Sizes of decision tables and decision trees

Hans Zantema and Hans L. Bodlaender
Department of Computer Science, Utrecht University
Padualaan 14, P.O. box 80.089, 3508 TB Utrecht
The Netherlands
e-mail: hansz@cs.uu.nl, hansb@cs.uu.nl

Abstract

Decision tables provide a natural framework for knowledge acquisition and representation in the area of knowledge based information systems. Decision trees provide a standard method for inductive inference in the area of machine learning. In this paper we show how decision tables can be considered as a special kind of decision trees. On the other hand every decision tree can be represented as a decision table, but we show that in worst case the size of this decision table is exponential in the size of the decision tree.

Our main result states that finding a decision table of minimal size that represents the same function as a given decision table is an NP-hard problem; in earlier work we obtained a similar result for decision trees.

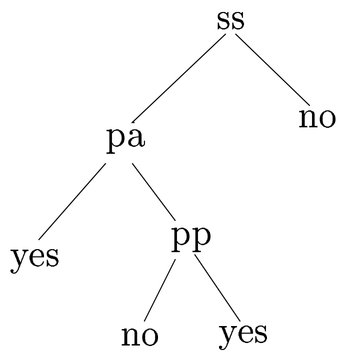
1 Introduction

The concept of a decision tabel is best explained by an example. Assume an order comes in from a client, and the company has to decide whether to execute this order or not. It has to be executed only if the stock is sufficient, and either the client pays in advance or he had never paying problems before. This can be described by the following decision table:

stock sufficient?	yes			no
pays in advance?	yes	no		-
paying problems?	-	yes	no	-
execute order?	yes	no	yes	no

The meaning of such a decision table is clear: handle the questions from up to down and choose the column with the correct answer. In case a bar (-) appears the question may be omitted. Finally the resulting action is found in the lowest line of the corresponding column.

Exactly the same can be described by a decision tree. Using the abbreviations ss, pa and pp for the three questions to be asked, and using the convention that the left branch means 'yes' and the right branch means 'no', we obtain:



Every decision table can be presented in this way as a decision tree, where the columns of the decision table have a one-to-one correspondence to the paths from the root to a leaf in the decision tree. Conversely one can wonder whether every decision tree can be expressed as a decision table. This is not the case: if in one path of the decision tree the question p is above the question q, and in another path it is the other way around, no straightforward decision table representation is possible. It is possible to represent the same function by a decision table, but we prove that in worst case the size of this decision table is exponential in the size of the original decision tree. Here the size of a table or a tree means the total number of decisions occurring in the structure: for a decision tree this is the number of internal nodes and for a decision table this is the number of splitting points. Through the rest of this paper we consider a decision table to be a decision tree in which the questions in a path from the root to a leaf occur in a fixed order.

An earlier comparison of decision tables and decision trees has been given in [7, 8]. However, there the emphasis is on the conceptual level and theoretical questions are not posed. Both notions appear in quite separated areas of research. Decision tables provide a natural framework for knowledge acquisition and representation in the area of knowledge based information systems. Decision trees provide a standard method for inductive inference in the area of machine learning ([6, 5]).

Two decision tables are called decision equivalent if they represent the same function, i.e., they yield the same result for every possible input. As soon as the order in which the questions have to occur is fixed, it is easy to find a minimal representation for a corresponding decision table, simply by omitting all redundant questions. However, if this order is not yet fixed it is a hard problem to find a small decision table that is decision equivalent to a given one. Our main result states that given a decision table and a number, to decide if there is a decision equivalent decision table of size at most that number is NP-complete.

This already holds for decision tables with only binary questions and a binary result value. As a consequence, finding a decision table of minimal size that is decision equivalent to a given decision table is an NP-hard problem. A similar result for decision trees was obtained in [10].

2 Basic definitions and properties

We restrict to the case of binary questions and binary result values. The binary questions are called *attributes*, and the set of attributes is denoted by A. A *decision tree* over A is a binary tree in which every internal node is labelled by an attribute and every leaf is labelled either 1 or 0. More formally, the set D of decision trees is defined to be the smallest set of strings satisfying

- $1 \in D$, and
- $0 \in D$, and
- if $p \in A$ and $T, U \in D$ then $p(T, U) \in D$.

For any decision tree T let $\mathsf{attr}(T)$ be the set of attributes occurring in T, defined inductively by

$$\mathsf{attr}(1) = \mathsf{attr}(0) = \emptyset,$$

$$\mathsf{attr}(p(T,U)) = \{p\} \cup \mathsf{attr}(T) \cup \mathsf{attr}(U) \quad \text{for all } p \in A \text{ and all } T, U \in D.$$

The size $\#T$ of a decision tree T is defined to be the number of internal nodes of T, inductively defined by

$$\#0 = \#1 = 0, \quad \#p(T,U) = 1 + \#T + \#U.$$

Every attribute can have either the value 1 or the value 0. If every attribute has such a value we speak about an *instance*, Introducing the convention that in a decision tree the left branch of a node p corresponds to p taking the value 1 and the right branch corresponds to 0, a boolean value $\phi(T, s)$ can be assigned to every decision tree T and every instance s, inductively defined as follows

$$\begin{array}{rcll} \phi(1,s) & = & 1 & \\ \phi(0,s) & = & 0 & \\ \phi(p(T,U),s) & = & \phi(T,s) & \text{if } s(p) = 1 \\ \phi(p(T,U),s) & = & \phi(U,s) & \text{if } s(p) = 0. \end{array}$$

Alternatively, in propositional notation the last two lines can be written as

$$\phi(p(T,U),s) \;=\; (s(p) \wedge \phi(T,s)) \vee (\neg s(p) \wedge \phi(U,s)),$$

or equivalently as $\phi(p(T,U),s) \;=\; (s(p) \to \phi(T,s)) \wedge (\neg s(p) \to \phi(U,s))$.

Definition 1 *Two decision trees T and U are called* decision equivalent, *or shortly* equivalent, *denoted as $T \simeq U$, if*

$$\phi(T,s) = \phi(U,s) \quad \textit{for all } s : A \to \{1,0\}.$$

In [9] it was proved that the equations

$$\begin{array}{rcl} p(x,x) & = & x \\ p(q(x,y),q(z,w)) & = & q(p(x,z),p(y,w)) \\ p(p(x,y),z) & = & p(x,z) \\ p(x,p(y,z)) & = & p(x,z) \end{array}$$

for all $p, q \in A$ comprise a sound and complete finite equational axiomatization for decision equivalence. Also a quadratic algorithm was given for determining whether two decision trees are equivalent.

Definition 2 *A decision tree T is called* compatible *with a total order $<$ on $\mathsf{attr}(T)$ if for every non-root node of T its label is bigger than the label of its parent with respect to $<$.*

A decision tree T is called a decision table *if there exists a total order $<$ on $\mathsf{attr}(T)$ such that T is compatible with $<$.*

A set $\{T_1, \ldots, T_n\}$ of decision tables is called compatible *if there exists a total order $<$ on $\bigcup_{i=1}^n \mathsf{attr}(T_i)$ such that T_i is compatible with $<$ restricted to $\mathsf{attr}(T_i)$, for all $i = 1, \ldots, n$.*

For a decision tree T and an attribute $p \in \mathsf{attr}(T)$ we define inductively:

$$\begin{array}{lcll} 0^{\leftarrow p} & = & 0; & \\ 1^{\leftarrow p} & = & 1; & \\ p(T,U)^{\leftarrow p} & = & T^{\leftarrow p}; & \\ q(T,U)^{\leftarrow p} & = & q(T^{\leftarrow p}, U^{\leftarrow p}) & \text{for } q \neq p; \\ 0^{\rightarrow p} & = & 0; & \\ 1^{\rightarrow p} & = & 1; & \\ p(T,U)^{\rightarrow p} & = & U^{\rightarrow p}; & \\ q(T,U)^{\rightarrow p} & = & q(T^{\rightarrow p}, U^{\rightarrow p}) & \text{for } q \neq p. \end{array}$$

Hence $T^{\leftarrow p}$ is obtained from T by removing all occurrences of p and their right arguments, while $T^{\rightarrow p}$ is obtained from T by removing all occurrences of p and their left arguments.

For a decision tree T and a total order $<$ on a set A of attributes satisfying $\mathsf{attr}(T) \subseteq A$ we define inductively its *normal form* $N(T,<)$:

$$\begin{array}{rcll} N(0,<) & = & 0; & \\ N(1,<) & = & 1; & \\ N(T,<) & = & p(N(T^{\leftarrow p},<), N(T^{\rightarrow p},<)) & \text{if } N(T^{\leftarrow p},<) \neq N(T^{\rightarrow p},<); \\ N(T,<) & = & N(T^{\leftarrow p},<) & \text{if } N(T^{\leftarrow p},<) = N(T^{\rightarrow p},<); \end{array}$$

where in the last two lines p denotes the smallest element of $\mathsf{attr}(T)$ with respect to the order $<$.

Clearly $N(T,<)$ is a decision table equivalent to T and compatible with $<$, and it is the smallest decision table having this property.

The following theorem states that for every total order on the attributes the normal form is a unique representative for the equivalence class of a decision tree. In order to prove it we need a lemma.

Lemma 3 *Let decision trees T and U satisfy $T \simeq U$, and let p be an attribute. Then $T^{\leftarrow p} \simeq U^{\leftarrow p}$ and $T^{\rightarrow p} \simeq U^{\rightarrow p}$.*

Proof: By symmetry it suffices to prove $T^{\leftarrow p} \simeq U^{\leftarrow p}$. Let s be an arbitrary instance. Define instance s_1 by $s_1(p) = 1$ and $s_1(q) = s(q)$ for $q \neq p$. Then

$$\phi(T^{\leftarrow p}, s) = \phi(T, s_1) = \phi(U, s_1) = \phi(U^{\leftarrow p}, s).$$

□

Theorem 4 *Let $<$ be a total order on the set of attributes. Then*

- *$N(T,<) \simeq T$ for every decision tree T;*
- *$T \simeq U$ if and only if $N(T,<) = N(U,<)$ for decision trees T, U.*

Proof: The first assertion follows by induction on the structure of T, using that $T \simeq p(T^{\leftarrow p}, T^{\rightarrow p})$ and $T \simeq p(T,T)$ for all T, p.

The 'if'-part of the second assertion follows from the first assertion.

For the 'only if'-part of the second assertion we apply induction on the number of attributes occurring in T and U. First assume that the smallest occurring attribute p occurs both in T and in U. From $T \simeq U$ we conclude from Lemma 3 that $T^{\leftarrow p} \simeq U^{\leftarrow p}$ and $T^{\rightarrow p} \simeq U^{\rightarrow p}$; from the induction hypothesis we conclude $N(T^{\leftarrow p},<) = N(U^{\leftarrow p},<)$ and $N(T^{\rightarrow p},<) = N(U^{\rightarrow p},<)$. If $N(T^{\leftarrow p},<) = N(T^{\rightarrow p},<)$ we also have $N(U^{\leftarrow p},<) = N(U^{\rightarrow p},<)$, hence

$$N(T,<) = N(T^{\leftarrow p},<) = N(U^{\leftarrow p},<) = N(U,<);$$

if not then

$$N(T,<) = p(N(T^{\leftarrow p},<), N(T^{\rightarrow p},<)) = p(N(U^{\leftarrow p},<), N(U^{\rightarrow p},<)) = N(U,<).$$

In both cases we have $N(T,<) = N(U,<)$, which we had to prove.

In the remaining case the smallest occurring attribute p occurs only in one of the two decision trees T and U. By symmetry we may assume that p occurs in T and p does not occur in U. Since $T \simeq U$ we obtain from Lemma 3 that $T^{\leftarrow p} \simeq U^{\leftarrow p} = U = U^{\rightarrow p} \simeq T^{\rightarrow p}$. From the induction hypothesis we conclude that

$N(T^{\leftarrow p}, <) = N(T^{\rightarrow p}, <)$. From the definition of N and the induction hypothesis applied on $T^{\leftarrow p} \simeq U$ we obtain

$$N(T, <) = N(T^{\leftarrow p}, <) = N(U, <),$$

which we had to prove. □

3 Comparing sizes of tables and trees

Since every decision tree can be considered as a decision table, clearly every decision table can be represented as a decision tree of the same size. In this section we address the converse question: given a decision tree of some size, what can be said about the size of an equivalent decision table? It turns out that in worst case the minimal size of an equivalent decision table is exponential in the size of the original decision tree. In order to prove this for any natural number n we choose distinct attributes $p_1, p_2, \ldots, p_n, q_1, q_2, \ldots, q_n, r$ and we inductively define

$$T_0 = U_0 = 0, \quad T_i = p_i(q_i(1,0), T_{i-1}), \quad U_i = q_i(p_i(1,0), U_{i-1}),$$

for $i = 1, \ldots, n$, and $V = r(T_n, U_n)$. Clearly V is a decision tree of size $\#V = 4n + 1$, but V is not a decision table. In fact, we will show that for every decision table that is equivalent to V the size exceeds a function that is exponential in n. First we need a lemma.

Lemma 5 *Let T be a decision table compatible with an order $<$ on the attributes satisfying $T \simeq T_i$. Let $k = \#\{j \in \{1, \ldots, i\} | q_j < p_j\}$. Then $\#T \geq 2^k - 1$.*

Proof: We apply induction on i. For $i = 0$ we have $k = 0$ and $\#T \geq 0 = 2^k - 1$.

Assume $i > 0$. Then the root of T is either p_m or q_m for some $m \in \{1, \ldots, i\}$.

Assume the root of T is p_m. Then $T = p_m(T', T'')$, where $T'' \simeq T_i^{\rightarrow p_m}$ by Lemma 3. Since $T_i^{\rightarrow p_m}$ equals T_{i-1} up to renaming of indices of p and q, we may apply the induction hypothesis on T''. Since p_m is the smallest attribute with respect to $<$, omitting m from $\{1, \ldots, i\}$ does not affect the corresponding value of k. Hence by the induction hypothesis we have $\#T'' \geq 2^k - 1$. We conclude

$$\#T > \#T'' \geq 2^k - 1.$$

For the remaining case assume the root of T is q_m. Then $T = q_m(T', T'')$, where $T' \simeq T_i^{\leftarrow q_m}$ and $T'' \simeq T_i^{\rightarrow q_m}$ by Lemma 3. Since $(T_i^{\leftarrow q_m})^{\rightarrow p_m} = (T_i^{\rightarrow q_m})^{\rightarrow p_m}$ equals T_{i-1} up to renaming of indices of p and q, we may apply the induction hypothesis on both $T'^{\rightarrow p_m}$ and $T''^{\rightarrow p_m}$. Now the corresponding value for k decreases by one since p_m and q_m satisfying $q_m < p_m$ have been removed. By the induction hypothesis we obtain $\#T'^{\to p_m} \geq 2^{k-1} - 1$ and $\#T''^{\to p_m} \geq 2^{k-1} - 1$. We conclude

$$\#T = 1 + \#T' + \#T'' \geq 1 + \#T'^{\to p_m} + \#T''^{\to p_m} \geq 1 + 2 * (2^{k-1} - 1) = 2^k - 1.$$

□

Theorem 6 *Let the decision tree V of size $4n+1$ be defined as above and let T be a decision table satisfying $T \simeq V$. Then $\#T \geq 2^{n/2}$.*

Proof: Let $<$ be the order on the attributes such that T is compatible with $<$. By symmetry between p's and q's and between T_n and U_n, without loss of generality we may assume that $k = \#\{j \in \{1, \ldots, n\} | q_j < p_j\} \geq n/2$. From Lemma 3 we obtain $T^{\leftarrow r} \simeq V^{\leftarrow r} = T_n$. From Lemma 5 we conclude $\#T^{\leftarrow r} \geq 2^k - 1 \geq 2^{n/2} - 1$. Since $T \simeq V = r(T_n, U_n)$ and $T_n \not\simeq U_n$ the symbol r occurs in T. Hence

$$\#T \geq 1 + \#T^{\leftarrow r} \geq 2^{n/2}.$$

□

4 Finding small decision tables is hard

Theorem 7 *Given a number $T_1, \ldots, T_n$ of compatible decision tables and a number k, it is NP-complete to decide whether a total order $<$ on $\bigcup_{i=1}^{n} \mathsf{attr}(T_i)$ exists satisfying*

$$\sum_{i=1}^{n} \#N(T_i, <) \leq k.$$

Proof: Clearly, the problem belongs to NP.

To prove that the problem is NP-complete, we use a transformation from the Feedback Arc Set problem. In this problem, we ask for a given directed graph $G = (V, E)$ without self-loops and an integer K, whether there exists a subset of arcs $E' \subseteq E$ with $|E'| \leq K$, such that every directed cycle in G contains at least one arc in E'. This problem was proved to be NP-complete by Karp in 1972 [3].

Suppose we are given a directed graph $G = (V, E)$ without self-loops, i.e., $(v, v) \notin E$ for all $v \in V$, and an integer K. Let $n = |E|$. We will create a collection $\mathcal{T} = \{T_1, \ldots, T_n\}$ of decision tables, in the following way. Each vertex in V corresponds to one attribute to be used in the tables. In addition, we have one attribute $p \notin V$.

For every edge $(v, w) \in E$ we define $T^1_{(v,w)} = v(w(p(0,1),0),1)$ as shown left in Figure 1 and $T^2_{(v,w)} = w(v(p(0,1),1), v(0,1))$ as shown right in Figure 1. Note that $T^1_{(v,w)} \simeq T^2_{(v,w)}$.

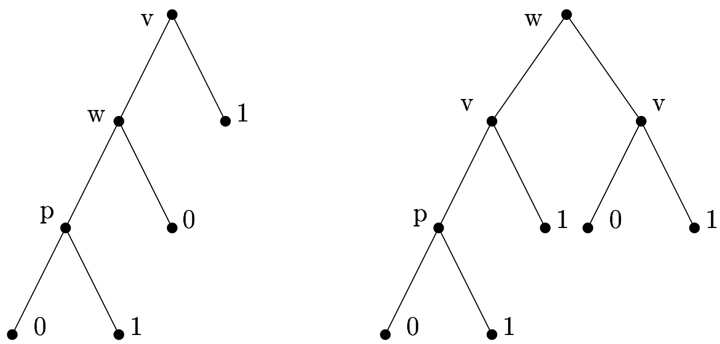


Figure 1: Decision tables $T^1_{(v,w)}$ and $T^2_{(v,w)}$

For every edge $(v, w) \in E$ we construct one table T_i in the collection $\mathcal{T}$. This construction depends on a fixed arbitrary total order $\ll$ on V. If $v \ll w$, then we choose T_i to be $T^1_{(v,w)}$, otherwise, if $w \ll v$, we choose T_i to be $T^2_{(v,w)}$. Note that every T_i is a decision table compatible with $\ll$, where $\ll$ is extended to $V\{p\}$ by defining $v \ll p$ for all $v \in V$. Hence $\mathcal{T} = \{T_1, \ldots, T_n\}$ is a compatible set of decision tables.

Claim 8 *Let for an arbitrary directed graph $G = (V, E)$ without self-loops the set $\mathcal{T} = \{T_1, \ldots, T_n\}$ be defined as above. Then the following statements are equivalent.*

1. *There is a set $E' \subseteq E$, such that $|E'| \leq K$, and every directed cycle in G contains at least one vertex from E'.*

2. *There is a total order $\prec$ on the vertices of V, such that*

$$|\{(v, w) \in E \mid w \prec v\}| \leq K.$$

3. *There exists a total order $<$ on $\bigcup_{i=1}^{n} \mathsf{attr}(T_i)$, satisfying*

$$\sum_{i=1}^{n} \#N(T_i, <) \leq 3 \cdot |E| + K.$$

Proof: (1) $\Rightarrow$ (2): Remove all edges from E'. Then the remaining graph $(V, E \setminus E')$ is acyclic. Topological sorting of this remaining graph yields a total order $\prec$ on V satisfying $v \prec w$ for all $(v, w) \in E \setminus E'$. Since $|E'| \leq K$ we obtain $|\{(v, w) \in E \mid w \prec v\}| \leq K$.

(2) $\Rightarrow$ (1): Define $E' = \{(v, w) \in E \mid w \prec v\}$. Assume there is a cycle $v_1, v_2, \ldots v_k, v_1$ not containing an edge from E'. Since the graph does not contain self-loops and the order $\prec$ is total we obtain $v_1 \prec v_2 \prec \cdots \prec v_k \prec v_1$, contradiction.

(2) $\Rightarrow$ (3): Note that $\bigcup_{i=1}^{n} \mathsf{attr}(T_i) = V \cup \{p\}$. Suppose $\prec$ is a total order on V with $|\{(v,w) \in E \mid w \prec v\}| \leq K$. Now, take the total order $<$ on attribute set $V \cup \{x\}$, by taking for each pair $v, w \in V$, $v < w \Leftrightarrow v \prec w$, and for each $v \in V$, $v < x$. For arbitrary $T_i \in \mathcal{T}$ let T_i correspond to edge (v,w). By definition T_i is either $T^1_{(v,w)}$ or $T^2_{(v,w)}$; from the definition of $N(-,<)$ we obtain that $N(T_i,<) = T^1_{(v,w)}$ if $v < w$, and $N(T_i,<) = T^2_{(v,w)}$ if $v > w$. As the latter case occurs at most K times, and $\#T^1_{(v,w)} = 3$ and $\#T^2_{(v,w)} = 4$ we have that $\sum_{i=1}^{n} \#N(T_i,<) \leq 3 \cdot |E| + K$.

(3) $\Rightarrow$ (2): Let $<$ be a total order on $\bigcup_{i=1}^{n} \mathsf{attr}(T_i)$ with $\sum_{i=1}^{n} \#N(T_i,<) \leq 3 \cdot |E| + K$. Let $\prec$ be the total order on V with for all $v, w \in V$: $v \prec w \Leftrightarrow v < w$. For every edge $(v,w) \in E$ corresponding to tree $T_i \in \mathcal{T}$, we have that $\#N(T_i,<) \geq 3$, as $N(T_i,<)$ must contain attributes v, w, and p. Now consider edge $(v,w) \in E$ with $w \prec v$. Let T_i be the corresponding decision tree in $\mathcal{T}$. There are three cases: $p < w < v$, $w < p < v$, and $w < v < p$, respectively yielding $N(T_i,<) = p(v(0,1), w(1, v(0,1)))$, $N(T_i,<) = w(p(v(0,1),1), v(0,1))$ and $N(T_i,<) = w(v(p(0,1),1), v(0,1))$. In all of these cases we have $\#N(T_i,<) = 4$. We obtain

$$3 \cdot |E| + |\{(v,w) \in E \mid w \prec v\}| \leq \sum_{i=1}^{n} \#N(T_i,<) \leq 3 \cdot |E| + K,$$

hence $|\{(v,w) \in E \mid w \prec v\} \leq K$. □

From the equivalence (1) $\Leftrightarrow$ (3) from this claim, the NP-completeness of the Feedback Arc Set problem, and the fact that the construction of the tables can be done in polynomial time, Theorem 7 now follows. □

In order to prove that for a single decision tree T and a number k it is NP-complete to decide whether a total order $<$ exists satisfying $\#N(T,<) \leq k$, we give a construction to code the number $T_1, \ldots, T_n$ of decision trees in a single decision tree. In order to show that the extra attributes for giving this construction do not affect the desired sizes of decision tables we give a number of lemmas.

Lemma 9 *Let T be a decision table having $p_0(0,1)$ as a subtable for a particular attribute p_0 and in which no other occurrence of p_0 exists. Let $<$ be a total order on $\mathsf{attr}(T)$. Let $<'$ be the total order on $\mathsf{attr}(T)$ obtained from $<$ by forcing p_0 to be the greatest element, more precisely: $q <' r$ if and only if*

$$(r = p_0 \wedge q \neq p_0) \vee (q < r \wedge q \neq p_0 \neq r).$$

Then $\#N(T,<') \leq \#N(T,<)$.

Proof: We apply induction on $\#\mathsf{attr}(T)$. If $\#\mathsf{attr}(T) = 1$ then $T = p_0(0,1)$ and $<=<'$, hence $\#N(T,<') = \#N(T,<)$.

For the induction step write $N(T,<) = q(T_1, T_2)$.

First assume $q \neq p_0$. Then $q(N(T^{\leftarrow q},<'), N(T^{\rightarrow q},<'))$ is a decision table compatible with $<'$ and equivalent to T, hence $\#N(T,<') \leq \#q(N(T^{\leftarrow q},<'), N(T^{\rightarrow q},<'))$. If p_0 occurs in $T^{\leftarrow q}$ we may apply the induction hypothesis on $T^{\leftarrow q}$, yielding $\#N(T^{\leftarrow q},<') \leq \#N(T^{\leftarrow q},<)$. If p_0 does not occur in $T^{\leftarrow q}$ then $N(T^{\leftarrow q},<') = N(T^{\leftarrow q},<)$. Hence in all cases we have $\#N(T^{\leftarrow q},<') \leq \#N(T^{\leftarrow q},<)$. Similarly applying the induction hypothesis on $T^{\rightarrow q}$ yields $\#N(T^{\rightarrow q},<') \leq \#N(T^{\rightarrow q},<)$. Finally we have $T_1 = N(T^{\leftarrow q},<)$ and $T_2 = N(T^{\rightarrow q},<)$. Combining all these observations gives

$$\begin{aligned}
\#N(T,<') &\leq \#q(N(T^{\leftarrow q},<'), N(T^{\rightarrow q},<')) \\
&\leq \#q(N(T^{\leftarrow q},<), N(T^{\rightarrow q},<)) \\
&= \#q(T_1, T_2) \\
&= \#N(T,<)
\end{aligned}$$

which we had to prove.

In the remaining case we have $N(T,<) = p_0(T_1, T_2)$ and p_0 is the smallest attribute with respect to $<$. Let r be the smallest but one attribute with respect to $<$. First assume that both T_1 and T_2 have r as the root, then we can write $T_1 = r(U_1, V_1)$ and $T_2 = r(U_2, V_2)$. Then $p_0(U_1, U_2)$ is compatible with $<$ and equivalent to $T^{\leftarrow r}$, hence $\#N(T^{\leftarrow r},<) \leq \#p_0(U_1, U_2)$. The decision table $p_0(V_1, V_2)$ is compatible with $<$ and equivalent to $T^{\rightarrow r}$, hence $\#N(T^{\rightarrow r},<) \leq \#p_0(V_1, V_2)$. Since $r(N(T^{\leftarrow r},<'), N(T^{\rightarrow r},<'))$ is a decision table compatible with $<'$ and equivalent to T we have $\#N(T,<') \leq \#r(N(T^{\leftarrow r},<'), N(T^{\rightarrow r},<'))$. As above we apply the induction hypothesis to obtain $\#N(T^{\leftarrow r},<') \leq \#N(T^{\leftarrow r},<)$ and $\#N(T^{\rightarrow r},<') \leq \#N(T^{\rightarrow r},<)$. Combining all these observations gives

$$\begin{aligned}
\#N(T,<') &\leq \#r(N(T^{\leftarrow r},<'), N(T^{\rightarrow r},<')) \\
&\leq \#r(N(T^{\leftarrow r},<), N(T^{\rightarrow r},<)) \\
&\leq \#r(p_0(U_1, U_2), p_0(V_1, V_2)) \\
&= \#p_0(r(U_1, V_1), r(U_2, V_2)) \\
&= \#N(T,<)
\end{aligned}$$

which we had to prove.

In the remaining case we have that not both T_1 and T_2 have r as the root. If none of them has r as the root then r does not occur in $N(T,<)$ and T is equivalent to a decision table in which r does not occur, hence $N(T,<) = N(T^{\leftarrow r},<)$ and $N(T,<') = N(T^{\leftarrow r},<')$. Applying the induction hypothesis on $T^{\leftarrow r}$ yields

$$\#N(T,<') = \#N(T^{\leftarrow r},<') \leq \#N(T^{\leftarrow r},<) = \#N(T,<).$$

In the remaining case one of T_1 and T_2 has r as the root and the other has not. By symmetry we may assume it is T_1, hence we may write $T_1 = r(U_1, V_1)$

and r does not occur in T_2. So r occurs in $N(T^{\leftarrow p_0}, <) = T_1$ and r does not occur in $N(T^{\rightarrow p_0}, <) = T_2$. Hence in all occurrences of r in $T^{\rightarrow p_0}$ the left argument is equivalent to the right argument, while this does not hold for all occurrences of r in $T^{\leftarrow p_0}$. Since $T^{\leftarrow p_0}$ is obtained from T by replacing the single occurrence of p_0 in T by 0 and $T^{\rightarrow p_0}$ is obtained from T by replacing the single occurrence of p_0 in T by 1, we conclude that there is an occurrence of r in T such that the single occurrence of p_0 in T is either in the left or in the right argument of that occurrence of r. Hence either p_0 is not in $T^{\leftarrow r}$ or p_0 is not in $T^{\rightarrow r}$, which is the last case analysis to conclude the proof.

Assume that p_0 is not in $T^{\leftarrow r}$. Since $T^{\leftarrow r} \simeq p_0(U_1, T_2)$ we conclude that $U_1 \simeq T_2$. Now $r(U_1, N(T^{\rightarrow r}, <'))$ is a decision table compatible with $<'$ that is equivalent to T, hence $\#N(T, <') \leq \#r(U_1, N(T^{\rightarrow r}, <'))$. Applying the induction hypothesis yields $\#N(T^{\rightarrow r}, <') \leq \#N(T^{\rightarrow r}, <)$. Since $p_0(V_1, T_2)$ is a decision table compatible with $<$ and equivalent to $T^{\rightarrow r}$ we obtain $\#N(T^{\rightarrow r}, <) \leq \#p_0(V_1, T_2)$. Combining these observations gives

$$\begin{array}{rcl} \#N(T, <') & \leq & \#r(U_1, N(T^{\rightarrow r}, <')) \\ & \leq & \#r(U_1, N(T^{\rightarrow r}, <)) \\ & \leq & \#r(U_1, p_0(V_1, T_2)) \\ & = & \#p_0(r(U_1, V_1), T_2) \\ & = & \#N(T, <) \end{array}$$

which we had to prove. In the last case where p_0 is not in $T^{\rightarrow r}$ we similarly obtain

$$\begin{array}{rcl} \#N(T, <') & \leq & \#r(N(T^{\leftarrow r}, <'), V_1) \\ & \leq & \#r(N(T^{\leftarrow r}, <), V_1) \\ & \leq & \#r(p_0(U_1, T_2), V_1) \\ & = & \#p_0(r(U_1, V_1), T_2) \\ & = & \#N(T, <), \end{array}$$

concluding the proof. $\square$

Let $\{T_1, \ldots, T_n\}$ be a set of compatible decision tables. Let $p_0, p_1, \ldots, p_n$ be attributes not occurring in $\bigcup_1^n \mathsf{attr}(T_i)$. Define inductively

$$\begin{array}{rcl} S_0 & = & p_0(0, 1), \\ S_k & = & p_k(S_{k-1}, T_k) \quad \text{for } k = 1, \ldots, n. \end{array}$$

Lemma 9 shows that in a decision table equivalent to S_n the attribute p_0 can be thrown downwards without increase of size. The following lemma states that in a decision table equivalent to S_n the attribute p_n can be forced to occur only as the root without increase of size.

Lemma 10 *Let $\{T_1, \ldots, T_n\}$ be a set of compatible decision tables for $n \geq 1$ and let S_n be defined as above. Let $<$ be a total order on $\mathsf{attr}(S_n)$ for which p_0 is the* *greatest element. Let* $\tilde{<}$ *be the total order on* $\mathsf{attr}(S_n)$ *obtained from* $<$ *by forcing* p_n *to be the smallest element, more precisely:* $q\tilde{<}r$ *if and only if*

$$(q = p_n \wedge r \neq p_n) \vee (q < r \wedge q \neq p_n \neq r).$$

Then $\#N(S_n, \tilde{<}) = 1 + \#N(S_{n-1}, <) + \#N(T_n, <) \leq \#N(S_n, <)$.

Proof: We apply induction on $\#\mathsf{attr}(S_n)$. If $\#\mathsf{attr}(S_n) = 1$ then $\tilde{<} =<$ and the statement clearly holds.

Let $\#\mathsf{attr}(S_n) > 1$. Let q be the root symbol of $N(S_n, <)$. If $q = p_0$ then $N(S_n, <) = p_0(0, 1)$ since p_0 is the greatest element with respect to $<$. This is not true since S_n is not equivalent to $p_0(0, 1)$. Hence we have $q \neq p_0$.

By definition we have $N(S_n, <) = q(N(S_n^{\leftarrow q}, <), N(S_n^{\rightarrow q}, <))$. A number of times we will apply the following claim that is immediate from the definition:

> **Claim:** If r is the smallest attribute with respect to $\prec$, then for every decision tree U we have
>
> $$\#N(U, \prec) \leq 1 + \#N(U^{\leftarrow r}, \prec) + \#N(U^{\rightarrow r}, \prec),$$
>
> while equality holds if $U^{\leftarrow r} \not\approx U^{\rightarrow r}$.

As a consequence of this claim and the definition of S_n we obtain the required equality

$$\#N(S_n, \tilde{<}) = 1 + \#N(S_{n-1}, <) + \#N(T_n, <).$$

If $q = p_n$ then $\tilde{<} =<$ and we are done. For the remaining possibilities for q we distinguish two cases: $q \in \{p_1, \ldots, p_{n-1}\}$ and $q \in \bigcup_{i=1}^{n} \mathsf{attr}(T_i)$.

Assume $q \in \{p_1, \ldots, p_{n-1}\}$. Then we have

$$\begin{array}{lll}
\multicolumn{2}{l}{\#N(S_n, \tilde{<})} & \\
= & 1 + \#N(S_{n-1}, <) + \#N(T_n, <) & \\
\leq & 2 + \#N(S_{n-1}^{\leftarrow q}, <) + \#N(S_{n-1}^{\rightarrow q}, <) + \#N(T_n, <) & \text{(the claim)} \\
\leq & 2 + \#N(S_{n-1}^{\leftarrow q}, <) + \#N(S_n^{\rightarrow q}, <) + \#N(T_n, <) & \\
= & 2 + \#N((S_n^{\leftarrow q})^{\leftarrow p_n}, <) + \#N((S_n^{\leftarrow q})^{\rightarrow p_n}, <) + \#N(S_n^{\rightarrow q}, <) & \text{(def. of } S_n) \\
= & 2 + \#N((S_n^{\leftarrow q})^{\leftarrow p_n}, \tilde{<}) + \#N((S_n^{\leftarrow q})^{\rightarrow p_n}, \tilde{<}) + \#N(S_n^{\rightarrow q}, <) & \\
= & 1 + \#N(S_n^{\leftarrow q}, \tilde{<}) + \#N(S_n^{\rightarrow q}, <) & \text{(the claim)} \\
\leq & 1 + \#N(S_n^{\leftarrow q}, <) + \#N(S_n^{\rightarrow q}, <) & \text{(ind. hyp.)} \\
= & \#N(S_n, <). &
\end{array}$$

For the remaining case we have $q \in \bigcup_{i=1}^{n} \mathsf{attr}(T_i)$. By the induction hypothesis we both have $\#N(S_n^{\leftarrow q}, \tilde{<}) \leq \#N(S_n^{\leftarrow q}, <)$ and $\#N(S_n^{\rightarrow q}, \tilde{<}) \leq \#N(S_n^{\rightarrow q}, <)$.

Applying the claim a number of times we obtain

$$\begin{array}{rcl}\#N(S_n, \tilde{<}) & \leq & 1 + \#N(S_n^{\leftarrow p_n}, \tilde{<}) + \#N(S_n^{\rightarrow p_n}, \tilde{<}) \\ & \leq & 3 + \#N((S_n^{\leftarrow p_n})^{\leftarrow q}, \tilde{<}) + \#N((S_n^{\leftarrow p_n})^{\rightarrow q}, \tilde{<}) \\ & & \quad + \#N((S_n^{\rightarrow p_n})^{\leftarrow q}, \tilde{<}) + \#N((S_n^{\rightarrow p_n})^{\rightarrow q}, \tilde{<}) \\ & = & 3 + \#N((S_n^{\leftarrow q})^{\leftarrow p_n}, \tilde{<}) + \#N((S_n^{\leftarrow q})^{\rightarrow p_n}, \tilde{<}) \\ & & \quad + \#N((S_n^{\rightarrow q})^{\leftarrow p_n}, \tilde{<}) + \#N((S_n^{\rightarrow q})^{\rightarrow p_n}, \tilde{<}) \\ & = & 1 + \#N(S_n^{\leftarrow q}, \tilde{<}) + \#N(S_n^{\rightarrow q}, \tilde{<}) \\ & \leq & 1 + \#N(S_n^{\leftarrow q}, <) + \#N(S_n^{\rightarrow q}, <) \\ & = & \#N(S_n, <),\end{array}$$

concluding the proof. □

Lemma 11 *Let $\{T_1, \ldots, T_n\}$ be a set of decision trees and let S_n be defined as above. Let $<$ be an arbitrary total order on $\mathsf{attr}(S_n)$ and let $\hat{<}$ be the total order on $\mathsf{attr}(S_n)$ defined by*

$$p_n \mathrel{\hat{<}} p_{n-1} \mathrel{\hat{<}} \cdots \mathrel{\hat{<}} p_1 \mathrel{\hat{<}} q \mathrel{\hat{<}} p_0$$

for every $q \in \bigcup_{i=1}^n \mathsf{attr}(T_i)$, and $q \mathrel{\hat{<}} r$ if and only if $q < r$ for every $q, r \in \bigcup_{i=1}^n \mathsf{attr}(T_i)$. Then

$$\#N(S_n, \hat{<}) \leq \#N(S_n, <).$$

Proof: By Lemma 9 it suffices to prove this lemma for the case that p_0 is the greatest element of $\mathsf{attr}(S_n)$ with respect to $<$. By induction on j we prove that $\#N(S_j, \hat{<}) \leq \#N(S_j, <)$ for every $j = 0, \ldots, n$, from which the lemma follows by taking $j = n$. For $j = 0$ this holds since then $N(S_j, \hat{<}) = p_0(0,1) = N(S_j, <)$.

For the induction step let $j > 0$. Then we obtain

$$\begin{array}{rcll}\#N(S_j, \hat{<}) & = & \#p_j(N(S_{j-1}, \hat{<}), N(T_j, \hat{<})) & \\ & = & \#p_j(N(S_{j-1}, \hat{<}), N(T_j, <)) & \\ & = & 1 + \#N(S_{j-1}, \hat{<}) + \#N(T_j, <) & \\ & \leq & 1 + \#N(S_{j-1}, <) + \#N(T_j, <) & \text{(ind. hypoth.)} \\ & \leq & \#N(S_j, <) & \text{(Lemma 10),}\end{array}$$

concluding the proof. □

Lemma 12 *Let $\{T_1, \ldots, T_n\}$ be a set of decision trees and let S_n be defined as above. Let $\prec$ be a total order on $\mathsf{attr}(S_n)$ satisfying*

$$p_n \prec p_{n-1} \prec \cdots \prec p_1 \prec q \prec p_0$$

for every $q \in \bigcup_{i=1}^n \mathsf{attr}(T_i)$. Then

$$\#N(S_n, \prec) = n + 1 + \sum_{i=1}^{n} \#N(T_i, \prec).$$

Proof: Observe that $\#N(S_0, \prec) = \#p_0(0,1) = 1$, and

$$\#N(S_j, \prec) = \#p_i(N(S_{j-1}, \prec), N(T_j, \prec))1 + \#N(S_{j-1}, \prec) + \#N(T_j, \prec)$$

for every $j = 1, \ldots, n$ By induction on j we obtain $\#N(S_j, \prec) = j + 1 + \sum_{i=1}^{j} \#N(T_i, \prec)$ for every $j = 0, \ldots, n$, from which the lemma follows by taking $j = n$. □

Theorem 13 *Let $\{T_1, \ldots, T_n\}$ be a set of decision trees and let S_n be defined as above. Let k be a natural number. Then there is a total order $<$ on $\bigcup_{i=1}^{n} \mathsf{attr}(T_i)$ satisfying $\sum_{i=1}^{n} \#N(T_i, <) \leq k$ if and only if there is a total order $\prec$ on $\mathsf{attr}(S_n)$ satisfying $N(S_n, \prec) \leq n + 1 + k$.*

Proof: For the 'only if'-part let $<$ be a total order on $\bigcup_{i=1}^{n} \mathsf{attr}(T_i)$ satisfying $\sum_{i=1}^{n} \#N(T_i, <) \leq k$. Define the total order $\prec$ on $\mathsf{attr}(S_n)$ by

$$p_n \prec p_{n-1} \prec \cdots \prec p_1 \prec q \prec p_0$$

for every $q \in \bigcup_{i=1}^{n} \mathsf{attr}(T_i)$, and $q \prec r$ if and only if $q < r$ for every $q, r \in \bigcup_{i=1}^{n} \mathsf{attr}(T_i)$. Then by Lemma 12 we obtain

$$\#N(S_n, \prec) = n + 1 + \sum_{i=1}^{n} \#(T_i, \prec) = n + 1 + \sum_{i=1}^{n} \#N(T_i, <) \leq n + 1 + k.$$

Conversely, for the 'if'-part let $\prec$ be a total order on $\mathsf{attr}(S_n)$ satisfying $N(S_n, \prec) \leq n + 1 + k$. Defining $\hat{\prec}$ as in Lemma 11 the order $\hat{\prec}$ satisfies the requirements of Lemma 12. We obtain

$$\begin{array}{rcll} n + 1 + \sum_{i=1}^{n} \#N(T_i, \hat{\prec}) & = & \#N(S_n, \hat{\prec}) & \text{(by Lemma 12)} \\ & \leq & \#N(S_n, \prec) & \text{(by Lemma 11)} \\ & \leq & n + 1 + k & \end{array}$$

by which we obtain $\sum_{i=1}^{n} \#N(T_i, \hat{\prec}) \leq k$, concluding the proof. □

Combining Theorems 7 and 13 we now arrive at our main result.

Theorem 14 *Given a decision table T and a number k, it is NP-complete to decide whether a decision table U exists satisfying $U \simeq T$ and $\#U \leq k$.*

Proof: From the definition of normal form it is clear that a decision table U exists satisfying $U \simeq T$ and $\#U \leq k$ if and only if a total order $<$ on $\mathsf{attr}(T)$ exists satisfying $\#N(T, <) \leq k$. □

5 Concluding remarks

In this paper we showed how decision tables can be considered as a special kind of decision trees. We showed that for a given decision tree the size of any decision table representing the same function can be exponential in the size of the decision tree. Our main result states that finding a decision table of minimal size that represents the same function as a given decision table is an NP-hard problem; in earlier work we obtained a similar result for decision trees.

If we allow sharing in the representation, i.e., we have dags instead of trees, then the notion of decision tree correspond to the notion of binary decision diagram (BDD), while the notion of decision table corresponds to the notion of ordered binary decision diagram (OBDD). For these notions we refer to [2, 4]. For both BDDs and OBDDs NP-hardness results for finding a minimal representation hold too: for BDDs this is straightforward from NP-hardness of satifiability; for OBDDs this has been proved in [1].

References

[1] Bollig, B., and Wegener, I. Improving the variable ordering of OBDDs is NP-complete. *IEEE Transactions on Computers 45* (1996), 993–1002.

[2] Bryant, R. E. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers C-35*, 8 (1986), 677–691.

[3] Karp, R. M. Reducibility among combinatorial problems. In *Complexity of Computer Computations* (1972), R. E. Miller and J. W. Thatcher, Eds., Plenum Press, pp. 85 – 104.

[4] Meinel, C., and Theobald, T. *Algorithms and Data Structures in VLSI Design: OBDD — Foundations and Applications.* Springer, 1998.

[5] Mitchell, T. M. *Machine Learning.* McGraw-Hill, 1997.

[6] Quinlan, J. R. Induction of decision trees. *Machine Learning 1* (1986), 81–106.

[7] Subramanian, G. H., Nosek, J., Raghunathan, S. P., and Kanitkar, S. S. A comparison of the decision table and tree. *Communications of the ACM 35*, 1 (1992), 89–94.

[8] Vanthienen, J. A more general comparison of the decision table and tree. *Communications of the ACM 37*, 2 (1994), 109–113.

[9] Zantema, H. Decision trees: Equivalence and propositional operations. In *Proceedings 10th Netherlands/Belgium Conference on Artificial Intelligence (NAIC'98)* (November 1998), H. L. Poutré and J. van den Herik, Eds., pp. 157 – 166. Extended version appeared as report UU-CS-1998-14, Utrecht University.

[10] Zantema, H., and Bodlaender, H. L. Finding small equivalent decision trees is hard. Tech. Rep. UU-CS-1999-02, Utrecht University, Department of Computer Science, January 1999. Available via `http://www.cs.uu.nl/docs/research/publication/TechRep.html`.