Lattices of labelled ordered trees (I)

Nicolae Ţăndăreanu

Abstract

In this paper we introduce a kind of labelled ordered tree. We obtain a lattice of such trees considered up to a natural equivalence. In the second part of this paper, which is in preparation, we study several algebraic properties of this lattice ([9]). These results will be used in a forthcoming paper to study the properties of the answer function for a knowledge representation and reasoning system based on inheritance property.

Keywords: labelled tree, lattice, knowledge representation

AMS Subject Classification: 06B05, 06A12

1 Introduction

The concepts of *object, frame*, and *inheritance* are used and interpreted in various ways. In the most cases the concept of object is used in conjunction with the object oriented databases ([3]) or the object oriented problem solving ([1]). The concept of frame is encountered in knowledge representation, but it is applied in various domains such as the theory of space-filling curves and Lindenmayer systems ([4]). There are several major implications of these concepts in knowledge representation languages, systems of knowledge processing and image synthesis. There are today a lot of implemented systems using these concepts. Most of them are described in literature (for example CAKE [5], NETL [7] etc) and other systems can be taken from Internet.

Knowledge representation is a major research domain in artificial intelligence. The study of the knowledge representation and reasoning systems ([8], [10]) is a main direction of research. An interesting problem related to this subject is the study of the computability of the answer function for such systems. By the results specified in the current paper we inaugurate a research line to study the properties of the answer function of a knowledge system based on inheritance property. The final paper will specify the computability of this function. Moreover, an algorithm will be given such that the answers *unknown* and *undefined* are identified at once.

All properties specified in this paper are proved in a separate appendix.

2 Pairwise mappings and ω-labelled trees

In this section we introduce the concept of ω-labelled tree, which is based on the concept of pairwise mapping.

A *directed ordered graph* is a pair $G = (A, R)$, where

- A is a finite set of elements, which are called *nodes*
- R is a nonempty finite set of elements of the form $[(i, i_1), \ldots, (i, i_n)]$, where $n \geq 1$ and $i, i_1, \ldots, i_n \in A$
- R satisfies the following condition: if $[(i, i_1), \ldots, (i, i_n)] \in R$ and $[(j, j_1), \ldots, (j, j_s)] \in R$ then $i \neq j$

We observe that for an element $[(i, i_1), \ldots, (i, i_n)] \in R$ we may have $i_j = i_k$ for some $j \neq k$.

We can represent a directed ordered graph as follows. We represent, as usual, a node of the graph by a point. If $[(i, i_1), \ldots, (i, i_n)] \in R$ then we draw *an arc* from the node i to the node i_j for every $j \in \{1, \ldots n\}$. The elements $i_1, \ldots, i_n$ are called *the direct descendants* of i. We shall consider that all the direct descendants of i are linear ordered and the order is given by the place of i_j in the element $[(i, i_1), \ldots, (i, i_n)]$.

If $G = (A, R)$ is a directed ordered graph then we associate it a *directed graph* $G' = (A, R')$, where

$$R' = \{(i, j) \mid \exists [(i, i_1), \ldots, (i, i_n)] \in R, \exists r \in \{1, \ldots, n\} : j = i_r\}$$

An *ordered tree* is a directed ordered graph $G = (A, R)$ satisfying the following properties:

- if $[(i, i_1), \ldots, (i, i_n)] \in R$ then $i_j \neq i_r$ for $j \neq r$
- the associated graph G' is a tree

We consider a finite set L and a decomposition $L = N_L \cup T_L$, where $N_L \cap T_L = \emptyset$. The elements of N_L are called *nonterminal labels* and those of T_L are called *terminal labels*. The elements of L are called *labels*.

Definition 2.1 *Let $L = N_L \cup T_L$ be a set of labels. A pairwise mapping ω on L is a mapping $\omega : N_L \longrightarrow \bigcup_{k \geq 1} \{k\} \times L^k$. For each $b \in N_L$ we shall denote $\omega(b) = (\omega_1(b), \omega_2(b))$ and therefore $\omega_1(b)$ gives us the number of components for $\omega_2(b)$.*

This means that for every $b \in N_L$ we have $\omega_2(b) \in L^{\omega_1(b)}$, that is $\omega_2(b) = (c_1, \ldots, c_s)$, where $s = \omega_1(b)$ and $c_i \in L$ for $i \in \{1, \ldots, s\}$.

Definition 2.2 *Let $\omega : N_L \longrightarrow \bigcup_{k \geq 1} \{k\} \times L^k$ be a pairwise mapping on L. An ω-labelled tree is a tuple $t = (A, R, h)$, where*

- *(A, R) is an ordered tree*
- *$h : A \longrightarrow L$ is a mapping satisfying the following condition: if $[(i, i_1), \ldots, (i, i_s)] \in R$ then $h(i) \in N_L$, $s = \omega_1(h(i))$ and $\omega_2(h(i)) = (h(i_1), \ldots, h(i_s))$*

In other words, if a node of t has at least one descendant then its label is a nonterminal given by the labelling function h; on the other hand there is a connection between ω and h, which is specified by the number of the direct descendants and their labels.

If $t = (A, R, h)$ is an ω-labelled tree then by $root(t)$ we denote the element of A which gives the root of t. Frequently in this paper we suppose that $A = \{1, \ldots, n\}$ for some natural number n and $root(t) = 1$.

We consider an element $b_0 \in N_L$ and denote by $Tree_\omega(b_0)$ the set of all ω-labelled trees $t = (A, R, h)$ such that $h(root(t)) = b_0$. If $u = [(i, i_1), \ldots, (i, i_s)] \in R$ then we denote $pr_{r_1,\ldots,r_m} u = [(i, i_{r_1}), \ldots, (i, i_{r_m})]$ where $1 \leq r_1 < r_2 < \ldots < r_m \leq s$. We shall write $v \subseteq u$ if there are $r_1, \ldots, r_m$ such that $v = pr_{r_1,\ldots,r_m} u$.

Let be $t_1 = (A_1, R_1, h_1) \in Tree_\omega(b_0)$ and $t_2 = (A_2, R_2, h_2) \in Tree_\omega(b_0)$. If $\alpha : A_1 \longrightarrow A_2$ is an arbitrary mapping then for every $u = [(i, i_1), \ldots, (i, i_s)] \in R_1$ we denote

$$\overline{\alpha}(u) = [(\alpha(i), \alpha(i_1)), \ldots, (\alpha(i), \alpha(i_s))]$$

Definition 2.3 *Let be* $t_1 = (A_1, R_1, h_1) \in Tree_\omega(b_0)$ *and* $t_2 = (A_2, R_2, h_2) \in Tree_\omega(b_0)$. *We define the following binary relation on* $Tree_\omega(b_0)$: $t_1 \preceq t_2$ *if there is an injective mapping* $\alpha : A_1 \longrightarrow A_2$ *such that:*

1) $\alpha(root(t_1)) = root(t_2)$

2) *if* $u \in R_1$ *then there is* $v \in R_2$ *such that* $\overline{\alpha}(u) \subseteq v$

3) $h_1(i) = h_2(\alpha(i))$ *for every* $i \in A_1$

A characterization of the relation $\preceq$ is given in **Proposition** 6.3.

By **Proposition 6.4** the relation $\preceq$ is reflexive and transitive. It is not antisymmetric as we can see in **Figure 1**. Therefore $\preceq$ is not a partial order.

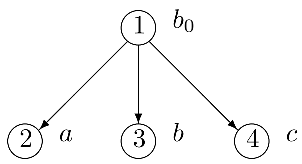

Figure 1: $t_1 \preceq t_2$, $t_2 \preceq t_1$, $t_1 \neq t_2$

For this reason we introduce now an equivalence relation on $Tree_\omega(b_0)$ and then we define a partial order on the set of all equivalence classes.

Definition 2.4 *Let be* $t_1, t_2 \in Tree_\omega(b_0)$. *We define* $t_1 \approx t_2$ *iff* $t_1 \preceq t_2$ *and* $t_2 \preceq t_1$.

Obviously, the relation $\approx$ is a symmetric relation and therefore according to **Proposition 6.4** we obtain an equivalence relation.

Because $\approx$ is an equivalence relation, we can consider the factor set $Tree_\omega(b_0)/_\approx$ of the equivalence classes. For every $t \in Tree_\omega(b_0)$ we denote by $[t]$ the equivalence class of t.

Definition 2.5 *We define the following relation on* $Tree_\omega(b_0)/_\approx$:

$$[t_1] \ll [t_2] \quad \textit{if and only if} \quad t_1 \preceq t_2$$

By **Proposition 6.6** the definition of $\ll$ is not dependent on the representatives and by **Proposition 6.7** the relation $\ll$ is a partial order.

3 A suitable representation for ω-labelled trees

We shall give now a particular representation for the elements of $Tree_\omega(b_0)$, which will permit us to obtain several algebraic properties for the factor set $Tree_\omega(b_0)/_\approx$. We consider an element $t = (A, R, h) \in Tree_\omega(b_0)$. The set of all the paths of t is denoted by $Path(t)$. Because t is an ordered tree, the pair (i, j) is an arc of t if and only if there is an element and only one $u = [(i, i_1), \ldots, (i, i_s)] \in R$ such that $j = i_r$ for some $r \in \{1, \ldots, s\}$. We shall denote this fact by $(i, j) \in R^{(r)}$. This means that (i, j) appears as the r^{th} element of some list of R.

We denote by N the set of all natural numbers and an element from $N^p \times L^p$ will be denoted by $((l_1, \ldots, l_p), (b_1, \ldots, b_p))$, where $l_i \in N$ and $b_i \in L$ for $i \in \{1, \ldots, p\}$. We suppose $root(t) = 1$ for every $t \in Tree_\omega(b_0)$.

Definition 3.1 *Let be $t = (A, R, h) \in Tree_\omega(b_0)$. We denote by $S(t)$ the following subset of $\bigcup_{p\geq 1} N^p \times L^p$:*

$$((l_1, \ldots, l_s), (b_1, \ldots, b_s)) \in S(t)$$

iff there is $(1, i_1, \ldots, i_s) \in Path(t)$ such that $(1, i_1) \in R^{(l_1)}, \ldots, (i_{s-1}, i_s) \in R^{(l_s)}$ and $h(i_1) = b_1, \ldots, h(i_s) = b_s$. We define $H_t : S(t) \longrightarrow Path(t)$ as follows:

$$H_t((l_1, \ldots, l_s), (b_1, \ldots, b_s)) = (1, i_1, \ldots, i_s)$$

Let us show that the mapping H_t is well defined and is bijective. Suppose that $(1, i_1, \ldots, i_s) \in Path(t)$ and $(1, j_1, \ldots, j_s) \in Path(t)$ are two paths satisfying **Definition 3.1**. From $(1, i_1) \in R^{l_1}$ and $(1, j_1) \in R^{l_1}$ we deduce $i_1 = j_1$. Similarly we have $i_2 = j_2$ and so on. Therefore H_t is well defined. Let us assume

$$H_t((l_1, \ldots, l_s), (b_1, \ldots, b_s)) = H_t((m_1, \ldots, m_q), (c_1, \ldots, c_q))$$

It follows that $s = q$ and if we denote by $(1, i_1, \ldots, i_s)$ the corresponding value then $(1, i_1) \in R^{(l_1)} \cap R^{(m_1)}$, …, $(i_{s-1}, i_s) \in R^{(l_s)} \cap R^{(m_s)}$, $h(i_1) = b_1 = c_1$, …, $h(i_s) = b_s = c_s$. Therefore $l_1 = m_1$, …, $l_s = m_s$.

According to **Proposition 6.8** and **Proposition 6.9** we have $t_1 \preceq t_2$ if and only if $S(t_1) \subseteq S(t_2)$. Taking into consideration the definition of $\approx$ we deduce now that $t_1 \approx t_2$ if and only if $S(t_1) = S(t_2)$.

We consider a finite set $X \subseteq \bigcup_{k\geq 1} N^k \times L^k$. For $x = (b_1, \ldots, b_k) \in L^k$ and $i \in \{1, \ldots, k\}$ we denote $pr_i x = b_i$. For every $k \geq 1$, the level k of X is defined as $lev_k(X) = X \cap (N^k \times L^k)$. We shall use the following notation: if $Y \subseteq lev_{k+1}X$ then

$$pr_{1,\ldots,k}Y = \{((l_1, \ldots, l_k), (b_1, \ldots, b_k)) \mid \exists l, b : ((l_1, \ldots, l_k, l), (b_1, \ldots, b_k, b)) \in Y\}$$

Definition 3.2 *Let $\omega : N_L \longrightarrow \bigcup_{k\geq 1}\{k\} \times L^k$ be a pairwise mapping and $b_0 \in N_L$. We say that $X \subseteq \bigcup_{k\geq 1} N^k \times L^k$ satisfies the $\omega - b_0$- conditions if the following conditions ω_1 and ω_2 are fulfilled:*

(ω1) $((l), (b_l)) \in X$ *iff* $l \in \{1, \ldots, \omega_1(b_0)\}$ *and* $b_l = pr_l \omega_2(b_0)$

(ω2) *If* $((l_1, \ldots, l_k, l), (b_1, \ldots, b_k, a)) \in X$ *then*

- $((l_1, \ldots, l_k), (b_1, \ldots, b_k)) \in X$

- $l \in \{1, \dots, \omega_1(b_k)\}$ *and* $a = pr_l\omega_2(b_k)$
- $((l_1, \dots, l_k, i), (b_1, \dots, b_k, pr_i\omega_2(b_k))) \in X$ *for all* $i \in \{1, \dots, \omega_1(b_k)\}$

Directly from this definition we observe that:
1) the condition ω_1 can be written equivalently

$$lev_1(X) = \{((l), (pr_l\omega_2(b_0)))\}_{l=1,\dots,\omega_1(b_0)}$$

2) if $((l_1, \dots, l_k), (b_1, \dots, b_k)) \in X$ and $((l_1, \dots, l_k), (a_1, \dots, a_k)) \in X$ then $b_j = a_j$ for $j \in \{1, \dots, k\}$.

The sets $S(t)$ for $t \in Tree_\omega(b_0)$ are identified by **Proposition 6.10** with the sets satisfying the $\omega - b_0$ conditions and this result is used in the next subsection. Moreover, the proof of **Proposition 6.10** includes an algorithm which is used to obtain a tree t such $X = S(t)$, where X is a given set satisfying the $\omega - b_0$ conditions.

4 The lattice $Tree_\omega(b_0)/_\approx$

In this section we introduce two binary operations on the set $Tree_\omega(b_0)/_\approx$ such that a lattice structure is obtained.

Definition 4.1 *We define the algebraic operations*

$$\vee : Tree_\omega(b_0)/_\approx \times Tree_\omega(b_0)/_\approx \longrightarrow Tree_\omega(b_0)/_\approx$$

$$\wedge : Tree_\omega(b_0)/_\approx \times Tree_\omega(b_0)/_\approx \longrightarrow Tree_\omega(b_0)/_\approx$$

as follows:

$$[t_1] \vee [t_2] = [t], \textit{ where } S(t) = S(t_1) \cup S(t_2)$$

$$[t_1] \wedge [t_2] = [t], \textit{ where } S(t) = S(t_1) \cap S(t_2)$$

The above definitions are correctly given. Really, if $t_1, t_2 \in Tree_\omega(b_0)$ then by **Proposition 6.10**, $S(t_1)$ and $S(t_2)$ satisfy the $\omega - b_0$ conditions. By **Proposition 6.11**, $S(t_1) \cup S(t_2)$ and $S(t_1) \cap S(t_2)$ satisfy also these conditions. According to **Proposition 6.10** it follows that there are $t_3, t_4 \in Tree_\omega(b_0)$ such that $S(t_3) = S(t_1) \cup S(t_2)$ and $S(t_4) = S(t_1) \cap S(t_2)$. By **Definition 4.1** we take $[t_1] \vee [t_2] = [t_3]$ and $[t_1] \wedge [t_2] = [t_4]$. Now, if t'_3 and t'_4 are two elements such that $S(t'_3) = S(t_1) \cup S(t_2)$ and $S(t'_4) = S(t_1) \cap S(t_2)$ then $S(t_3) = S(t'_3)$ and $S(t_4) = S(t'_4)$, therefore by **Proposition 6.8** and **Proposition 6.9** we have $t_3 \approx t'_3$ and $t_4 \approx t'_4$. Thus $[t_3] = [t'_3]$ and $[t_4] = [t'_4]$.

In **Theorem 6.1** we prove that $(Tree_\omega(b_0)/_\approx, \vee, \wedge)$ is a lattice.

5 Conclusions and future work

Based on the concept of pairwise mapping some kind of labelled ordered tree is introduced in this paper. Several properties of these trees are established, a natural equivalence relation is introduced and a lattice structure of labelled ordered trees is obtained. This research work will be continued as follows: in a forthcoming paper

([9]) several algebraic properties of the lattice introduced in this paper will be presented; in order to study the computability of the answer function of a knowledge representation system based on inheritance property, these properties will be used and this is the aim of of a paper in preparation.

6 APPENDIX: Theoretical results

In this section we give the formal proofs for the properties concerning the ω-labelled trees.

Proposition 6.1 *Let be $t_1, t_2 \in Tree_\omega(b_0)$. If $t_1 \preceq t_2$ then for every $u \in R_1$ we have $\overline{\alpha}(u) \in R_2$.*

Proof. Let be $u = [(i, i_1), \ldots, (i, i_s)] \in R_1$. Since $t_1 \in Tree_\omega(b_0)$ we have $s = \omega_1(h_1(i))$. But $\overline{\alpha}(u) = [(\alpha(i), \alpha(i_1)), \ldots, (\alpha(i), \alpha(i_s))]$ and $h_1(i) = h_2(\alpha(i))$ therefore $s = \omega_1(h_2(\alpha(i)))$. There is $v \in R_2$ such that $\overline{\alpha}(u) \subseteq v$. It follows that there is $k \geq s$ and there are $j_1, \ldots, j_k$ such that $v = [(\alpha(i), j_1), \ldots, (\alpha(i), j_k)]$. From $v \in R_2$ we deduce $k = \omega_1(h_2(\alpha(i)))$, therefore $k = s$. Thus $\overline{\alpha}(u) = v \in R_2$. ∎

If $t = (A, R, h) \in Tree_\omega(b_0)$ then we denote $level_0(t) = \{root(t)\}$ and $level_k(t)$ is the set of all nodes $i \in A$ such that there is a path of length k from 1 to i.

Proposition 6.2 *Let be $t_1, t_2 \in Tree_\omega(b_0)$. If $t_1 \preceq t_2$ then for every k we have*

$$\alpha(level_k(t_1)) \subseteq level_k(t_2)$$

Proof. We proceed by induction on k. For $k = 0$ the property is true because we have $\alpha(level_0(t_1))$=$\alpha(\{root(t_1)\})$=$\{root(t_2)\}$. Suppose the property is true for $k = m-1$, where $m \geq 1$. If $i \in level_m(t_1)$ then there is $j_i \in level_{m-1}(t_1)$ such that i is a direct descendant of j_i in t_1. By inductive assumption we have $\alpha(j_i) \in level_{m-1}(t_2)$. By **Proposition 6.1** the node $\alpha(i)$ is a direct descendant of $\alpha(j_i)$ in t_2. Therefore $\alpha(i) \in level_m(t_2)$. ∎

Proposition 6.3 *Let be $t_1, t_2 \in Tree_\omega(b_0)$. We have $t_1 \preceq t_2$ iff there is an injective mapping $\alpha : A_1 \longrightarrow A_2$ such that*

- $\alpha(root(t_1)) = root(t_2)$
- $\overline{\alpha}(u) \in R_2$ *for every* $u \in R_1$

Proof. The direct implication is obtained from **Proposition 6.1**. Let us prove the converse implication. We verify the property $h_1(i) = h_2(\alpha(i))$ for every $i \in A_1 = \bigcup_{m \geq 0} level_m(t_1)$. We proceed by induction on m. For $m = 0$ we have $level_m(t_1) = \{root(t_1)\}$, $h_1(root(t_1)) = b_0 = h_2(root(t_2)) = h_2(\alpha(root(t_1)))$. We suppose $h_1(i) = h_2(\alpha(i))$ for every $i \in level_{k-1}(t_1)$. Let be $i \in level_k(t_1)$. There is $u = [(j, r_1), \ldots, (j, r_s)] \in R_1$ and there is $p \in \{1, \ldots, s\}$ such that $r_p = i$. The values of $\overline{\alpha}$ are taken from R_2, therefore $\overline{\alpha}(u) \in R_2$. But t_1 and t_2 belong to $Tree_\omega(b_0)$ and $\overline{\alpha}(u) = [(\alpha(j), \alpha(r_1)), \ldots, (\alpha(j), \alpha(r_s))]$. It follows that $\omega_2(h_1(j)) = (h_1(r_1), \ldots, h_1(r_s))$, $\omega_2(h_2(\alpha(j))) = (h_2(\alpha(r_1)), \ldots, h_2(\alpha(r_s)))$. On the other hand $j \in level_{k-1}t_1$ and by the inductive assumption we have $h_1(j) = h_2(\alpha(j))$. It follows that $\omega_2(h_1(j)) = \omega_2(h_2(\alpha(j)))$ therefore $h_1(r_p) = h_2(\alpha(r_p))$. Thus we have $h_1(i) = h_2(\alpha(i))$. ∎

Proposition 6.4 *The binary relation $\preceq$ is reflexive and transitive.*

Proof. Immediate, by the fact that the superposition of two injective function is an injective one. ∎

Proposition 6.5 *Let be $t_1 = (A_1, R_1, h_1) \in Tree_\omega(b_0)$, $t_2 = (A_2, R_2, h_2) \in Tree_\omega(b_0)$. We have $t_1 \approx t_2$ iff there is a bijective mapping $\alpha : A_1 \longrightarrow A_2$ such that:*

- $\alpha(root(t_1)) = root(t_2)$
- $\overline{\alpha}(u) \in R_2$ *for every* $u \in R_1$

Proof. Suppose $t_1 \approx t_2$. By **Proposition 6.3** we deduce that there are the injective mappings $\alpha_1 : A_1 \longrightarrow A_2$ and $\alpha_2 : A_2 \longrightarrow A_1$ such that $\overline{\alpha}_1(u) \in R_2$ for every $u \in R_1$ and $\overline{\alpha}_2(v) \in R_1$ for every $v \in R_2$. It follows that $Card(A_1) \leq Card(A_2) \leq Card(A_1)$, therefore α_1 is a bijective mapping because A_1 and A_2 are finite sets. The direct implication is proved. Let us consider a mapping $\alpha : A_1 \longrightarrow A_2$ such that $\alpha(root(t_1)) = root(t_2)$ and $\overline{\alpha}(u) \in R_2$ for every $u \in R_1$. We shall prove that $t_1 \approx t_2$. By **Proposition 6.3** we have $t_1 \preceq t_2$. There is $\alpha^{-1} : A_2 \longrightarrow A_1$ and moreover, we have $\alpha^{-1}(root(t_2)) = \alpha^{-1}(\alpha(root(t_1))) = root(t_1)$. We shall prove that $\overline{\alpha^{-1}}(v) \in R_1$ for every $v \in R_2$. We take $v = [(i, i_1), \ldots, (i, i_s)] \in R_2$, where $i, i_1, \ldots, i_s \in A_2$. We obtain $\overline{\alpha^{-1}}(v) = [(\alpha^{-1}(i), \alpha^{-1}(i_1)), \ldots, (\alpha^{-1}(i), \alpha^{-1}(i_s))]$. Taking the direct descendants of $\alpha^{-1}(i)$ in t_1 we obtain $u = [(\alpha^{-1}(i), k_1), \ldots, (\alpha^{-1}(i), k_p)] \in R_1$. By the properties verified by α we have $\overline{\alpha}(u) = [(i, \alpha(k_1)), \ldots, (i, \alpha(k_p))] \in R_2$ and we have also $v = [(i, i_1), \ldots, (i, i_s)] \in R_2$. It follows $p = s$ and $\alpha(k_1) = i_1, \ldots, \alpha(k_p) = i_p$. Equivalently we have $k_1 = \alpha^{-1}(i_1), \ldots, k_p = \alpha^{-1}(i_p)$ therefore $\overline{\alpha^{-1}}(v) = [(\alpha^{-1}(i), k_1), \ldots, (\alpha^{-1}(i), k_p)] = u \in R_1$. By **Proposition 6.3** we have $t_2 \preceq t_1$ and the proof is complete. ∎

Proposition 6.6 *The definition of the relation $\ll$ on $Tree_\omega(b_0)/_\approx$ does not depend on the representatives.*

Proof. Let us suppose $t_1 \preceq t_2$ and $t'_1 \approx t_1$, $t'_2 \approx t_2$. We have $t'_1 \preceq t_1$, $t_1 \preceq t_2$ and $t_2 \preceq t'_2$. By transitivity we have $t'_1 \preceq t'_2$. ∎

Proposition 6.7 *The binary relation $\ll$ on $Tree_\omega(b_0)/_\approx$ is reflexive, antisymmetric and transitive, therefore it is a partial order.*

Proof. If $[t_1] \ll [t_2]$ and $[t_2] \ll [t_1]$ then $t_1 \preceq t_2$ and $t_2 \preceq t_1$. Thus we have $[t_1] = [t_2]$. The reflexivity and transitivity is obtained from the corresponding properties for the relation $\preceq$ on $Tree_\omega(b_0)$. ∎

Proposition 6.8 *Let be $t_1, t_2 \in Tree_\omega(b_0)$. If $t_1 \preceq t_2$ then $S(t_1) \subseteq S(t_2)$.*

Proof. Let be $t_1 = (A_1, R_1, h_1)$ and $t_2 = (A_2, R_2, h_2)$ such that $t_1 \leq t_2$. If $((l_1, \ldots, l_s), (b_1, \ldots, b_s)) \in S(t_1)$ then there is $(1, i_1, \ldots, i_s) \in Path(t_1)$ such that $(1, i_1) \in R_1^{(l_1)}$, $(i_1, i_2) \in R_1^{(l_2)}$, …, $(i_{s-1}, i_s) \in R_1^{(l_s)}$. Using **Proposition 6.1**, from $t_1 \preceq t_2$ it follows that $(\alpha(1), \alpha(i_1)) \in R_2^{(l_1)}$, $(\alpha(i_1), \alpha(i_2)) \in R_2^{(l_2)}$, …, $(\alpha(i_{s-1}), \alpha(i_s)) \in R_2^{(l_s)}$. Since $\alpha(1) = 1$ we deduce that $(1, \alpha(i_1), \ldots, \alpha(i_s)) \in Path(t_2)$. Moreover, $h_2(\alpha(i_1)) = h_1(i_1) = b_1$, …, $h_2(\alpha(i_s)) = h_1(i_s) = b_s$. Thus $((l_1, \ldots, l_s), (b_1, \ldots, b_s)) \in S(t_2)$. ∎

Proposition 6.9 *Let be $t_1, t_2 \in Tree_\omega(b_0)$. If $S(t_1) \subseteq S(t_2)$ then $t_1 \preceq t_2$.*

Proof. We use the same notations as in the previous proposition. In order to prove that $t_1 \preceq t_2$ we must define an injective mapping $\alpha : A_1 \longrightarrow A_2$ satisfying **Definition 2.3**. We define $\alpha(1) = 1$. Let be $i \in A_1 \setminus \{1\}$ an arbitrary element. There is $(1, i_1, \ldots, i_s) \in Path(t_1)$ such that $i_s = i$. We consider $H_{t_2}(H_{t_1}^{-1}(1, i_1, \ldots, i_s)) = (1, j_1, \ldots, j_s)$ and define $\alpha(i) = j_s$. The mapping α is well defined since for every $i \in A_1 \setminus \{1\}$ there is only one path $(1, i_1, \ldots, i_s)$ such that $i_s = i$ and on the other hand $S(t_1) \subseteq S(t_2)$, therefore H_{t_2} can be composed with $H_{t_1}^{-1}$.
It is not difficult to verify by induction on s that if $(1, i_1, \ldots, i_s) \in Path(t_1)$ then

$$H_{t_2}(H_{t_1}^{-1}(1, i_1, \ldots, i_s)) = (1, \alpha(i_1), \ldots, \alpha(i_s))$$

Moreover, if $(i_{r-1}, i_r) \in R_1^{(l_r)}$ then $(\alpha(i_{r-1}), \alpha(i_r)) \in R_2^{(l_r)}$, where $r \in \{1, \ldots, s\}$ and $i_0 = 1$. Really, if $((l_1, \ldots, l_s), (b_1, \ldots, b_s)) = H_{t_1}^{-1}(1, i_1, \ldots, i_s)$ and $H_{t_2}((l_1, \ldots, l_s), (b_1, \ldots, b_s)) = (1, \alpha(i_1), \ldots, \alpha(i_s))$ then by the definition of H_{t_2} we have $(1, \alpha(i_1)) \in R_2^{(l_1)}, \ldots, (\alpha(i_{s-1}), \alpha(i_s)) \in R_2^{(l_s)}$.
Let us prove that α satisfies the conditions of **Definition 2.3**. Since $t_1, t_2 \in Tree_\omega(b_0)$ we have $h_1(1) = h_2(1) = b_0$ therefore $h_2(\alpha(1)) = h_1(1)$. Let us verify that $h_2(\alpha(i)) = h_1(i)$ for every $i \in A_1 \setminus \{1\}$. Let be $((l_1, \ldots, l_s), (b_1, \ldots, b_s)) = H_{t_1}^{-1}(1, i_1, \ldots, i_s)$, where $i_s = i$. Then $H_{t_2}((l_1, \ldots, l_s), (b_1, \ldots, b_s)) = (1, \alpha(i_1), \ldots, \alpha(i_s))$ and by the definition of H_{t_2} we have $h_2(\alpha(i_s)) = b_s$, therefore $h_2(\alpha(i)) = h_1(i)$.
We verify now that if $u \in R_1$ then $\overline{\alpha}(u) \subseteq v$ for some $v \in R_2$. Let be $u = [(i, i_1), \ldots, (i, i_s)] \in R_1$. Then $(i, i_j) \in R_1^{(j)}$ for $j \in \{1, \ldots, s\}$. Therefore $(\alpha(i), \alpha(i_j)) \in R_2^{(j)}$ for $j \in \{1, \ldots, s\}$. It follows that there is $v \in R_2$ such that $\overline{\alpha}(u) = pr_{1,\ldots,s}v$, that is $\overline{\alpha}(u) \subseteq v$.
Let us verify that α is an injective mapping. Let be $i, k \in A_1 \setminus \{1\}$ such that $\alpha(i) = \alpha(k)$. There are $(1, i_1, \ldots, i_p, i)$, $(1, j_1, \ldots, j_q, k)$ in $Path(t_1)$, therefore

$$H_{t_2}(H_{t_1}^{-1}(1, i_1, \ldots, i_p, i)) = (1, \alpha(i_1), \ldots, \alpha(i_p), \alpha(i)) \in Path(t_2)$$

$$H_{t_2}(H_{t_1}^{-1}(1, j_1, \ldots, j_q, k)) = (1, \alpha(j_1), \ldots, \alpha(j_q), \alpha(k)) \in Path(t_2)$$

Since t_2 is a tree and $\alpha(i) = \alpha(k)$ we deduce $p = q$, $\alpha(i_r) = \alpha(j_r)$ for $r = 1, \ldots, p$. But H_{t_2} and H_{t_1} are bijective mappings, therefore $(1, i_1, \ldots, i_p, i) = (1, j_1, \ldots, j_p, k)$. Thus $i = k$.
We have also $\alpha(i) \neq 1$ for every $i \neq 1$. Really, if $\alpha(i) = \alpha(1)$ for $i \neq 1$ then there is a path $(1, \ldots, i) \in Path(t_1)$, therefore $(1, \ldots, \alpha(i)) \in Path(t_2)$, which is not true since $\alpha(i) = 1$. ■

Proposition 6.10 *Let $X \subseteq \bigcup_{k \geq 1} N^k \times L^k$ be a finite set. There is $t \in Tree_\omega(b_0)$ such that $S(t) = X$ iff X satisfies the $\omega - b_0$-conditions.*

Proof. Let be $t = (A, R, h) \in Tree_\omega(b_0)$. We shall verify that $S(t)$ satisfies the $\omega - b_0$-conditions. The element $((l, b_l))$ belongs to $S(t)$ iff there is $(1, i) \in R^{(l)}$ such that $h(i) = b_l$. Let be $[(1, i_1), \ldots, (1, i_s)] \in R$. Since t is an ω-labelled tree and moreover, $t \in Tree_\omega(b_0)$, it follows that $s = \omega_1(h(1)) = \omega_1(b_0)$ and $(h(i_1), \ldots, h(i_s)) = \omega_2(h(1)) = \omega_2(b_0)$. It follows that there is $(1, i) \in R^{(l)}$ such that $h(i) = b_l$ iff

$l \in \{1, \ldots, \omega_1(b_0)\}$ and $b_l = pr_l\omega_2(b_0)$. Thus the condition ω_1 is satisfied.
Let us consider an element $((l_1, \ldots, l_k, l), (b_1, \ldots, b_k, a)) \in S(t)$ and let us denote

$$H_t((l_1, \ldots, l_k, l), (b_1, \ldots, b_k, a)) = (1, i_1, \ldots, i_k, i)$$

By the definition of $S(t)$ we deduce that $(1, i_1) \in R^{(l_1)}$, $(i_1, i_2) \in R^{(l_2)}$, …, $(i_{k-1}, i_k) \in R^{(l_k)}$, $(i_k, i) \in R^{(l)}$, $h(i_1) = b_1$, …, $h(i_k) = b_k$, $h(i) = a$. Obviously $((l_1, \ldots, l_k), (b_1, \ldots, b_k)) \in S(t)$. Since $(i_k, i) \in R^{(l)}$ we deduce that there is $u = [(i_k, j_1), \ldots, (i_k, j_s)] \in R$ such that $i = j_l$. Because t is an ω-labelled tree we have $s = \omega_1(h(i_k)) = \omega_1(b_k)$ and $l \in \{1, \ldots, s\} = \{1, \ldots, \omega_1(b_k)\}$. Moreover, $\omega_2(b_k) = \omega_2(h(i_k)) = (h(j_1), \ldots, h(j_s))$, therefore $pr_l\omega_2(b_k) = h(j_l) = h(i) = a$. Since $(1, i_1, \ldots, i_k) \in Path(t)$ and $[(i_k, j_1), \ldots, (i_k, j_s)] \in R$ we deduce $\{(1, i_1, \ldots, i_k, j_1), \ldots, (1, i_1, \ldots, i_k, j_s)\} \subseteq Path(t)$. The mapping H_t is a bijective one, therefore $\{H_t^{-1}(1, i_1, \ldots, i_k, j_1), \ldots, H_t^{-1}(1, i_1, \ldots, i_k, j_s)\} \subseteq S(t)$. But

$$H_t^{-1}(1, i_1, \ldots, i_k, j_m) = ((l_1, \ldots, l_k, m), (b_1, \ldots, b_k, h(j_m)))$$

for $m \in \{1, \ldots, s\}$, $\omega_2(b_k) = (h(j_1), \ldots, h(j_s))$ and $s = \omega_1(b_k)$.
Conversely, we suppose that X satisfies the ω-b_0-conditions. We build a tree $t \in Tree_\omega(b_0)$ recursively, taking into account the levels of the set X. We apply the following algorithm:

Algorithm 6.1

$A := \{1\}$; $h(1) = b_0$; $R := \emptyset$; $n_1 := \omega_1(b_0) + 1$;

$A := A \cup \{2, \ldots, n_1\}$; $(b_1, \ldots, b_{n_1-1}) := \omega_2(b_0)$;

for $i \in \{1, \ldots, n_1 - 1\}$ **do**

$H_1((i), (b_i)) := (1, i + 1)$; $h(i + 1) := b_i$;

endfor

$R := R \cup \{[(1, 2), \ldots, (1, n_1)]$; $m := max\{k \mid lev_k(X) \neq \emptyset\}$;

for $k = 1$ **to** $m - 1$ **do**

$B_k = pr_{1,\ldots,k}lev_{k+1}(X)$; $pk := Card(B_k)$; **denote** $B_k = \{z_1, \ldots, z_{pk}\}$;
for $s = 1$ **to** pk **do**
denote $z_s = ((l_1, \ldots, l_k), (b_1, \ldots, b_k))$;
$n := Card(A)$; $(1, i_1, \ldots, i_k) := H_k(z_s)$;
$A := A \cup \{n + 1, \ldots, n + \omega_1(b_k)\}$; $(a_1, \ldots, a_{\omega_1(b_k)}) := \omega_2(b_k)$;
for $l \in \{1, \ldots, \omega_1(b_k)\}$ **do**
$H_{k+1}((l_1, \ldots, l_k, l), (b_1, \ldots, b_k, a_l)) := (1, i_1, \ldots, i_k, n + l)$;
$h(n + l) := a_l$;
endfor
$R := R \cup \{[(i_k, n + 1), \ldots, (i_k, n + \omega_1(b_k))]\}$;
endfor

endfor

End of algorithm

We obtain the tree $t = (A, R, h) \in Tree_\omega(b_0)$. We observe that for every $l \in \{1, \ldots, \omega_1(b_k)\}$ we have

$$H_{k+1}((l_1, \ldots, l_k, l), (b_1, \ldots, b_k, a_l)) = (1, i_1, \ldots, i_k, j_l)$$

iff $H_k((l_1, \ldots, l_k), (b_1, \ldots, b_k)) = (1, i_1, \ldots, i_k)$, $h(j_l) = a_l$ and $(i_k, j_l) \in R^{(l)}$. Now, we can verify by induction on k that $lev_k(X) = lev_k(S(t))$ for every $k \in \{1, \ldots, m\}$, therefore $X = S(t)$. ∎

Proposition 6.11 *If X_1 and X_2 satisfy the ω-b_0-conditions then $X_1 \cap X_2$ and $X_1 \cup X_2$ satisfy also the ω-b_0-conditions.*

Proof. Obviously, the conditions ω_1 and ω_2 are satisfied. ∎

Theorem 6.1 *$(Tree_\omega(b_0)/_\approx, \vee, \wedge)$ is a lattice.*

Proof. Let be $[t_1], [t_2] \in Tree_\omega(b_0)/_\approx$. We take $X = S(t_1) \cap S(t_2)$. Since $S(t_1)$ and $S(t_2)$ statisfy the ω-b_0-conditions it follows that X satisfies also these conditions. Thus there is $t^* \in Tree_\omega(b_0)$ such that $X = S(t^*)$. Applying **Proposition 6.9** we deduce $t^* \preceq t_1$ and $t^* \preceq t_2$, therefore $[t^*] \ll [t_1]$ and $[t^*] \ll [t_2]$. Let be $t_0 \in Tree_\omega(b_0)$ such that $[t_0] \ll [t_1]$ and $[t_0] \ll [t_2]$. Applying **Proposition 6.8** we obtain $S(t_0) \subseteq S(t_1)$ and $S(t_0) \subseteq S(t_2)$, therefore $S(t_0) \subseteq X$. By **Proposition 6.9** we have $[t_0] \ll [t^*]$. Thus we have $[t^*] = inf\{[t_1], [t_2]\}$. But $[t^*] = [t_1] \wedge [t_2]$, therefore $[t_1] \wedge [t_2] = inf\{[t_1], [t_2]\}$. Similarly, we have $sup\{[t_1], [t_2]\}=[t^{**}]$, where $S(t^{**}) = S(t_1) \cup S(t_2)$ and $[t_1] \vee [t_2] = sup\{[t_1], [t_2]\}$.

∎

References

[1] R. Morelli, Java, Java, Java. Object-Oriented Problem Solving (Prentice- Hall, 2000)

[2] R. Nado and R. Fikes, Saying more with frames: Slots as classes, Computers Math. Applic., 23(6-9) (1992) 719-731

[3] T. Ozsu and P. Valduriez, Principles of Distributed Database Systems (Prentice Hall, 1999)

[4] P. Prusinkiewicz and A. Lindenmayer, The Algorithmic Beauty of Plants (Springer Verlag, 1990)

[5] C. Rich and Y.A. Feldman, Seven Layers of Knowledge Representation and Reasoning in Support of Software Development, IEEE Transactions on Software Engineering, 18(6), (1992) 451-469

[6] Sergiu Rudeanu, Lattice Functions and Equations, Springer, 2001

[7] R. H. Thomason, NETL and subsequent path-based inheritance theories, Computers Math. Applic., Vol. 23(2-5 (1992) 179-204

[8] N. Ţăndăreanu, Knowledge Bases with Output, Knowledge and Information Systems 2 (2000), 4, 438-460

[9] N. Ţăndăreanu, Lattices of labelled ordered trees (II), in preparation

[10] G. Wagner, Vivid Logic, Knowledge-Based Reasoning with Two Kinds of Negation, Lecture Notes in Artificial Intelligence 764 (Springer Verlag, 1994).

Author's address:

Nicolae Ţăndăreanu
Faculty of Mathematics and Computer Science
University of Craiova
13, Al.I. Cuza st., 1100, Craiova, România
Tel/Fax: 40-51413728
e-mail: ntand@oltenia.ro