $\sphericalangle(n_1, n_1'') = \langle i, d, j\rangle = \sphericalangle(n_2, n_2'')$.
- Suppose $d \neq 0$ and $j > f(i, d)$. By condition (2) of Definition 8, we now indeed have $\sphericalangle(n_1, n_1'') = \langle i, d, f(i, d)\rangle = \sphericalangle(n_2, n_2'')$.

□

Lemma 16. Let n_1 and n_1'' be nodes in a tree X_1, and suppose that n_1'' is not a leaf. Similarly, let n_2 and n_2'' be nodes in a tree X_2, and suppose that n_2'' is not a leaf. Suppose also $\sphericalangle(n_1, n_1'') = \sphericalangle(n_2, n_2'')$. Then for any belt-selector s, we have $n_1'' \in s(X_1, n_1) \Leftrightarrow n_2'' \in s(X_2, n_2)$.

Proof. We suppose exactly what is suggested above in the text of the lemma and set out to verify that for any s, it is the case that $n_1'' \in s(X_1, n_1) \Leftrightarrow n_2'' \in s(X_2, n_2)$.

As depicted in Figure 8, we let n_1' [respectively, n_2'] denote any such leaf of X_1 [respectively, X_2] that is also a proper descendant of n_1'' [respectively, n_2'']. (Because neither n_1'' nor n_2'' is a leaf, such n_1' and n_2' do exist.) It is now easy to see that there exists a tree X_0, sketched on the right-hand side of Figure 8, with such nodes n_0, n_0'', $n_{0,1}'$, and $n_{0,2}'$ that meet the following conditions.

1. n_0'' is not a leaf.
2. $\sphericalangle(n_0, n_0'') = \sphericalangle(n_1, n_1'')$.
3. $\sphericalangle(n_0, n_0'') = \sphericalangle(n_2, n_2'')$. (This is a trivial consequence of the previous condition, since it is supposed that $\sphericalangle(n_1, n_1'') = \sphericalangle(n_2, n_2'')$.)
4. Both $n_{0,1}'$ and $n_{0,2}'$ are such leaves that are proper descendants of n_0''.
5. $\sphericalangle(n_0, n_{0,1}') = \sphericalangle(n_1, n_1')$.
6. $\sphericalangle(n_0, n_{0,2}') = \sphericalangle(n_2, n_2')$.

First, by the uniangularity stated in Definition 6, conditions (5) and (2) above together ensure that $n_0'' \in s(X_0, n_0) \Leftrightarrow n_1'' \in s(X_1, n_1)$. Second, again by uniangularity, conditions (6) and (3) ensure that $n_0'' \in s(X_0, n_0) \Leftrightarrow n_2'' \in s(X_2, n_2)$. The claim now trivially follows from the combination of these two facts. □

Definition 17. Let s be a given belt-selector, and let L denote the link set that consists of every such link $\langle i^*, d^*, j^*\rangle$ that meets the following condition: there are such a tree X and such nodes n and n'' of X that $\sphericalangle(n, n'') = \langle i^*, d^*, j^*\rangle$, $n'' \in s(X, n)$, and n'' is not a leaf. We say that a given comb f is a ***natural comb*** of s if the following conditions are met for every $\langle i, d\rangle \in \mathbb{N}^+ \times \{-1, 1\}$.

1. $f(i, d) = \infty$ if and only if $\langle i, d, j\rangle \notin L$ for every $j \in \mathbb{N}^+$.
2. If $f(i, d) < \infty$, then $\langle i, d, f(i, d)\rangle \in L$.

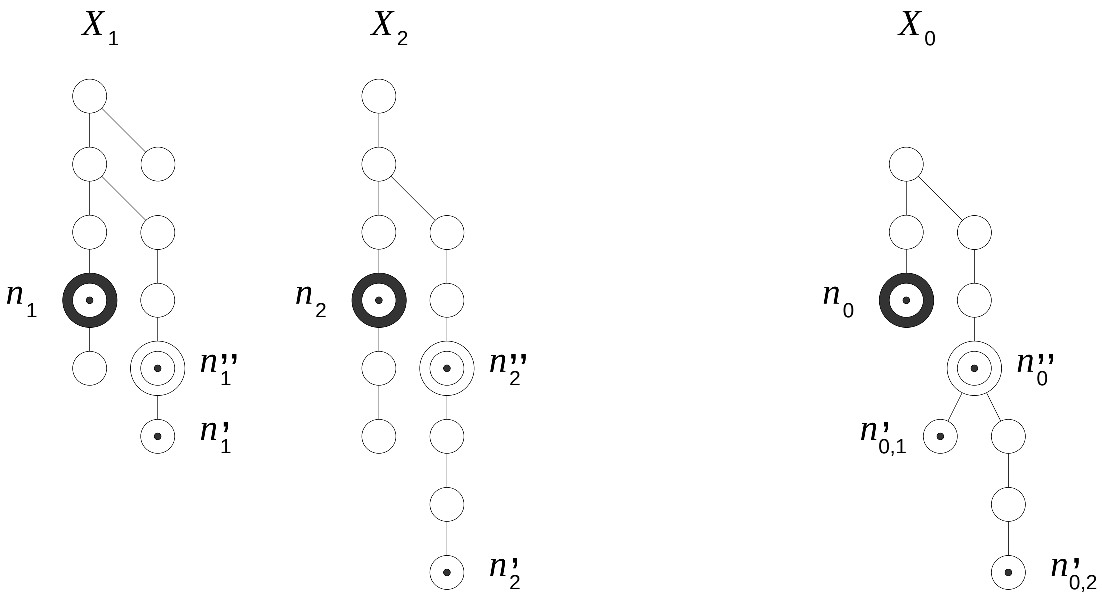

Figure 8: Proving Lemma 16.

Lemma 18. Each belt-selector has at least one natural comb.

Proof. Obvious from Definition 17. (Notice that for any given link set L, even if it is different from the particular link set constructed in Definition 17, there is at least one such comb f that meets the two conditions (1) and (2) of Definition 17 for every $\langle i, d\rangle \in \mathbb{N}^+ \times \{-1, 1\}$.) $\square$

Lemma 19. If a belt-selector has a natural comb, then the natural comb is also a characteristic comb of the belt-selector.

Proof. Let a comb f be a natural comb of a belt-selector s. To find out whether f is necessarily also a characteristic comb of s, we set out to examine whether the conditions of Definition 8 are met for a given tree X, for a given node n of X, and for a given leaf n' of X. We denote $\lhd(n, n')$ as $\langle i, d, j\rangle$ and the unique ancestor of n' that belongs to $s(X, n)$ as n''. The examination may be divided into the following three cases. (Of the two conditions of Definition 8, condition (1) is covered by cases (1) and (2) below, and condition (2) is covered by case (3).)

1. Suppose $d \neq 0$ and $j < f(i, d) = \infty$. By condition (1) of Definition 17, we must have $n'' = n'$. Hence, the appropriate condition (1) of Definition 8 is indeed met.
2. Suppose $d \neq 0$ and $j \leq f(i, d) < \infty$. By condition (2) of Definition 17, there are such a tree X_0 and such nodes n_0 and n_0'' of X_0 that $\lhd(n_0, n_0'') = \langle i, d, f(i, d)\rangle$ and $n_0'' \in s(X_0, n_0)$. (Here we need not be interested in whether n_0'' is or is not a leaf.) This means that for any proper ancestor n^* of n', there is such a proper ancestor n_0^* of n_0'' that the following conditions are met.
   - $\lhd(n, n^*) = \lhd(n_0, n_0^*)$.
   - Neither n^* nor n_0^* is a leaf.
   - $n_0^* \notin s(X_0, n_0)$.

Lemma 16 now implies that $n'' \neq n^*$ for any n^*, and so we must have $n'' = n'$. Hence, the appropriate condition (1) of Definition 8 is indeed met.
3. Suppose $d \neq 0$ and $f(i,d) < j < \infty$. By condition (2) of Definition 17, there are, again, such a tree X_0 and such nodes n_0 and n''_0 of X_0 that $\lhd(n_0, n''_0) = \langle i, d, f(i,d) \rangle$, $n''_0 \in s(X_0, n_0)$, and n''_0 is not a leaf. Lemma 16 now implies that n'' must be the unique proper ancestor (which obviously cannot be a leaf) of n' for which $\lhd(n, n'') = \lhd(n_0, n''_0) = \langle i, d, f(i,d) \rangle$. Hence, the appropriate condition (2) of Definition 8 is indeed met.

□

Proof of Theorem 10. By Lemmas 18 and 19, every belt-selector has a characteristic comb; and by Lemma 15, no such belt-provider that is not a belt-selector has a characteristic comb. Hence, a belt-provider has a characteristic comb if and only if the belt-provider is a belt-selector, and so the claim now follows from Lemma 14. □

References

[1] A. V. Aho, R. Sethi, and J. D. Ullmann. *Compilers: Principles, Techniques, and Tools.* Addison-Wesley, 1986.

[2] P. J. Brown. A survey of macro processors. *Annual Review in Automatic Programming*, vol. 6, part 2, pp. 37–88. Pergamon Press, 1969.

[3] P. J. Brown. *Macro Processors and Techniques for Portable Software.* Wiley, 1974.

[4] M. Campbell-Kelly. *An Introduction to Macros.* Macdonald, London (UK), 1973.

[5] A. J. Cole. *Macro Processors* (second edition). Cambridge University Press, 1981.

[6] F. Gécseg and M. Steinby. Tree languages. In [22], vol. 3, pp. 1–68.

[7] G. T. Herman and G. Rozenberg. *Developmental Systems and Languages.* North-Holland, 1975.

[8] L. Kari, G. Rozenberg, and A. Salomaa. L systems. In [22], vol. 1, pp. 253–328.

[9] E. Lassila. ReFlEx—an experimental tool for special-purpose processor code generation. Report B15, Digital Systems Laboratory, Helsinki University of Technology. Espoo (Finland), March 1996.

[10] E. Lassila. A macro expansion approach to embedded processor code generation. *Proceedings of the 22nd EUROMICRO Conference*, pp. 136–142. IEEE Computer Society Press, 1996.

[11] E. Lassila. Towards optimizing code generation by domain-sensitive macro expansion. Report A42, Digital Systems Laboratory, Helsinki University of Technology. Espoo (Finland), January 1997.

[12] P. Marwedel and G. Goossens (eds.). *Code Generation for Embedded Processors.* Kluwer, 1995.

[13] R. Měch. Modeling and simulation of the interaction of plants with the environment using L-systems and their extensions. Ph. D. thesis, Department of Computer Science, The University of Calgary. Calgary (Alberta, Canada), November 1997.

[14] R. Měch and P. Prusinkiewicz. Visual models of plants interacting with their environment. *Proceedings of SIGGRAPH 96 Conference*, pp. 397–410. ACM SIGGRAPH, 1996.

[15] R. Morgan. *Building an Optimizing Compiler*. Butterworth-Heinemann, 1998.

[16] S. S. Muchnick. *Advanced Compiler Design and Implementation*. Morgan Kaufmann, 1997.

[17] P. Prusinkiewicz. Simulation modeling of plants and plant ecosystems. *Communications of the ACM*, vol. 43, no. 7, pp. 84–93. 2000.

[18] P. Prusinkiewicz, M. Hammel, J. Hanan, and R. Měch. Visual models of plant development. In [22], vol. 3, pp. 535–597.

[19] P. Prusinkiewicz, M. James, and R. Měch. Synthetic topiary. *Proceedings of SIGGRAPH 94 Conference*, pp. 351–358. ACM SIGGRAPH, 1994.

[20] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer, 1990.

[21] G. Rozenberg and A. Salomaa. *The Mathematical Theory of L Systems*. Academic Press, 1980.

[22] G. Rozenberg and A. Salomaa (eds.). *Handbook of Formal Languages* (vols. 1–3). Springer, 1997.