Compiling Relational Data into Disjunctive Structure: Empirical Evaluation. *

Rina Dechter
Information and Computer Science
University of California, Irvine, CA 92717
dechter@ics.uci.edu

Eddie Schwalb
Information and Computer Science
University of California, Irvine, CA 92717
eschwalb@ics.uci.edu

Abstract

Recent work in knowledge compilation suggests that relations which can be described precisely by either *Horn theories* or *tree constraint networks* are identifiable in output polynomial time. Algorithms for computing approximations using these languages were also proposed. Upon testing such approximations on artificially generated and real life data, it was immediately observed that they yield numerous superfluous models. As a result, although certain entailment queries can be answered reliably, these methods may be ineffective for a large class of membership queries.

To improve the approximation quality, we investigate here the *k-decomposition problem*, that is, determining whether a relation can be described by a *disjunction* of k tractable theories. The paper discusses the complexity of this task, outlines several algorithms for computing both exact and approximate k-decompositions, and evaluates the potential of this approach empirically. We focus on the class of *tree constraint networks* and *Horn theories* and report results on artificially generated relations and on three real life cases. Our experiments show that for uniform random relations, the quality of upper bound approximations improves as k increases. However, when we require very high accuracy, decomposition is not effective since k grows linearly with the size of the data. When the data comes from a near-tractable source, the approach is useful. Experiments show that for the King Rook King problem the generalizing power of such methods is comparable to that of recently developed learning algorithms.

1 Introduction

Recently, frameworks for approximating intractable theories and relations using tractable languages were proposed using the notions of identifiability [2] and knowledge compilation [9]. The goal is to replace an intractable theory, or its set of models, with a tight upper and lower bound tractable language, thus allowing efficient query processing using the tractable approximations.

*This work was partially supported by NSF grant IRI-9157636, by Air Force Office of Scientific Research grant AFOSR 900136, by TOSHIBA of america and by Xerox grant.

In this paper, following [2] and in contrast to [9], we assume that the input theory is given by its set of models (or tuples) representing, perhaps, a set of observations and the task is to describe these observations using a tractable language. In [1, 2] it was shown that relations that can be described precisely by either *Horn theories* or *tree networks* can be identified in polynomial time. Otherwise, tight upper bound Horn theory or tree network approximations can be computed. In this paper we investigate empirically the effectiveness of such approximations on artificially generated relations as well as some real life data.

The effectiveness of an approximation should be measured with respect to a class of queries. There are two common types of queries on a theory φ: entailment queries (whether a formula holds in all models of φ), and membership queries (whether a given tuple is a model of φ). The first is common in automated reasoning while the second appears often in learning and classification tasks. Each query type dictates a different evaluation measure. We use, respectively, two measures: the fraction of clauses correctly entailed by the approximation, and the number of superfluous models in the approximation.

In our experiments it became immediately apparent that when the relations cannot be compiled into a language (as is often the case) the resulting tightest upper bound approximation is effective for certain entailment queries, but still yields numerous superfluous models.

To improve effectiveness relative to membership queries, we propose here to extend the model one step further by considering, as our target language, a *disjunction* of a fixed number of tractable theories. Specifically we address the following questions: Given a relation ρ and a class of theories Ω, can ρ be decomposed exactly into k subrelations, each represented by theories from Ω ? If not, does the approximation quality improve as the number of theories in the disjunction increases ? What are the complexity issues involved?

Our experiments show that for relations generated randomly and uniformly, the accuracy of an upper bound approximation improves significantly with the disjunction size. When we require the number of superfluous models to be small, decomposition is not effective because the number of theories required grows linearly with the size of the input relation. However, when the relation comes from a near-tractable source, the approach is useful.

The paper is organized as follows. Section 2 contains definitions and preliminaries, section 3 discusses the framework of k-disjunctive approximations and presents the algorithms used, and section 4 presents our empirical results. Concluding remarks are given in section 5.

2 Definitions and Preliminaries

We denote propositional symbols, also called *variables*, by uppercase letters $P, Q, R, X, Y, Z, ...$, propositional literals (i.e., $P, \neg P$) by lowercase letters $p, q, r, x, y, z, ...$, and disjunctions of literals, or *clauses*, by $\alpha, \beta, ...$. A *formula* in conjunctive normal form (cnf) is a set of clauses $\varphi = \{\alpha_1, ..., \alpha_t\}$, implying their conjunction. The *models* of a formula φ, $M(\varphi)$, is the set of all satisfying truth assignments to all of the formula's symbols. A clause α is *entailed* by φ, written $\varphi \models \alpha$, iff α is true in all models of φ. A Horn formula is a cnf formula whose clauses all have at most one positive literal.

A *relation* associates a set of multivalued variables, also called attributes, with a set of tuples specifying their allowed combinations of values. A *constraint network* is a set of such relations, each defined on a subset of the variables. Taken together, this set represents a conjunction of constraints that restricts value assignments to comply with each and every constituent relation. The theory of relations has been studied extensively in the database literature [6].

Definition 1: (Relations and Networks)
Given a set of multivalued variables $X = \{X_1, ..., X_n\}$, each associated with a domain of discrete values $D_1, ..., D_n$ respectively, a *relation* (or, alternatively, a *constraint*) $\rho = \rho(X_1, ..., X_n)$ is any subset $\rho \subseteq D_1 \times D_2 \times ... \times D_n$. A *constraint network* N over X is a set $\rho_1, \ldots, \rho_t$ of relations each defined on a subset of variables $S_i \subseteq X$. Each relation ρ_i specifies the set of allowed assignments of the variables in S_i. A solution is an assignment of a value to each variable that satisfies all the constraints, and the network N represents the relation $rel(N)$ of all its solutions. If $rel(N) = \rho$ we say that N *describes* ρ. A constraint network in which all constraints involve pairs of variables ($|S_i| = 2$), is called a *binary constraint network*. A *constraint graph* associates each variable with a node and connects pair of variables that appear in the same constraint. *Tree networks* are binary networks whose constraint graph is a tree.

A cnf formula can be viewed as a special kind of constraint network, where the domains are bi-valued ($|D_i| = 2$) and each clause specifies a constraint on its propositional symbols. The set of models of the formula are the set of solutions of the corresponding constraint network. A bi-valued relation $\rho = \rho(x_1, ..., x_n)$ is *described* by a cnf formula $\varphi = \varphi(x_1, ..., x_n)$ iff $M(\varphi) = \rho$. We will use the term *theory* to denote either a network or a propositional formula.

Frequently, a relation cannot be precisely described by a theory from a given language, in which case we use approximations. We will examine primarily upper bound approximations.

Definition 2: (Upper bounds)
Given a class of theories, Ω, a theory $T \in \Omega$ is said to be an *upper bound* of ρ relative to Ω if $\rho \subseteq M(T)$. T is a tightest upper bound if $\rho \subseteq M(T)$ and there is no $T' \in \Omega$ such that $\rho \subseteq M(T') \subset M(T)$.

Clearly, if φ is a theory describing ρ ($M(\varphi) = \rho$), and T is an upper bound of φ, then if $T \models \gamma$ we can infer $\varphi \models \gamma$. Alternatively, if $t \notin M(T)$ then $t \notin \rho$ ($\rho \subseteq M(T)$).

Some languages admit a unique *tightest upper bound*. In this case, $U_\Omega(\rho)$ denotes the unique relation associated with this *tightest upper bound* expressed within this language. It is known that *Horn theories* allow a *unique* tightest upper bound [1, 2] while there may be many *tight upper bound tree networks* [2].

In [2, 3] it is shown that Horn theories and tree networks are identifiable, namely there is a polynomial algorithm that can decide whether any given relation can be described precisely by a Horn theory or a tree-network, and also finds the corresponding description whenever possible. Otherwise, the algorithm computes an upper bound. For Horn theories the algorithm generates the tightest Horn upper-bound, but it is no longer polynomial in the input relation. For tree-networks the algorithm is always polynomial but does not necessarily generate the tightest upper bound. For completeness sake, we present the algorithms for computing tight upper bounds for tree constraint networks and Horn theories.

2.1 Computing a tight tree network

The tree algorithm [2], finds a tree-network representation to a given relation, if such exists, otherwise it computes a tight upper bound.

Given an arbitrary relation, ρ, let $n(x_i)$ be the number of tuples in ρ for which $X_i = x_i$, and let $n(x_i, x_j)$ be the number of tuples for which $X_i = x_i \ \wedge \ X_j = x_j$. Let us define weights $w(X_i, X_j)$ as

$$w(X_i, X_j) = \frac{1}{|\rho|} \sum_{(x_i, x_j) \in \prod_{X_i X_j}(\rho)} n(x_i, x_j) log \frac{n(x_i, x_j)}{n(x_i) n(x_j)}$$

The constraint graph of the tree approximation is computed as the maximum weight spanning tree formed with the arc-weights $w(X_i, X_j)$. Once the structure of the tree is determined, the constraints of the network can be obtained by projecting ρ onto the pairs of connected variables in the tree.

2.2 Computing the tightest Horn upper bound

In [1, 2] it was shown that the models of a Horn theory are closed under intersection when intersection is defined as follows. Let $x = \{x_1, x_2, \ldots, x_n\}$ be a tuple where $x_i \in \{0, 1\}$. Then $true(x)$ is the set of variables assigned to 1 and $false(x)$ is the set of variables assigned to 0. The intersection $z = x \cap y$ is defined as $true(z) = true(x) \cap true(y)$ and $false(z) = false(x) \cup false(y)$. A bi-valued relation is said to be *closed under intersection* iff $\forall x, y \in \rho \ \ x \cap y \in \rho$. The closure of ρ is the set of models of the tightest Horn theory bounding ρ.

We compute the set of models of the *tightest Horn upper bound* of a given relation by computing its intersection closure. The procedure is polynomial in the size of the output relation but not necessarily polynomial in the size of its input. Once the set of its models is computed, the

Horn theory can be extracted by algorithms presented in [1, 2].

3 Computing k-decompositions

We now extend the notion of identifiability to a disjunction of theories. We will assume (except when otherwise noted) throughout this paper that our languages admit a unique upper bound.

Definition 3: A relation ρ is k-decomposable relative to a class of theories Ω iff there exist a set of relations $Q = \{\rho_1, \ldots, \rho_k\}$ such that ρ_i is described in Ω and $\rho = \bigcup_{i=1}^{k} \rho_i$. A language, Ω, is k-identifiable if for every relation ρ, deciding if ρ is k-decomposable relative to Ω is polynomial.

Clearly, for any language that can describe a single tuple, every theory is k-decomposable for $k = |\rho|$. The interesting task is to find the smallest k for which a theory is *k-decomposable*. The following paragraphs provide the necessary and sufficient conditions for k-decomposability.

Definition 4: Let $U_\Omega(\rho)$ be the unique tightest upper bound of ρ relative to Ω. We define a graph $G_\Omega(\rho)$ as follows. Each *tuple*, $x \in \rho$, is mapped to a *node* and an *arc* between two nodes $x, y \in \rho$ exists iff $U_\Omega(\{x,y\}) \not\subseteq \rho$.

Theorem 1: *Given ρ and Ω,*

1. *If $G_\Omega(\rho)$ is not k-colorable then ρ is not k-decomposable.*
2. *If $G_\Omega(\rho)$ is k-colorable then ρ is k-decomposable iff there exists a k-coloring of $x_1 \ldots x_{|\rho|}$ (a value of $1 \ldots k$ assigned for every tuple in ρ) for which $\forall i \leq k$ the sets $\rho'_i = \{x \mid color(x) = i\}$ satisfy $U_\Omega(\rho'_i) \subseteq \rho$.*

Consequently, a lower bound on k is the size of each clique in G_Ω.

Theorem 1 suggests a brute force algorithm for computing a k-decomposition. Enumerate all k-colorings for $G_\Omega(\rho)$, and, for each coloring, check whether condition (2) is satisfied. If condition 2 can be tested in polynomial time (true for Horn theories), then the algorithm's complexity is dominated by the complexity of enumerating all k-colorings of a graph. Since finding even one coloring is NP-complete, the problem is clearly intractable. However, for the special case of $k = 2$, enumerating all possible colorings can be done in time linear in the number of colorings [4]. Moreover, for $k = 2$, it can be shown that every connected component of $G_\Omega(\rho)$ (bi-partite for $k = 2$) can be colored in at most two ways[1] and, therefore, $2^{\#components}$ possible colorings need to be checked.

Corollary 1: *Given a language Ω such that G_Ω can be computed in polynomial time, then 2-decomposability can be decided in time polynomial in $2^{\#components}$ of $G_\Omega(\rho)$.*

3.1 Approximated Decomposition

Because computing a k-decomposition is a difficult task, we examine polynomial approximation algorithms for two related formulation of this problem: (1) (minimization) given a theory φ and a language Ω, find the minimal k for which φ is *k-decomposable* relative to Ω. (2) (upper bound decomposition) given Ω and k, find a k-disjunctive upper-bound of φ relative to Ω that minimizes the number of superfluous models.

For the first task, we describe a greedy approximation algorithm. The algorithm can be viewed as a variant of the algorithm suggested by theorem 1. It starts from two arbitrary models $x, y \in \rho$, and computes $U_\Omega(\{x,y\})$. If $U_\Omega(\{x,y\}) \not\subseteq \rho$ it concludes that x and y must participate in different relations; otherwise, the algorithm adds $U_\Omega(\{x,y\})$ to ρ_i and deletes $U_\Omega(\{x,y\})$ from ρ. The algorithm continues with a third and fourth tuple, until all models in ρ are covered (see Figure 1).

Lemma 1: *Algorithm GreedyDecompose (Figure 1) terminates in $O(|\rho| \cdot k \cdot t_\rho)$ steps where k is the number of resulting theories and t_ρ is the number of steps required to test whether $U_\Omega(\rho) \subseteq \rho$.*

In the second task, the disjunction size k, is fixed in advance. The *partitioning* algorithm for that task, divides the input relation ρ into k equal partitions, $\rho_1, \ldots, \rho_k$, and outputs their tightest upper bounds, $U_\Omega(\rho_1), \ldots, U_\Omega(\rho_k)$. Clearly, $\rho \subseteq \cup_i U_\Omega(\rho_i) \subseteq U_\Omega(\rho)$. The complexity of partitioning is $O(k \cdot t_\rho)$ where t_ρ is the time required to compute $U_\Omega(\rho)$. Note that for Horn theories the greedy algorithm is always polynomial while the partitioning algorithm can be exponential. Note also that the partitioning algorithm has no control over the number of superfluous models.

To control the number of superfluous models we define ε approximations and show how such approximations can be computed by the greedy algorithm.

Definition 5: ((k,ε)-approximations) A relation ρ is *(k,ε)-decomposable* relative to Ω iff there exists a set of relations $Q = \{\rho_1, \rho_2 \ldots \rho_k\}$ such that ρ_i is described in Ω, and,

$$\rho \subseteq \bigcup_{i=1}^{k} \rho_i \qquad and \qquad \left| \bigcup_{i=1}^{k} \rho_i - \rho \right| \leq \varepsilon |\rho| \tag{1}$$

Q is called a (k,ε) upper bound.

A (k,ε) upper bound can be computed by *GreedyDecompose* if we allow only a bounded number of models in each subrelation ρ_i to fall outside the input relation, namely $|\rho_i - \rho| \leq \frac{\varepsilon}{k'}|\rho|$, and if the actual disjunction size generated (k) happens to be smaller than k'. This can be implemented by modifying line 10 of *GreedyDecompose* (Figure 1) to accommodate some models not in ρ (e.g. "**if** $|U_\Omega(\rho_i \cup \{x\}) - \rho| \leq \frac{\varepsilon}{k'}|\rho|$ **then**...").

4 Experiments

In this section we evaluate the quality of the approximation obtained. We use two measures: the number of superfluous models divided by the whole tuple space (2^n) and the fraction of clausal entailment queries answered correctly.

Notice that the unique tightest upper bound with respect a class Ω is guaranteed to correctly answer entailment

[1] We thank Dan Roth for this observation.

```
GreedyDecompose
Input(ρ,  a class of theories Ω  and  a polynomial algorithm to compute U_Ω(ρ))
Output(Q = {ρ_1, ..., ρ_k})          ; The disjunction of the relations - see Definition 3
Begin
    Q ← {}                           ; Initialize an empty disjunction.
    ρ' ← ρ                           ; We use ρ' to preserve ρ for comparisons.
    while ρ' ≠ {} do
        choose arbitrary x ∈ ρ'
        flag ← false                 ; flag detects whether this tuple requires a new relation ρ_i
        if Q ≠ {} then
            foreach ρ_i ∈ Q do
                if U_Ω(ρ_i ∪ {x}) ⊆ ρ then      ; If we can add the tuple to ρ_i, then
                    ρ_i ← U_Ω(ρ_i ∪ {x})        ; add it, and
                    ρ' ← ρ' − ρ_i               ; don't iterate on tuples already in ρ_i,
                    flag ← true                 ; and signal that no new relation ρ_i is needed.
                end-if
            end-for
        end-if
        if flag = false then         ; Here, we add to the disjunction Q
            Q ← Q ∪ {x}              ; a new relation which consists of
            ρ' ← ρ' − {x}            ; a single tuple x.
        end-if
    end-while
End.
```

Figure 1: The greedy algorithm for decomposing a relation.

queries of formulas expressed in Ω. In particular, a Horn tightest upper bound will infer correctly all Horn queries.

Observation 1: *Let $U_\Omega(\varphi)$ be the unique tightest upper bound of φ. For every $\alpha \in \Omega$, $U_\Omega(\varphi) \models \alpha$ iff $\varphi \models \alpha$.*

Proof: Clearly, if $U_\Omega(\varphi) \models \alpha$ then $\varphi \models \alpha$. If $\varphi \models \alpha$ then α is an upper bound of φ. Since $U_\Omega(\varphi)$ is the tightest upper bound, it also entails α. □

Consequently, it is meaningless to measure the effectiveness of the approximation with respect to queries from the bounding language since we are guaranteed correct answers.

We evaluate the effectiveness of the approximation on artificially generated relations and three real life databases: the KRK problem from the chess domain, the "politicians" relation that represents voting records of politicians, and the "breast-cancer" relation that represents medical records of patients.

4.1 Horn upper bound

Tables 1 and 2 summarize the results for random input relations. In Table 1, the input is a relation (whose number of attributes and models are given) and the output is the number of superfluous models in the tightest Horn upper bounding relation. Table 2 reports the fraction of entailment queries correctly answered as a function of the clauses size, for relations having 10 variables and 50 models (also reported in the 2nd row of Table 1). Additional details are provided in Figure 3(a) by the curve labeled "Single Partition". Note that most of the 3-literal clauses were *not* entailed by the theory nor by its upper bound.

Real Life Data: The "politicians" relation is defined over 16 bi-valued attributes and consists of 125 tuples. The tightest Horn upper bound consists of 1160 tuples.

As observed, although tightest Horn upper bounds exclude many non-models, they also contain numerous superfluous models and thus may be unacceptable for answering membership queries.

Table 1
membership queries

Num of variables	Num of models	Superfluous models
9	32	140
10	50	275
11	200	1052
15	50	1527

Table 2
entailment queries

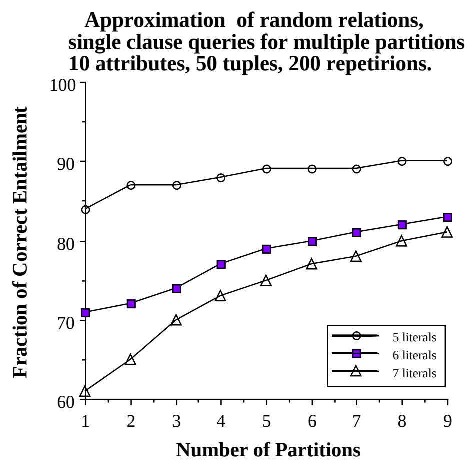
Num of literals	Entailment accuracy
3	99%
5	82%
6	67%
7	60%

4.2 Disjunctive Horn approximations

We next show the improvement (over tightest Horn upper bound) obtained using disjunction of Horn theories. Given a constant k, the *partitioning algorithm* computes a k-disjunctive Horn theory. As described earlier, the algorithm partitions the input relation ρ into k disjoint subrelations of equal size, $\rho_1, \ldots, \rho_k$, and compute the Horn upper bound of each ρ_i yielding $U_\Omega(\rho_1), \ldots, U_\Omega(\rho_k)$. The results are summarized in Figures 2 and 3.

In Figure 2 we plot the size $|\cup_i U_\Omega(\rho_i)|$ (number of models) as a function of k, the disjunction size. The input relations have 9 and 11 attributes with 32 and 200 models respectively. We show, for instance, that when approximating with five theories, for 9 and 11 attributes the fraction of superfluous models (with respect to $2^9, 2^{11}$) was reduced from 29%,51% to 8%,33% respectively.

In Figure 3 we report the results obtained on relations having 10 attributes and 50 models. In this case, the tightest single upper bound Horn approximation had 325 models on average. Every point is obtained by testing all clauses of a fixed length and computing the fraction of correctly answered queries. This is averaged over 50 relations. We observe in Figure 3(a) that the approximation obtained using disjunction of nine Horn theories was significantly better than the tightest upper bound. In Figure 3(b) we report the accuracy as it improves when



(a) effectiveness vs clause size,

(b) effectiveness vs number of partitions.

Figure 3

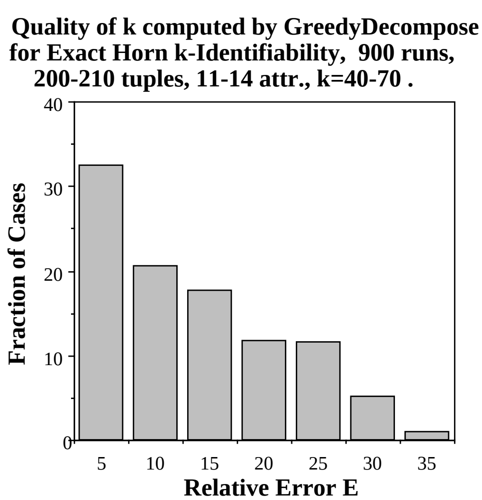


Figure 4: Stability of GreedyDecompose.

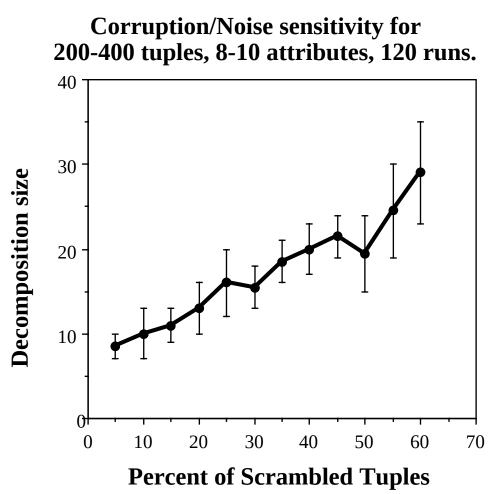


Figure 5: Near-Horn relations.

the number of theories in the disjunction increases. We plot results for clauses with 5,6,7 literals. Note that 50% is the lowest accuracy possible since guessing yields at least 50%. Figure 3(b) suggests that for small clause entailment, disjunctive upper bounds do not improve much over single Horn bounds, because for small clauses the single Horn upper bound is already quite effective. As the clause size increases, there are more non-Horn clauses and therefore single Horn upper bounds are less effective and consequently disjunctive bounds can be more cost-effective.

The same experiments were performed on the "politicians" relation and roughly the same results were obtained.

In the remainder of this section we report results of experiments using *GreedyDecompose* for computing both exact decomposition and approximate disjunctive upper bounds.

We next focus on the behavior *GreedyDecompose*. Since there exists an ordering of models for which the greedy algorithm yields an optimal decomposition size k, we evaluate its effectiveness by observing the variations of k as a function of the order by which models are processed. For every relation generated, *GreedyDecompose* was run 30 times, each time with a different random ordering. Q_0 denotes the smallest size decomposition out of the 30 computed, and $Q_1, \ldots, Q_{29}$ denote the other 29 decompositions. The stability, is measured by the *relative error* $E_i = \frac{|Q_i|-|Q_0|}{|Q_0|}$, namely the fraction of cases the size of the decomposition, $k_i = |Q_i|$, was larger than $k_0 = |Q_0|$. As shown in Figure 4, in about 32% of the cases the size of the decomposition computed was less than 5% larger than $k_0 = |Q_0|$ and the largest variation was 35%. Thus the algorithm is reasonably stable.

We next report results of experiments with exact decompositions. The experiments were performed on small bi-valued random relations, having about 30 models, and larger relations, having 300 models. As Table 3(a) shows, for relations with 30-34 models having 10 attributes, about 15 theories were required on the average (i.e., k=15). For larger relations with 300-325 models having 14 attributes, about 130 theories were needed. Clearly, since the algorithm is not optimal, we do not know whether smaller decompositions exist.

Tree networks (constraint networks whose constraint graph is a tree) were examined next. We observed in Table 3(b) that, as in the case of *Horn theories*, k might be arbitrary large. However, it does not increas as fast as the number of attributes.

To evaluate sensitivity to noise, we take a relation that can be described by a Horn theory and corrupt randomly selected models; the degree of noise introduced is measured by the percentage of models corrupted. A model is corrupted by flipping each of its bits with 0.5 probability. As shown in Figure 5, when only a few models were corrupted, the corrupted relation admits a relatively small k.

We next examine the effectiveness of bounded overflow on k. Namely, we compute a (k, ε) upper bound. To demonstrate the effect on k, we show (Figure 6(a)) the dependence of the *Horn* disjunct size, k, on the overflow fraction, defined as $\xi = \frac{|\cup_i U_\Omega(\rho_i) - \rho|}{|\rho|}$. As expected, we see that increasing overflow reduces the decomposition size.

Finally, we measure the trade-off between noise and overflow. We plotted the number of overflow models with respect to the number of corrupted models while holding k under 5. We observe that k, which was increased by corruption or noise, can be decreased by allowing overflow proportional to the degree of corruption (Figure 6(b)).

4.2.1 Real Life Data

We report results of experiments made with three real life databases taken from the machine learning repository at U.C. Irvine. We first examine Horn k-decompositions of a bi-valued relation that represents voting records of politicians. The relation can be represented exactly by 48 Horn theories. However, by allowing overflow, this number can be reduced (see Figure 7(a)).

The breast cancer relation represents records of symptoms and diagnosis (i.e. whether the patient had breast cancer or not). The task is to derive a theory that enables efficient processing queries that involve symptoms and diagnosis. The relation can be described exactly by 38 tree networks; however, by allowing overflow, this number can be reduced (see Figure 7(b)).

4.3 Learning with Horn upper bound

We next suggest that perhaps compilation methods that aim at providing a tractable and concise representation when all the data is available can be modified and used for learning when only part of the data is available.

The experiments are performed on the King Rook King (KRK) problem from the chess domain. The task is to learn a predicate that classifies board positions as either legal or illegal, given a small training set with positive and negative examples. Each example is a tuple with 6 attributes that specify the coordinates of the white king, the black rook and the black king. The multi-valued training relation $\rho_{training}$ is transformed into bi-valued relation $\rho'_{training}$ using a set of predicates provided by the expert, and $U_\Omega(\rho'_{training})$ is computed. To classify an unseen instance x, we map it to a bi-valued instance x' and check whether $x' \in U_\Omega(\rho'_{training})$. If $x' \in U_\Omega(\rho'_{training})$, then the classification can be determined accordingly. Otherwise, we guess the most frequent class.

We compare performance with a recent algorithm for learning prolog programs, called FOCL [7]. Figure 4 shows that $U_\Omega(\rho_{training})$ was able to correctly classify about 95% of the unseen examples when trained on 300 examples while FOCL was able to achieve better accuracy with only 200 examples. The curve labeled by "HORN known" shows the fraction of unseen examples found in the closure and that were correctly classified[2]. The curve labeled "HORN" shows the final accuracy

[2] Since the target concept in the KRK domain is not Horn, we can only approximate it.

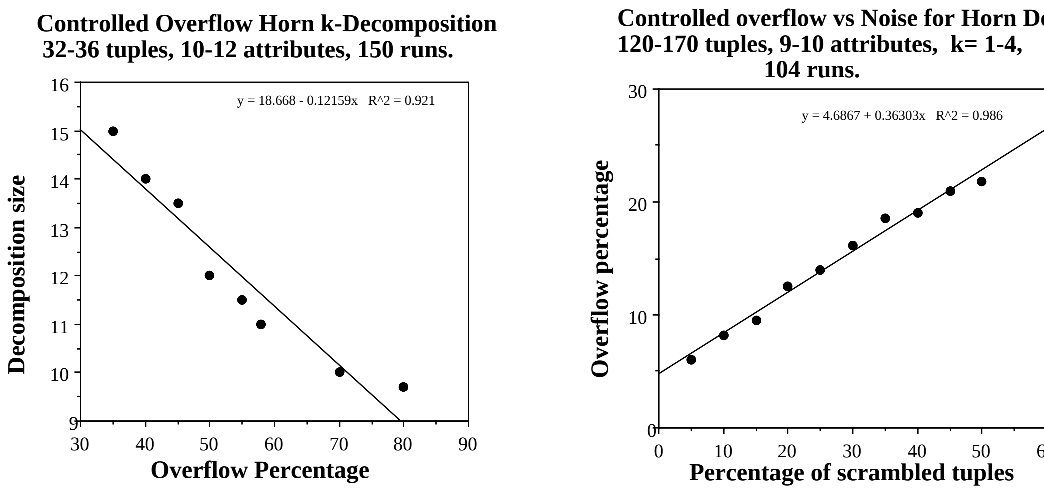
Table 3: Exact decomposability.

(a) k-Horn

Number of Attributes	*Number of Disjuncts*	*Min Num. Disjuncts*	*Max Num Disjuncts*
Horn $k-decomposition$, $30-34$ *models*, 250 *runs*			
6	5	4	6
7	8	6	10
8	11	7	14
9	14	9	16
10	15	9	17
Horn $k-decomposition$, $300-325$ *models*, 1040 *runs*			
11	70	68	72
12	90	84	98
13	113	105	122
14	133	126	140

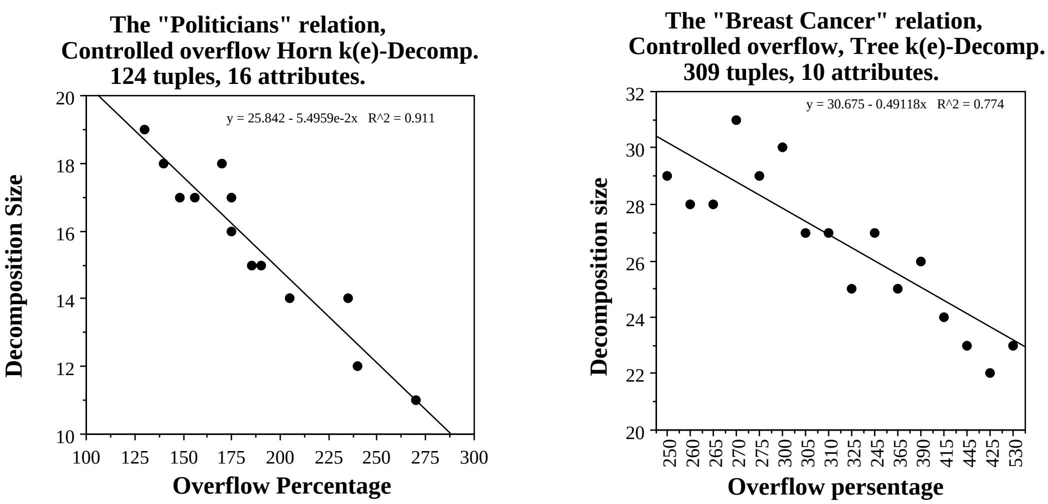
(b) k-Tree

Number of Attributes	*Number of Disjuncts*	*Min Num. Disjuncts*	*Max Num Disjuncts*
Tree $k-decomposition$, $32-36$ *models*, 125 *runs*			
8	9.4	8	11
9	11.0	9	13
10	11.5	10	13
11	12.0	11	13
12	13.1	12	14
Tree $k-decomposition$, $200-202$ *models*, 105 *runs*			
10	39	35	44
12	52	48	58
14	63	59	65
16	69	64	73



(a) (b)

Figure 6: Sensitivity to noise of Exact k-Horn decomposability.



(a) (b)

Figure 7: Decomposability of "real data"

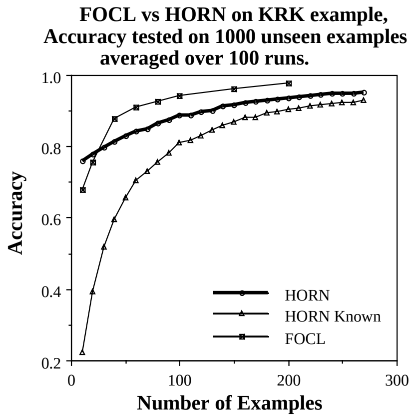


Figure 8: Learning with tightest Horn upper bound approximations.

achieved by guessing the most frequent class when the unseen instance is not in $U_\Omega(\rho'_{training})$. For more details on the use of single-upper bounds on learning see [8].

5 Conclusion

This paper builds upon prior investigations into the prospects of compiling empirical data into structures that allow efficient processing of queries [2]. Previous work had presented algorithms for describing or approximating data by Horn theories [1, 2] or tree constraint networks [2, 3]. The effectiveness of these approximation should be measured with respect to the class of queries to be asked. We focus on entailment queries, which are common in automated reasoning, and membership queries, which are common in classification tasks. Upon testing these approximations on artificially generated and real life data, it was immediately observed that, although effective for some entailment queries, they are ineffective for membership queries since they yield numerous supefluous models.

We therefore propose to improve on the single tightest upped bound approximation by approximating with a disjunction of theories. We define the *k-decomposition problem*: given an integer k and a relation ρ, determine whether ρ can be described by a *disjunction* of k tractable theories. The paper presents the necessary and sufficient conditions for a relation to be k-decomposable and identifies cases in which determining 2-decomposability is polynomial.

Because computing a k-decomposition is a difficult task, we examine polynomial approximation algorithms for two related formulation of this problem: (1) (minimization) given a theory φ and a language Ω, find the minimal k for which φ is *k-decomposable* relative to Ω. (2) (upper bound decomposition) given Ω and k, find a k-disjunctive upper-bound of φ relative to Ω that minimizes the number of superfluous models. We evaluate the effedctiveness of these approximations empirically with respect to both entailment and membership queries.

In our experiments we focus on the class of *tree constraint networks* and *Horn theories* and report results on artificially generated relations and on three real life cases. For the second task, we observe that the quality approximation obtained by upper bound decomposition improves as k increases. For the first task, when the data comes from a near-tractable source, or when the overflow is proportional to the level of noise, the approach is useful since k is small. However, when the input relation is not generated by a near-tractable source and we require very high quality approximations in which the number of superfluous models is bounded, decomposition is not effective since k grows almost linearly with the size of the input relation.

Fianlly we suggest that perhaps compilation methods, that aim at providing a tractable and concise representation when all the data is available, can be modified and used for learning when only part of the data is available. Experiments show that for the King Rook King problem the generalizing power of the tightest upped bound Horn approximation is comparable to that of recently developed learning algorithms. For more details on the use of single-upper bounds on learning see [8].

Acknowledgments

We would like to thank Dan Roth for valuable comments on an earlier version of this paper. The breast cancer database was obtained from the University Medical Center, Institute of Oncology, Ljubljana, Yugoslavia. Thanks go to M. Zwitter and M. Soklic for providing the breast cancer data.

References

[1] Angluin, D., Frazier, M., Pitt, L., 1991. Learning conjunctions of Horn Clauses, *Machine Learning* 9:147-164.

[2] Dechter, R., Pearl, J., 1992. Structure Identification in Relational Data, *Artificial Intelligence* 58:237-270.

[3] Dechter, R., 1990. "Decomposing a relation into a tree of binary relations", *Journal of Computer and Systems Sciences*, special issue on the Theory of Relational Databases, Vol 41, 2-24 (1990)

[4] Dechter, R., Itai, A., 1992. Finding all solutions if you can find one. UCI Tech Rep. 92-61.

[5] Kautz, H., Kearns, M.J., Selman, B.S., 1993. Reasoning with Characteristic Models, *Proceedings of AAAI-93*, 34-39.

[6] Maier, D., 1983. The theory of Relational Databases Computer Science press, Rockville, MD, 1983.

[7] Pazzani, M.,, Kibler, D., 1992. The utility of knowledge in inductive learning, *Machine Learning* 9 (1992), 57-94

[8] Schwalb, E., Dechter, R., Pazzani, M, 1993. Using identifiability for learning Horn logic programs, UCI Tech-Rep. 94-13.

[9] Selman, B., Kautz, H., 1992. Knowledge compilation using Horn approximation *Proceedings of AAAI-91*, 904-909.