On learnability and predicate logic (Extended Abstract)

W. Maass* Gy. Turán†

1 Introduction

Several applications of learning in artificial intelligence use a *predicate logic* formalism. The theoretical study of efficient learnability in this area, in the framework of *computational learning theory* started relatively recently, considering, for example, the *PAC (Probably Approximately Correct)* learnability of *logic programs* and *description logic* (see Cohen and Hirsh [6] , the survey of Kietz and Dzeroski [11] and the further references in these papers).

In this paper we discuss a *model theoretic approach* to learnability in predicate logic. Results in this direction were obtained by Osherson, Stob and Weinstein [15] . It is assumed that there is a *first-order structure* given. *Instances* are tuples of elements of the universe of the model, and *concepts* are *relations* that are *definable* in the model by formulas from some prespecified class. The goal of the *learner* is to identify an unknown *target concept* in some specific model of learning. (A standard example is that of a *finite database* containing data describing the relations *father* and *mother*. The goal is to learn the relation *grandparent* , given positive and negative examples of this concept.) Properties of classes of definable sets are well studied in model theory. It is interesting to note that one of the several simultaneous sources for the notion of the *Vapnik-Chervonenkis dimension (*

*Institute for Theoretical Computer Science, Technische Universitaet Graz. Email: maass@igi.tu-graz.ac.at

†Department of Mathematics, Statistics and Computer Science, University of Illinois at Chicago ; Automata Theory Research Group of the Hungarian Academy of Sciences, Szeged. Partially supported by NSF Grant CCR-9208170 , ESPRIT BRA ILP, Project 6020 and OTKA T 14228. Email: u11557@uicvm.bitnet

VC-dimension) that plays a central role in computational learning theory is the work of Shelah [17] in stability theory. In view of the strong connections between the VC-dimension and PAC learnability, model theoretic results on the VC-dimension have direct implications for learnability.

Learnability also raises *new types of related questions* as well, for instance when one studies learnability with different kinds of *queries* such as equivalence queries. Another general question (that is not discussed in the present paper) is the complexity of finding a *hypothesis* that is *consistent* with a set of positive and negative examples.

The results of this paper give a rough classification of the difficulty of learning problems depending on the *language*, i.e on the number and arities of its predicate and function symbols, and on the number of *quantifiers* allowed.

Given a first-order language $\mathcal{L}$, a class Φ of formulas over $\mathcal{L}$, a model M and a model of learning such as PAC learning , we consider the problem of learning a target concept in M defined by a formula $\varphi \in \Phi$. We look at the following question. When is the *complexity* of this learning problem (such as the number of examples or the number of queries) bounded *independently* of the size of M ? If this is the case then Φ is called an *easy* class of formulas with respect to the learning model considered. As we do *not* take into consideration any additional information that might be available about the database (besides the arities of its predicate and function symbols) , it is to be expected that the range of positive results is rather limited.

If the language contains *at least one predicate or function of arity greater than one* then a standard example described in Section 2 shows that the complexity of learning *quantifier free* formulas in the learning models considered can *grow* with the size of the universe. Hence positive results about easiness in the sense mentioned above can be hoped for only if the language contains only *unary* predicates and functions. Even in this case, if there are *at least two unary functions* , then the example mentioned above can be modified to show that a positive result does not hold for formulas that contain *at least one quantifier*. It turns out that in the two cases left open by these considerations one can indeed prove positive results. These cases are discussed in Sections 3 and 4.

In Section 3 it is shown that if the language contains only *unary* predicates and functions then the class of *quantifier free formulas in conjunctive normal form* , having bounded size , is easy for learning with *equivalence queries*. The algorithm described uses an idea of Blum [4] (introduced in the context of learning with infinitely many attributes) to handle

negations .

In Section 4 it is shown that if the language contains only *unary* predicates and *a single unary function* then *arbitrary* first-order formulas of bounded size are easy for *PAC learning* . In view of the general results on PAC learnability and the VC - dimension this follows from the following known result in model theory: the VC -dimensions of the concept classes involved have a finite upper bound (Korec and Rautenberg [12] , Shelah [16]) . Here we give a proof of this result that provides an *explicit* (although very large) upper bound for the VC - dimension and uses only *finite combinatorial methods* that are useful in the study of finite models in general. These are Ramsey's theorem, Fraissé - Ehrenfeucht games , a result of Marcus [14] and Gaifman [7] on *local properties* and a result of Laskowski [13] on definability with a single parameter (see also Shelah [16] , Baldwin and Shelah [3]) . We note that some related questions on *definability in finite models* are discussed in Babai and Turán [2] and Hella, Kolaitis and Luosto [8] .

2 Preliminaries

A first-order *language* $\mathcal{L}$ contains *predicate* or *relation* symbols $P_1, \ldots, P_r$ and *function* symbols $f_1, \ldots, f_s$. *Terms* and *atomic formulas* are as usual. A *literal* is an atomic formula or its negation. A *structure* or *model* for $\mathcal{L}$ is a pair $M = (A, I)$, where A is a set and I is the interpretation of predicate and function symbols in $\mathcal{L}$ as relations and functions on A . For the results of this paper it does not have to be assumed that A is finite. A *formula* φ with free variables from $\mathbf{x} = (x_1, \ldots, x_n)$ is written as $\varphi(x_1, ..., x_n)$ or $\varphi(\mathbf{x})$. The *complexity* of a formula is the number of symbols it contains. If $\varphi(\mathbf{x})$ is a formula and M is a structure then the relation, or *concept* defined by φ on M is $C_{\varphi,M} := \{\mathbf{a} = (a_1, \ldots, a_n) \in A^n \ : \ M \models \varphi(\mathbf{a})\}$.
More generally, let $\varphi(\mathbf{x}, \mathbf{y})$ be a formula, where $\mathbf{x} = (x_1, ..., x_n)$ and $\mathbf{y} = (y_1, ..., y_m)$. Thus the free variables of φ are from $x_1, ..., x_n, y_1, ..., y_m$. Here $x_1, ..., x_n$ are called the *variables* of φ and $y_1, ..., y_m$ are called the *parameters* of φ . We write $par(\varphi) = m$ for the number of parameters of φ . Then for $\mathbf{b} = (b_1, ..., b_m) \in A^m$, the relation or *concept* defined by φ on M with *parameters* $\mathbf{b}$ is $C_{\varphi,\mathbf{b},M} := \{\mathbf{a} = (a_1, \ldots, a_n) \in A^n \ : \ M \models \varphi(\mathbf{a}, \mathbf{b})\}$.

Let Φ be a class of formulas. Then the class of relations defined by formulas in Φ on M , or the *concept class* corresponding to Φ on M is $C_\Phi(M) := \{C_{\varphi,\mathbf{b},M} : \varphi \in \Phi, \mathbf{b} \in A^{par(\varphi)}\}$. Thus, we consider the class of

concepts defined by formulas in Φ with all possible choices of the parameters. It will be assumed that formulas in Φ have the *same number of variables* , although they may have a *different number of parameters.*

Now we give a brief description of the *learning models* considered in this paper. For a general introduction to *computational learning theory* we refer to Kearns and Vazirani [10] . In the general framework a *learning problem* is specified by a *domain* X and a *concept class* $\mathcal{C}$ consisting of some subsets of X. Thus in our case the domain is A^n for some n, and $\mathcal{C}$ is of the form $\mathcal{C}_\Phi(M)$ as above. It is assumed that there is an unknown *target concept* C_T belonging to $\mathcal{C}$. The goal of the *learner* is to *identify* the target concept.

In the model of *learning with equivalence queries* (Angluin [1]) the learning process consists of a sequence of equivalence queries. An *equivalence query* is a concept $C \in \mathcal{C}$. The *response* to such a query is either *yes* if $C_T = C$, or a *counterexample* from $C_T \triangle C$. The learning process is completed when the response *yes* is received. The *complexity* of a learning algorithm is the number of queries it asks in the worst case.

In the *PAC (Probably Approximately Correct)* learning model of Valiant [18] it is assumed that there is an unknown probability distribution P on the domain. A learning algorithm draws a set of m *examples*, i.e. a set of pairs of the form $(\mathbf{a}, \epsilon)$, where $\mathbf{a}$ is an element of the domain and ϵ is $+$ or $-$, indicating whether $\mathbf{a}$ belongs to the target concept. The learning algorithm then outputs a *hypothesis* $H \in \mathcal{C}$. A learning algorithm is an $(\epsilon, \delta)-$*learning algorithm* with *sample complexity* m if for every target concept $C_T \in \mathcal{C}$ and for every probability distribution P it holds that $Prob(error_P(H \triangle C_T) \geq \epsilon) < \delta$. Here *Prob* is probability with respect to the product distribution P^m on samples of size m and $error_P(H \triangle C_T)$ is the probability of $H \triangle C_T$ under the distribution P .

A subset Y of the domain X is *shattered* if for every $Z \subseteq Y$ there is a concept $C \in \mathcal{C}$ such that $Z = Y \cap C$. The *Vapnik - Chervonenkis dimension (VC-dimension)* $VC(\mathcal{C})$ of the concept class $\mathcal{C}$ is the size of a largest shattered set.

The Vapnik - Chervonenkis dimension $VC(\mathcal{C})$ provides a *lower bound* for the complexity of every algorithm that learns $\mathcal{C}$ with equivalence queries.

The main result on PAC - learnability and the VC - dimension is due to Blumer, Ehrenfeucht, Haussler and Warmuth [5] , based on the work of Vapnik and Chervonenkis [19] . It states the following: if $VC(\mathcal{C})$ is finite, then any learning algorithm that draws $O(\frac{1}{\epsilon} \log \frac{1}{\delta} + \frac{VC(\mathcal{C})}{\epsilon} \log \frac{1}{\epsilon})$ examples and outputs a consistent hypothesis is an (ϵ, δ) -learning algorithm. (The theorem assumes some measure theoretic conditions that are omitted. These

conditions always hold, for example, for finite structures.) On the other hand, if C is a non-trivial concept class then every (ϵ, δ) -learning algorithm must use samples of size $\Omega(\frac{1}{\epsilon} \log \frac{1}{\delta} + \frac{VC(C)}{\epsilon})$. (A concept class is non-trivial if it either consists of a single concept, or of two concepts partitioning the domain.)

Now let us introduce the notion of *easy* classes of formulas that is used for convenience throughout this paper.

Let $\mathcal{L}$ be a first-order language and Φ be a class of formulas over $\mathcal{L}$. Then Φ is *easy for learning with equivalence queries* if there is an N such that for every model M the concept class $C_\Phi(M)$ can be learned with at most N equivalence queries. Similarly, Φ is *easy for PAC learning* if for every $\epsilon, \delta > 0$ there is an $N = N(\epsilon, \delta)$ such that for every model M there is an (ϵ, δ)-learning algorithm for the concept class $C_\Phi(M)$ of sample complexity at most N . Easiness for learning with equivalence queries implies easiness for PAC learning.

In the remainder of this section we describe the examples showing that the class of subsets defined by simple formulas can have high VC-dimension if the language contains at least one binary predicate or at least two unary functions.

Let R be a binary predicate and $\Phi := \{R(x, y)\}$. Here x is a variable and y is a parameter. The following example is standard. Let M be a structure with $A = A_1 \cup A_2$, $|A_1| = n$, $|A_2| = 2^n$, such that the sets $\{a \in A_1 : R(a, b)\}$ are different for every $b \in A_2$. Then $C_\Phi(M)$ shatters A_1 and so $VC(C_\Phi(M)) \geq n$.

A similar example can be given if the language contains two unary functions f and g . Let $\Psi := \{\exists z(f(z) = x \wedge g(z) = y)\}$. Let M' be the structure obtained from M above as follows. If $R(a, b)$ holds in M for $a \in A_1, b \in A_2$ then a new element c is added such that $f(c) = a$ and $g(c) = b$. Then again $C_\Psi(M')$ shatters A_1 and so $VC(C_\Psi(M')) \geq n$.

In view of the general lower bounds mentioned above these examples indicate that for positive results about easiness with respect to learning with equivalence queries or PAC learning one should consider languages with only *unary* predicates and functions. If there are at least *two* unary functions then the formulas should be *quantifier free.*

3 Quantifier free unary formulas are easy for learning with equivalence queries

In this section we consider languages that contain only unary predicate and function symbols. *Quantifier free formulas* are Boolean combinations of atomic formulas. We will consider quantifier free formulas in *conjunctive normal form* , i.e. as a conjunction of *clauses*, where each clause is a disjunction of literals.

Let $CNF_{n,c,l,d}$ be the class of quantifier free formulas in conjunctive normal form, with variables from $x_1, \ldots, x_n$, consisting of at most c clauses, each containing at most l literals and assuming that each term occurring has depth at most d . For fixed n, c, l and d formulas belonging to $CNF_{n,c,l,d}$ will be referred to as *permissible* formulas. Atomic formulas containing terms of depth at most d will be called *permissible atomic formulas.*

Theorem 1 *For every n, c, l and d the class of formulas $CNF_{n,c,l,d}$ is easy for learning with equivalence queries.*

Outline of the proof. Let M be a model. The goal of the learning algorithm to be described is to identify a permissible defining formula for the target concept. The only variables occurring in the defining formula are $x_1, \ldots, x_n$, as the parameters are replaced by elements of the universe.

The first observation is that every n-tuple $\mathbf{a} = (a_1, \ldots, a_n) \in A^n$ satisfies $O(rs^d n + s^{2d} n^2)$ permissible atomic formulas, denoted by $\theta_{\mathbf{a},i}$, $i = 1, \ldots, N_{\mathbf{a}}$.

The learning algorithm can be thought of as building and traversing a search tree of hypotheses based on the counterexamples it receives. Each node is labelled by a permissible formula, corresponding to the equivalence query asked when the algorithm gets to that node. When used as node labels, permissible formulas are allowed to contain *multiple* occurrences of the same clause. In addition, each inner node (already being queried) is labelled by the counterexample obtained and the sign (+ or −) of the counterexample. The next equivalence query is the label of the first leaf. If the response to the query is *yes* then the algorithm terminates. If a counterexample is received then the node is labelled by the counterexample and the tree is updated according to the rules specified below. The node just queried will either become an inner node or it will be removed. If an inner node has no further children that are leaves then it is removed. Initially the tree consists of a single node labelled by the hypothesis $\emptyset$ corresponding to an empty disjunction.

Thus the first equivalence query of the algorithm is $\emptyset$. If the response is *yes* then we are done. Otherwise, let **a** be the positive counterexample received. Let $\theta_{\mathbf{a},i}$, $i = 1, \ldots, N_{\mathbf{a}}$ be the permissible atomic formulas satisfied by **a** . Then for every multiset $\{\theta_{\mathbf{a},i_1}, \ldots, \theta_{\mathbf{a},i_p}\}$, $0 \leq p \leq c$, the root will have a child labelled by $\theta_{\mathbf{a},i_1} \wedge \ldots \wedge \theta_{\mathbf{a},i_p}$. For $p = 0$ we get the empty conjunction, corresponding to the hypothesis A^n .

In general, let a node v be labelled by a permissible formula $\varphi = \varphi_1 \wedge \ldots \wedge \varphi_p$, $p \leq c$ and let us assume that the equivalence query φ is asked. If the response is *yes* then we are done. Otherwise, let the counterexample received be **a** and let the permissible atomic formulas satisfied by **a** be $\theta_{\mathbf{a},i}$, $i = 1, \ldots, N_{\mathbf{a}}$.

If **a** is a *positive* counterexample, then for every $j = 1, \ldots, p$ and every $i = 1, \ldots, N_{\mathbf{a}}$ such that $\varphi_1 \wedge \ldots \wedge \varphi_{j-1} \wedge (\varphi_j \vee \theta_{\mathbf{a},i}) \wedge \varphi_{j+1} \wedge \ldots \wedge \varphi_p$ is permissible, v will have a child labelled by this formula. If there are no such formulas then v is removed.

If **a** is a *negative* counterexample, then for every nonemtpy set $\{\theta_{\mathbf{a},i_1}, \ldots, \theta_{\mathbf{a},i_q}\}$ such that $\varphi_1 \wedge \ldots \wedge \varphi_p \wedge (\overline{\theta_{\mathbf{a},i_1}} \vee \ldots \vee \overline{\theta_{\mathbf{a},i_q}})$ is permissible, v will have a child that is labelled by that formula. If there are no such formulas then v is removed.

This completes the description of the algorithm.

Now we turn to proving the correctness of the algorithm. Let the target concept be ψ and write ψ in the form

$$\psi = \psi_1^+ \wedge \ldots \wedge \psi_{b_1}^+ \wedge \psi_1^- \wedge \ldots \wedge \psi_{b_2}^-$$

where $\psi_1^+, \ldots, \psi_{b_1}^+$ contain only unnegated atomic formulas and $\psi_1^-, \ldots, \psi_{b_2}^-$ each contain at least one negated atomic formula.

We claim that after the first query the tree always contains a node labelled by a formula φ such that the following holds:

- $\varphi = \varphi_1^+ \wedge \ldots \wedge \varphi_{b_1}^+ \wedge \varphi_1^- \wedge \ldots \wedge \varphi_{b_3}^-$ where $b_3 \leq b_2$,
- the literals in φ_i^+ form a subset of the literals in ψ_i^+ for $i = 1, \ldots, b_1$,
- the literals in φ_i^- form a subset of the literals in ψ_i^- that contains *all the negated literals* from ψ_i^- for $i = 1, \ldots, b_3$.

This claim can be proved by induction and it implies the correctness of the algorithm. The easiness of $CNF_{n,c,l,d}$ then follows directly. We note that the number of equivalence queries required by the algorithm is *polynomial* in n , the number of variables.

□

4 First-order formulas over unary predicates and a single unary function are easy for PAC learning

The examples described in Section 2 indicate that easiness results for classes of formulas containing *quantifiers* can only be hoped for if the language contains unary predicates and a *single* unary function. Let $\mathcal{L}$ consist of unary predicate symbols $P_1, \ldots, P_r$ and a single unary function symbol f. Let $\Phi_{n,s}$ denote the class of first-order formulas of size at most s with free variables from $x_1, \ldots, x_n$.

Theorem 2 *For every n and s, the class of formulas $\Phi_{n,s}$ is easy for PAC learning.*

Outline of the proof. Let $M = (A, I)$ be a model over $\mathcal{L}$. In view of the general upper bound on PAC learning mentioned in Section 2, it is sufficient to show that $VC(\mathcal{C}_{\Phi_{n,s}}(M))$ is bounded from above by some constant depending only on n and s.

Let the language $\mathcal{L}_1$ consist of $P_1, \ldots, P_r$ and a binary relation R. Let the interpretation of R on A be defined by $R(a,b) \Leftrightarrow f(a) = b$ and let M_1 be the model over the language $\mathcal{L}_1$ defined on A by replacing f by R (R can also be viewed as the edge set of a directed graph). Thus M_1 is the relational version of M. Then every formula of the original language $\mathcal{L}$ can be transformed into a formula over $\mathcal{L}_1$ that represents the same concept. In what follows we consider the class $\Psi_{n,s}$ of first-order formulas over $\mathcal{L}_1$ of size at most s, with free variables from $x_1, \ldots, x_n$. In order to prove the theorem it is sufficient to show that $VC(\mathcal{C}_{\Psi_{n,s}}(M_1))$ is bounded from above by some constant depending only on n and s.

The *distance* $d(a,b)$ of $a, b \in A$ is the graph theoretic distance of a and b in the undirected graph formed by deleting the directions of the edges in R. For $a \in A$ let $N^{(p)}(a)$ be the $\mathcal{L}_1$-substructure of M_1 containing all elements within distance p to a. $N^{(p)}(a)$ is called the *neighborhood* of a of *radius* p, or the *p-neighborhood* of a. If $A' \subseteq A$ then $N^{(p)}(A') = \bigcup_{a \in A'} N^{(p)}(a)$ is the p-neighborhood of A'. If $\mathbf{a} = (a_1, \ldots, a_n) \in A^n$ then the *q-type* of the *p-neighborhood* of $\mathbf{a}$ is the set of formulas $\varphi(x_1, \ldots, x_n)$ having at most q quantifiers such that $N^{(p)}(\mathbf{a}) \models \varphi(a_1, \ldots, a_n)$. The number of different q -types is bounded from above by some function of n and q. Let $\mathbf{a} = (a_1, \ldots, a_n)$, $\mathbf{b} = (b_1, \ldots, b_n) \in A^n$. If the p-neighborhoods $N^{(p)}(\mathbf{a})$ and $N^{(p)}(\mathbf{b})$ have the *same* q-type then $\mathbf{a}$ and $\mathbf{b}$ are called *q-equivalent.* This is written as $N^{(p)}(\mathbf{a}) \equiv_q N^{(p)}(\mathbf{b})$. Let $Q_1(M_1, p, q)$ be the number of different q - types of the neighborhoods $N^{(p)}(a)$ for $a \in A$.

If an element a does not belong to a cycle of R then one can consider the subtree T_a rooted at a , containing all elements a' such that $f^i(a') = a$ for some i . If a' is such an element then the q *-type* of the pair (T_a, a') is the set of formulas $\chi(x_1, x_2)$ having quantifier depth at most q such that $T_a \models \chi(a, a')$. The number of different q -types is bounded from above by a constant depending only on q. Two pairs (T_a, a') and (T_b, b') are q *-equivalent* if they have the same q *-type* . Let $Q_2(M_1, q)$ be the number of different q -types of the pairs (T_a, a') for $a, a' \in A$ such that $f^i(a') = a$ for some i .

The following lemma was proved by Marcus [14] . Gaifman [7] generalized the result and improved the bounds involved. Here we give the original formulation [14] as it is convenient for our purposes. The functions $\tilde{p}(n,k)$ and $\tilde{q}(n,k)$ can be explicitly given.

Lemma 3 *(Marcus [14], Gaifman [7]) There are functions $\tilde{p}(n,k)$ and $\tilde{q}(n,k)$ such that for every formula $\varphi(x_1, \ldots, x_n)$ with k quantifiers and every $\boldsymbol{a} = (a_1, \ldots, a_n), \boldsymbol{b} = (b_1, \ldots, b_n) \in A^n$ if $N^{\tilde{p}(n,k)}(\boldsymbol{a}) \equiv_{\tilde{q}(n,k)} N^{\tilde{p}(n,k)}(\boldsymbol{b})$ then $M_1 \models \varphi(\boldsymbol{a}) \leftrightarrow \varphi(\boldsymbol{b})$.* □

Let $p := \tilde{p}(s,s)$, $q := \tilde{q}(s,s)$, $Q := max(Q_1(M_1, p, q), Q_2(M_1, q), 3)$ and $N := (16psQ)^{(16psQ)}$. We note that $N \leq F(s)$ for an explicit function F .

Lemma 4 $VC(\mathcal{C}_{\Psi_{1,s}}(M_1)) < N$.

Outline of the proof of the lemma. Let S be a shattered subset of size N . Ramsey's theorem implies that either there are $(4s-1)Q+1$ elements in S with pairwise distance greater than $4p$ or there are $4p((4s-1)Q+3)^{4p}+1$ elements in S with pairwise distance at most $4p$. We consider the two cases separately.

Case 1 There are at least $(4s-1)Q+1$ elements in S with pairwise distance greater than $4p$.

By the definition of Q there is a set S_1 of $4s$ elements from S , such that their pairwise distances are greater than $4p$ and their p-neighborhoods have the same q-type. We claim that S_1 is not shattered by $\mathcal{C}_{\Psi_{1,s}}(M_1)$.

Let $\varphi(x, y_1, \ldots, y_m)$, $m \leq s$ be a formula from $\Psi_{1,s}$. Let $\mathbf{c} = (c_1, \ldots, c_m) \in A^m$ be an arbitrary setting of the parameters. For every c_i there can be at most one $a \in S_1$ such that $d(c_i, a) \leq 2p$. Hence there is a set S_2 of at least $3s$ a's in S_1 such that for every c_i it holds that $d(a, c_i) > 2p$. Thus for every $a \in S_2$ one has $N^{(p)}(a) \cap N^{(p)}(\mathbf{c}) = \emptyset$. A

standard argument implies that for every $a, b \in S_2$ it holds that $N^{(p)}(a, \mathbf{c}) \equiv_q N^{(p)}(b, \mathbf{c})$. Lemma 3 implies that $M_1 \models \varphi(a, \mathbf{c})$ iff $M_1 \models \varphi(b, \mathbf{c})$. Thus $C_{\varphi, \mathbf{c}, M_1}$ either contains all S_2 or it is disjoint from it. Therefore $|C_{\varphi, \mathbf{c}, M_1} \cap S_1|$ is either at least $3s$ or at most s . Hence, for example, $C_{\varphi, \mathbf{c}, M_1}$ cannot cut S_1 in half, and thus indeeed S_1 is not shattered by $\mathcal{C}_{\Psi_{1,s}}(M_1)$.

Case 2 There are $(4p-1)((4s-1)Q+3)^{4p}+1$ elements in S such that their pairwise distances are at most $4p$.

In this case a similar argument is used by finding several q - equivalent pairs (T_a, a') such that the roots are adjacent to the same vertex. The details are omitted.

□

The proof of the Theorem 2 is completed by using a general result of Laskowski [13] . The details are again omitted.

□

Acknowledgement We would like to thank John Baldwin and Tamás Horváth for their comments.

References

[1] D.Angluin: Queries and concept learning, *Machine Learning* **2** (1987), 319-342.

[2] L.Babai, Gy.Turán: The complexity of defining a relation on a finite graph, *Zeitschr. f. Math. Logik und Grundl. der Math.* **33** (1987), 277-288.

[3] J.T.Baldwin, S.Shelah: Second order quantifiers and the complexity of theories, *Notre Dame J. of Formal Logic* **26** (1985), 229-302.

[4] A.Blum: Learning Boolean functions in an infinite attribute space, *Machine Learning* **9** (1992), 373-386.

[5] A.Blumer, A.Ehrenfeucht, D.Haussler, M.K.Warmuth: Learnability and the Vapnik - Chervonenkis dimension, *J. of the ACM* **36** (1989), 929-965.

[6] W.W.Cohen, H.Hirsh: The learnability of description logics with equality constraints, *Machine Learning* **17** (1994), 169-199.

[7] H.Gaifman: On local and non-local properties, in: *Proc. of the Herbrand Symp., Logic Colloquium '81*, ed. J.Stern, 105-135. North - Holland, 1982.

[8] L.Hella, P.G.Kolaitis, K.Luosto: How to define a linear order on finite models?, *Proc. 9th IEEE Symp. on Logic in Computer Science* (1994), 40-49.

[9] T.Horváth, Gy.Turán: Learning logic programs with structured background knowledge, 1995, submitted.

[10] M.J.Kearns, U.V.Vazirani: *An Introduction to Computational Learning Theory.* ACM Press, 1994.

[11] J.-U. Kietz, S.Dzeroski: Inductive logic programming and learnability, *SIGART Bulletin* **5** (1994), 22-32.

[12] I.Korec, W.Rautenberg: Model - interpretability into trees and applications, *Archiv f. Math. Logik u. Grundlagenforschung* **17** (1976), 97-104.

[13] M.C.Laskowski: Vapnik - Chervonenkis classes of definable sets, *J. London Math. Soc.* **45** (1992), 377-384.

[14] L.Marcus: Minimal models of one function symbol, *Israel J. Math.* **18** (1974), 117-131.

[15] D.N.Osherson, M.Stob, S.Weinstein: New directions in automated scientific discovery, *Information Sciences* **57-58** (1991), 217-230.

[16] S.Shelah: Stability, the f.c.p. and superstability, *Annals of Mathematical Logic*, **3**(1971), 271-362.

[17] S.Shelah: A combinatorial problem; stability and order for models and theories in infinitary languages, *Pacific J. of Math.* **41** (1972), 241-261.

[18] L.G.Valiant: A theory of the learnable, *Comm. ACM* **27** (1984), 1134-1142.

[19] V.N.Vapnik, A.Y.Chervonenkis: On the uniform convergence of relative frequencies of events to their probabilities, *Theory of Prob. and its Appl.* **16** (1971), 264-280.