Expressing Linear Recursions as Graph Traversals

David A. Briggs

Technical Report 87-39

April 29, 1987

Expressing Linear Recursions as Graph Traversals

David A. Briggs

May 7, 1987

Abstract

The connection between relations implicitly defined by recursive rules and results of computations defined via graph traversals has received attention from a number of researchers. In this paper we demonstrate that any recursive relation defined by a single linear recursive rule can be expressed as the result of a graph traversal. Under certain conditions, the computation performed during the traversal is simply reachability, implying that the transitive closure of the graph encodes information sufficient for materializing the recursive relation without any operators beyond those provided by the relational algebra.

The expression and evaluation of queries against a relational data base that require some form of a fixed point operator, so-called recursive queries, have received a great deal of attention recently. Researchers have proposed a variety of language constructs for such queries and have offered a variety of algorithms to implement their proposals. Here we confine our attention to a specific class of recursive queries, linear recursive queries of the Datalog language with a single recursive rule, and show that the extension of the recursive relation can be realized essentially by a graph traversal over a graph definable solely in terms of the extensional data base. We believe that casting the computation in this form is useful for several reasons. First, graph traversal has been thoroughly studied, and results from previous investigations become immediately available for this context. Second, properties of graphs that facilitate the computation of a traversal by permitting a

simpler or faster algorithm translate into integrity constraints on the graph derived from the data base which, if present, permit the corresponding simplification in the query evaluation as a graph traversal. Identifying the graph that is implicitly traversed in a recursion determines access paths which support its evaluation, and thus serves in the design of the physical data base. In general, representing the recursion as a graph traversal offers a new vantage for the design and analysis of algorithms to support it.

The organization of the paper is as follows. In the first section we delimit the class of queries we address and describe the general strategy of the construction. In the second we detail the specifics of the construction and argue that the traversal over the derived graph does indeed preserve the semantics of the recursion. In the final section we discuss methods to handle recursions that violate some of the simplifying assumptions that we make, and relate the work described here to other research in the field.

1 Linear Recursion, Rule Expansions, and Blocked Decompositions

We confine our attention to a class of recursive relations expressible in Datalog, a language like Prolog, except that function complexes are not allowed as arguments of predicates. We impose the reasonable constraint that all rules are *range restricted*, that is, any variable occurring in the consequent of a rule must occur in the antecedent. We will consider only recursions defined by a single *linear* recursive rule, that is, only one of the antecedent literals is recursive with the consequent literal, and for simplicitly we will assume that the recursive antecedent is another instance of the predicate of the consequent. We assume that there is additionally a single "exit" rule that equates the recursive relation with some base relation. For convenience we will initially assume that the antecedent of the rule consists of the recursive literal and a single non-recursive base relation literal and that the arguments of both recursive literals are all variables. We will discuss relaxing these conditions later. We make no assumptions about repetition of variables in the recursive literals, nor about the presence or absence of variables in the non-recursive antecedent, other than what is implied by the range restricted character of the rule.

Such a recursion is expressible by a pair of rules, the recursive rule, and the exit rule, in general,

$$\begin{aligned} P(X^0_{j_1},\ldots,X^0_{j_n}) &\Leftarrow P(X^0_{i_1},\ldots,X^0_{i_n}) \wedge Q(X^0_1,\ldots,X^0_t) \\ P(X^0_1,\ldots,X^0_n) &\Leftarrow P_0(X^0_1,\ldots,X^0_n) \end{aligned}$$

where P is the recursive relation and P_0 and Q are base relations, and all the variables are drawn from some finite set $V = \{X^0_1, X^0_2, \ldots\}$. We will use R to denote the recursive rule, P_A for the recursive antecedent literal, Q for the antecedent base literal, and P_C for the recursive consequent. For a concrete example consider

$$\begin{aligned} P(X^0_4, X^0_4, X^0_3, X^0_2, X^0_1, X^0_2) \Leftarrow\ & P(X^0_6, X^0_5, X^0_4, X^0_4, X^0_2, X^0_5) \wedge \\ & Q(X^0_1, X^0_2, X^0_3) \end{aligned}$$

We define the mth rule expansion, denoted R_m, to be the rule derived by resolving $m+1$ separated instances of the recursive rule. In [Brig87b] we defined sequences of variable substitutions $(\gamma_i)_{i\geq 0}$ and $(\sigma_i)_{i\geq 1}$ such that

$$\begin{aligned} R_m = & (\prod_{l'=1}^{m} \sigma_{l'}) P(X^0_{j_1},\ldots,X^0_{j_n}) \Leftarrow \\ & \gamma_m P(X^0_{i_1},\ldots,X^0_{i_n}) \wedge \bigwedge_{l=0}^{m} (\prod_{l'=l+1}^{m} \sigma_{l'}) \gamma_l Q(X^0_1,\ldots,X^0_t) \end{aligned}$$

Clearly, if we replace the letter 'P' in the antecedent of an expansion with 'P_0', we have a non-recursive rule whose evaluation contributes to the extension of P. We will use P^m_A to denote the recursive antecedent of R_m, P^m_C for the consequent, and Q^m_l for the literal $(\prod^m_{l'=l+1} \sigma_{l'})\gamma_l Q(X^0_1,\ldots,X^0_t)$, omitting the superscript m when it can be gathered from the context.

We describe Ioannidis's dynamic α-graph for a recursive rule R, denoted G_R or simply G, as the directed graph (V, E), where $V = \{X^0_1, \ldots\}$, the variables occurring in the rule, and $E = \{(v, w) : \exists k 1 \leq k \leq n \wedge v = X^0_{i_k} \wedge w = X^0_{j_k}\}$. The dynamic α-graph for the example rule is given in Figure 1. Note that within rule expansion R_m, all predicate arguments are of the form $\sigma_m \cdots \sigma_{l+1}\gamma_l(X^0_r)$, with $X^0_r \epsilon V$ and $0 \leq l \leq m$. In [Brig87b] we proved the following.

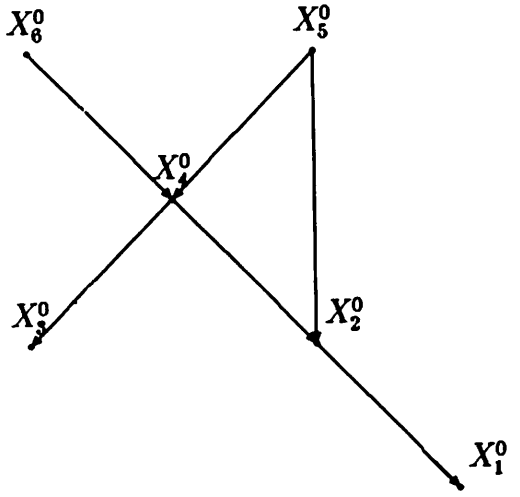

Figure 1: An example α-graph

The n-k-l Theorem. For all n, k, and l in **N**, and all variables X_r^0 and X_s^0 in the rule,

$$\sigma_{n+k+l} \cdots \gamma_n(X_r^0) = \sigma_{n+k+l} \cdots \gamma_{n+k}(X_s^0)$$

if and only if there is a path in the dynamic α-graph from X_r^0 to X_s^0, that may traverse directed arcs in either the forward or reverse direction, such that

1. There are k more traversals of arcs in the forward direction than there are traversals of arcs in the reverse direction.

2. If one keeps track of the number of forward and the number of reverse traversals as one proceeds through the path, then the number of reverses less the number of forwards is never more than n, and never less than $-k-l$.

We may picture the statement of the theorem by imagining a path between two variables plotted on a two-dimensional grid. As we move from node to node along the path we proceed from left to right on the grid. If we

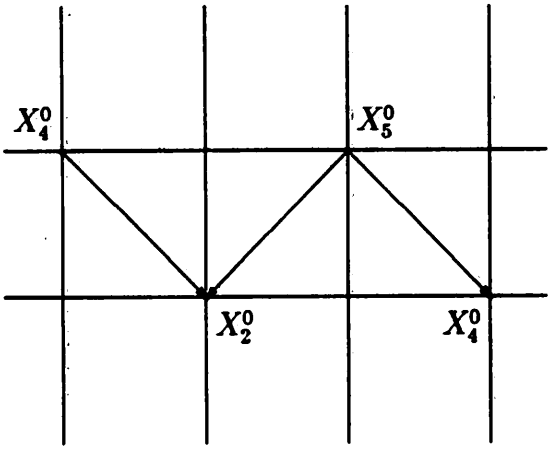

Figure 2: Example path from X_4^0 to X_4^0

traverse a directed arc in the reverse direction, we gain a level in height, while traversing an arc in the forward direction drops a level. The theorem states that the two expressions are equal if and only if there is a path that drops k levels and stays within the bounds imposed by n and l. Figure 2 shows such a plot for a path from X_4^0 to itself from the graph for the example rule. Its characteristics imply, among other things, that $\sigma_1\gamma_0(X_4^0) = \gamma_1(X_4^0)$.

A useful visual aid for comprehending the implications of the n-k-l theorem for variable equivalencing in the m-th rule expansion is to imagine a table of $m+1$ rows, each containing all of the variables occurring in the rule, with the rows indexed from top to bottom by the numbers 0 through m. We make the table a graph by connecting a variable entry in one row to a variable entry in the next by a directed arc if there is a directed arc from the variable of the first to that of the second in the dynamic α-graph. Now, for two variable entries x and y, occurring in rows i_1 and i_2, respectively, $\sigma_m \cdots \gamma_{i_1}(x) = \sigma_m \cdots \gamma_{i_2}(y)$ if and only if the two variable entries are connected in the table-graph by some sequence of arcs, without concern for the direction of the arcs.

The strategy for the construction of a graph whose traversal computes the recursion is to group the literals of R_m into blocks that can be interpreted as functions, as in Figure 3, where

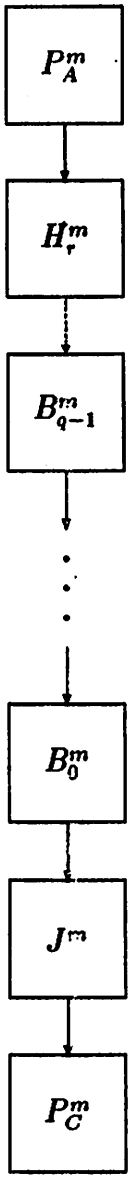

Figure 3: Blocked Decomposition of R_m

$$\begin{aligned} \lfloor \frac{m+1-j-h}{p} \rfloor &= q \geq 1 \\ (m+1-j-h) \bmod p &= r \\ H_r^m &= \bigwedge_{k=m-h-r+1}^{m} Q_k^m \\ B_l^m &= \bigwedge_{k=j+lp}^{j+(l+1)p-1} Q_k^m \\ J^m &= \bigwedge_{k=0}^{j-1} Q_k^m \end{aligned}$$

If the Q_k^m literals are ordered by subscript, the block sequence

$$J^m, B_0^m, \ldots, B_{q-1}^m, H_r^m$$

partitions the sequence into consecutive, contiguous subsequences. The number of literals within the B blocks is the same for each, namely p, which we shall call the *blocking factor*. The arcs connecting blocks indicate the passage of arguments from the block at the tail to the block at the head.

We interpret each block as a function that receives a set of bindings from its predecessor block and provides a set of bindings to its successor block. Specifically, the bindings a block receives are for variables it has in common with its predecessor, and the bindings it passes on are for variables it shares with its successor. For each individual tuple of bindings it receives, it finds all bindings for any other variables appearing in its conjunction of literals that make the conjunction true, and passes out as a result the bindings for variables appearing in its successor. For example, consider the following three block sequence

Block 1 $Q_1(X,Y) \wedge Q_2(X,Z)$

Block 2 $Q_3(X,Z,W,U)$

Block 3 $Q_4(Z,V) \wedge Q_5(U,Z)$

The first block passes to the second the set

$$S = \{(X,Z) : \exists Y Q_1(X,Y) \wedge Q_2(X,Z)\}$$

The second passes to the third the set

$$T = \{(Z, U) : \exists X \exists W (X, Z) \epsilon S \wedge Q_3(X, Z, W, U)\}$$

To be useful for evaluation our decomposition must possess the following properties.

- **Mediation.** All overlap among blocks containing base literals is mediated by intervening blocks, that is, if two blocks C_1 and C_2, with at least one consisting of a conjunction of Q literals, have a variable x in common, then all blocks between C_1 and C_2 contain an occurrence of that variable. The proviso that one of the blocks consists of Q literals excuses P_A and P_C from this constraint, and we shall later show why we cannot force their overlap to be mediated and discuss the problems posed by unmediated overlap between the antecedent and consequent.

- **Iteration.** The B blocks all compute the same function. A sufficient condition for this is that they are all isomorphic, that is, each is a renaming of the other, and that the variable overlap between each and its adjacent blocks occurs at the same argument positions.

- **Stability.** The decomposition should be relevant to subsequent rule expansions. Specifically, if the blocks are regarded as functions the functions should not change in subsequent expansions, and rule R_{m+p} should in effect differ from R_m by the insertion of a single additional B block function.

If we can establish these properties for our decomposition, then the recursive relation can be computed by a procedure that evaluates finitely many of the rule expansions and unions these results with those of another computation involving a traversal of a graph derived from the blocked rule expansion. The nodes of the graph are tuples of bindings for the distinct variables of a B block that satisfy the conjunction. We have a directed arc from v to w if v and w agree at those positions corresponding to overlap between adjacent B blocks. If we substitute 'P_0' for 'P' in P_A^m, we obtain a function that depends only on the contents of the base relations. The

union, over all residues r, of the composition of H_r with this function provides a set of bindings that selects a subset of the nodes of the derived graph, based on the overlap of the H_r block with B_{q-1}. The nodes in the graph reachable from this subset provide bindings to be passed to the J block function, which produces bindings for the consequent. It should be clear that the graph traversal mimics the iterated expansion of the recursive rule, and so the result obtained from it will be the same as that obtained by a more conventional algorithm. There is one hazard which we will discuss later corresponding to unmediated overlap between P_A^m and P_C^m.

In the next section we show that there exist choices for j, h, and p to force the above conditions, for sufficiently large m.

2 Variable Equivalencing in Rule Expansions

We begin with a number of definitions.

Definition. For a rule R and associated graph G, define the *opposite of* G, G^{op}, to be the graph obtained from G by reversing the direction of the arcs.

The opposite of G is the graph that would be associated with the rule R^{op} obtained from R by exchanging the recursive consequent with the recursive antecedent. Many of the concepts and arguments we use below have analogs in the opposite, and we will sometimes claim a result by this duality. In particular, note that by the n-k-l theorem, $\sigma_{n+k+l} \cdots \gamma_n(X_r^0) = \sigma_{n+k+l} \cdots \gamma_{n+k}(X_s^0)$ in an expansion for R if and only if $\sigma_{n+k+l} \cdots \gamma_l(X_s^0) = \sigma_{n+k+l} \cdots \gamma_{l+k}(X_r^0)$ in the expansion for R^{op}.

Variables that are distinct in Q may be equivalenced in an instantiation of Q, and we are particularly interested in which variables become equivalenced and in which instantiations. By the n-k-l theorem, we know that distinct variables will be equivalenced within the same instantiation if there is a path connecting them that places them on the same horizontal grid line, the special case of the theorem with $k = 0$. Our interest in the values for n and l that effect any within literal equivalencing that can ever occur motivates the following definitions.

Definition. For a graph G, define the *attic of* G, N_G, as

$$N_G = \mu n \forall r \forall s \; \gamma_{n+1}(X_r^0) = \gamma_{n+1}(X_s^0) \Longrightarrow \gamma_n(X_r^0) = \gamma_n(X_s^0)$$

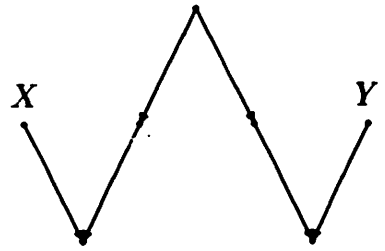

Figure 4: Example requiring space above and below

By the n-k-l theorem, any two variables that can be connected by a path that places them on the same level, and never drops below that level, can be so connected by a path that requires no more than N_G levels. We will omit the subscript when G is understood.

We have the following dual concept.

Definition. For a graph G, define the *cellar of G*, L_G,

$$\begin{aligned} L_G \;=\; & \mu l \forall r \forall s \\ & \sigma_{l+1} \cdots \gamma_0(X_r^0) = \sigma_{l+1} \cdots \gamma_0(X_s^0) \Longrightarrow \\ & \sigma_l \cdots \gamma_0(X_r^0) = \sigma_l \cdots \gamma_0(X_s^0) \end{aligned}$$

Not every pair of variables that can be connected by a path that places them on the same level can be so connected by a path that does not require some space both above and below that level. For example, consider the graph of Figure 4. The variables X and Y are connected by a path that places them on the same level, but it obviously requires space on both sides. To cover such cases, we define the number of additional levels that might prove useful for equivalencing variables.

Definition. For a graph G, define the *relative attic of G*, N'_G,

$$\begin{aligned} N'_G \;=\; & \mu n \forall r \forall s \\ & \sigma_{L_G+n+1}\gamma_{n+1}(X_r^0) = \sigma_{L_G+n+1} \cdots \gamma_{n+1}(X_s^0) \Longrightarrow \\ & \sigma_{L_G+n} \cdots \gamma_n(X_r^0) = \sigma_{L_G+n} \cdots \gamma_n(X_s^0) \end{aligned}$$

The relative attic is the fewest number of additional levels above the line that are useful for equivalencing variables if we have provided L_G levels below the line. Its dual is

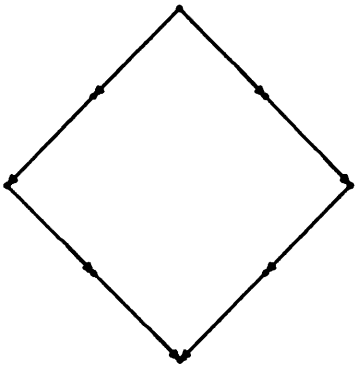

Figure 5: $N = L = 3$ and $N' = L' = 0$

Definition. For a graph G, define the *relative cellar of* G, L'_G, as

$$\begin{aligned} L'_G \;=\; & \mu l \forall r \forall s \\ & \sigma_{N_G+l+1} \cdots \gamma_{N_G}(X^0_r) = \sigma_{N_G+l+1} \cdots \gamma_{N_G}(X^0_s) \Longrightarrow \\ & \sigma_{N_G+l} \cdots \gamma_{N_G}(X^0_r) = \sigma_{N_G+l} \cdots \gamma_{N_G}(X^0_s) \end{aligned}$$

Evidently, $N'_G \leq N_G$ and $L'_G \leq L_G$. As usual, we will omit the subscripts when the context provides the graph. Consider the graph of Figure 5. For this graph, $L = N = 3$ and $L' = N' = 0$.

The following is easily derived from the definitions and the n-k-l theorem.

Lemma 1.

$$\begin{aligned} & \forall r \forall s \forall k \exists n \exists l \\ \sigma_{n+k+l} \cdots \gamma_n(X^0_r) \;&=\; \sigma_{n+k+l} \cdots \gamma_{n+k}(X^0_s) \Longleftrightarrow \\ \sigma_{N'+k+L} \cdots \gamma_{N'}(X^0_r) \;&=\; \sigma_{N'+k+L} \cdots \gamma_{N'+k}(X^0_s) \Longleftrightarrow \\ \sigma_{N+k+L'} \cdots \gamma_N(X^0_r) \;&=\; \sigma_{N+k+L'} \cdots \gamma_{N+k}(X^0_s) \end{aligned}$$

If we let $k = 0$, this lemma tells us that if $l \geq N' \wedge m - l \geq L$ or $l \geq N \wedge m - l \geq L'$, the variables of Q^m_l are "fully equivalenced", by which we mean that no other instantiation will have any fewer distinct variables. The example of Figure 6 shows that the sum $N + L'$ may be different from the

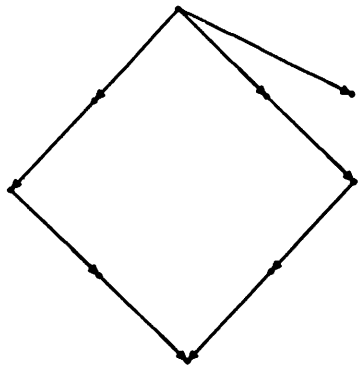

Figure 6: $N = L = 3$ and $L' = 0$, but $N' = 1$

sum $N' + L$. We will later require that the rule expansion index be greater than or equal to both of these sums, so for convenience, let M denote the larger of the two.

The above discussion suggests that for large enough m, there will be a band of Q literal instantiations in the "interior" of the expansion that will be isomorphic. We will need some additional notions to analyze the literals at the extrema and to determine a satisfactory blocking factor.

We define three equivalence relations for the nodes of G. Let v and w be elements of V.

Definition. $v \equiv_N w \iff \gamma_N(v) = \gamma_N(w)$.

Definition. $v \equiv_L w \iff \sigma_L \cdots \gamma_0(v) = \sigma_L \cdots \gamma_0(w)$.

Definition. $v \equiv w \iff \sigma_{N+L} \cdots \gamma_N(v) = \sigma_{N+L} \cdots \gamma_N(w)$.

Clearly, $\equiv_N$ and $\equiv_L$ are both refinements of $\equiv$.

For each relation we define an associated graph obtained from G by merging equivalent nodes and erasing redundant directed arcs, and we denote these graphs, G_L, G_N, and $G_{\equiv}$. We will denote nodes of these graphs with $[v]$, standing for the equivalence class of the node v under the appropriate relation. We will say that $[v]$ occurs in P_A, P_C, or Q to mean that there is some w equivalent to v that occurs in P_A, P_C, or Q.

The following facts are easy to see.

1. The indegree of every node of G_L is less than or equal to 1.

2. The outdegree of every node of G_N is less than or equal to 1.

3. The indegree and outdegree of every node of $G_{\equiv}$ are both less than or equal to 1.

4. $G_L = ((G^{op})_N)^{op}$

5. $G_N = ((G^{op})_L)^{op}$

6. $G_{\equiv} = ((G^{op})_{\equiv})^{op}$

The following lemmas relate the existence of paths in the collapsed graphs to the existence of paths in the original graph. Since in the n-k-l theorem we allow paths that traverse arcs in either the forward or reverse direction, we refer to a "forward" path in the lemmas to mean a path that traverses arcs in only the forward direction.

Lemma 2. $\forall n \forall k \forall l \forall r \forall s$ if $\sigma_{n+k+l} \cdots \gamma_n(X_r^0) = \sigma_{n+k+l} \cdots \gamma_{n+k}(X_s^0)$ then there is a k-arc forward path from $[X_r^0]$ to $[X_s^0]$ in $G_{\equiv}$.

The converse holds, provided that $n \geq N' \wedge l \geq L$ or $n \geq N \wedge l \geq L'$.

Lemma 3. $\forall n \forall k \forall r \forall s$ if $\sigma_{n+k} \cdots \gamma_n(X_r^0) = \gamma_{n+k}(X_s^0)$ then there is a k-arc forward path from $[X_r^0]$ to $[X_s^0]$ in G_N.

The converse holds, provided that $n \geq N$. The dual for G_L is as follows.

Lemma 4. $\forall l \forall k \forall r \forall s$ if $\sigma_{k+l} \cdots \gamma_0(X_r^0) = \sigma_{k+l} \cdots \gamma_k(X_s^0)$ then there is a k-arc forward path from $[X_r^0]$ to $[X_s^0]$ in G_L.

The converse holds if $l \geq L$. We also have the following.

Lemma 5. If $\sigma_{n+k+l} \cdots \gamma_n(X_r^0) = \sigma_{n+k+l} \cdots \gamma_{k+l}(X_s^0)$ then in G_N there exists a node $[v]$ and some t, $t \geq 0$ with a t-arc forward path from $[X_s^0]$ to $[v]$, and a $t+k$-arc forward path from $[X_r^0]$ to $[v]$.

Proof. By the n-k-l theorem, there is a path connecting X_r^0 and X_s^0. Let v be a node on the lowest level of this connecting path and apply Lemma 3.

Lemma 6. The premiss is as for Lemma 5, and we claim the existence of a node $[v]$ in G_L, and a t such that there is a t-arc forward path from $[v]$ to $[X_r^0]$ and a $t+k$-arc forward path from $[v]$ to $[X_s^0]$.

Proof. Let v be a variable on the highest level of the path and apply Lemma 4.

We alluded earlier to a sequence of Q literal instances within a rule expansion whose variables will be "fully equivalenced", and we now identify the instances of this sequence. Consider, within the rule expansion R_m for

some m, $m \geq M$, the subsequence of literals $Q_{N'}, Q_{N'+1}, \ldots, Q_{m-L'}$. They possess a property that we call *k-interval overlap consistency*, by which we mean that the variable overlap between two instances within this sequence that differ by k in their subscripts occurs at the same argument positions for *all* instances that differ by k. We first show that all of these instances are fully equivalenced.

Lemma 7. For any m and l, and any variables X_r^0 and X_s^0, if $m \geq M$ and $N' \leq l \leq m - L'$, then

$$\sigma_m \cdots \gamma_l(X_r^0) = \sigma_m \cdots \gamma_l(X_s^0) \Longleftrightarrow X_r^0 \equiv X_s^0$$

Proof. The only-if direction of the conclusion is immediate. We prove the reverse direcetion by induction on l. For $l = N'$, the implication is immediate since under the conditons of the premiss $m - l \geq L$. Assume it is true for l, and let $l + 1 \leq m - L'$. Consider an equivalence class, $[v_0] = \{v_0, \ldots, v_t\}$. We must show that for any choice of v_i and v_j, $\sigma_m \cdots \gamma_{l+1}(v_i) = \sigma_m \cdots \gamma_{l+1}(v_j)$. If both occur in P_C, then there exist w_i and w_j, with $w_i \equiv w_j$, $\sigma_m \cdots \gamma_{l+1}(v_i) = \sigma_m \cdots \sigma_{l+1}\gamma_l(w_i)$, and $\sigma_m \cdots \gamma_{l+1}(v_j) = \sigma_m \cdots \sigma_{l+1}\gamma_l(w_j)$. The result then follows from the inductive hypothesis. So all members of the equivalence class that occur in P_C are taken care of. Let v_i be a member that does not occur in P_C. If v_i does not occur in P_A, then it is the only member of the equivalence class, since there can be no paths from it to another node in the graph, and the implication follows trivially. So assume that v_i occurs in P_A. Now, either there is a member of the class that occurs in P_C or there is not. If there is not, then every member of the class can only be equivalenced by a path that uses levels below the line, for a path that used a level above the line as well would force the existence of a member in the class that occurs in P_C. Clearly, by the definition of L', L' levels must be adequate to equivalence any of the variables in the class, and $l + 1 \leq m - L'$ insures that we have enough levels to accomodate the path. On the other hand, if there are some members of the class that do occur in P_C, then as we argued above, the equality expression holds for any two of them. Now, there must be some of these that are connected to v_i by a path that uses only levels below the line, since any path starting from v_i must immediately drop a level, and the first time it returns to the initial line, we have a variable that is in the class. Clearly, the number of levels required by the shallowest such path can be no more than L', by the

definition of L', and since $l+1 \leq m - L'$, we again have enough levels at our disposal to accomodate the path.

The lemma demonstrates that the variables of the literals $Q_{N'}, \ldots, Q_{m-L'}$ are fully equivalenced, and k-interval overlap consistency follows quickly.

Lemma 8. Let $m \geq M$, $l, k, k' \epsilon \mathbf{N}$, $l \geq N'$, and $l + k + k' \leq m - L'$. Then for all variables X_r^0 and X_s^0

$$\begin{aligned} \sigma_m \cdots \gamma_l(X_r^0) &= \sigma_m \cdots \gamma_{l+k}(X_s^0) \Longleftrightarrow \\ \sigma_m \cdots \gamma_{l+k'}(X_r^0) &= \sigma_m \cdots \gamma_{l+k'+k}(X_s^0) \end{aligned}$$

Proof. Assuming either side of the if-and-only-if, by Lemma 2 there is a k-arc forward path from $[X_r^0]$ to $[X_s^0]$ in $G_{\equiv}$. Each arc in this path indicates the existence in the tail equivalence class of a variable $X_{i_l}^0$ occurring in P_A whose correspondent $X_{j_l}^0$ of P_C is a member of the equivalence class at the head of the arc. By the result of the previous lemma, provided we are within the given bounds, we can construct a path that does the trick by piecing together paths that equivalence members within a class and the forward arc from $X_{i_l}^0$ to $X_{j_l}^0$.

We will derive the B blocks from this interior sequence of Q literal instances. It can be shown that for any choice of blocking factor, the B blocks are renamings of each other, but to obtain mediation we must be more fussy. Consider the following example rule

$$P(X_2^0, X_3^0) \Longleftarrow P(X_1^0, X_2^0) \wedge Q(X_1^0, X_3^0)$$

The Q literals for R_3 are

$$\begin{array}{c} Q(X_1^3, X_1^1) \\ Q(X_1^2, X_1^0) \\ Q(X_1^1, X_2^0) \\ Q(X_1^0, X_3^0) \end{array}$$

If the blocking factor were 1, the overlap wouldn't be mediated, since every instance shares a variable with an instance that is at a distance of two from it. If one draws the graph for this example, the problem derives from the fact that there is a "gap" of length two between two nodes that occur in Q, that is, a path of length two from a node that occurs in Q to another node that occurs in Q with the intermediate node not occurring in Q. We will choose a block size that is as big as the largest gap to force mediation.

Formally, in $G_{\equiv}$, let p be the length of the longest non-null path from a node occurring in Q to another node occurring in Q with no intermediate node along the path occurring in Q, if such a path exists, or 1 if no such path exists. It is not hard to show

Lemma 9. Let p be determined as described above, and $m \geq M$ with $\lfloor \frac{m+1-N'-L'}{p} \rfloor = q \geq 2$. For $k \epsilon \{0, \ldots, q-1\}$, let $B_k = \bigwedge_{l=N'+kp}^{N'+(k+1)p-1} Q_l^m$. Then for $k_1, k_2 \epsilon \{0, \ldots, q-1\}$, $k_1 < k_2$, if a variable occurs in both B_{k_1} and B_{k_2} it occurs in B_{k_1+1}.

We use a similar ploy to force that the overlap between P_A^m and any block containing Q literal instances is mediated by H_r^m, only for this case the gap we are concerned with is defined differently.

In G_N, for each $[v]$ occurring in P_A, define gap($[v]$) to be 0, if $[v]$ occurs in Q or there is no $[w]$ occurring in Q with a forward path from $[w]$ to $[v]$, else the length of the shortest forward path from a node $[w]$ occurring in Q to $[v]$. Let h_0 be the maximum for all $[v]$ occurring in P_A of gap($[v]$). Then the following can be shown.

Lemma 10. For all $m \geq N + h_0$, if a variable of P_A^m occurs in some literal Q_l^m, $0 \leq l \leq m$, then it occurs in some literal Q_k^m with $m - h_0 \leq k \leq m$.

Proof. Use Lemma 3 and the definition of h_0.

In a similar vein, we define j_0 to be the value of h_0 for $(G^{op})_N$, and claim by duality,

Lemma 11. For all $m \geq j_0 + L$ if a variable occurs in P_C^m and in some Q_l^m for $0 \leq l \leq m$, then it occurs in some Q_l^m for some l with $0 \leq l \leq j_0$.

From the foregoing discussion, we have a value for p, and our value for j and h should be large enough to achieve full equivalencing for the B blocks and mediated overlap with the instances of the recursive literals. If we let h be the larger of $h_0 + 1$ and L', and j be the larger of $j_0 + 1$ and N', we guarantee almost all of the aspects of mediation. The exceptions are that J block overlap should be mediated by B_0 and H block overlap should be mediated by B_{q-1}. Both of these can be established by arguments similar to those outlined above.

We know that the B blocks are isomorphic, by the k-interval overlap consistency property. To complete the demonstration of iteration, we must show that the overlap of each with its predecessor occurs at the same argument positions, and similarly, the overlap of each with its successor occurs

at the same argument positions. This is obviously true for the block overlap within the fully equivalenced zone, but it is not so evident that the overlap between B_0 and J must occur at exactly the same "output" positions, or that the overlap of B_{q-1} with H_r must also occur at exactly the same "input" positions. We prove one of these, and claim the other by duality.

Lemma 12. Assume j and p are determined from the graph as described above. Let m be greater than or equal to M, X_r^0 be a variable occurring in Q, and $l_1 \epsilon \{0, 1, \ldots, p-1\}$. Then there exists X_s^0 occurring in Q and l_2, $0 \leq l_2 \leq j-1$ with

$$\sigma_m \cdots \gamma_{j+l_1}(X_r^0) = \sigma_m \cdots \gamma_{l_2}(X_s^0)$$

if and only if there exists X_t^0 occurring in Q and $l_3 \epsilon \{0, 1, \ldots, p-1\}$ with

$$\sigma_m \cdots \gamma_{j+p+l_1}(X_r^0) = \sigma_m \cdots \gamma_{j+l_3}(X_t^0)$$

Proof.($\Longrightarrow$). If $j - l_2 \leq p$, then let t be s and l_3 be $p + l_2 - j$. If $j - l_2 > p$, then certainly $l_1 + j - l_2 > p$ as well. By Lemma 2 we have a $l_1 + j - l_2$ arc forward path from $[X_s^0]$ to $[X_r^0]$ in $G_{\equiv}$, so let the sequence of nodes on this path be be $[X_s^0] = [v_0]$, $[v_1], \ldots, [v_{l_1+j-1}] = [X_r^0]$. By the definition of p, there must be some node among $[v_{j-l_2-p}], \ldots, [v_{j-l_2-1}]$ that occurs in Q, so let $[v_k]$ be the node. Let t be chosen so that $X_t^0 \equiv v_k$, and X_t^0 occurs in Q. There must be a choice for t, since $[v_k]$ occurs in Q. If we let l_3 be $k - j + l_2 + p$ the conclusion can be obtained.

($\Longleftarrow$) By Lemma 6 we know there exists in G_L a node $[v]$ and a value t_1, $t_1 \geq 0$, with a t_1-arc forward path from $[v]$ to $[X_t^0]$, and a $(t_1 + p - l_3 + l_1)$-arc forward path from $[v]$ to $[X_r^0]$. Trace back $l_1 + 1$ arcs from $[X_r^0]$ to a node we will call $[u]$. The purpose of this initial backward movement is to take us out of block B_0. Next, trace back arcs from $[u]$ until one of the following conditions is met.

1. a node occurring in Q is encountered
2. j_0 arcs have been traversed, where j_0 is the maximum gap value described above
3. a node with no predecessor is encountered

Regardless of the stopping condition, we claim that in the trace we encounter a node $[w]$ for which there exist numbers k_1 and k_2 and a variable X_s^0 such that

1. $0 \leq k_1 \leq k_2 \leq j_0$
2. X_s^0 occurs in Q
3. there is a k_1-arc forward path in G_L from $[w]$ to $[X_s^0]$
4. there is a k_2-arc forward path in G_L from $[w]$ to $[u]$(and thus, a $k_2 + l_1 + 1$-arc forward path to $[X_r^0]$)

We consider each stopping condition in turn.

1. If we reach a node occurring in Q after backing up less than or equal to j_0 arcs, let $[w]$ be the node we reached, k_2 be the number of arcs from $[w]$ to $[u]$, k_1 be 0, and X_s^0 be a variable occurring in Q that is equivalent to w.
2. If we traverse j_0 arcs backwards, the node we reach either has a predecessor or it does not. If it does, then by the definition of j_0, it must have a descendant node that occurs in Q and is within j_0 arcs of it. We let $[w]$ be the node we reached, pick X_s^0 as a variable occurring in Q that is in the descendant node, and let k_1 be the number of arcs from $[w]$ to the descendant. Of course, k_2 is j_0. If the node we reached after backing up j_0 arcs from $[u]$ has no predecessor, then $[X_t^0]$ must be descended from one of the nodes we encountered along the way. Call the first common ancestor of $[u]$ and $[X_t^0]$ encountered in the trace $[w]$. Let k_2 be the number of arcs between $[w]$ and $[u]$, k_1 be the distance from $[w]$ to $[X_t^0]$, and X_s^0 be X_t^0.
3. If we reach a node with no predecessors, then $[X_t^0]$ must be descended from one of the nodes we encountered along the way, and it reduces to the case we just considered.

Now, it can be shown that

$$\sigma_m \cdots \gamma_{j-1-k_2+k_1}(X_s^0) = \sigma_m \cdots \gamma_{j+l_1}(X_r^0)$$

by using the variable w as a connection. Since $j \geq j_0 + 1$ and $0 \leq k_1 \leq k_2 \leq j_0$, $0 \leq j - 1 - k_2 + k_1 \leq j - 1$, so we have a value for l_2.

This lemma and its dual complete the proof that our iteration condition is met. The stability condition is actually not hard to show, and we omit the proof. Taken together these three conditions insure that the expansions of any linear recursive rule can be organized in this fashion, and consequently viewed as a graph traversal.

We earlier stated that we could not force overlap between the recursive antecedent and the recursive consequent in a rule expansion to be mediated by the Q literal instances, and the following example shows why.

$$P(X_2^0, X_3^0, X_4^0) \Longleftarrow P(X_1^0, X_4^0, X_3^0) \wedge Q(X_1^0, X_2^0)$$

This rule falls within the class of range-restricted rules that we are considering, but because one of the connected components of the graph contains no variable occurring in Q, there will always be some overlap between the consequent and the antecedent that is unmediated. To see what happens to these variables in a rule expansion, one can form a dummy predicate Q' that contains all and only these variables, and analyze it separately. It is easy to see that the range restricted character of the rule forces that they will form cycles in $G_{\equiv}$, and each successive rule expansion induces a cyclic permutation of their locations in argument positions of Q'. If the blocking factor p is a multiple of the period of all these cycles, they pose no problems for the computation as a transitive closure, for the permutation induced by a B block will be the identity. The permutations induced by the blocks J and H_r are fixed, and can be correctly simulated. If the blocking factor is not a multiple of all of the periods, either the blocking factor must be changed to be a multiple, or the graph traversal must take into account not only the reachability of a node from another, but also note at what distance, to correctly mimic the permutations these variables would undergo in the expansion.

3 Extensions and Related Work

We have imposed the restrictions that the arguments of the recursive literals can only be variables and that there be a single non-recursive literal in the

antecedent. The former is simple to relax. If there are constants in the recursive literals, and any two distinct constants are connected in the graph, we know at once that the number of rule expansions necessary to calculate the relation is bounded, since at some point the two distinct constants will be equivalenced, a contradiction. Otherwise, any time a variable is equivalenced to a constant, the variable must become the constant, so we have a selection condition on the relation containing the variable, and the graph we traverse is simpler than what it would be if the constant were replaced by a variable.

If there are several non-recursive literals in the antecedent, although there is always a single relation definable in terms of the many, computing it might involve taking an unrestricted Cartesian product, potentially a very expensive operation. The full α-graph of Ioannidis includes undirected arcs between variables that occur in the same non-recursive literal, and it seems that the analysis performed here could be applied to the connected components of the full α-graph. Exactly what additional computation must be be performed to coalesce the results is a subject of future investigation, but it would appear that the results would have to be "tagged" with identifiers to be used to join them up, the identifiers corresponding to a source tuple in P_0 and a distance.

This work derives from a number of sources. The author's interest was first piqued by the work reported in [Hens84]. The notion of connecting the recursive computation to a graph traversal via the pattern of the rule expansions helped clarify some issues in that algorithm(see [Brig87a]). The work of Rosenthal, et al, described in [Rose86] uses graph traversal as a starting point for recursive computation over data bases, and the question of the relation of recursion as graph traversal and recursion as inference rule gained interest. Ioannidis's dissertation, although primarily directed at the question of boundedness of recursive rules, employed the graphical representation of the recursive rule we have adopted, and it has proved a convenient vehicle for recognizing and characterizing the pattern of variable overlap in rule expansions. Jagadish and Agrawal realized this in the paper [Jaga86], and the work presented here is closely related to theirs. The chief differences of this work are that the analysis here is explicitly tied to a formal representation of the rule expansions, the analysis here is deeper, and the graph constructed here is, in general, different from the graph

constructed by their method. In effect, they choose a blocking factor that forces something much stronger than mediation. For their blocking factor, if $i < j$ and a variable x occurs in both block B_i and B_j, then either $j = i + 1$, or x occurs in all of the B blocks. Their purpose is to permit the graph to be put in a form like the familiar graphical representation of a binary relation, but this is not essential to define a graph traversal that mimics the recursion. Their blocking factor also avoids the problem noted for unmediated overlap between the recursive consequent and the recursive antecedent, since it is always a multiple of the periods of all the cycles in $G_{\equiv}$.

We have shown how a simple linear recursion with a single recursive rule and a single exit rule can be viewed as a graph traversal by a close analysis of the rule expansions that a single linear rule can generate. It is the connection of the graph to the rule expansions that gives the construction a secure formal basis. We are hopeful that the general ideas employed here may serve in the analysis of more complicated recursions. We are also investigating the potential for supporting the recursion by maintaining the transitive closure of the graph in stored form.

References

[Agra86] Agrawal, R., and Jagadish, H., "On Bounded Linear Recursion", AT&T Bell Laboratories Technical Memorandum, 1986.

[Banc86] Bancilhon, F. , and Ramakrishnan, R., "An Amateur's Introduction to Recursive Query Processing", *Proceedings of SIGMOD '86 International Conference on Management of Data*, pps. 16–52.

[Brig87a] Briggs, D., "A Reconsideration of the Termination Condition of the Henschen-Naqvi Technique", COINS TR 87-11, University of Massachusetts, 1987.

[Brig87b] Briggs, D., "Towards Determining Variable Overlap in Recursive Rule Expansions", COINS TR 87-33, University of Massachusetts, 1987.

[Hens84] Henschen, L., and Naqvi, S., "On Compiling Queries on Recursive First-Order Data Bases", *JACM*, Vol. 31, January 1984, pps. 47–85.

[Ioan86] Ioannidis, Y., *Processing Recursion in Database Systems*, Ph. D. Thesis, University of California at Berkeley, 1986.

[Jaga86] Jagadish, H., and Agrawal, R., "A Study of Transitive Closure as a Recursion Mechanism", unpublished manuscript.

[Lloy84] Lloyd, J., *Foundations of Logic Programming*, Springer-Verlag, 1984.

[Rose86] Rosenthal, A., Heiler, S., Dayal, U., and Manola, F., "Traversal Recursion: A Practical Approach to Supporting Recursiye Applications", *Proceedings of SIGMOD '86 International Conference on Management of Data*,pps. 166–176,1986.