MATHEMATICAL PROBLEMS IN COMPUTATION THEORY
BANACH CENTER PUBLICATIONS, VOLUME 21
PWN – POLISH SCIENTIFIC PUBLISHERS
WARSAW 1988

AN AXIOMATIC THEORY OF INFORMATION TREES

KEH-HSUN CHEN
ZBIGNIEW W. RAS

University of North Carolina at Charlotte, U.S.A.

In our previous paper (see [1]) we define a storage cost measure on decision trees and propose heuristic algorithms to construct minimal trees with respect to that measure. For the current paper we develop some rules on decision trees which allow us to manipulate them into more convenient forms. The completeness theorem proved by us gives the certainty that there is a way of applying our rules to a tree to construct a minimal tree for the one we start with.

Introduction

In this paper we define a notion of an information tree on the universe U, where U is a set of attributes. Information trees are similar in structure to decision trees applied to identification problems. The main difference between information trees and decision trees lies in the interpretations and applications. Their structures and methods of constructions are often transferable to each other. In decision trees nodes are labeled by queries, edges by responses to these queries and leaves by some objects uniquely identified by the path from the root to the leaf. Now, having some item which we want to classify we pick up queries one after another starting from the root of a decision tree and being controlled by responses to them. In an information tree, internal nodes are labeled by attributes and terminal nodes by sets of objects. A path from the root to a leaf is interpreted as a description of objects labeling that leaf.

Decision trees have been investigated in the literature by many authors

Key words. *Information tree, storage cost, minimization problems, completeness theorem, formal theory.*

and have applications in decision table programming, pattern recognition, taxonomy and identification, switching theory, expert systems and analysis of algorithms. Some decision trees are equivalent with respect to the information they keep, so we may ask for an optimal tree in the class of equivalent trees. To ask for a minimal tree we have to develop first some measure on decision trees. A storage cost and a testing cost measures are the most natural ones. In our paper (see [1]) we propose a heuristic polynomial algorithm to construct a minimal tree with respect to the storage cost. It is worth to note that the problem of constructing even a minimal binary tree with respect to the storage cost is to be known as NP-complete. The method proposed by us in [1] requires for a given tree T to construct a matrix representation of T and next on the basis of this matrix we look for a minimal tree. Questions we would like to state in this paper say: can we look for this minimal tree for T without using its matrix representation? Can we manipulate information trees algebraically into more convenient forms?

To answer these questions we devise a representation of information trees as terms in a formal theory (theory of information trees) and present rules to manipulate them. Next we show that the rules proposed by us are complete in constructing equivalent information trees and deducing weaker information trees. Completeness theorem proved by us clearly does not give any new (heuristic) method for constructing minimal trees. However, it gives us the certainty that applying system's rules to T we will arrive to a minimal tree.

1. Basic definitions

In this section we recall the definition of an information tree and introduce the notion of equivalence of two information trees and the notion of one tree being covered by another.

Let U be a finite set of attributes called the *universe*. For each $A \in U$, let V_A be the set of attribute values of A. We assume that V_A is finite, for any $A \in U$. By an information tree on the universe U, we mean a tree $T = (N, E)$ such that:

(a) each interior node is labeled by an attribute from U,

(b) each edge is labeled by an attribute value of the attribute that labels the initial node of the edge,

(c) along a path, all nodes (except the leaf) are labeled with different attributes,

(d) all edges leaving a node are labeled with different attribute values (of the attribute that labels the node),

(e) a subset N_l of N is given, each node in N_l is called an *object node*.

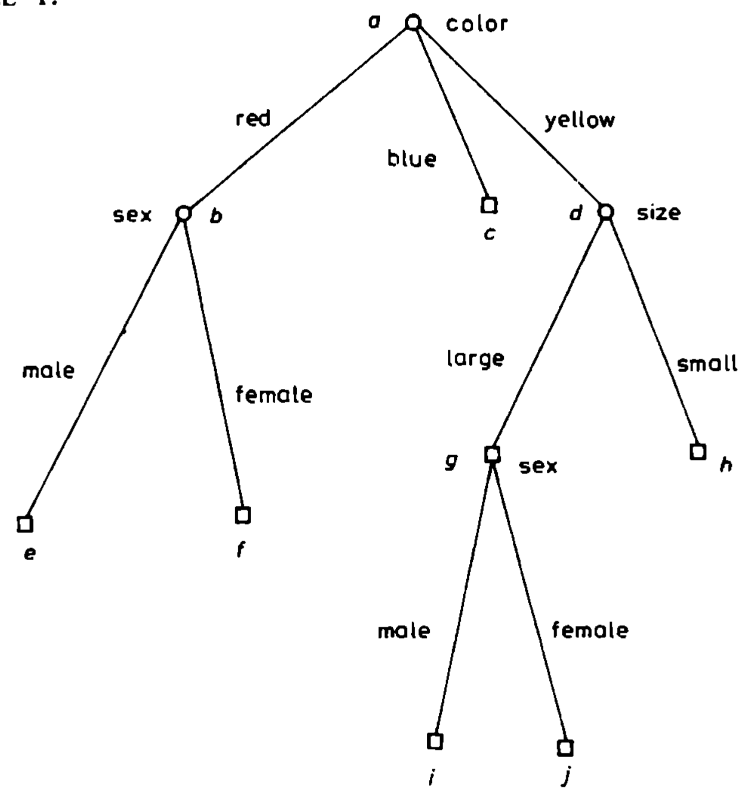
So an information tree can be thought of as a triple (T, l, N_l) where T

$=(N, E)$ is a tree, $N_l \subseteq N$ and l is the labeling function from $N_l \cup E$ into $U \cup (\bigcup_{A \in U} V_A)$. The set N_l is a set of internal nodes in T.

Let m be an object node, $n_1, n_2, n_3, \ldots, n_k$ with $n_k = m$ be the path from the root n_1 to m. Objects node m determines an object type $O(m) = \{[l(n_i), l([n_i, n_{i+1}])]: i = 1, 2, \ldots, k-1\}$ where $l(n_i)$ is the label of the node n_i, which is an attribute. $l([n_i, n_{i+1}])$ is the label of the edge $[n_i, n_{i+1}]$, which is an attribute value of the attribute $l(n_i)$.

An information tree $S = ((N, E), l, N_l)$ determines a set of object types $O(S) = \{O(m)\colon m \in n_l\}$. Two information trees S_1, S_2 are said to be *equivalent* if and only if $O(S_1) = O(S_2)$. If $O(S_1) \subseteq O(S_2)$, we say that S_1 is covered by S_2.

EXAMPLE 1.



The above figure represents an information tree $S = ((N, E), l, N_l)$, where $N = \{a, b, c, d, e, f, g, h, i, j\}$, $E = \{[a, b], [b, e], [b, f], [a, c], [a, d], [d, g], [g, i], [g, j], [d, h]\}$,

$l(a) = $ color, $l([a, b]) = $ red, $l([a, c]) = $ blue,

$l(b) = $ sex, $l([a, d]) = $ yellow, $l([b, e]) = $ male, $l([b, f]) = $ female,

$l(d) = $ size, $l([d, g]) = $ large, $l([d, h]) = $ small, $l(g) = $ sex,

$l([g, i]) = $ male, $l([g, j]) = $ female,

$N_l = \{e, f, c, g, i, j, h\}$.

The object type of the node f is {[color, red], [sex, female]}. This information tree classifies seven different object types (determined by the seven nodes in N_I).

2. Formal theory of information trees

In this section we shall define a formal syntax for representing information trees which is motivated by the LISP representation in [3] and coalgebra representation in [4]. We will introduce axioms and rules of inference for the formal theory of information trees.

Let us use the information tree from Example 1 as a starting point in this section. Assume V_{color} = {red, blue, green, yellow} is ordered as red, blue, green, yellow; V_{sex} = {male, female} is ordered as male, female; V_{size} = {large, medium, small} is ordered as large, medium, small. Then the following term

$$\text{Color}\big(\text{Sex}(-,\ -),\ -,\ *,\ \text{Size}\big(\overline{\text{Sex}}(-,\ -),\ *,\ -\big)\big)$$

retains all the information about the information tree from Example 1. A bar means the node is an object node. A star means the corresponding subtree is empty.

To describe an information tree, we use the general scheme

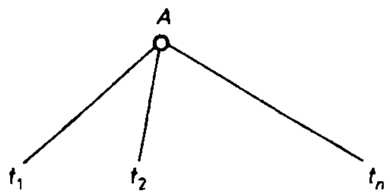
$$\text{attribute}(\text{subtree}_1, \text{subtree}_2, \ldots, \text{subtree}_n)$$

assuming the attribute has n different values.

Now we are ready to introduce a formal theory of information trees over an attribute universe U. There are two constant symbols $*$, $-$ which have the standard interpretation: empty tree and single node tree (tree with one node being an object node) respectively.

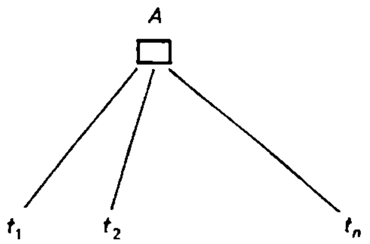
For each atrribute A in U with $|V_A| = n$ (where $|V_A|$ denotes the number of attribute values of A), there are two n-ary function symbols f_A, $\bar{f}_A$. The standard interpretation of $f_A(t_1, t_2, \ldots, t_n)$ is the information tree with the root labeled A and next level subtrees $t_1, t_2, \ldots, t_n$.

We represent $f_A(t_1, t_2, \ldots, t_n)$ by the following graph



The standard interpretation of $\bar{f}_A(t_1, t_2, \ldots, t_n)$ is the information tree with the root labeled A which is an object node and next level subtrees $t_1, t_2, \ldots, t_n$.

We represent $\bar{f}_A(t_1, t_2, \ldots, t_n)$ by the following graph



Function symbol f_A is called a *type* 0 *function symbol*, $\bar{f}_A$ is called a *type* 1 *function symbol*.

There is one predicate symbol $\equiv$. Statement $t_1 \equiv t_2$ in the standard interpretation says that t_1 and t_2 are equivalent.

Terms are defined by the following recursive definition:

DEFINITION OF TERMS. (a) constant symbols are terms,

(b) if g is n-ary function symbol, $t_1, t_2, \ldots, t_n$ are terms not containing g or its dual type function symbol, then $g(t_1, t_2, \ldots, t_n)$ is a term.

Intuitively, each term represents an information tree.

If a term does not contain any type 1 function symbol or the constant symbol $-$, it is called *null object term*.

The nested level $h(t)$ of a term t is defined as follows:

(1) $$h(*) = h(-) = 0,$$

(2) $$h\big(f_A(t_1, t_2, \ldots, t_n)\big) = h\big(\bar{f}_A(t_1, t_2, \ldots, t_n)\big) = \max_{i \leqslant n} h(t_i) + 1.$$

Let t be a term, we use $I(t)$ to denote the standard interpretation of t, i.e., the information tree that t represents.

We have:

$$h(t) = n \quad \text{if and only if the height of } I(t) \text{ is } n.$$

If t is a term then by $\bar{t}$ we mean a new term defined below:

$$\bar{t} = \begin{cases} - & \text{if } t \text{ is } *, \\ \bar{f}_A(t_1, t_2, \ldots, t_n) & \text{if } t \text{ is } f_A(t_1, t_2, \ldots, t_n), \\ t & \text{otherwise.} \end{cases}$$

The formulas are defined by the following recursive definition:

DEFINITION OF FORMULAS. (a) $t_1 \equiv t_2$ is a formula for any two terms t_1, t_2,

(b) $p \wedge q$, $p \vee q$, $p \rightarrow q$, $p \leftrightarrow q$, $\sim p$ are formulas if p, q are formulas.

Our formal theory has the following axiom schemata:

A1. (reflexive) $t \equiv t$ is an axiom for any term t, A2. (nullity) $* \equiv t$ for any null object term t,

A3. (change the order of branching)

$$f(g(t_{1,1}, t_{1,2}, \ldots, t_{1,m}), g(t_{2,1}, t_{2,2}, \ldots, t_{2,m}), \ldots, g(t_{n,1}, t_{n,2}, \ldots, t_{n,m}))$$
$$\equiv g(f(t_{1,1}, t_{2,1}, \ldots, t_{n,1}), f(t_{1,2}, t_{2,2}, \ldots, t_{n,2}), \ldots, f(t_{1,m}, t_{2,m}, \ldots, t_{n,m}))$$

is an axiom for any two type 0 function symbols f, g, where f is n-ary, g is m-ary and for any $n \cdot m$ terms $t_{i,j}$ $(i \leqslant n, j \leqslant m)$ not containing f, g or their type 1 duals,

A4. $p \rightarrow (q \rightarrow p)$ for any formulas p, q,

A5. $(p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$ for any formulas p, q, r,

A6. $(\sim p \rightarrow \sim q) \rightarrow (q \rightarrow p)$ for any formulas p, q.

The rules of inference for our formal system are the following:

R1. from $p \rightarrow q$ and p we can deduce q for any formulas p, q,

R2. from $t_1 \equiv t_2$ we can deduce $t(t_1) \equiv t(t_2)$, where $t(t_1)$ is a term containing t_1 as a subterm and $t(t_2)$ comes from $t(t_1)$ by replacing some of the occurrences of t_1 with t_2,

R3. from $t_1 \equiv t_2$, we can deduce $\bar{t}_1 \equiv \bar{t}_2$.

3. Completeness of the formal theory

In this section we shall prove that the formal theory defined in the second section is complete with respect to the predicate $\equiv$.

Let t be a term, we shall use $I(t)$ to denote the information tree represented by t under the standard interpretation. Then we have the following completeness theorem.

THEOREM 1. $\vdash t_1 \equiv t_2$ *if and only if* $I(t_1)$ *is equivalent to* $I(t_2)$.

Proof (*only if*). All axioms A1, A2, A3 are valid under standard interpretation and rules of inference R1, R2, R3 preserve validity under standard interpretation. Hence all theorems are valid under standard interpretation, so $t_1 \equiv t_2$ implies that $I(t_1)$ is equivalent to $I(t_2)$.

(*if*) By induction on $k = h(t_1)$. Recall that $h(t_1)$ is the nested level of t_1.

Base. $k = 0$. Term t_1 is either $*$ or $-$.

Case 1. Assume that t_1 is $*$. Hence $I(t_1)$ is the empty tree, so $I(t_2)$ cannot have any object nodes, i.e., t_2 is a null object term. By the axiom A2, we have $\vdash t_1 \equiv t_2$.

Case 2. Assume now that t_1 is $-$. Hence t_2 must be of the form $\bar{t}$, where t is a null object term. Since $* \equiv t$ is an axiom for a null object term t, then by R3 we have $- \equiv \bar{t}$.

Induction step. Assume that:

$[I(t_1)$ is equivalent to $I(t_2)]$ implies $\vdash t_1 \equiv t_2$ holds for all terms t_1 with $h(t_1) \leqslant k$.

Consider a term t_1 with $h(t_1) = k+1$. The term t_1 is of the form either $f_A(u_1, u_2, \ldots, u_n)$ or $\bar{f}_A(u_1, u_2, \ldots, u_n)$, where $u_1, u_2, \ldots, u_n$ satisfy the condition $h(u_i) \leqslant k$, $i \leqslant n$.

Let us assume first that t_1 is of the form $f_A(u_1, u_2, \ldots, u_n)$. Assume that t_2 is of the form $f_B(v_1, v_2, \ldots, v_m)$. Observe that for each i, u_i is either a null object term or u_i has an f_B above every type 1 function symbol and an f_B above $-$. So, we may use repeatedly A2, A3 and R2 to get w_i such that $\vdash w_i \equiv u_i$ and w_i is of the form $f_B(z_1, z_2, \ldots, z_m)$. Hence $\vdash t_1 \equiv f_A(w_1, w_2, \ldots, w_n)$. Applying axiom A3 we have: $\vdash f_A(w_1, w_2, \ldots, w_n) \equiv f_B(y_1, y_2, \ldots, y_m)$ for some $y_1, y_2, \ldots, y_m$ with $h(y_i) < h(t_1)$. Hence $I(t_1)$ is equivalent to $I(f_B(y_1, y_2, \ldots, y_m))$. Since $I(t_1)$ is equivalent to $I(f_B(v_1, v_2, \ldots, v_m))$, we have $I(y_i)$ is equivalent to $I(v_i)$ for any i, where $i \leqslant m$. By induction hypothesis we have, $\vdash v_i \equiv y_i$ for $i = 1, 2, \ldots, m$. Hence $\vdash f_B(y_1, y_2, \ldots, y_m) \equiv f_B(v_1, v_2, \ldots, v_m)$ by the axiom A1 and the rule R2. Thus $\vdash t_1 \equiv f_B(v_1, v_2, \ldots, v_m)$.

If t_1 is of the form $\bar{f}_A(u_1, u_2, \ldots, u_n)$, then t_2 must be of the form $\bar{f}_B(v_1, v_2, \ldots, v_m)$ or of the form $-$. The rule R3 will push this case through.

Recall (see [1]) that an optimal information tree for $I(t_0)$ with respect to the storage cost is an information tree with a fewest number of edges among all equivalent information trees to $I(t_0)$. This optimal tree corresponds to a term with a fewest occurrences of function symbols and the constant symbol $-$ among terms in $\{t\colon \vdash t \equiv t_0\}$.

Note that in A3, the left-hand side has $l+n+k$ such symbols (l for f, n for g, k for the function symbols and the constant symbol $-$ in $t_{i,j}$'s) and the right-hand side has $l+m+k$ such symbols. So in simplifying terms, we have the heuristic guidance of moving out a function symbol with fewer arguments, which corresponds to moving up a node with fewer branches in an information tree.

4. An expansion of the formal theory

In this section, we shall introduce a new predicate symbol $\leqslant$. We will prove that the expansion of the formal theory defined in the second section is complete with respect to that predicate symbol.

Statement $t_1 \leqslant t_2$ has the standard interpretation $I(t_1)$ is covered by $I(t_2)$, which means $O(I(t_1)) \subseteq O(I(t_2))$.

We need the following additional axiom schemata:

A7. $t \leqslant t'$ for any null object term t and an arbitrary term t',

A8. $t \leqslant t$ for any term t,

A9. $t_1 \leqslant t_2 \wedge t_2 \leqslant t_1 \leftrightarrow t_1 \equiv t_2$,

A10. $t_1 \leqslant t_2 \wedge t_2 \leqslant t_3 \rightarrow t_1 \leqslant t_3$,

A11. $t \leqslant \bar{t}$,

and the following additional inference rules:

R4. from $t_1 \leqslant t_2$, we can deduce $t(t_1) \leqslant t(t_2)$, where t_1, t_2 are terms, $t(t_2)$ is a term which comes from the term $t(t_1)$ by replacing one or more occurrences of t_1 with t_2.

R5. from $t_1 \leqslant t_2$, we can deduce $\bar{t}_1 \leqslant \bar{t}_2$.

We can prove the following completeness theorem with respect to $\leqslant$.

THEOREM 2. *$\vdash t_1 \leqslant t_2$ if and only if $I(t_1)$ is covered by $I(t_2)$.*

Proof (*only if*). The additional axioms are all valid under standard interpretation. The proof is below:

Axiom A7 holds, since $O(I(t)) = \emptyset \subseteq O(I(t'))$.

Axiom A8 holds, since $O(I(t)) \subseteq O(I(t))$.

Axiom A9 holds, since $O(I(t_1)) \subseteq O(I(t_2))$ and $O(I(t_2)) \subseteq O(I(t_1))$ if and only if $O(I(t_1)) = O(I(t_2))$.

Axiom A10 holds, since $O(I(t_1)) \subseteq O(I(t_2))$ and $O(I(t_2)) \subseteq O(I(t_3))$ imply $O(I(t_1)) \subseteq O(I(t_3))$.

Axiom A11 holds, since $O(I(t)) \subseteq O(I(t)) \cup \{\emptyset\}$.

The additional rules preserve validity under standard interpretation. The proof is below:

Rule R4. $O(I(t_1)) \subseteq O(I(t_2))$ implies $O(I(t(t_1))) \subseteq O(I(t(t_2)))$.

Rule R5. $O(I(t_1)) \subseteq O(I(t_2))$ implies $O(I(t_1)) \cup \{\emptyset\} \subseteq O(I(t_2)) \cup \{\emptyset\}$.

Hence $\vdash t_1 \leqslant t_2$ implies $O(I(t_1)) \subseteq O(I(t_2))$.

(*if*) By induction on $k = h(t_1)$.

Base. $k = 0$. If t_1 is $*$, then by A7 we have $\vdash t_1 \leqslant t_2$. If t_1 is $-$, then t_2 must be $\bar{t}$ for some t. By the axiom A7 we have $\vdash * \leqslant t$ and using the rule R5, $\vdash - \leqslant \bar{t}$.

Induction step. Assume that:

$[I(t_1)$ is covered by $I(t_2)]$ implies $\vdash t_1 \leqslant t_2$ holds for all terms t_1 with $h(t_1) \leqslant k$.

Consider a term t_1 with $h(t_1) = k+1$. Assume t_1 is of the form $f_A(u_1, u_2, \ldots, u_n)$, t_2 is of the form $f_B(v_1, v_2, \ldots, v_m)$. All other cases can be extended from this case without difficulty. For each i, u_i is either a null object term or has an f_B above every type 1 function symbol or constant symbol $-$, so we may repeatedly use A2, A3 and R2 to get w_i such that $\vdash w_i \equiv u_i$ and w_i is of the form $f_B(z_1^i, z_2^i, \ldots, z_m^i)$. Hence $\vdash t_1 \equiv f_A(w_1, w_2, \ldots, w_n)$. By axiom A3, $\vdash f_A(w_1, w_2, \ldots, w_n) \equiv f_B(y_1, y_2, \ldots, y_m)$

for some $y_1, y_2, \ldots, y_m$ with $h(y_i) < h(t_1)$. Since $I(t_1)$ is equivalent to $I(f_B(y_1, y_2, \ldots, y_m))$ and $I(t_1)$ is covered by $I(f_B(v_1, v_2, \ldots, v_m))$ then $I(f_B(y_1, y_2, \ldots, y_m))$ is covered by $I(f_B(v_1, v_2, \ldots, v_m))$. Hence $I(y_i)$ is covered by $I(v_i)$ for each i, $i \leqslant m$. So by induction hypothesis we have, $\vdash y_i \leqslant v_i$ for each $i \leqslant m$.

By rule R4 we have:

$$\vdash f_B(y_1, y_2, \ldots, y_m) \leqslant f_B(v_1, y_2, \ldots, y_m)$$

$$\vdash f_B(v_1, y_2, \ldots, y_m) \leqslant f_B(v_1, v_2, \ldots, y_m)$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\vdash f_B(v_1, v_2, \ldots, v_{m-1}, y_m) \leqslant f_B(v_1, v_2, \ldots, v_m).$$

Thus, $\vdash f_B(y_1, y_2, \ldots, y_m) \leqslant f_B(v_1, v_2, \ldots, v_m)$ by repeated use of A10. Hence $\vdash t_1 \leqslant f_B(v_1, v_2, \ldots, v_m)$ by A9.

5. Conclusion

Information trees investigated by us satisfy the assumption that on the path from the root of a tree to a leaf there cannot be two nodes labeled by the same attribute. We used such trees to implement an experimental dynamic hierarchical data base holding the course grades of Computer Science students at UNC-Charlotte. Trees allowing repeated attributes on a path from the root of a tree to a leaf were investigated by Cockett (see [4]) and used in the implementation of CASCADE system at the University of Tennessee, Knoxville.

References

[1] K. Chen, Z. Ras, *Homogeneous Information Trees*, Fundamenta Informaticae 8 (1985), 123–149.

[2] –, –, *DDH Approach to Information Systems*, Proceedings of the 1982'CISS in Princeton, N. J., 521–526.

[3] –, –, *Dynamic Hierarchical Data Bases*, Proceedings of the 1983'ICAA in Taipei, Taiwan, 450–456.

[4] R. Cockett, *The Algebraic co-Theory of Decision Processes*, Technical Report CS-84-58, University of Tennessee, Knoxville.

[5] B. E. Jacobs, "*On Database Logic*", Journal of the ACM, Vol. 29, No. 2, April 1982, 310–332.

[6] B. M. Moret, M. G. Thomason, R. C. Gonzales, *The Activity of a Variable and its Relation to Decision Trees*, ACM Transactions on Programming Languages and Systems, Vol. 2, No. 4, October, 1980, 580–595.

Presented to the semester
Mathematical Problems in Computation Theory
September 16–December 14, 1985