Algorithmica (1986) 1: 361-385

Algorithmica

A Time Bound on the Materialization of Some Recursively Defined Views[1]

Yannis E. Ioannidis[2]

Abstract. A virtual relation (or view) can be defined with a recursive Horn clause that is a function of one or more base relations. In general, the number of times such a Horn clause must be applied in order to retrieve all the tuples in the virtual relation depends on the contents of the base relations of the definition. However, there exist Horn clauses for which there is an upper bound on the number of applications necessary to form the virtual relation, independent of the contents of the base relations. Considering a restricted class of recursive Horn clauses, we give necessary and sufficient conditions for members of the class to have this bound.

Key Words. Logic and databases, Horn clauses, Recursively defined relations, Uniformly bounded recursion.

1. Introduction. In the past few years major attempts have been made to improve the power of database systems, in particular those based on the relational model [Codd]. A significant part of this effort has been in the formalization, design, and development of deductive databases. Deductive databases are defined as "databases in which new facts may be derived from facts that were explicitly introduced" [Gall]. A major difference between a deductive and a conventional relational database is that in the former new facts may be derived recursively. This very characteristic of deductive databases is what makes query processing a difficult task in such an environment. The main problem that arises is how to detect the point at which further processing gives no more answers to a given query. Many researchers have studied and proposed solutions to this termination problem for various cases [Reit], [Chan 2], [Hens], [Viei].

A common characteristic among all the proposed solutions that we are aware of is that the termination condition relies on the data explicitly stored in the database. In general this is necessary. However, there are some cases where a termination condition exists, which is independent of the particular instance of the database. The purpose of this paper is to identify and characterize these

[1] This research was supported by the National Science Foundation under Grant ECS-8300463.

[2] Department of Computer Science, 1210 West Drayton St., University of Wisconsin, Madison, WI 53706, USA.

Received March 15, 1986; revised July 10, 1986. Communicated by Dale Skeen.

cases. Restricted to a particular class of recursive formulas, we give necessary and sufficient conditions for the existence of a data-independent termination condition.

The paper is organized as follows. In Section 2 the basic definitions for the study of deductive databases are given. Our investigation is restricted to a subset of all possible deductive databases. The restrictions imposed on the database are outlined and justified. Uniformly bounded recursion is introduced in Section 3. Some general results together with related work is presented as well. Section 4 contains the description of the graph model used as a tool to derive the results presented. Using this graph model, necessary and sufficient conditions for uniformly bounded recursion are given in Section 5. The theorems are illustrated with a number of characteristic examples. In Section 6 we characterize uniform boundedness for a slightly more general class of recursive formulas. Finally, in Section 7, the results are summarized and more problems for future work in the area are discussed.

2. Assumptions.

The following definitions about first-order formulas [Ende] are useful in the forthcoming analysis.

DEFINITION 1. A first-order formula is equivalent to a *Horn clause* if it is of the form $A_1 \wedge A_2 \wedge \cdots \wedge A_n \to C$. All the variables appearing in the formula are (implicitly) universally quantified. The formula to the left of $\to$ is called the *antecedent* and that to the right of $\to$ the *consequent*.

Each one of $C, A_1, A_2, \ldots, A_n$ is an atomic formula [Ende], i.e., it is of the form $P(t_1, t_2, \ldots, t_n)$, where P is a relation (predicate) symbol and t_i, $1 \leq i \leq n$, is a term (a variable symbol or a constant symbol or a function symbol "applied" on terms). A Horn clause is *recursive* if the relation that appears in the consequent appears at least once in the antecedent as well.

DEFINITION 2. The sole relation appearing in the consequent of a recursive Horn clause is called the *recursive* relation. Any other relation in the Horn clause is called *nonrecursive.*

DEFINITION 3. A recursive Horn clause is called *linear* if its recursive relation appears only once in the antecedent.

DEFINITION 4. Two variables x, y *appear under the same relation* in a Horn clause if there is an atomic formula $P(\ldots, x, \ldots, y, \ldots)$ appearing in the Horn clause, where P is a relation symbol.

DEFINITION 5. A variable is called *distinguished* if it appears under the recursive relation in the consequent of a Horn clause. Otherwise it is called *nondistinguished.*

A deductive database is a relational database [Codd] enhanced with a set of Horn clauses. If there is some recursive Horn clause (or a set of mutually recursive Horn clauses) appearing in the database, then the termination problem mentioned in Section 1 arises. We address the problem with respect to the processing of a single recursive Horn clause only (immediate recursion).

We restrict our attention to recursive Horn clauses that satisfy the following conditions:

R1 The recursive Horn clause is *linear.*
R2 There are *no function symbols* in the Horn clause.
R3 There are *no constant symbols* in the Horn clause.
R4 There are *no repeated* variables in the consequent.
R5 No subsequence of distinguished variables in the consequent is a *permutation* of the corresponding subsequence of the variables under the recursive relation in the antecedent.

The motivation behind restriction R1 is simplicity. Moreover, most of the recursive Horn clauses in a real world system are expected to be linear. Function symbols in a recursive Horn clause may lead to infinite relations. Situations like that are not easily handled in a database environment, if at all. Restriction R2 is imposed to avoid them. The last three restrictions are imposed for the sole purpose of getting a uniform result. It is the goal of our future research to remove them, thereby generalizing our results.

Finally, we assume that there are no equalities in the Horn clause. Any equality may be removed by replacing one of its variables with the other, wherever it appears in the Horn clause. It is clear that the new Horn clause is equivalent to the initial one.

Definition 6. A recursive Horn clause is called *simple* if it satisfies conditions R1–R5 and does not contain any equality symbol.

3. Bounded Recursion. Let the following be a simple recursive Horn clause.

$$P(x_1, x_2, \ldots, x_m) \wedge \beta \rightarrow P(y_1, y_2, \ldots, y_m). \tag{1}$$

Subformula β is a conjunction of atomic formulas, with relations other than P. The following infinite sequence of nonrecursive Horn clauses is equivalent to (1):

$$P_0(x_1^{(0)}, x_2^{(0)}, \ldots, x_m^{(0)}) \wedge \beta^{(0)} \rightarrow P_1(y_1, y_2, \ldots, y_m)$$
$$P_0(x_1^{(1)}, x_2^{(1)}, \cdots, x_m^{(1)}) \wedge \beta^{(1)} \wedge \beta^{(0)} \rightarrow P_2(y_1, y_2, \ldots, y_m)$$
$$P_0(x_1^{(2)}, x_2^{(2)}, \ldots, x_m^{(2)}) \wedge \beta^{(2)} \wedge \beta^{(1)} \wedge \beta^{(0)} \rightarrow P_3(y_1, y_2, \ldots, y_m)$$

In the above, it is $\beta^{(i)} = \beta[s_i]$ and $P_0(x_1^{(i)}, x_2^{(i)}, \ldots, x_m^{(i)}) = P_0(x_1, x_2, \ldots, x_m)[s_i]$, for some substitution s_i of the variables in (1). Note that s_0 substitutes each variable for itself. The ith Horn clause above is called the $(i-1)$th *expansion* of (1). So, the first one of these Horn clauses is the 0th expansion. Each one of these expansions is applied on P_0, the initial contents of P. P is equal to $\bigcup_{i=0}^{\infty} P_i$. The ith expansion of a recursive Horn clause α is denoted by α_i. Whenever this creates no confusion, the same notation is used to denote the recursive Horn clause produced by removing the subscripts from P in α_i.

In a database environment all the relations are finite. Furthermore, only function-free recursive Horn clauses are considered. Hence, even though a recursive Horn clause is equivalent to an infinite sequence of nonrecursive Horn clauses, the latter stop producing new tuples for P after some point. The process terminates exactly when some Nth expansion of (1) fails to produce any new tuples for the first time. In general, N depends on the database contents. Our goal is to identify and characterize simple recursive Horn clauses, for which this is not true, i.e., the number of expansions needed to materialize the relation defined is independent of the database contents.

The problem can be addressed in two frameworks. In the first, P_0 is produced by a given nonrecursive Horn clause. In the second, P_0 is stored in the database. The question in the first case is one of *boundedness*, whereas in the second it is one of *uniform boundedness*. In this paper we address the question of uniform boundedness only.

DEFINITION 7. A simple recursive Horn clause is called *uniformly bounded* if it is equivalent to a finite number of its expansions.

DEFINITION 8. Let a uniformly bounded simple recursive Horn clause α be equivalent to its first N expansions, $\alpha_0, \alpha_1, \ldots, \alpha_{N-1}$. The smallest such N is called the *order* of α.

EXAMPLE 1. Consider the following simple recursive Horn clause α:

$$\alpha: \quad reachable(x) \wedge edge(x, y) \rightarrow reachable(y).$$

If *edge* represents the edges of a directed graph, then *reachable* denotes the nodes of the graph reachable from the ones originally contained in it. In general, the number of times α has to be applied to get the materialization of *reachable* is not known. It depends on the initial contents of *edge* (the graph) and *reachable*.

EXAMPLE 2. As another example of a simple recursive Horn clause, consider β:

$$\beta: \quad P(z) \wedge NP\_c(z) \wedge NP\_c(y) \rightarrow P(y).$$

If NP_c is the set of NP-complete problems and P is the set of problems evaluated in polynomial time, then β states the well-known theorem that if one NP-complete problem is in P, then all are [Lewi]. Clearly, one application of β is enough to materialize P, regardless of the initial contents of the relations P and NP_c. If there is one tuple in P that joins with (that is, is equal to) some tuple in NP_c, then β produces for P all the tuples in NP_c. Any further applications of β fail to produce new tuples for P. So β, unlike α, is uniformly bounded with order equal to 1.

Uniform boundedness of Horn clauses has been addressed in the past. Minker and Nicolas give a sufficient condition for a Horn clause to be uniformly bounded [Mink]. In particular, they define a restricted class of Horn clauses, called *singular*, and show that any singular Horn clause is uniformly bounded. They do allow nonlinearity, but singular Horn clauses are restricted in the way relations share variables.

The problem has also been addressed under a tableau formulation [Sagi 2]. Representing a Horn clause by a tableau, Sagiv gives necessary and sufficient conditions for a set of Horn clauses to be uniformly bounded. The restrictions imposed are that there exists only one relation symbol in the Horn clauses, and that the Horn clauses are *typed*, i.e., no variable appears in more than one column of the relation. Similar results have been given by Cosmadakis and Kanellakis also [Cosm].

More recently, Naughton has addressed the same problem [Naug]. The class of Horn clauses he considers is similar to the one of simple Horn clauses. He does not impose restriction R5, but he does not allow a nonrecursive relation to appear more than once in the antecedent either. Besides giving a necessary and sufficient condition for uniform boundedness for this class, he also addresses boundedness as well as the case with multiple recursive Horn clauses.

Some tools developed for the study of conjunctive queries [Chan 1] and tableaux [Aho] are helpful in addressing the uniform boundedness problem of Horn clauses also.

DEFINITION 9. A *valuation* θ is a function from variables to constants. Applying θ on an atomic formula $Q(x_1, \ldots, x_n)$ gives the tuple $\langle\theta(x_1), \ldots, \theta(x_n)\rangle$.

DEFINITION 10. Consider two nonrecursive Horn clauses α and β. A *homomorphism* $h: \alpha \to \beta$ is a mapping from the variables of α into those of β, such that:

(i) If x, y are distinguished variables appearing in the same argument position in the consequent of α and β, respectively, then $h(x) = y$.
(ii) If $Q(x_1, \ldots, x_n)$ appears in the antecedent of α, then $Q(h(x_1), \ldots, h(x_n))$ appears in the antecedent of β.

When it is well defined, composition of homomorphisms h_1 and h_2 is denoted by $h_1 \circ h_2$.

DEFINITION 11. For two nonrecursive Horn clauses α and β, α is *more restrictive* than β, denoted $\alpha \leq_r \beta$, if for any database instance the relation produced by α is a subset of that produced by β. Clearly, $\leq_r$ is a partial order.

DEFINITION 12. For two nonrecursive Horn clauses α and β, α is *equivalent* to β, denoted $\alpha =_r \beta$, iff there exists an isomorphism $h: \alpha \to \beta$, that is a homomorphism that is one-to-one and onto.

The following lemma is a slight extension of a similar result on typed tableaux [Abo].

LEMMA 1. *For two nonrecursive Horn clauses α and β, it is $\alpha \leq_r \beta$ iff there exists a homomorphism $h: \beta \to \alpha$.*

PROOF. Let d_i (D_i), $0 \leq i \leq m$, be the distinguished variable in the ith argument position in the consequent of α (β).

(*if*) Assume that there exists a homomorphism $h: \beta \to \alpha$. Let θ be a valuation on the variables of α. Consider a database instance such that relation Q contains the tuple $\langle \theta(x_1), \ldots, \theta(x_n) \rangle$ iff an atomic formula $Q(x_1, \ldots, x_n)$ appears in the antecedent of α. The composition of θ and h, $\theta' = \theta \circ h$, is a valuation on β. Property (ii) of homomorphisms assures that for an atomic formula $Q(y_1, \ldots, y_n)$ in the antecedent of β, the tuple $\langle \theta'(y_1), \ldots, \theta'(y_n) \rangle \in Q$. Property (i) of homomorphisms guarantees that $\theta(d_i) = \theta'(D_i)$, $1 \leq i \leq m$. Thus, any tuple in the relation produced by α is in the one produced by β was well. So, $\alpha \leq_r \beta$.

(*Only if*) Consider a one-to-one valuation θ from the variables in α onto some set of constants C. Consider a database instance such that relation Q contains the tuple $\langle \theta(x_1), \ldots, \theta(x_n) \rangle$ iff an atomic formula $Q(x_1, \ldots, x_n)$ appears in the antecedent of α. Then the tuple $\langle \theta(d_1), \ldots, \theta(d_n) \rangle$ is in the relation produced by α. Since it is $\alpha \leq_r \beta$, it has to be in the relation produced by β as well. Thus, a valuation θ' from the variables of β into the set of constants C exists, such that $\theta'(D_i) = \theta(d_i)$, $0 \leq i \leq m$, and for any atomic formula $Q(y_1, \ldots, y_n)$ in the antecedent of β, $\langle \theta'(y_1), \ldots, \theta'(y_n) \rangle \in Q$. θ is one-to-one and onto, so its inverse θ^{-1} is defined. Taking the composition $h = \theta^{-1} \circ \theta'$, it is easy to verify that it is a homomorphism from the variables of β into the variables of α. □

LEMMA 2. *Let α_s and α_t, $0 \leq s \leq t$, be two expansions of some simple recursive Horn clause α, such that $\alpha_t \leq_r \alpha_s$. Then it is $\alpha_{t+k} \leq_r \alpha_{s+k}$, for all $k \geq 0$.*

PROOF. It is shown by induction on k. The basis case $k = 0$ is given. Assume that $\alpha_{t+k-1} \leq_r \alpha_{s+k-1}$ is true. By Lemma 1, there exists a homomorphism $h: \alpha_{s+k-1} \to \alpha_{t+k-1}$. Consider α_{s+k} and α_{t+k}. From the way expansions are formed, it is clear that the new part in the antecedent of α_{s+k} is isomorphic to the new part in the antecedent of α_{t+k} (they are both equivalent to α_0). Let h_1 be this isomorphism from the part of α_{s+k} onto the part of α_{t+k}. Consider the mapping $h': \alpha_{s+k} \to \alpha_{t+k}$ defined as follows:

$$h'(x) = \begin{cases} h(x) & \text{if } x \text{ appeared in } \alpha_{s+k-1} \\ h_I(x) & \text{if } x \text{ is new.} \end{cases}$$

Clearly, h' is a homomorphism from α_{s+k} into α_{t+k}. By Lemma 1, it is $\alpha_{t+k} \leq_r \alpha_{s+k}$. □

LEMMA 3. *A simple recursive Horn clause α is uniformly bounded iff there exist n, N, $n < N$, such that $\alpha_N \leq_r \alpha_n$.*

PROOF. (*If*) By Lemma 2 and the transitivity of $\leq_r$, if $\alpha_N \leq_r \alpha_n$ then for all $k \geq N$ there exists some $n' < N$ such that $\alpha_k \leq_r \alpha_{n'}$. Hence, α is equivalent to its first N expansions. Its order is at most N.

(*Only if*) This part follows the proof of [Sagi 1], and is similar to Lemma 1. Let α be equivalent to its N first expansions, α_0 through α_{N-1}. Assume that all of them share the same distinguished variables d_i, $1 \leq i \leq m$. Consider a one-to-one valuation θ from the variables of α_N onto some set of constants C. Consider a database instance such that relation Q contains the tuple $\langle \theta(x_1), \ldots, \theta(x_l) \rangle$ iff an atomic formula $Q(x_1, \ldots, x_l)$ appears in the antecedent of α_N. Then the tuple $\langle \theta(d_1), \ldots, \theta(d_m) \rangle$ is in the relation produced by α_N. Since α is equivalent to its first N expansions, it has to be in the relation produced by α_n, for some $n < N$, as well. Thus, a valuation θ' from the variables of α_n into the set of constants C exists, such that for any atomic formula $Q(y_1, \ldots, y_l)$ in the antecedent of α_n, $\langle \theta'(y_1), \ldots, \theta'(y_l) \rangle \in Q$. θ is one-to-one and onto, so its inverse θ^{-1} is defined. It is easy to verify that the composition $h = \theta^{-1} \circ \theta'$ is a homomorphism from the variables of α_n into the variables of α_N. Hence, by Lemma 1, it is $\alpha_N \leq_r \alpha_n$. □

Consider a uniformly bounded simple recursive Horn clause of order N. By Lemma 3, there exists n, $n < N$, such that $\alpha_N \leq_r \alpha_n$. Whenever $\alpha_k \leq_r \alpha_l$ implies $(k - l) = c(N - n)$, for some integer $c \geq 0$, following the terminology of [Clif], we define n to be the *index* and $N - n$ the *period* of α.

4. The Model. The examples given in Section 3 indicate that the way in which the variables of a Horn clause are connected with each other through the relations, plays an important role in whether the Horn clause is uniformly bounded or not. In this section a graph model is developed for simple recursive Horn clauses. The form of the graph reflects the connection among the variables in the Horn clause.

Let α be a simple recursive Horn clause. It is modeled by a labeled, weighted, directed graph constructed as follows:

(i) There is a node in the graph for every variable in α.
(ii) If two variables x, y appear under some nonrecursive relation Q in α then an undirected edge $(x - y)$ is put in the graph between the corresponding two nodes x, y. The label of the edge is Q and its weight is 0.
(iii) If two variables x, y appear in the same argument position of the recursive relation P in the antecedent and the consequent, respectively, then a directed edge $(x \rightarrow y)$ is put in the graph from node x to node y with weight 1 and its inverse edge $(y \rightarrow x)$ with weight -1. Each directed edge has label P.

The graph corresponding to a simple recursive Horn clause α is called the *α-graph*. The subgraph induced on the α-graph by the undirected edges defined in (ii) is called the *static α-graph*. The spanning subgraph of the α-graph having the directed edges defined in (iii) as its edge set is called the *dynamic α-graph.* The weight of a path (cycle) in the graph is the sum of the weights of the edges along the path (cycle). Regarding static edges, they can be traversed in both directions, as if there were two opposite directed edges.

EXAMPLE 3. Consider the following simple recursive Horn clause:

$$\alpha: \quad P(z, w) \wedge Q(z, x) \wedge R(w, u) \wedge S(u, x, y) \rightarrow P(x, y).$$

The α-graph is shown in Figure 1.

In terms of the graph model, restrictions R4 and R5 together may be stated as

R4, R5 The dynamic graph (restricted on the positive edges) is a forest.

According to the definition of the graph model, there is a one-to-one correspondence between the positive and the negative dynamic edges. The positive ones alone are enough to carry all the information captured by the dynamic edges in the graph. Hereafter, we refer to the dynamic graph as containing the positive edges only, the negative ones implicitly assumed only whenever the weight of a path is discussed. Likewise, in all the figures only the positive edges are drawn. Finally, since the weight of some edge is determined from whether it is static (weight zero) or dynamic (weight one), no weight is put on the edges.

5. Characterizing Uniform Boundedness. Uniform boundedness for simple recursive Horn clauses is characterized by the following theorem:

THEOREM 1. *A simple recursive Horn clause α is uniformly bounded iff the α-graph contains no cycle of nonzero weight. In that case the order of α is equal to the maximum path-weight in the α-graph.*

Without loss of generality, we restrict our attention to Horn clauses that involve only binary nonrecursive relations. Any n-ary relation R, $n > 2$, is equivalent to n binary relations R_i, constructed as follows. Each tuple of R is assigned a unique

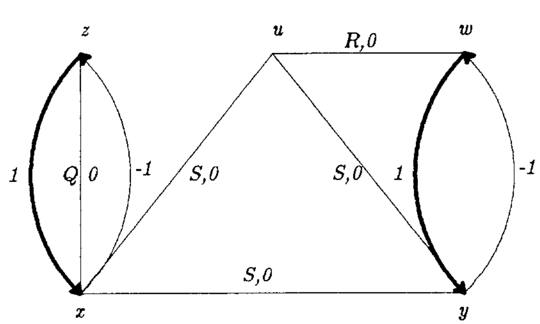

Fig. 1. The α-graph.

identifier, called *tid* [Ston]. Relation R_i is created by combining tids with the corresponding values in the ith column of R. Clearly, R may be reconstructed from the R_i's by joining all of them on the tid column. That generality is not lost when considering only binary nonrecursive relations is justified by the following proposition:

PROPOSITION 1. *Let α be a simple recursive Horn clause and α' the Horn clause produced by replacing all n-ary, $n>2$, nonrecursive relations of α with binary relations according to the description above. The α-graph contains no nonzero weight cycle iff the α'-graph contains no nonzero weight cycle.*

PROOF. Any two variables that appear under some n-ary, $n>2$, relation in α are connected through a zero weight edge in the α-graph and through two zero weight edges in the α'-graph. The common node of the two edges is the one corresponding to the tid column. Hence, for any path in the α-graph there exists another one in the α'-graph with the same weight and vice-versa. □

Also, every (connected) component of the graph of some simple recursive Horn clause α, expands independently of the other. Hence, uniform boundedness of α is equivalent to uniform boundedness of all of its components. Lemma 4 formalizes the above.

LEMMA 4. *Let α be a simple recursive Horn clause and the α-graph consist of M connected components β^1 through β^M. If, for all $1\le i\le M$, β^i is uniformly bounded then α is uniformly bounded as well. Moreover, if β^i is of order N_i and period P_i, α is of period $P=\mathrm{lcm}\{P_i\}$ (the least common multiple of $\{P_i\}$) and order $N=\max_{1\le i\le M}\{N_i-P_i\}+\mathrm{lcm}\{P_i\}$.*

PROOF. Since each component expands without any interactions with the other, uniform boundedness of all of them implies uniform boundedness of the whole graph also. Since the period is well defined for all β^i, it is well defined for α also. Let N be the order and P the period of α. This means that $\alpha_N \le_r \alpha_{N-P}$, and N is the minimum such number. Moreover, this is the case for each β^i, that is $(\beta^i)_N \le_r (\beta^i)_{N-P}$. This, Lemma 2, and the transitivity of $\le_r$ imply that for all $1\le i\le M$ there exist integers c_i, c_i', and r_i, with $0\le c_i<c_i'$ and $0\le r_i\le P_i-1$, such that

$$(N_i-P_i)+c_iP_i+r_i=N-P, \tag{2}$$

$$(N_i-P_i)+c_i'P_i+r_i=N. \tag{3}$$

Subtracting (2) from (3) yields $(c_i'-c_i)P_i=P$, which implies that P_i divides P, for all $1\le i\le M$. Hence, $P=\mathrm{lcm}\{P_i\}$. Since $r_i\ge 0$ and $c_i\ge 0$, (2) yields $N-P\ge N_i-P_i$, for all $1\le i\le M$. Hence, $N-P=\max_{1\le i\le M}\{N_i-P_i\}$ or equivalently $N=\max_{1\le i\le M}\{N_i-P_i\}+\mathrm{lcm}\{P_i\}$. Notice that this implies that the index of α is the maximum of the indices of the β^i's. □

For the proof of Theorem 1, a regular naming scheme for variables is established. Consider a simple recursive Horn clause α. Restrictions R4 and R5 denote that the dynamic α-graph is a forest; therefore, every connected component in the graph is a tree. For every node in such a tree there is a unique path from it to the root of the tree. Each variable is subscripted by the weight of this path, which is nonpositive. Hence, for a variable x_j, it is:

$$x_j = \begin{cases} \text{nondistinguished variable} & \text{if } j = 0, \\ \text{distinguished variable} & \text{if } j < 0. \end{cases} \tag{4}$$

Variables that belong to the same root-to-leaf path in the dynamic α-graph are denoted by the same symbol (with different subscripts).

EXAMPLE 4. Figure 2 illustrates the established notation for the variables of some simple recursive Horn clause α. The Horn clause corresponding to the figure is

$$P(x_o, x_{-1}, x_{-2}, y_0) \wedge Q(x_{-1}, y_o) \wedge R(y_{-1}, x_{-3}) \rightarrow P(x_{-1}, x_{-2}, x_{-3}, y_{-1}).$$

In the first expansion of some Horn clause α, each distinguished variable is replaced by the corresponding variable appearing under the recursive relation in the antecedent. Due to the variable naming convention, this means that a distinguished variable x_{-k} is replaced by x_{-k+1}. In addition, some new variables are introduced to replace the nondistinguished variables. The convention is that each new variable has the name of the one it replaces with the subscript increased by 1. According to the above it may be inductively shown that in the nth expansion x_i is replaced by x_{i+n}. So, the variable substitution for the expansions of α given in Section 3 is

$$s_n(x_i) = x_{i+n} \tag{5}$$

meaning that x_{i+n} is substituted for x_i.

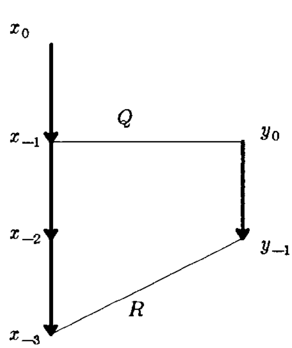

Fig. 2. Example of variable naming.

LEMMA 5. *Consider two variables x_k, x_l, with $k > l$. There is a path of weight $c > 0$ from x_k to x_l in the dynamic α_n-graph, $n \geq 0$, iff the following hold*:

(a) $k - l = c(n+1)$ *the distance of the two variables in the dynamic α-graph is a multiple of* $(n+1)$.
(b) $k, l \leq n$ *both variables do appear in α_n.*

PROOF. (*If*) Let x_k, x_l be two variables satisfying (a) and (b). The conditions imply that

$$k - l = c(n+1) \Rightarrow l = k - c(n+1) \leq n - c(n+1) = (1-c)n - c < 0.$$

Hence, (4) implies that x_l is a distinguished variable. The lemma is shown by induction on c.

Basis: For $c = 1$, it is $k - l = n + 1$. Since x_l is distinguished, there is an edge from x_{l+1} to x_l in the α-graph. The substitution in (5) implies that in the α_n-graph there is an edge from $x_{l+1+n} = x_k$ to x_l. Hence the two are connected with a path of weight 1.

Induction step: Assume that the above is true for all integers less than c. Take variable x_j, with $j - l = (n+1)$. From the induction hypothesis there is an edge from x_j to x_l in the α_n-graph. Also, since $k - j = k - l - (n+1) = (c-1)(n+1)$, the induction hypothesis implies that there is a path of weight $(c-1)$ from x_k to x_j. Hence there is a path of weight c from x_k to x_l.

(*Only if*) Let x_k, x_l be two variables that are connected through a path of weight c in the α_n-graph. Substitution (5) implies that the first expansion variable x_k (x_l) appears is $\alpha_{\max\{k, 0\}}$ ($\alpha_{\max\{l, 0\}}$). Hence, it is $k, l \leq n$. Furthermore, (5) implies that the path of weight 1 from x_k leads to $x_{k-(n+1)}$. An easy induction shows that the path of weight c from x_k leads to $x_{k-c(n+1)}$. The last variable is equal to x_l. Hence it is $k - c(n+1) = l \Leftrightarrow k - l = c(n+1)$. □

Consider a path in the α_n-graph of some expansion of α. In general, this path corresponds to a *walk* in the α-graph (when traversing a walk, nodes and edges may be visited multiple times [Bond]). Static edges met in a traversal of the path in the α_n-graph are met in the same order in a traversal of the walk in the α-graph. Likewise, a cycle in the α_n-graph corresponds to a cyclic walk in the α-graph (its end nodes coincide). The following lemma relates the path in the α_n-graph and the corresponding walk in the α-graph.

LEMMA 6. *Let x_k, y_l be two variables connected through a path of weight L_n in the α_n-graph. Let $x_{k'}$, $y_{l'}$ be the end variables of the corresponding walk in the α-graph, of weight L. Then the following holds*:

(6) $$(k - k') = (l - l') + (n+1)L_n - L.$$

PROOF. Let $x_k = x_{1,k_1}$, $x_{k'} = x_{1,k'_1}$, $y_l = x_{m,l_m}$, and $y_{l'} = x_{m,l'_m}$. Figure 3 shows the path from x_k to y_l and the walk from $x_{k'}$ to $y_{l'}$ in the α_n- and the α-graph, respectively. By Lemma 5, each section in the dynamic α_n-graph is of length $(k_i - l_i)/(n+1)$. Hence it is

(7) $$L_n = \frac{1}{n+1} \sum_{i=1}^{m} (k_i - l_i).$$

Likewise for the walk in the α-graph it is

(8) $$L = \sum_{i=1}^{m} (k'_i - l'_i).$$

Consider $Q_i(x_{i,l_i}, x_{i+1,k_{i+1}})$, $1 \le i \le m-1$, in the α_n-graph. Let n_i be the expansion that this was formed from $Q_i(x_{i,l'_i}, x_{i+1,k'_{i+1}})$ in α. Substitution (5) implies that

(9) $$\left.\begin{aligned} l_i &= l'_i + n_i \\ k_{i+1} &= k'_{i+1} + n_i \end{aligned}\right\} \Rightarrow l_i - l'_i = k_{i+1} - k'_{i+1}.$$

Adding (9) for all $1 \le i \le m-1$ gives $\sum_{i=1}^{m-1} (l_i - l'_i) = \sum_{i=2}^{m} (k_i - k'_i)$. By (7) and (8), since it is $k = k_1$, $k' = k'_1$, $l = l_m$, and $l' = l'_m$, the above gets transformed into

$$(k - k') = (l - l') + (n+1)L_n - L. \qquad \square$$

COROLLARY 1. *If there is a cycle of weight L_n in the α_n-graph, then there is a cyclic walk of weight $L = (n+1)L_n$ in the α-graph.*

PROOF. Assume that in Figure 3 there is one more static edge Q_m between x_{m,l_m} and x_{1,k_1} (see Figure 4). The corresponding edge in the α-graph is between x_{m,l'_m} and x_{1,k'_1}, which creates a cycle in the α-graph as well. Like in (9), substitution (5) implies that $l_m - l'_m = k_1 - k'_1$. Applying the above on (6) yields $L = (n+1)L_n$. $\square$

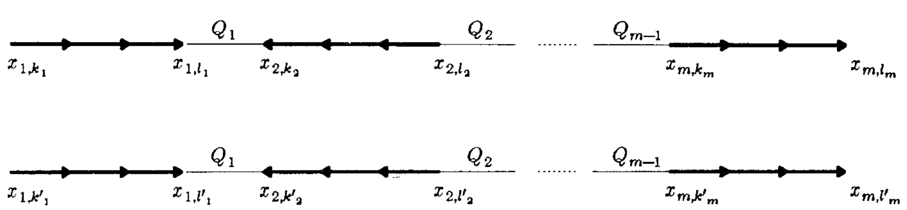

Fig. 3. Path in the α_n-graph and corresponding walk in the α-graph.

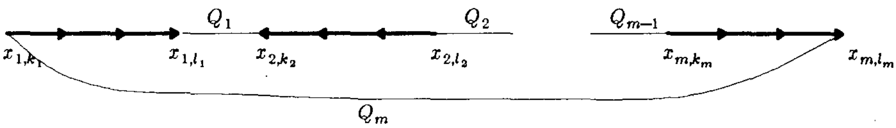

Fig. 4. Cycle in the α_n-graph.

5.1. Sufficiency of the Condition

LEMMA 7. *Let α be a simple recursive Horn clause such that the α-graph is connected and has no nonzero weight cycles. If N, $N \geq 1$, is the maximum path-weight in the α-graph, then $\alpha_N \leq_r \alpha_{N-1}$.*

PROOF. Consider α_{N-1}. Let $Q(x_m, x'_{m'})$ appear in its antecedent. Assume that α_n is the first expansion in which it appeared, corresponding to $Q(x_{-k}, x'_{-k'})$, $k, k' \geq 0$, in α. In this case, (5) implies that $n = k + m = k' + m'$. Consider a node z_0 in the α-graph whose distance from some other node in the graph is N, i.e., there is a path in the graph starting at that node whose weight is the maximum possible (see Figure 5).

Let r be the distance of z_0 from x_{-k}. Consider the following mapping h: $\alpha_{N-1} \rightarrow \alpha_N$:

$$h(x_m) = \begin{cases} x_{m+1} & \forall m, r+1 \leq k+m \leq N-1, \\ x_m & \text{otherwise.} \end{cases}$$

If x_m does not map to itself then $m \geq 0$. Otherwise, it is

$$m < 0 \Rightarrow k + m < k \Rightarrow r < k \Rightarrow r_1 + r_2 < k \Rightarrow r_1 + r_3 < k - r_2 + r_3 \Rightarrow N < k - r_2 + r_3,$$

i.e., a path of length greater than N exists (see Figure 5). Hence, looking at (4) also, we can see that h is meaningful, i.e., it affects only nondistinguished variables. Since $k + m = k' + m'$, h affects x_m and $x'_{m'}$ the same way. Distinguished variables map to themselves. For every atomic formula $Q(x_m, x'_{m'})$ in α_{N-1} there is another

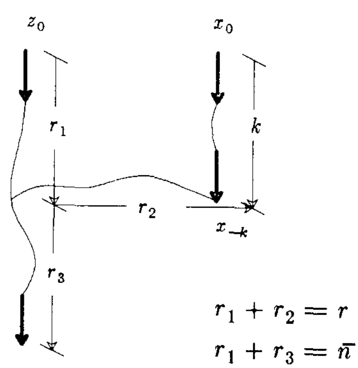

Fig. 5. Distance between variables in the α-graph.

one $Q(x_{m+1}, x'_{m'+1})$ in α_N, which appeared in α_{n+1} for the first time (see substitution (5)). The same is true for the recursive relation also. Hence, h is indeed a homomorphism from α_{N-1} into α_N. Lemma 1 implies that $\alpha_N \leq_r \alpha_{N-1}$. □

THEOREM 2. *Let α be a simple recursive Horn clause. If the α-graph contains no cycle of nonzero weight then α is uniformly bounded.*

PROOF. By Lemmas 3 and 7, each component of the α-graph is uniformly bounded. Lemma 4 implies that α is uniformly bounded as well. □

5.2. Order of Uniformly Bounded Simple Horn Clauses

LEMMA 8. *Let α be a simple recursive Horn clause. Consider α_s and α_t, $0 \leq s < t$, with $\alpha_t \leq_r \alpha_s$. Then the maximum path-weight in the dynamic α_s-graph is* 1.

PROOF. Lemma 1 implies that there exists a homomorphism h: $\alpha_s \to \alpha_t$. This induces a homomorphism h for the corresponding graphs also. Consider a single component G of the α_s-graph. Partition the nodes of G into two subgraphs H and H' as follows:

$$H = \{x \in G: h(x) \neq x\},$$
$$H' = \{x \in G: h(x) = x\}.$$

Distinguished variables map to themselves, so all heads of dynamic edges are in H'. Furthermore, restrictions R3–R5 imply that no variable appears in the same argument position of the same relation in two different expansions (see s_n in (5)). Since the recursive relation appears exactly once in the antecedent of each expansion, no variable appearing under it in α_s can map to itself. Hence, all tails of dynamic edges are in H. The two points above imply that H is connected to H' as shown in Figure 6. Clearly, the maximum path-weight in the dynamic α_s-graph is 1. □

For the following analysis it is necessary to define the following family of functions $f_n: \mathbb{Z} \to \{0, 1, \ldots, n-1\}$, $\mathbb{Z}$ the set of integers, defined for all $n > 0$ as follows:

$$f_n(x) = (x \bmod n).$$

In the above, $(x \bmod n) = r$ if and only if n divides $(x - r)$ and $0 \leq r < n$.

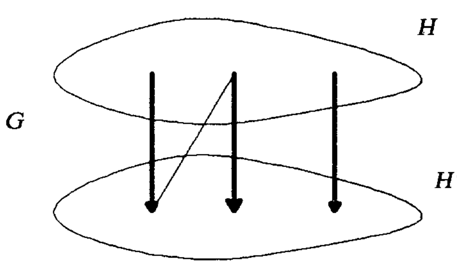

Fig. 6. General form of the α_s-graph with $\alpha_t \leq_r \alpha_s$, for $t > s$.

LEMMA 9. *For all integers x, y and all positive integers n it is $(f_n(x)+y)-f_n(x+y)=cn$ for some integer c.*

PROOF. Obvious from the definition of f_n [Ioan]. □

LEMMA 10. *Let α be a simple recursive Horn clause. If the α-graph contains a path of weight L with ε static edges, then the α_n-graph contains a path of weight $\lfloor (n+L)/(n+1) \rfloor^3$ with ε static edges.*

PROOF. Consider a path in the α-graph of weight L and $\varepsilon = m-1$ (see Figure 7). From Figure 7 the weight of the path from x_{1,k_1} to x_{m,l_m} is equal to

$$L=\sum_{i=1}^{m}(k_i-l_i). \tag{10}$$

From the way the expansions are formed, α_n contains $n+1$ instances of Q_i, $1 \le i \le m-1$, each one created at a different expansion. The one created at expansion r_i, $0 \le r_i \le n$, is of the form $Q_i(x_{i,r_i+l_i}, x_{i+1,r_i+k_{i+1}})$. We claim that the combination of appropriately chosen instances of the Q_i's in α_n creates a path in the α_n-graph. Let σ_i be

$$\sigma_i=\sum_{j=1}^{i}(k_j-l_j) \qquad \text{for} \quad 0 \le i < m.$$

In Figure 7 σ_i denotes the distance of the variables appearing under Q_i from x_{1,k_1}. The Q_i's are chosen so that $r_i = f_{n+1}(n+\sigma_i)$. For every Q_{i-1}, Q_i chosen as above the two variables $x_{i,r_{i-1}+k_i}$ and x_{i,r_i+l_i} are connected with a path in the dynamic α_n-graph. This is shown as follows. Lemma 9 implies that

$$(f_{n+1}(n+\sigma_{i-1})+k_i)-(f_{n+1}(n+\sigma_i)+l_i)=c(n+1) \tag{11}$$

for some integer c. Furthermore, from the definition of f_{n+1} and k_i, $l_i \le 0$ it is

$$f_{n+1}(n+\sigma_{i-1})+k_i<n+1 \quad \text{and} \quad f_{n+1}(n+\sigma_i)+l_i<n+1.$$

Hence, the conditions of Lemma 5 are satisfied. Therefore, the given variables are connected in the dynamic α_n-graph by a path of weight c (as given in (11)). Since this is true for all $1 \le i \le m$, the Q_i's chosen as above form a path in the α_n-graph with $\varepsilon = m-1$ static edges.

Let L_n be the weight of the path. Using (11) for the weight of each individual subpath, L_n becomes equal to

$$L_n=\frac{1}{n+1}\sum_{i=1}^{m}[f_{n+1}(n+\sigma_{i-1})+k_i-f_{n+1}(n+\sigma_i)-l_i] \Leftrightarrow$$

$$L_n=\frac{1}{n+1}\left[f_{n+1}(n+\sigma_0)-f_{n+1}(n+\sigma_m)+\sum_{i=1}^{m}(k_i-l_i)\right].$$

[3] By $\lfloor x \rfloor$ we denote the smallest integer greater than or equal to x.

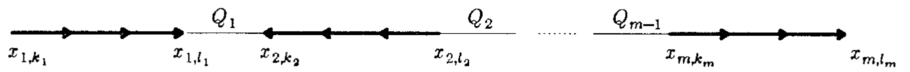

Fig. 7. Typical path in the graph of some simple recursive Horn clause.

From (10), which gives the weight L of the original path, and the fact that $\sigma_0 = 0$ we get

$$L_n = \frac{1}{n+1}[n+L-f_{n+1}(n+L)] \Leftrightarrow L_n = \left\lfloor \frac{n+L}{n+1} \right\rfloor.$$

The last equivalence is an obvious implication of the definition of f_{n+1}. □

THEOREM 3. *Let α be a simple recursive Horn clause such that the α-graph contains no nonzero weight cycles. The order of α is equal to the maximum path-weight in the α-graph.*

PROOF. Suppose that the maximum path-weight in the α-graph is N. Theorem 2 shows that α is uniformly bounded. Assume that the α-graph is connected. By Lemma 7, N is an upper bound on the order of α. This upper bound is tight, i.e., N is indeed the order of α.

Assume to the contrary that for some s, t, with $s < t < N$, it is $\alpha_t \leq_r \alpha_s$. Lemma 1 implies that there exists a homomorphism $h: \alpha_s \to \alpha_t$. By Lemma 8, the α_s-graph is of the form of Figure 8. The variables that h maps to themselves are the ones in H'. Since $s < N-1$, Lemma 10 guarantees the existence of some path of weight greater than 1, in the α_s-graph. Hence, there exists a path of weight zero with one end in H and the other, a tail of a dynamic edge, in H'. In Figure 8 x_m to y_{l_1} is such a path. Because of h, there exists a path of weight zero in the α_t-graph also, like the one from x_m to y_{l_2} in Figure 8. Substitution (5) implies that for l_1 and l_2 it is

$$(12) \qquad l_1 = s + l + 1, \qquad l_2 = t + l + 1.$$

By Lemma 6, in the α-graph there exist two walks between the dynamic components where x_m and y_l belong (see Figure 9). Let L_1 and L_2 be the weight of

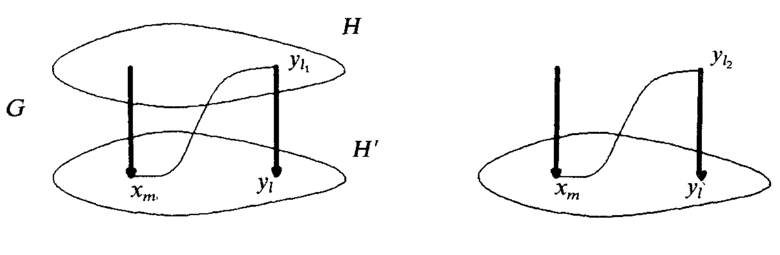

Fig. 8. Expansions α_s and α_t with $\alpha_t \leq_r \alpha_s$ and maximum path-weight in the corresponding graphs greater than 1.

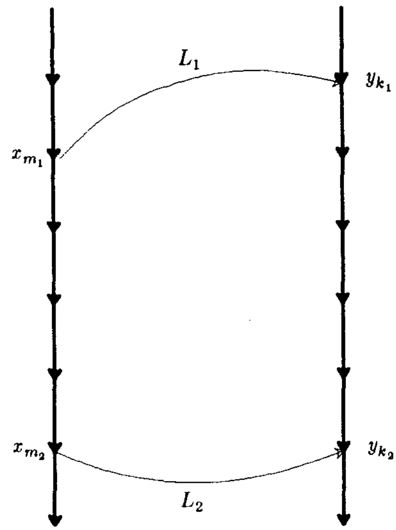

Fig. 9. Cyclic walk in the α-graph.

these walks, respectively. Figure 9 shows that a cyclic walk of weight

$$L=(m_1-m_2)+L_2-(k_1-k_2)-L_1 \tag{13}$$

exists in the α-graph. Furthermore, Lemma 6 implies that the following hold.

$$(m-m_1)=(l_1-k_1)+(s+1)0-L_1,$$

$$(m-m_2)=(l_2-k_2)+(t+1)0-L_2.$$

Subtracting the two above we get $(m_1-m_2)+L_2-(k_1-k_2)-L_1=(l_2-l_1)$. Making the substitution in (13) and using (12) yields $L=(t-s)$. Since $t>s$, the cyclic walk in the α-graph has nonzero weight. This implies that there exists some cycle in the α-graph with nonzero weight also, which contradicts the hypothesis. Thus, $\alpha_t \leq_r \alpha_s$ holds for no s, t, with $s<t<N$. The smallest such numbers are $s=N-1$ and $t=N$. Hence, the order of α is N, the index is $N-1$, and the period is 1.

If the α-graph is not connected the above is true for each one of its components. Applying Lemma 4, gives again that the order of α is equal to the maximum path-weight in the complete α-graph. □

EXAMPLE 5. The proofs of Theorems 2 and 3 are illustrated with the following example. Consider the simple recursive Horn clause α:

$$\alpha:\quad P(u_1, u_2, u_4, u_4, y) \wedge Q(u_1, u_2) \wedge R(u_2, u_3, x) \wedge S(w, z) \wedge T(v)$$
$$\rightarrow P(v, w, x, y, z).$$

The α-graph appears in Figure 10. All the cycles in the α-graph have zero weight.

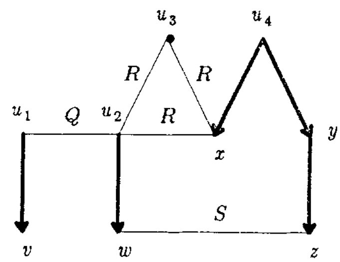

Fig. 10. The α-graph.

Hence, according to Theorem 2, α is uniformly bounded. The maximum path-weight in the graph being 2, it implies that α_2 is redundant, i.e., α is equivalent to α_0 and α_1. This becomes apparent by looking at α_1 and α_2:

$$\begin{aligned}
\alpha_1\colon\quad & P(u_1', u_2', u_4', u_4', u_4) \wedge Q(u_1', u_2') \wedge R(u_2', u_3', u_4) \wedge S(u_2, y) \wedge T(u_1) \\
& \wedge Q(u_1, u_2) \wedge R(u_2, u_3, x) \wedge S(w, z) \\
& \wedge T(v) \rightarrow P(v, w, x, y, z), \\
\alpha_2\colon\quad & P(u_1'', u_2'', u_4'', u_4'', u_4') \wedge Q(u_1'', u_2'') \wedge R(u_2'', u_3'', u_4') \wedge S(u_2', u_4) \wedge T(u_1') \\
& \wedge Q(u_1', u_2') \wedge R(u_2', u_3', u_4) \wedge S(u_2, y) \wedge T(u_1) \\
& \wedge Q(u_1, u_2) \wedge R(u_2, u_3, x) \wedge S(w, z) \\
& \wedge T(v) \rightarrow P(v, w, x, y, z).
\end{aligned}$$

The corresponding graphs are shown in Figures 11 and 12, respectively. The α_1-graph in Figure 11 has two (connected) components, none of which can be the image of the α-graph under any homomorphism. This implies that there are some instances of the relations in α that make α_1 produce some tuples that are not produced by α_0. Hence, α_1 is necessary. On the other hand the α_2-graph in Figure 12 has three components. Two of them are homomorphic images of those in the α_1-graph. Therefore, α_2 is not necessary.

5.3. Necessity of the Condition. The converse of Theorem 2 is proved using the following lemmas.

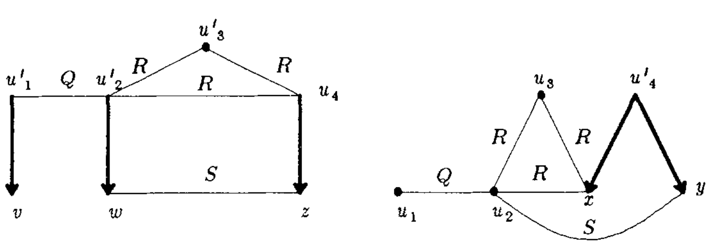

Fig. 11. The α_1-graph.

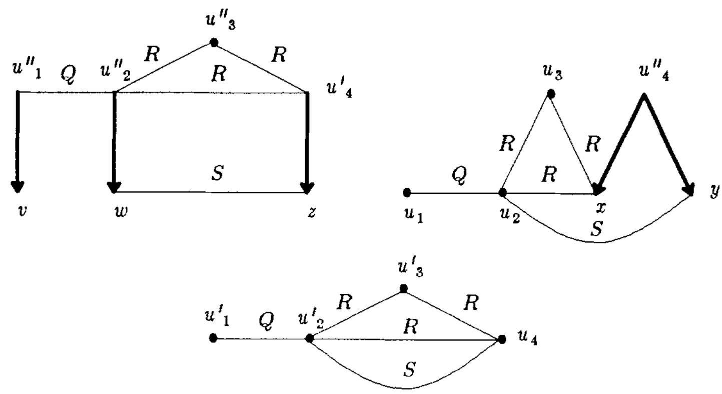

Fig. 12. The α_2-graph.

LEMMA 11. *Let α be a recursive Horn clause. If α is uniformly bounded then, for all $k \geq 0$, α_k is uniformly bounded as well.*

PROOF. In what follows the fact that $(\alpha_k)_l = (\alpha_l)_k = \alpha_{(k+1)(l+1)-1} = \alpha_{k+l(k+1)}$ is used. Lemma 3 implies that $\alpha_N \leq_r \alpha_n$, for some $0 \leq n \leq N-1$. Consider α_k for some $k \geq 0$. It is

$$\begin{aligned}
(\alpha_k)_N = \alpha_{N+k(N+1)} &\leq_r \alpha_{n+k(N+1)} && \text{from Lemma 2} \\
&= \alpha_{n+k(n+1)+k(N-n)} \\
&\leq_r \alpha_{n+k(n+1)} && \text{from Lemma 2 and the transitivity of } \leq_r \\
&= (\alpha_k)_n.
\end{aligned}$$

Lemma 3 implies that α_k is uniformly bounded. □

Consider a typical cycle in the α-graph (see Figure 13). It is essentially the path of Figure 7, with one more static edge labeled Q_m between x_{m,l_m} and x_{1,k_1}. The following two lemmas are similar to Lemma 10.

LEMMA 12. *Let α be a simple recursive Horn clause. If the α-graph contains a cycle of weight $n+1$, $n \geq 0$, with ε static edges, then the α_n-graph contains a cycle of weight 1 with ε static edges also.*

PROOF. Adding the static edge labeled Q_m to the path of Figure 7 creates the cycle in Figure 13, of weight equal to that of the original path with $\varepsilon = m$ static

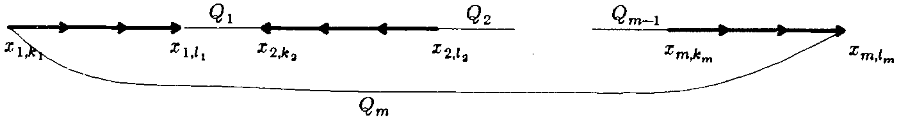

Fig. 13. Typical cycle in the α-graph.

edges. Hence, Lemma 10 is directly applicable. Since the α-graph contains a cycle of weight $n+1$, it implies that the α_n-graph contains a cycle of weight

$$\left\lfloor \frac{n+(n+1)}{n+1} \right\rfloor = \left\lfloor \frac{2n+1}{n+1} \right\rfloor = 1.$$

The number of the static edges in the cycle remains the same, that is $\varepsilon = m$. □

LEMMA 13. *Let α be a simple recursive Horn clause. If the α-graph contains a cycle of weight 1 with ε static edges, then the α_n-graph, $n \geq 0$, contains a cycle of weight 1 with $\varepsilon(n+1)$ static edges.*

PROOF. The proof is similar to that of Lemma 10, so it is only sketched here. The Q_i's are partitioned into $n+1$ partitions, each partition having exactly one instance of each Q_i, $1 \leq i \leq m$. The rth partition, $0 \leq r \leq n$, contains $Q_i(x_{i,r_i+l_i}, x_{i+1,r_i+k_{i+1}})$, such that $r_i = f_{n+1}(r+\sigma_i)$. Each one of these partitions is shown to form a path. Furthermore, the last node of partition r and the first node of partition $r+1$ are connected by a dynamic path, and so are the last node of partition n and the first node of partition 0. Hence a cycle is formed. Its weight is calculated as in Lemma 10 and is equal to 1. The static edges in the cycle are those of all the partitions. Each partition has $\varepsilon = m$ edges, and there are $n+1$ partitions. Hence, the cycle formed has $\varepsilon(n+1)$ static edges. □

THEOREM 4. *Let α be a simple recursive Horn clause. If α is uniformly bounded then the α-graph contains no cycle of nonzero weight.*

PROOF. Suppose to the contrary that the α-graph contains some nonzero weight cycles. Consider the one with the smallest number of static edges, say ε. Let $n+1$, $n \geq 0$, be its weight. By Lemma 12, the α_n-graph contains a cycle of weight 1, with ε static edges also. Lemma 11 implies that α_n is uniformly bounded. Hence, there are two expansions $(\alpha_n)_s$ and $(\alpha_n)_t$, $s < t$, such that $(\alpha_n)_t \leq_r (\alpha_n)_s$. By Lemma 1, there exists a homomorphism $h:(\alpha_n)_s \to (\alpha_n)_t$. Lemma 13 implies that the $(\alpha_n)_s$-graph contains a cycle of weight 1 with $(s+1)\varepsilon$ static edges, which h maps to another cycle of weight 1 in the $(\alpha_n)_t$-graph (see Figure 14). The cycle contains

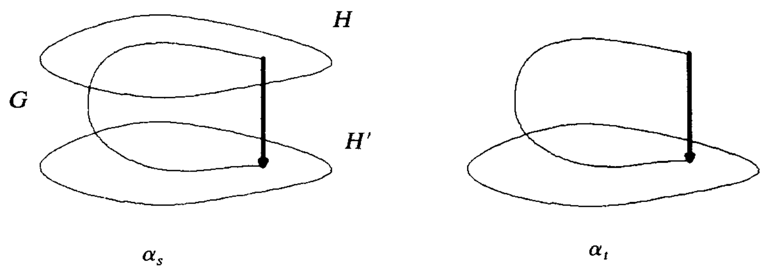

Fig. 14. Expansions α_s and α_t with $\alpha_t \leq_r \alpha_s$ and cycles in the corresponding graphs of weight 1.

at most as many static edges as the cycle of the $(\alpha_n)_s$-graph. It may contain fewer static edges if h is not one-to-one but it can never contain more. By Corollary 1, there exists a cyclic walk in the α-graph of weight $(n+1)(t+1)-1+1=(n+1)(t+1)$. Without loss of generality, assume that it is formed by traversing c times a cycle of weight n' with ε' static edges (the case that it is formed by traversing multiple cycles connected with each other is a trivial extension of what follows). Hence, it is

$$cn' = (n+1)(t+1). \tag{14}$$

There are $c\varepsilon'$ static edges in the cyclic walk, and as mentioned above it is

$$c\varepsilon' \leq (s+1)\varepsilon. \tag{15}$$

Combining (14) and (15) yields

$$\frac{\varepsilon}{\varepsilon'} \geq \frac{(n+1)}{n'} \frac{(t+1)}{(s+1)}. \tag{16}$$

However, from Lemma 2 and the transitivity of $\leq_r$, t may be chosen arbitrarily large. In (16) s and n are fixed, whereas there is an upper bound on the value of n', imposed by the form of the α-graph. Hence, there exists some t satisfying the desired properties, such that $[(n+1)/n'][(t+1)/(s+1)]>1$. This combined with (16) yields $\varepsilon > \varepsilon'$, that is there exists a cycle in the α-graph, with fewer static edges than ε. This contradicts the hypothesis. Hence, all the cycles in the α-graph are of weight zero. □

EXAMPLE 6. Theorem 4 is illustrated with an example. Consider the Horn clause below:

$$\beta: \quad P(u_1, w, u_2, u_3) \wedge Q(w, u_2) \wedge R(y, u_3) \wedge S(x, z) \rightarrow P(w, x, y, z).$$

The β-graph appears in Figure 15. The β-graph contains a cycle of weight 1,

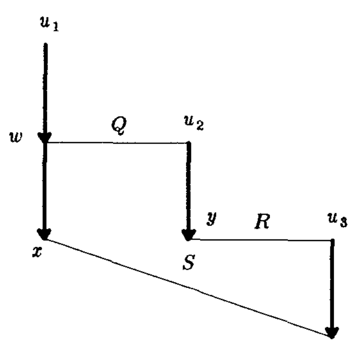

Fig. 15. The β-graph.

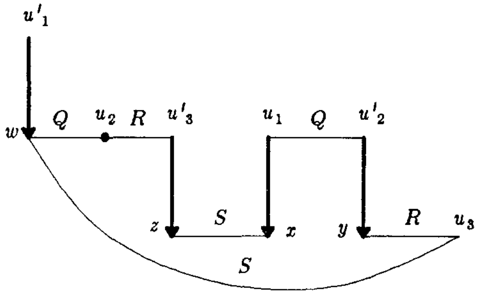

Fig. 16. The β_1-graph.

namely $(w \to u_2 \to y \to u_3 \to z \to x \to w)$. According to Theorem 4, β is not uniformly bounded. This becomes apparent by looking at the graphs of the expansions of β. The β_1- and β_2-graphs appear in Figures 16 and 17, respectively. Contrary to what happened to the graphs of uniformly bounded Horn clauses, the graphs of the expansions of β continue to have a single component, but the number of static edges in the original cycle increases (Lemma 13). This continues, no matter how many expansions are taken.

THEOREM 1. *A simple recursive Horn clause α is uniformly bounded iff the α-graph contains no cycle of nonzero weight. In that case the order of α is equal to the maximum path-weight in the α-graph.*

PROOF. The proof follows immediately from Theorems 2, 3, and 4. □

Testing the condition of Theorem 1 requires time linear in the number of edges and nodes of the graph. A depth-first search in the graph is sufficient [Ioan].

6. Transitive Closure. Unfortunately some very useful recursive Horn clauses are not simple. A characteristic example is the transitive closure P of a binary relation Q expressed by the Horn clause

$$P(x, z) \wedge Q(z, y) \to P(x, y).$$

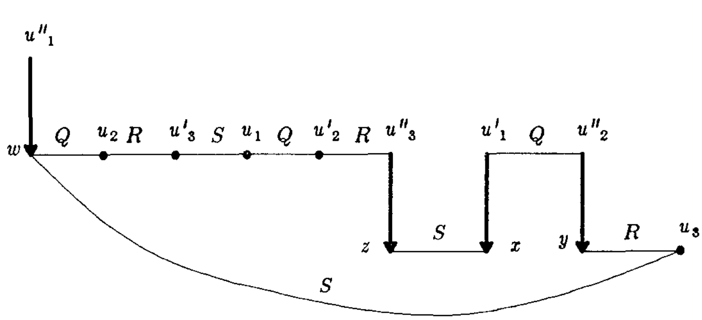

Fig. 17. The β_2-graph.

This is clearly unbounded and one would expect to be able to characterize uniform boundedness for Horn clauses of this form. To achieve that we relax restriction R5. We define a *permutation* Horn clause to be one whose corresponding graph *is* a dynamic cycle. The dynamic α-graph of a simple recursive Horn clause is a forest. Restriction R5 is relaxed by allowing components of the dynamic graph to *be* cycles, as long as there are no static edges attached to them in the complete graph. That is, each component of the α-graph is either simple or permutation.

EXAMPLE 7. Let α be the above given Horn clause representing the transitive closure of Q.

$$P(x, z) \wedge Q(z, y) \rightarrow P(x, y).$$

The α-graph is shown in Figure 18. Clearly, α belongs in the new extended class of Horn clauses, since one component in the α-graph is simple and the other is permutation.

The following theorem characterizes uniform boundedness for the new extended class of recursive Horn clauses.

THEOREM 5. *Let α be a recursive Horn clause in the extended class. It is uniformly bounded iff the simple components of the α-graph contain no cycle of nonzero weight. In that case, α has order $N = N_s + \text{lcm}\{P_i\}$ and period $P = \text{lcm}\{P_i\}$, where N_s is the maximum path-weight in the simple components of the α-graph, and $\{P_i\}$ is the set of weights of the cycles in the permutation components.*

PROOF. Consider a recursive Horn clause α in the extended class. Assume that the α-graph contains M permutation components, with P_i dynamic edges in the ith one, $1 \leq i \leq M$. Each such component is clearly uniformly bounded, since it simply permutes distinguished variables. Apparently, the period is well defined for permutation components, and for the ith component it is P_i. Its order is $P_i - 1$. Notice that the index is equal to $(P_i - 1) - P_i = -1$. This actually represents the identity Horn clause, with antecedent equal to the consequent, and may be denoted as α_{-1} for consistency. Furthermore, by Theorem 4, each simple component is uniformly bounded iff it contains no nonzero weight cycles. In that case it has been shown that the order of the component is the maximum path-weight in the graph of the component and the period is 1. Applying Lemma 4 yields that α is uniformly bounded if the simple components of the α-graph

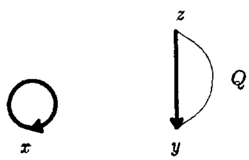

Fig. 18. Transitive closure graph.

contain no nonzero weight cycles. In that case, if the maximum path-weight in the simple components is N_s, then Lemma 4 implies that the period P of α is equal to $P = \mathrm{lcm}\{P_i\}$ and the order N is $N = N_s + \mathrm{lcm}\{P_i\}$. □

7. Conclusions. We have considered a restricted class of recursive Horn clauses in the context of a deductive database. We have demonstrated that some such Horn clauses are equivalent to a finite set of nonrecursive ones. By modeling such a Horn clause with a weighted graph, we have shown that the uniform boundedness property of Horn clauses is equivalent to the property that the graph has no cycles of nonzero weight.

Recently, it has been shown that both boundedness and uniform boundedness are undecidable in the presence of multiple Horn clauses [Gaif]. However, the question is open for the case of a single recursive Horn clause. We are currently working in this direction, trying to characterize uniform boundedness for more general classes of recursive Horn clauses, by removing some of the restrictions R1–R5 of Section 2.

Acknowledgments. I would like to give thanks to Professor Eugene Wong for all his valuable help and guidance. I am also deeply indebted to Timos Sellis for the innumerable discussions I had with him on the subject. Finally, many thanks go to the anonymous referees, whose comments improved the quality of the text immensely.

References

[Aho] A. Aho, Y. Sagiv, and J. Ullman, Equivalences among relational expressions. *SIAM J. Comput.*, **8** (1979), 218–246.

[Bond] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications.* North Holland: Amsterdam, 1976.

[Chan 1] A. K. Chandra and P. M. Merlin, Optimal implementation of conjunctive queries in relational databases. *Proceedings of the Ninth Annual ACM Symposium on Theory of Computing*, Boulder, CO, 1976, pp. 77–90.

[Chan 2] C. L. Chang, On evaluation of queries containing derived relations in a relational data base, *Advances in Data Base Theory*, vol. 1 (H. Gallaire, J. Minker, and J. M. Nicolas, eds.), Plenum Press, New York, 1981, pp. 235–260.

[Clif] A. H. Clifford and G. B. Preston, *The Algebraic Theory of Semigroups.* American Mathematical Society: Providence, RI, 1961.

[Codd] E. F. Codd, A relational model of data for large shared data banks. *Comm. ACM*, **13** (1970), 377–387.

[Cosm] S. Cosmadakis and P. Kanellakis, Parallel evaluation of recursive rule queries. *Proceedings of the Fifth ACM SIGMOD-SIGACT Symposium on Principles of Database Systems*, Boston, MA, 1986, pp. 280–293.

[Ende] H. B. Enderton, *A Mathematical Introduction to Logic.* Academic Press: New York, 1972.

[Gaif] H. Gaifman, *NAIL communication*, February 1986.

[Gall] H. Gallaire, J. Minker, and J. M. Nicolas, Logic and databases: a deductive approach. *ACM Comput. Surveys*, **16** (1984), 153–185.

[Hens] L. Henschen and S. Naqvi, On compiling queries in recursive first-order databases. *J. Assoc. Comput. Mach.*, **31** (1984), 47–85.

[Ioan] Y. E. Ioannidis, Bounded recursion in deductive databases Memorandum No. UCB/ERL M85/6, University of California, Berkeley, 1985.

[Lewi] H. Lewis and C. Papadimitriou, *Elements of the Theory of Computation.* Prentice-Hall: Englewood Cliffs, NJ, 1981.

[Mink] J. Minker and J. M. Nicolas, On recursive axioms in deductive databases. *Inform. Systems,* **8** (1983), 1-13.

[Naug] J. Naughton, Data independent recursion in deductive databases. *Proceedings of the Fifth ACM SIGMOD-SIGACT Symposium on Principles of Database Systems,* Boston, MA, 1986, pp. 267-279.

[Reit] R. Reiter, Deductive question-answering on relational databases. *Logic and Data Bases* (H. Gallaire and J. Minker, eds.), Plenum Press: New York, 1978, pp. 149-177.

[Sagi 1] Y. Sagiv and M. Yannakakis, Equivalences among relational expressions with the union and difference operators. *J. Assoc. Comput. Mach.,* **27** (1980), 633-655.

[Sagi 2] Y. Sagiv, On computing restricted projections of representative instances. *Proceedings of the Fourth ACM SIGMOD-SIGACT Symposium on Principles of Database Systems,* Boston, MA, 1985, pp. 171-180.

[Ston] M. Stonebraker, E. Wong, P. Kreps, and G. Held, The design and implementation of INGRES. *ACM Trans. Database Systems,* **1** (1976), 189-22.

[Viei] L. Vieille, Recursive axioms in deductive databases: the query/subquery approach. *Proceedings of the First International Conference on Expert Database Systems,* Charleston, SC, 1986, pp. 179-193.