On the Convergence Rate of Linear Datalog° over Stable Semirings

Sungjin Im
University of California, Merced, CA, USA

Benjamin Moseley
Carnegie Mellon University, Pittsburgh, PA, USA

Hung Ngo
RelationalAi, Berkeley, CA, USA

Kirk Pruhs
University of Pittsburgh, Pittsburgh, PA, USA

Abstract

Datalog° is an extension of Datalog, where instead of a program being a collection of union of conjunctive queries over the standard Boolean semiring, a program may now be a collection of sum-product queries over an arbitrary commutative partially ordered pre-semiring. Datalog° is more powerful than Datalog in that its additional algebraic structure alows for supporting recursion with aggregation. At the same time, Datalog° retains the syntactic and semantic simplicity of Datalog: Datalog° has declarative least fixpoint semantics. The least fixpoint can be found via the naïve evaluation algorithm that repeatedly applies the immediate consequence operator until no further change is possible.

It was shown in [10] that, when the underlying semiring is p-stable, then the naïve evaluation of any Datalog° program over the semiring converges in a finite number of steps. However, the upper bounds on the rate of convergence were exponential in the number n of ground IDB atoms.

This paper establishes polynomial upper bounds on the convergence rate of the naïve algorithm on **linear** Datalog° programs, which is quite common in practice. In particular, the main result of this paper is that the convergence rate of linear Datalog° programs under any p-stable semiring is $O(pn^3)$. Furthermore, we show a matching lower bound by constructing a p-stable semiring and a linear Datalog° program that requires $\Omega(pn^3)$ iterations for the naïve iteration algorithm to converge. Next, we study the convergence rate in terms of the number of elements in the semiring for linear Datalog° programs. When L is the number of elements, the convergence rate is bounded by $O(pn \log L)$. This significantly improves the convergence rate for small L. We show a nearly matching lower bound as well.

2012 ACM Subject Classification Theory of computation → Database query languages (principles)

Keywords and phrases Datalog, convergence rate, semiring

Digital Object Identifier 10.4230/LIPIcs.ICDT.2024.11

Funding Moseley was supported in part by a Google Research Award, an Inform Research Award, a Carnegie Bosch Junior Faculty Chair, and NSF grants CCF-2121744 and CCF-1845146. Pruhs was supported in part by NSF grants CCF-1907673, CCF-2036077, CCF-2209654 and an IBM Faculty Award. Im was supported in part by NSF grants CCF-1844939 and CCF-2121745.

1 Introduction

In order to express common recursive computations with aggregates in modern data analytics while retaining the syntactic and semantic simplicity of Datalog, [10] introduced Datalog°, an extension of Datalog that allows for aggregation and recursion over an arbitrary *commutative partially ordered pre-semiring* (POPS). Datalog is exactly Datalog° over the Boolean semiring. Like Datalog, Datalog° has a declarative least fixpoint semantics, and the least fixpoint can

be found via the naïve iteration algorithm that repeatedly applies the immediate consequence operator until no further change is possible. Moreover, its additional algebraic structure allows for common recursions with aggregations.

However unlike Datalog, the naïve evaluation of a Datalog° program may not always converge in a finite number of steps. The convergence of a Datalog° program over a given POPS can be studied through its "core semiring", which is where we focus our attention on in this paper. This paper will only study Datalog° programs over commutative semirings, referring the readers to [10] for the generality of POPS.

It is known that the commutative semirings for which the iterative evaluation of Datalog° programs is guaranteed to converge are exactly those semirings that are stable [10]. A semiring is p-stable if the number of iterations required for any one-variable recursive linear Datalog° program to reach a fixed point is at most p, and a semiring is stable if there exists a p for which it is p-stable. Further, every non-stable semiring has a simple (linear) Datalog° program with one variable with the property that the iterative evaluation of this program over that semiring will not converge. Thus it is natural to concentrate on Datalog° programs over stable semirings. Previously, the best known upper bound on the convergence rate, which is the number of iterations until convergence, is $\sum_{i=1}^{n}(p+2)^i = \Theta(p^n)$ steps, where n is the number of ground atoms for the IDB's that ever have a nonzero value at some point in the iterative evaluation of the Datalog° program, and the underlying semiring is p-stable. In contrast there are no known lower bounds that show that iterative evaluation requires an exponential (in the parameter n) number of steps to reach convergence.

The exact general upper bound on the convergence rate of Datalog° programs over p-stable semirings is open, even for the special case of linear Datalog° programs. Linear Datalog° programs are quite common in practice.[1] Thus, in this paper we focus on this "easiest" case where the exact convergence rate is not known.

Currently, the best known upper bound on the convergence rate of linear Datalog° programs over p-stable semirings is $\sum_{i=1}^{n}(p+1)^i$ steps. This bound is unsatisfactory in the following sense. The prototypical example of a p-stable semiring is the tropical semiring Trop_p^+, (see Section 2 for a definition of Trop_p^+); in this case, it is known that the naïve algorithm converges in $O(pn)$ steps for linear Datalog° programs [10]. These results leave open the possibility that the convergence rate of the naïve algorithm on linear Datalog° programs over p-stable semirings could be exponentially smaller than the best known guarantee.

This paper seeks to obtain tighter bounds on the convergence rate of naïve evaluation of **linear** Datalog° programs. As the iterative evaluation of Datalog° programs is a reasonably natural and important algorithm/process, bounding the running time of this process is of both theoretical interests and practical interests. In practice, a known upper bound on the convergence rate allows the database system to determine *before hand* an upper bound on the number of iterations that will be required to evaluation a particular Datalog° program.

1.1 Background

Before stating our main results, we need to back up to set the stage a bit. A (traditional) Datalog program Π consists of a set of rules of the form:

$$R_0(\boldsymbol{X}_0) \text{ :- } R_1(\boldsymbol{X}_1) \wedge \cdots \wedge R_m(\boldsymbol{X}_m) \tag{1}$$

[1] For example, we can express transitive closure, all-pairs-shortest-paths, or weakly connected components in linear Datalog°.

where $R_0, \ldots, R_m$ are predicate names (not necessarily distinct) and each $\boldsymbol{X}_i$ is a tuple of variables and/or constants. The atom $R_0(\boldsymbol{X}_0)$ is called the head, and the conjunction $R_1(\boldsymbol{X}_1) \wedge \cdots \wedge R_m(\boldsymbol{X}_m)$ is called the body of the rule. Multiple rules with the same head are interpreted as a disjunction. A predicate that occurs in the head of some rule in Π is called an *intensional database predicate* or IDB, otherwise it is called an *extensional database predicate* or EDB. The EDBs form the input database, and the IDBs represent the output instance computed by Π. The finite set of all constants occurring in an EDB is called the *active domain*, and denoted ADom. A $\mathsf{Datalog}$ program is *linear* if every rule has at most one IDB predicate in the body.

There is an implicit existential quantifier over the body for all variables that appear in the body but not in the head, where the domain of the existential quantifier is ADom. In a linear Datalog program every conjunction has at most one IDB.

▶ **Example 1.** *The textbook example of a linear* $\mathsf{Datalog}$ *program is the following one, which computes the transitive closure of a directed graph, defined by the edge relation* $E(X, Y)$*:*

$$T(X, Y) \text{ :- } E(X, Y)$$
$$T(X, Y) \text{ :- } T(X, Z) \wedge E(Z, Y)$$

Here E *is an EDB predicate,* T *is an IDB predicate, and* ADom *consists of the vertices in the graph. The other way to write this program is to write it as a union of conjunctive queries (UCQs), where the quantifications are explicit:*

$$T(X, Y) \text{ :- } E(X, Y) \vee \exists_Z \ (T(X, Z) \wedge E(Z, Y)) \tag{2}$$

A $\mathsf{Datalog}$ program can be thought of as a function (called the *immediate consequence operator*, or *ICO*) that maps a set of ground IDB atoms to a set of ground IDB atoms. Every rule in the program is an inference rule that can be used to infer new ground IDB atoms from old ones. For a particular EDB instance, this function has a unique least fixpoint which can be obtained via repeatedly applying the ICO until a fixpoint is reached [1]. This least fixpoint is the semantics of the given $\mathsf{Datalog}$ program. The algorithm is called the *naïve evaluation algorithm*, which converges in a polynomial number of steps in the size of the input database, given that the program is of fixed size.

Like $\mathsf{Datalog}$ programs, a $\mathsf{Datalog}^\circ$ program consists of a set of rules, where the unions of conjunctive queries are now replaced with sum-product queries over a commutative semiring $\boldsymbol{S} = (S, \oplus, \otimes, 0, 1)$; see Section 2. Each rule has the form:

$$R_0(X_0) \text{ :- } \bigoplus R_1(\boldsymbol{X}_1) \otimes \cdots \otimes R_m(\boldsymbol{X}_m) \tag{3}$$

where sum ranges over the active ADom of the variables not in X_0. Further each ground atom in an EDB predicate or IDB predicate is associated with an element of the semiring $\boldsymbol{S}$, and the element associated with a tuple in an EDB predicate is specified in the input. The EDBs form the input database, and the ground atoms in the IDB's that have an associated semiring value that is nonzero represent the output instance computed by the $\mathsf{Datalog}^\circ$ program. Note that in a $\mathsf{Datalog}$ program the ground atom present in the input or output databases can be thought of as those that are associated with the element 1 in the standard Boolean semiring. A $\mathsf{Datalog}$ program is a $\mathsf{Datalog}^\circ$ program over the Boolean semiring $\boldsymbol{S} = (\{\mathsf{true}, \mathsf{false}\}, \vee, \wedge, \mathsf{false}, \mathsf{true})$. Again a $\mathsf{Datalog}^\circ$ program is linear if every rule has no more than one IDB prediciate in its body.

▶ **Example 2.** *A simple example of a linear* Datalog° *program is the following,*

$$T(X,Y) \text{ :- } E(X,Y) \oplus \bigoplus_Z T(X,Z) \otimes E(Z,Y), \tag{4}$$

which is (2) *with* $(\vee, \wedge, \exists_Z)$ *replaced by* $(\oplus, \otimes, \bigoplus_Z)$.

When interpreted over the Boolean semiring, we obtain the transitive closure program from Example 1. When interpreted over the tropical semiring $\mathsf{Trop}^+ = (\mathbb{R}_+ \cup \{\infty\}, \min, +, \infty, 0)$, *we have the* All-Pairs-Shortest-Path *(APSP) program, which computes the shortest path length* $T(X,Y)$ *between all pairs* X, Y *of vertices in a directed graph specified by an edge relation* $E(X,Y)$*, where the semiring element associated with* $E(X,Y)$ *is the length of the directed edge* (X,Y) *in the graph:*

$$T(X,Y) \text{ :- } \min\Big(E(X,Y), \min_Z(T(X,Z) + E(Z,Y))\Big) \tag{5}$$

A Datalog° program can be thought of as an immediate consequence operator (ICO). To understand how the ICO works in Datalog°, consider a rule with head R and let A be a ground atom for R with associated semiring value y, and assume that for A the body of this rule evaluates to the semiring element x. As a result of this, the ICO associates A with $x \oplus y$. Note that the functioning of the Datalog° ICO, when the semiring is the standard Boolean semiring, is identical to how the Datalog ICO functions. The iterative evaluation of a Datalog° program works in rounds/steps, where initially the semiring element 0 is associated with each possible ground atom in an IDB, and on each round the ICO is applied to the current state. So in the context of the Datalog° program in Example 1, if (a,b) was a ground atom in T with associated semiring element x, meaning that the shortest known directed path from vertex a to vertex b has length x, and (b,c) was a ground atom in E with associated semiring element y, meaning that there is a directed edge from b to c with length y, then the ICO would make the semiring element associated with A the minimum (as minimum is the addition operation in the tropical semiring) of its current value and $x + y$ (as normal additional is the multiplication operation in the tropical semiring).

Since the final associated semiring values of the ground atoms in an IDB are not initially known, it is natural to think of them as (IDB) variables. Then the grounded version of the ICO of a Datalog°program is a map $\boldsymbol{f} : S^n \to S^n$, where n is the number of ground atoms for the IDB's that ever have a nonzero value at some point in the iterative evaluation of the Datalog° program. For instance, in (5), there would be one variable for each pair (x,y) of vertices where there is a directed path from x to y in the graph. So the grounded version of the ICO of a Datalog° program has the following form:

$$\begin{aligned} X_1 &\text{ :- } f_1(X_1, \ldots, X_n) \\ &\quad \ldots \\ X_n &\text{ :- } f_n(X_1, \ldots, X_n) \end{aligned} \tag{6}$$

where the X_i's are the IDB variables, and f_i is the component of $\boldsymbol{f}$ corresponding to the IDB variable X_i. Note that each component function f_i is a multivariate polynomial in the IBD variables of degree at most the maximum number of terms in any product in the body of some rule in the Datalog° program. After q iterations of the iterative evaluation of a Datalog° program, the semiring value associated with the ground atom corresponding to X_i will be:

$$f_i^{(q)}(\mathbf{0}) \tag{7}$$

▶ **Definition 1** (p-stability). *Given a semiring $\boldsymbol{S} = (S, \oplus, \otimes, 0, 1)$ and $u \in S$, let $u^{(p)} := 1 \oplus u \oplus u^2 \oplus \cdots \oplus u^p$, where $u^i := u \otimes u \otimes \cdots \otimes u$ (i times). Then $u \in S$ is* p-stable *if $u^{(p)} = u^{(p+1)}$, and semiring $\boldsymbol{S}$ is* p-stable *if every element $u \in S$ is p-stable.*

▶ **Definition 2** (Stability index and convergence rate). *A function $f : S^n \to S^n$ is p-stable if $f^{(p+1)}(\mathbf{0}) = f^{(p)}(\mathbf{0})$, where $f^{(k)}$ is the k-fold composition of f with itself. The* stability index *of f is the smallest p such that f is p-stable. The* convergence rate *of a* Datalog$^\circ$ *program is the stability index of its ICO.*

The following bounds on the rate of convergence of a general (multivariate) polynomial function $f : S^n \to S^n$, where $\boldsymbol{S}$ is p-stable; this result naturally infers bounds on the convergence rate of Datalog$^\circ$ programs over p-stable semirings.

▶ **Theorem 3** ([10]). *The convergence rate of a* Datalog$^\circ$ *program over a p-stable commutative semiring is at most $\sum_{i=1}^{n}(p+2)^i$. Further, the convergence rate is at most $\sum_{i=1}^{n}(p+1)^i$ if the* Datalog$^\circ$ *program is linear. Finally, the convergence rate of a* Datalog$^\circ$ *program is at most n if $p = 0$.*

Thus the natural question left open by [10] was whether these upper bounds on the rate of convergence are (approximately) tight, and thus convergence can be exponential, or whether significantly better bounds are achievable.

1.2 Our Results

When a Datalog$^\circ$ program over a semiring $\boldsymbol{S}$ is *linear*, its ICO $f : S^n \to S^n$ is a *linear* function of the form $f(x) = A \otimes x \oplus b$, where A is an n by n matrix with entries from S, b is an $n \times 1$ column vector with entries from S, and the scalar multiplication and addition are from $\boldsymbol{S}$. To simplify notations, we will use $+$ and $\cdot$ to denote the operations $\oplus$ and $\otimes$ respectively. Further, following the convention, we may omit $\otimes$ if it is clear from the context. Then, we have

$$f(x) = Ax + b$$

▶ **Example 3.** *For the APSP* Datalog$^\circ$ *program, n is the number of edges in the graph, $b_{(u,v)}$ is $E(u,v)$, and the matrix A would be:*

$$A_{(u,v),(u,w)} = \begin{cases} E(v,w) & \text{if } u \neq v \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The stability index of f can easily be expressed in terms of the matrix A and vector b. Letting $A^0 = I$ where I is the identity matrix, we have

$$f^{(k)}(x) = A^k x + A^{(k-1)} b \qquad \text{where } A^{(k)} := \sum_{h=0}^{k-1} A^h.$$

Thus, the linear function $f = Ax + b$ is p-stable if and only if $A^{(p)}b = A^{(p+1)}b$. Using the simple form of f for the linear case, we can rewrite Definition 2 into the following simpler form.

▶ **Observation 4** (Convergence Rate of a Linear Datalog$^\circ$ Program).

- *The convergence rate for a particular linear* Datalog$^\circ$ *program, with associated matrix A and vector b, is the minimum natural number p such that $A^{(p)}b = A^{(p+1)}b$.*
- *The convergence rate for general linear* Datalog$^\circ$ *programs over a semiring S is then the maximum over all choices of A and b, of the convergence rate for that particular A and b.*

Our first result is an asymptotically tight bound of $\Theta(pn^3)$ on the rate of convergence for linear Datalog° programs.

▶ **Theorem 5.** *Every linear* Datalog° *program over a p-stable commutative semiring $\boldsymbol{S}$ converges in $O(pn^3)$ steps, where n is the total number of ground IDB atoms.*

The proof Theorem 5 can be found in Section 3, but we will give a brief overview of the main ideas of the proof here. Consider the complete n-vertex loop-digraph G where the edge from i to j is labeled with entry $A_{i,j}$. Then note that row i column j of $A^{(h)}$ is the sum – over all walks W from i to j of length $\leq h$ – of the product of the edge labels on the walk. We show that the summand corresponding to a walk W with more than $h = \Omega(pn^3)$ hops doesn't change the sum $A^{(h)}_{i,j}$. We accomplish this by rewriting the summand corresponding to W, using the commutativity of multiplication in S, as the product of a simple path and of multiple copies of at most $n^2 - n$ distinct closed walks. We then note that, by the pigeon hole principle, one of these closed walks, say C, must have mulitplicity greater than p. We then conclude that the summand corresponding to W will not change the sum $A^{(h)}_{i,j}$ by appealing to the stability of the semiring element that is the product of the edges in C.

In section 4 we establish a matching lower bound by finding a graph G and a commutative semiring $\boldsymbol{S}$ where the calculations in the upper bound are tight.

▶ **Theorem 6.** *For any $p, n \geq 1$, there is a linear* Datalog° *program over a p-stable semiring $\boldsymbol{S}$ that requires $\Omega(pn^3)$ steps to converge.*

In the lower bound instance that establishes Theorem 6 the size of the ground set S of $\boldsymbol{S}$ is of size $\Theta((p+1)^{n^2})$, so exponential in n. Thus one natural question is whether Datalog° programs over semirings with subexponentially sized ground sets will converge more quickly. In section 5 we answer this question in the affirmative by showing that the rate of convergence of a Datalog° program over a p-stable commutative semiring with L elements is $O(pn \log L)$.

▶ **Theorem 7.** *Every linear* Datalog° *program over a p-stable commutative semiring that contains L elements in its ground set converges in $O(pn \log L)$ steps.*

Let us explain the high level idea of the proof, with some simplifying assumptions, starting with the assumption that the stability of $\boldsymbol{S}$ is $p = 1$. Again we think of $A^{(k)}_{i,j}$ as the sum of products over walks W from i to j with at most k hops. Now consider a walk W from i to j consisting of $\Omega(n \log L)$ hops. Since there are n vertices, there exists a vertex v such that there are at least $\Omega(\log L)$ prefixes $P_1, P_2, \ldots, P_q$ of W ending at v. Let C_i be the closed walk starting and ending at v such that appending C_i to P_i produces P_{i+1}. Let us make the simplifying assumption that all of these closed walks C_i are distinct. The key observation is that if we delete any subset $\mathcal{C}$ of these $\Omega(\log L)$ closed walks from W, the result will still be a walk from i to j. Thus there are at least $2^{\Omega(\log L)}$ walks from i to j that can be formed by deleting a subset $\mathcal{C}$ of these closed walk. Since there are at most L distinct elements, by the pigeon hole principle, there must be some element e of S such that that are two subsets $\mathcal{C}, \mathcal{C}'$ where the product of the edges in them are the same. We then conclude by the stability of e that the summand corresponding to W does not change the sum $A^{(k)}_{i,j}$.

Finally in section 6 we establish a nearly matching lower bound by finding a graph G and a commutative semiring where the calculations in the upper bound are nearly tight.

▶ **Theorem 8.** *There are linear* Datalog° *programs over a p-stable commutative semiring that contains L elements that require $\Omega(pn\frac{\log L}{\log p})$ steps to converge.*

Finally in the appendix we consider a special case from [10], in which the commutative semiring $\boldsymbol{S}$ is naturally ordered.

2 Preliminaries

In this section, we introduce the notation and terminology used throughout the paper.

▶ **Definition 9** (Semiring). *A* semiring *[7] is a tuple* $\boldsymbol{S} = (S, \oplus, \otimes, 0, 1)$ *where*

- $\oplus$ *and* $\otimes$ *are binary operators on* S*,*
- $(S, \oplus, 0)$ *is a commutative monoid, meaning* $\oplus$ *is commutative and associative, and* 0 *is the identity for* $\oplus$*,*
- $(S, \otimes, 1)$ *is a monoid, meaning* $\otimes$ *is associative, and* 0 *is the identity for* $\oplus$*,*
- 0 *annilates every element* $a \in S$*, that is* $a \otimes 0 = 0 \otimes a = 0$*, and*
- $\otimes$ *distributes over* $\oplus$*.*

A commutative semiring $\boldsymbol{S} = (S, \oplus, \otimes, 0, 1)$ *is a semiring where additionally* $\otimes$ *is commutative.*

If A is a set and $p \geq 0$ a natural number, then we denote by $\mathcal{B}_p(A)$ the set of bags (multiset) of A of size p, and $\mathcal{B}_{\texttt{fin}}(A) := \bigcup_{p \geq 0} \mathcal{B}_p(A)$. We denote bags as in $\{\!\{a, a, a, b, c, c\}\!\}$. Given $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{B}_{fin}(\mathbb{R}_+ \cup \infty)$, define

$$\boldsymbol{x} \uplus \boldsymbol{y} := \text{bag union of } \boldsymbol{x}, \boldsymbol{y} \qquad \boldsymbol{x} + \boldsymbol{y} := \{\!\{u + v \mid u \in \boldsymbol{x}, v \in \boldsymbol{y}\}\!\}$$

▶ **Example 4.** *For any multiset* $\boldsymbol{x} = \{\!\{x_0, x_1, \ldots, x_n\}\!\}$*, where* $x_0 \leq x_1 \leq \ldots \leq x_n$*, and any* $p \geq 0$*, define:*

$$\min_p(\boldsymbol{x}) := \{\!\{x_0, x_1, \ldots, x_{\min(p,n)}\}\!\}$$

In other words, $\min_p$ *returns the smallest* $p+1$ *elements of the bag* $\boldsymbol{x}$*. Then, for any* $p \geq 0$*, the following is a semiring:*

$$\mathsf{Trop}_p^+ := (\mathcal{B}_{p+1}(\mathbb{R}_+ \cup \{\infty\}), \oplus_p, \otimes_p, \mathbf{0}_p, \mathbf{1}_p)$$

where:

$$\boldsymbol{x} \oplus_p \boldsymbol{y} \stackrel{def}{=} \min_p(\boldsymbol{x} \uplus \boldsymbol{y}) \qquad \mathbf{0}_p \stackrel{def}{=} \{\infty, \infty, \ldots, \infty\}$$

$$\boldsymbol{x} \otimes_p \boldsymbol{y} \stackrel{def}{=} \min_p(\boldsymbol{x} + \boldsymbol{y}) \qquad \mathbf{1}_p \stackrel{def}{=} \{0, \infty, \ldots, \infty\}$$

For example, if $p = 2$ *then* $\{\!\{3, 7, 9\}\!\} \oplus_2 \{\!\{3, 7, 7\}\!\} = \{\!\{3, 3, 7\}\!\}$ *and* $\{\!\{3, 7, 9\}\!\} \otimes_2 \{\!\{3, 7, 7\}\!\} = \{\!\{6, 10, 10\}\!\}$*. The following identities are easily checked, for any two finite bags* $\boldsymbol{x}, \boldsymbol{y}$*:*

$$\min_p(\min_p(\boldsymbol{x}) \uplus \min_p(\boldsymbol{y})) = \min_p(\boldsymbol{x} \uplus \boldsymbol{y}) \quad \min_p(\min_p(\boldsymbol{x}) + \min_p(\boldsymbol{y})) = \min_p(\boldsymbol{x} + \boldsymbol{y}) \quad (9)$$

Note that Trop_0^+ *is the natural "min-plus" semiring that we used in Example 2.*

In the following fact about p-stable commutative semiring will be useful.

▶ **Proposition 10.** *Given a* p*-stable commutative semiring* $\boldsymbol{S} = (S, \oplus, \otimes, 0, 1)$*, for any* $u \in S$*, we have* $pu = (p+1)u$*, where* pu *here is shorthand for* $\oplus_{i=1}^{p} u$*.*

Proof. This follows directly from the p-stability of 1, and the fact that $1^2 = 1$. ◀

Let us now explain the general procedure for creating a matrix A and a vector b from a linear $\mathsf{Datalog}^\circ$ program Q. Each ground tuple for each IDB predicate R can be viewed as a variable, and ground tuples of EDB predicates can be viewed as constants. Then a $\mathsf{Datalog}^\circ$ rule, where the head is IDB predicate R, can be converted into a collection of linear equations, one for each ground tuble of R. Since all rules that share IDB R as a the head can be combined via $\oplus$, we can compactly rewrite the entire set of $\mathsf{Datalog}^\circ$ rules as a collection of linear equations of the following form:

$$T_i \leftarrow \bigoplus_{j=1}^{n} (A_{ij} \otimes T_j) \oplus b_i$$

where $T_1, T_2, \ldots, T_n$ are the variables corresponding to ground tuples of IDB predicates. By using more familiar notation $+, \cdot$ in the lieu of $\otimes, \oplus$, we can model a linear Datalog° program Q by a linear function $f : S^n \to S^n$ of the form:

$$f(x) = Ax + b$$

where n is the total number of ground tuples across all IDB predicates, x is an n-dimensional vector with entries from S, A is an n by n matrix with entries from the S, and b is a dimensional column vector with entries from S. For some more examples of converting linear Datalog° programs into linear equations see [10].

3 Upper bounding the Convergence as a Function of the Matrix Dimension and the Semiring Stability

This section is devoted to proving Theorem 5, the main theorem of the paper. More specifically, we will show the following lemma. This lemma upper bounds the convergence of a p-stable semiring and implies Theorem 5.

▶ **Lemma 11.** *Let A be an $n \times n$ matrix over a p-stable semiring S. Then $A^{(k+1)} = A^{(k)}$, where $k = n(n^2 - n)(p + 2) + n - 1$.*

Consider the complete n-vertex loop-digraph G where the edge from i to j is labeled with entry $A_{i,j}$. Then

$$A^h_{i,j} = \sum_{W \in \mathcal{W}^h_{i,j}} \Phi(W)$$

where $\mathcal{W}^h_{i,j}$ is the collection of all h-hop walks from i to j in G, and

$$\Phi(W) = \prod_{(a,b) \in W} A_{a,b}$$

is the product off all the labels on all the directed edges in W. That is, row i column j of A^h is the sum over all h-hop walks W from i to j of the product of the labels on the walk. Similarly then,

$$A^{(h)}_{i,j} = \sum_{g=0}^{h} A^g_{i,j} = \sum_{g=0}^{h} \sum_{W \in \mathcal{W}^g_{i,j}} \Phi(W)$$

That is, row i column j of $A^{(h)}$ the sum over all walks W from i to j with at most h hops of the product of the labels on the walk. Further,

$$A^{(h+1)}_{i,j} = A^{(h)}_{i,j} + A^{h+1}_{i,j} = A^{(h)}_{i,j} + \sum_{W \in \mathcal{W}^{h+1}_{i,j}} \Phi(W)$$

Our proof technique is to show that, by the p-stability of S, it must be the case that for each $W \in \mathcal{W}^{p+1}_{i,j}$ it is the case that

$$A^{(k)}_{i,j} + \Phi(W) = A^{(k)}_{i,j}$$

The proof that $A_{i,j}^{(k)} = A_{i,j}^{(k+1)}$ then immediately follows by applying this fact to each $W \in \mathcal{W}_{i,j}^{k+1}$.

Fix $W = i, \ldots, j$ to be an arbitrary walk in $\mathcal{W}_{i,j}^{k+1}$. Our next intermediate goal is to rewrite $\Phi(W)$ using the commutativity of multiplication in S as the product of a simple path and at most $n^2 - n$ multicycles. That is,

$$\Phi(W) = \Phi(P) \prod_{h=1}^{\ell} \Phi(C_h^{z_h})$$

where P is a simple path from i to j in G, each C_h is a simple cycle in W that is repeated z_h times, and $\ell \leq n^2 - n$. We accomplish this goal via the following tail-recursive construction. The recursion is passed a collection of edges, and a parameter h. Initially, the edges are those in W and h is set to zero.

Recursive Construction. The base case is if W is a simple path. In the base case the path P is set to W and ℓ is set to 0. Otherwise:

- h is incremented
- C_h is set to be an arbitrary simple cycle in W.
- Let z_h be the minimum over all edges $e \in C_h$ of the number of times that e is traversed in W.
- Let W' be the collection of edges in W except that z_h copies of every edge in C_h are removed.
- The construction then recurses on W' and h.

In Lemma 13 we show that a particular statement about W is invariant through the recursive construction. A proof Lemma 13 requires the following lemma. The proof of the following lemma (or at least, the techniques needed for a proof) can be found in most introductory graph theory texts, e.g. [20] Theorem 23.1.

▶ **Lemma 12.**

- *A loop digraph G has a Eulerian walk from from a vertex i to a vertex j, where $i \neq j$, if and only if vertex i has out-degree one greater than its in-degree, vertex j has out-degree one less than its in-degree, every other vertex has equal in-degree and out-degree, and all of the vertices with nonzero degree belong to a single connected component of the underlying undirected graph.*
- *A loop digraph has an Eulerian cycle that includes a vertex i if and only if every vertex has equal in-degree and out-degree, vertex i has non-zero in-degree, and all of the vertices with nonzero degree belong to a single connected component of the underlying undirected graph.*

▶ **Lemma 13.** *Let W be a collection of edges that is passed at some point in the recursive construction. Let $D_1, \ldots, D_h$ be a partition of the edges of W with the property that if the edges in W were viewed as undirected, then the connected components would be $D_1, \ldots, D_h$.*

- *If $i \in D_f$ then D_f is a walk from i to j.*
- *If $i \notin D_f$ then D_f is a Eulearian circuit.*

Proof. The proof is by induction on the number of steps of the recursive construction. The statement is obviously true for the initial walk W, which is the base case. Now consider one step of the recursive construction. Removing copies of a cycle from W does not change the difference between the in-degree and out-degree of any vertex. Thus by Lemma 12 the only

issue we need to consider is vertex i and vertex j possibly ending up in different connected components of W'. Let D_f be the connected component of W that contains i (which also contains j by the induction hypothesis), let D'_a be the connected component of W' that contains i, and let D'_b be the connected component of W' that contains j, where $a \neq b$. Then the walk in D_f from i to j must cross the cut (in either direction) formed by the vertices in D'_a an odd number of times. But the edges in C_h must cross the cut (in either direction) formed by the vertices in D'_a an even number of times. Thus there must be an edge in D_f minus z_h copies of C_h that must cross the cut (in either direction) formed by the vertices in D'_a. However, then this is a contradiction to D'_a be a connected component in W'. ◀

We now make a sequence of observations, that will eventually lead us to our proof of Lemma 11.

▶ **Observation 14.**
- $1 \leq \ell \leq n^2 - n$.
- *The recursive construction terminates.*
- $\Phi(W) = \Phi(P) \prod_{h=1}^{\ell} \Phi(C_h^{z_h})$.

Proof. As W contains $k+1 = n(n^2-n)(p+2)+n$ edges, then it must contain a simple cycle, which implies $\ell \geq 1$. Consider one iteration of our recursive construction. There must be a directed edge $e \in C_h$ that appears exactly z_h times in W. Thus there are no occurrences of e in W'. Thus e can not appear in any future cycles, that is $e \notin C_g$ for any $g > h$. The first observation then follows because there are at most $n^2 - n$ different edges in G. The second observation is then an immediate consequence of the first observation, and the invariant established in Lemma 13 . The third observation follows because no edges are ever lost or created in the recursive construction. ◀

▶ **Observation 15.** *There is a cycle C_s, $1 \leq s \leq \ell$, such that z_s is at least $p+2$.*

Proof. Since P is a simple path it contains at most $n-1$ edges. Thus $W - P$ contains at least $n(n^2-n)(p+2)$ edges. As any simple cycle contains at most n edges, and as there are at most $\ell \leq n^2 - n$ cycles, then by applying the pigeon hole principle to the cycle decomposition of W we can conclude that there must be a cycle C_s that has multiplicity at least $p+2$, that is $z_s \geq p+2$. ◀

For convenience, consider renumbering the cycles so that $s = 1$ where $z_s \geq p+1$, which exists by Observation 15. For each h such that $1 \leq h \leq p+1$, let define W_h be the collection of edges in W minus h copies of every edge in C_1.

▶ **Observation 16.** *The edges in each W_h, $1 \leq h \leq p+1$ form a walk from i to j.*

Proof. As $z_1 \geq p+2$ and $h \leq p+1$, every edge that appears in W also appears in W_h. So W_h has the same connectivity properties as W, and W_h has the same vertices with positive in-degree as does W. Also as C_1 is a simple cycle, the difference between in-degree and out-degree for each vertex is the same in W_h as in W. Thus the result follows by appealing to Lemma 12. ◀

▶ **Observation 17.** *For each h such that $1 \leq h \leq p+1$ it is the case that*

$$\Phi(W_h) = \Phi(P)\Phi(C_1^{z_1-h}) \prod_{f=2}^{\ell} \Phi(C_f^{z_f})$$

Proof. This follows directly from the defintion of W_h. ◀

▶ **Observation 18.** *For all h such that $1 \leq h \leq p+1$, we have $W_h \in \bigcup_{f=0}^{k} \mathcal{W}_{i,j}^{f}$. That is $\Phi(W_h)$ appears as a term in $A_{i,j}^{(k)}$.*

Proof. This follows because W has $k+1$ edges and C_1 is non-empty, so removing edges in C_i strictly decreases the number of edges. ◀

We are now ready to prove Lemma 11. By Observation 18 we know that each $\Phi(W_h)$ is included in $A^{(k)}$, and thus there exists an element r in the semiring S such that

$$A_{i,j}^{(k)} = r + \sum_{h=1}^{p+1} \Phi(W_h)$$

Thus

$$\begin{aligned}
A_{i,j}^{(k)} + \Phi(W) &= r + \sum_{h=1}^{p+1} \Phi(W_h) + \Phi(W) \\
&= r + \sum_{h=1}^{p+1} \left(\Phi(P)\Phi(C_1^{z_1-h}) \prod_{f=2}^{\ell} \Phi(C_f^{z_f}) \right) + \Phi(P) \prod_{f=1}^{\ell} \Phi(C_f^{z_f}) \\
&= r + \left(\Phi(P) \prod_{f=2}^{\ell} \Phi(C_f^{z_f}) \right) \left(\sum_{h=1}^{p+1} \Phi(C_1^{z_1-h}) + \Phi(C_1^{z_1}) \right) && (10) \\
&= r + \left(\Phi(P) \prod_{f=2}^{\ell} \Phi(C_f^{z_f}) \right) \left(\Phi(C_1^{z_1-(p+1)}) \sum_{h=0}^{p+1} \Phi(C_1^{h}) \right) \\
&= r + \left(\Phi(P) \prod_{f=2}^{\ell} \Phi(C_f^{z_f}) \right) \left(\Phi(C_1^{z_1-(p+1)}) \sum_{h=0}^{p+1} [\Phi(C_1)]^{h} \right) && (11) \\
&= r + \left(\Phi(P) \prod_{f=2}^{\ell} \Phi(C_f^{z_f}) \right) \left(\Phi(C_1^{z_1-(p+1)}) \sum_{h=0}^{p} [\Phi(C_1)]^{h} \right) && (12) \\
&= r + \left(\Phi(P) \prod_{f=2}^{\ell} \Phi(C_f^{z_f}) \right) \left(\Phi(C_1^{z_1-(p+1)}) \sum_{h=0}^{p} \Phi(C_1^{h}) \right) \\
&= r + \left(\Phi(P) \prod_{f=2}^{\ell} \Phi(C_f^{z_f}) \right) \left(\sum_{h=1}^{p+1} \Phi(C_1^{z_1-h}) \right) \\
&= r + \sum_{h=1}^{p+1} \left(\Phi(P)\Phi(C_1^{z_1-h}) \prod_{f=2}^{\ell} \Phi(C_f^{z_f}) \right) \\
&= r + \sum_{h=1}^{p+1} \Phi(W_h) = A_{i,j}^{(k)}
\end{aligned}$$

The equality in line (10) follows from Observation 17. The equality in line (11) follows from the defintion of Φ. The key step in this line of equations is the equality in line (12), which follows from the stability of $\Phi(C_1)$. The rest of the equalities follow from basic algebraic properties of semirings, or by definition of the relavent term.

4 A Lower Bound for Convergence Rate of Linear Datalog° Programs

This section lower bounds the convergence rate of linear Datalog° programs using naive evaluation and establishes Theorem 6. In particular, this section will construct a semiring and a datalog program that requires $\Omega(pn^3)$ iterations to converge. The section first constructs a semiring then defines a matrix A and finally the converge rate is bounded. We remark that we only show this lower bound holds for this specific semiring and matrix.

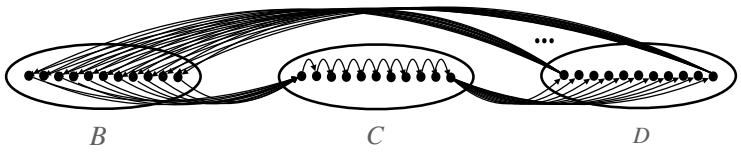
4.1 Constructing the Semiring

The ground set S consists of multi-sets of $\{1, \dots m\}$; later we will set $m = \Theta(n^2)$. To avoid confusion, we will refer to $1, 2, \dots m$ as items, which will be distinguished from the elements of S. An element in S can have up to $p \geq 1$ copies of each item $i \in [m]$. Additionally, there is a special element $\mathcal{O}$ in the semiring. Thus, the ground set S has $(p+1)^m + 1$ elements.

For a multiset A we denote the number of copies of item i in A as A_i. We now define the semiring operations. Consider two distinct elements (multisets) A and B in $S \setminus \{\mathcal{O}\}$. Define $C = A + B$, where $C_i = \max\{A_i, B_i\}$ for all $i \in [m]$. Further, $\mathcal{O} + A = A + \mathcal{O} = A$ for all $A \in S$. Multiplication $C = A \cdot B$ is defined as follows: $C_i = \min\{A_i + B_i, p\}$ for all $i \in [m]$. Further $\mathcal{O} \cdot A = A \cdot \mathcal{O} = \mathcal{O}$ for all $A \in S$.

We establish that this is a semiring where $\mathcal{O}$ is the additive identity and the empty multi-set $\emptyset$ is the multiplicative identity (i.e. 1). By definition, $\mathcal{O}$ is the annihilator element. We show in Appendix B that this is a commutative semiring. The proof easily follows from the definition of the semiring.

4.2 Defining the Matrix A



This section defines the matrix A. To do so, we first define a graph G. Let G be a directed graph where the vertex set is the integers from 1 to n inclusive. For simplicity assume n is divisible by 3. The vertices are partitioned into 3 parts, $B = \{1, \dots, n/3\}$, $C = \{n/3 + 1, \dots, 2n/3\}$ and $D = \{2n/3 + 1, \dots n\}$. There is a directed edge from each vertex in B to vertex $n/3 + 1$, there is a directed edge from vertex $2n/3$ to each vertex in D, and there is a directed edge from each vertex in D to each vertex in B. Finally, all vertices in C are sequentially connected from $n/3 + 1$ to $2n/3$, i.e., there is a directed edge from τ to $\tau + 1$ for all $\tau \in [n/3 + 1, 2n/3 - 1]$.

Index the edges by 1 through m and assign distinct labels (items) to them. So, m is exactly equal to the number of distinct items. Notice that m is $\Theta(n^2)$. If there is a directed edge (i, j) with label k in G then $A_{i,j} = \{k\}$. This is the element corresponding to the multiset with one copy of k. If there is no directed edge (i, j) in G then $A_{i,j} = \mathcal{O}$.

▶ **Lemma 19.** *The number of steps until convergence is* $\Omega(pn^3)$.

Proof. Note that there are $|B| \cdot |D| = n^2/9$ edges from D to B. Consider a long walk, say from $i := n/3+1$ to $j := n/3+2$. Observe that the walk must visit all edges within C (and the edge from $2n/3$ to $2n/3+1$) before visiting exactly one edge from D to B. Thus, to visit each

edge from D to B at least p times, the walk must have length at least $p|B|\cdot|D|\cdot|C| = pn^3/27$. Similarly, there is such a walk of length at most $p|B|\cdot|D|(|C|+2)+1 \leq 4pn^3$. Thus, we have shown that $A^{(k)}$ includes the multiset Q that has p copies of each item when $k \geq 4pn^3$. Further, we have shown that $A^{(k)}$ doesn't include Q when $k < p(n/3)^3$. This proves the lower bound on the convergence rate. ◀

5 Bounding Convergence in Terms of the Semiring Ground Set Size

This section investigates the convergence rate with the assumption that the semiring has a ground set of size at most L. With this assumption, we can prove significantly better upper bounds. This section's goal is to prove Theorem 7. We prove the following lemma, which will immediately imply Theorem 7. As before, we let $\Phi(W) = \prod_{e\in W} e$ for a walk W.

▶ **Lemma 20.** *Let A be an n by n matrix both a p-stable semiring S with a ground set consisting of L elements. Then $A^{(k+1)} = A^{(k)}$, where $k = \lceil 8p(\lg L+1)n\rceil + 1 = O(np\lg L)$.*

Proof. Fix $i,j \in [n]$. Consider any $W \in \mathcal{W}_{i,j}^{k+1}$ for the value of k stated in the lemma. To show the lemma it suffices to show $\Phi(W) + A_{i,j}^{(k)} = A_{i,j}^{(k)}$. By the pigeon hole principle, there must exist a vertex v visited at least $8p(\lg L+1)+1$ times. We now conceptually cut W at visits to v to form a cycle decomposition of W. That is, we can write W as $TC_1,\ldots,C_hT'$ where T is a walk from i to v such that the only time it visits vertex v is on the last step, each C_i is a closed walk that includes vertex v exactly once, and T' is a walk from v to j that doesn't visit v again after initially leaving v. Note by the definition of vertex v it must be the case that $h \geq 8p(\lg L+1)$. Let $\mathcal{C} = \{C_f \mid 1 \leq f \leq h\}$ be the collection of cycles in this cycle decomposition.

We now partition $\mathcal{C}$ into parts $\mathcal{C}_1,\ldots,\mathcal{C}_\ell,\mathcal{J}$ where for each closed walk C that has multiplicity m in $\mathcal{C}$ there are 2^f copies of C in $\mathcal{C}_f$ and $m-2^f$ copies of C in $\mathcal{J}$ where $f = \lfloor \lg m\rfloor$. For a collection $\mathcal{D}$ of cycles, it will be convenient to use $\Phi(\mathcal{D})$ to denote $\Phi(\bigcup\mathcal{D})$. Thus it then immediate that $\Phi(W) = \Phi(T)\Phi(\mathcal{C})\Phi(T')\Phi(\mathcal{J})$. Note that the cardinality of the multiset $\bigcup_{f=1}^{\ell}\mathcal{C}_f$ is at least $4p(\lg L+1)$.

We change $\mathcal{C}$ repeatedly without changing $\Phi(\mathcal{C})$. When changing $\mathcal{C}$ into $\mathcal{C}'$, we satisfy:

1. $\Phi(\mathcal{C}) = \Phi(\mathcal{C}')$.
2. $N(\mathcal{C}) = N(\mathcal{C}')$, where $N(\mathcal{C})$ denotes the size of multi-set $\mathcal{C}$. So, a cycle C contributes to $N(\mathcal{C})$ by the number of times it appears in $\mathcal{C}$. Further, $\mathcal{C}'$ has no more edges in total than $\mathcal{C}$ when we count 1 for each edge in one cycle appearance, i.e., $\sum_{C'\in\mathcal{C}'}|C'| \leq \sum_{C\in\mathcal{C}}|C|$.
3. Consider $\langle\ldots,N_3(\mathcal{C}'),N_2(\mathcal{C}'),N_1(\mathcal{C}')\rangle$ and $\langle\ldots,N_3(\mathcal{C}),N_2(\mathcal{C}),N_1(\mathcal{C})\rangle$. The first vector dominates the second lexicographically. Here $N_\ell(\mathcal{C})$ denotes the number of cycles in $\mathcal{C}$ of exponent 2^ℓ.
4. When we terminate, $N_\ell(\mathcal{C}) \leq 2\lfloor \lg L+1\rfloor$ for all $\ell \leq \lfloor\lg p\rfloor$.
5. Every cycle in $\mathcal{C}'$ also appears in $\mathcal{C}$.

We now describe the transformation from $\mathcal{C}$ into $\mathcal{C}'$. Consider the smallest ℓ such that $N_\ell > 2\lfloor\lg L+1\rfloor$. Consider every subset of cardinality $\lfloor\lg L+1\rfloor$, that consists of cycles of exponent 2^ℓ; so they are in $\mathcal{C}_\ell$. The number of such subsets is at least $\binom{2\lfloor\lg L+1\rfloor}{\lfloor\lg L+1\rfloor} > 2^{\lg L} = L$. Since the semiring has at most L distinct elements, there must exist distinct subsets A and B of cycles of exponent 2^ℓ such that $|A| = |B|$ and $\Phi(A) = \Phi(B)$. Assume wlog that B's cycles have no more edges in total than A's cycles. Then, we replace A with B in $\mathcal{C}$ and let $\mathcal{C}'$ be $\mathcal{C}$ after this change.

It is easy to see that $\Phi(\mathcal{C}) = \Phi(\mathcal{C}')$. This is because we replaced A with B such that $\Phi(A) = \Phi(B)$. More formally, $\Phi(\mathcal{C}) = \prod_{\ell'}\Phi(\mathcal{C}_{\ell'}) = (\Phi(\mathcal{C}_\ell\setminus A)\Phi(A))\prod_{\ell'\neq\ell}\Phi(\mathcal{C}_{\ell'}) = (\Phi(\mathcal{C}_\ell\setminus A)\Phi(B))\prod_{\ell'\neq\ell}\Phi(\mathcal{C}_{\ell'}) = \Phi(\mathcal{C}')$. The second property is also obvious because when

we replace A with B in the change, we ensured $|A| = |B|$, which implies that the multiset size remains unchanged. Also, it is immediate that the total number of edges don't increase as B has no more edges in total than A. The forth property, the termination condition, is immediate. The fifth property is also immediate since we do not create a new cycle when replacing A with B.

To see the third property, before replacing A with B, we had $A \cup B$ in $\mathcal{C}_\ell$, and their exponent was 2^ℓ. After the replacement, all cycles in $B \setminus A$ come to have exponent $2 \cdot 2^\ell = 2^{\ell+1}$, the cycles in $A \setminus B$ disappear from $\mathcal{C}_\ell$, and those in $A \cap B$ remain unchanged. Note that every cycle remains to have an exponent that is a power of two. Therefore, we have $N_{\ell+1}(\mathcal{C}') > N_{\ell+1}(\mathcal{C})$, and $N_{\ell'}(\mathcal{C}') = N_{\ell'}(\mathcal{C})$ for all $\ell' \geq \ell + 2$.

Observe that due to the second property and the third, the process must terminate. Thus, starting from $\mathcal{C}$, at the termination we have $\mathcal{C}'$ that satisfies the first, second, and fourth properties. We know $N(\mathcal{C}') = N(\mathcal{C}) > 4p(\lg L + 1)$. Further, we know that cycles of exponent at most p contribute to $N(\mathcal{C}')$ by at most $2(\lg L + 1)(1 + 2 + 4 + \ldots + 2^{\lfloor \lg p \rfloor}) < 4p(\lg L + 1)$. Thus, there must exist a cycle in $\mathcal{C}'$ of exponent greater than p. Let C denote the cycle. So, we have shown that

$$\Phi(W) = \Phi(T)\Phi(\mathcal{C}')\Phi(T') = \Phi(T)\Phi(C)^q\Phi(\mathcal{C}' \setminus C^q)\Phi(T'),$$

where $q \geq p + 1$. Here $\mathcal{C}' \setminus C^q$ implies the resulting collection of cycles we obtain after removing q copies of C from $\mathcal{C}'$. Now consider walk $W_{q'}$ that concatenates T, q' copies of C, $\mathcal{C}' \setminus C^{q'}$, and T'. Here, the walk starts with T and ends with T', and the cycles can be placed in an arbitrary order. This is because T is a walk from i to v, and all the cycles in $\mathcal{C}'$ start from v and end at v – due to the fifth property – and T' is a walk from v to j Further, they are all shorter than W because W_q is no longer than W due to the second property. Therefore, $W_0, W_1, \ldots W_{q-1} \in \mathcal{W}^{(k)}_{i,j}$. Thus, thanks to p-stability, we have $\Phi(W) + A^{(k)}_{i,j} = A^{(k)}_{i,j}$ as desired. ◀

Although we gave the full proof of Theorem 7, to convey intuition better, we also give some warm-up analyses in Appendix C by giving a looser bound for the general case and subsequently by considering a special case of $p = 1$.

6 Lower Bounds on Convergence in Terms of the Semiring Ground Set Size

This section constructs a lower bound of the convergence rate in terms of the size of the ground set. The goal is to show Theorem 8, which is implied by the following lemma.

▶ **Lemma 21.** *There exists an idempotent semiring on L elements and a matrix A of size n by n that requires $\Omega(nL)$ steps to converge to a fixed point.*

Proof. Consider a semiring that whose all powers of 2 from 1 to 2^L and the value 0. The value of 2^L is the largest value. Summation A and B in the semiring is $\min\{A \cdot B, 2^L\}$, where $\cdot$ is standard multiplication. Addition is standard maximum. By definition, addition is idempotent ensuring the semiring is idempotent.

The matrix A corresponds to a computation graph with a cycle of length n. All edges that are not in cycles are labeled 0. All edges of the cycle are labeled 1, the identity in standard multiplication, except for one edge, which is 2. Notice that multiplying all edges of the cycle i times results in the symbol 2^i.

The cycle needs to be traversed L times to reach 2^L. The walk is of length $\Theta(nL)$. ◀

▶ **Lemma 22.** *There exists a matrix A of size n by n which requires $\Omega(np\frac{\log L}{\log p})$ steps to find a fixed point over a semiring of L elements that is p-stable.*

Proof. Consider the following semiring. The L elements are over vectors of ℓ dimensions. Each position can be $0, 1, 2, \ldots, p$. Consider any two elements x and y. Let x_i and y_i be the ith dimension of x and y, respectfully. The addition operation on x and y returns whichever among x and y are lexicographically larger. Multiplication of x and y produces the vector z where $z_i = \min\{x_i + y_i, p\}$.

We first claim that this is a semiring. Notice that the all 0 vector is a monoid for multiplication. The vector of all p's is the multiplicative identity. The only case that is non-obvious is that multiplication distributes over addition. Consider three vectors a, b and c. Consider the expression $c \cdot (a + b)$. We aim to show that this is equal to $c \cdot a + c \cdot b$. Without loss of generality say a is lexicographically bigger than b. Then we have that $c \cdot (a + b) = c \cdot a$ because $a + b = a$ using that a is lexiographically bigger than c. Similarly, $c \cdot a + c \cdot b = c \cdot a$ because multiplication is standard addition and a is lexicographically bigger than b.

The matrix A corresponds to the following graph. There are two special nodes a and b. They are connected by ℓ one-hop path using two edges from a to b. The kth path goes via a node c_k. The edge from a to c_k is labeled with the 0 vector. The edge from c_k to b is labeled with a vector of all 0s except a 1 in dimension k. Additionally, b is connected to a via a directed path of length $n - \ell - 2$. The edges of this path are all labeled with the all 0 vector.

To collect the vector consisting of p in each dimension, one needs to walk a cycle starting at a at least $p\ell$ times. To see why, notice multiplication of the edges corresponding to a single time-around cycle increases a single dimension by at most one. Moreover, addition's definition ensures the final output is the vector that is lexicographically the biggest among each walk. Each cycle is of length $\Omega(n)$. The length of the walk required is $\Omega(p\ell n)$. Setting $\ell = \frac{\log L}{\log p}$ gives the lemma by noting that $L = p^\ell$ is the number of elements. ◀

7 Related Work

If the semiring is naturally ordered[2], then the least fixpoint of a Datalog° program is the least fixed point of f under the same partial order extended to S^n componentwise. This is the *least fixpoint* semantics of a Datalog° program. The naïve evaluation algorithm for evaluating Datalog programs extends naturally to evaluating Datalog° programs: starting from $\boldsymbol{x} = 0^n$, we repeatedly apply f to $\boldsymbol{x}$ until a fixpoint is reached $\boldsymbol{x} = f(\boldsymbol{x})$. The core semiring of a POPS is naturally ordered. Thus, we can find the least fixpoint of a Datalog° program by applying the naïve evaluation algorithm [10].

Computing the least fixpoint solution to a recursive Datalog° program boils down to solving fixpoint equations over semirings. In particular, we are given a multi-valued polynomial function $f : S^n \to S^n$ over a commutative semiring, and the problem is to compute a (pre-) fixpoint of f, i.e. a point $x \in S^n$ where $x = f(x)$. As surveyed in in [10], this problem was studied in a very wide range of communities, such as in automata theory [12], program analysis [4, 16], and graph algorithms [3, 14, 15] since the 1970s. (See [7, 8, 13, 17, 21] and references thereof).

When $f = Ax + b$ is linear, as shown in the paper $f^{(k)}(x) = A^{(k-1)}b$ and thus at fixpoint the solution is $A^{(\omega)}b = \lim_{k\to\infty} A^{(k-1)}b$, interpreted as a formal power series over the semiring. If there is a finite k for which $A^{(k)} = A^{(k+1)}$, then it is easy to see that

[2] $\boldsymbol{S}$ is naturally ordered if the relation $x \preceq_S y$ defined as $\exists z : x \oplus z = y$ is a partial order.

$A^{(\omega)} = A^{(k)}$. The problem of computing $A^{(\omega)}$ is called the *algebraic path problem* [17], which unifies many problems such as transitive closure [19], shortest paths [5], Kleene's theorem on finite automata and regular languages [11], and continuous dataflow [4, 9]. If A is a real matrix, then $A^{(\omega)} = I + A + A^2 + \cdots$ is exactly $(I - A)^{-1}$, if it exists [2, 6, 18].

There are several classes of solutions to the algebraic path problem, which have pros and cons depending on what we can assume about the underlying semiring (whether or not there is a closure operator, idempotency, natural orderability, etc.). We refer the reader to [7, 17] for more detailed discussions.

References

1 Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995. URL: `http://webdam.inria.fr/Alice/`.

2 R. C. Backhouse and B. A. Carré. Regular algebra applied to path-finding problems. *J. Inst. Math. Appl.*, 15:161–186, 1975.

3 Bernard Carré. *Graphs and networks*. The Clarendon Press, Oxford University Press, New York, 1979. Oxford Applied Mathematics and Computing Science Series.

4 Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In Robert M. Graham, Michael A. Harrison, and Ravi Sethi, editors, *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los Angeles, California, USA, January 1977*, pages 238–252. ACM, 1977. `doi:10.1145/512950.512973`.

5 Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345, 1962. `doi:10.1145/367766.368168`.

6 M. Gondran. Algèbre linéaire et cheminement dans un graphe. *Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Verte*, 9(V-1):77–99, 1975.

7 Michel Gondran and Michel Minoux. *Graphs, dioids and semirings*, volume 41 of *Operations Research/Computer Science Interfaces Series*. Springer, New York, 2008. New models and algorithms.

8 Mark W. Hopkins and Dexter Kozen. Parikh's theorem in commutative kleene algebra. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*, pages 394–401. IEEE Computer Society, 1999. `doi:10.1109/LICS.1999.782634`.

9 John B. Kam and Jeffrey D. Ullman. Global data flow analysis and iterative algorithms. *J. ACM*, 23(1):158–171, 1976. `doi:10.1145/321921.321938`.

10 Mahmoud Abo Khamis, Hung Q. Ngo, Reinhard Pichler, Dan Suciu, and Yisu Remy Wang. Convergence of datalog over (pre-) semirings. In Leonid Libkin and Pablo Barceló, editors, *PODS '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, pages 105–117. ACM, 2022. `doi:10.1145/3517804.3524140`.

11 S. C. Kleene. Representation of events in nerve nets and finite automata. In *Automata studies*, Annals of mathematics studies, no. 34, pages 3–41. Princeton University Press, Princeton, N. J., 1956.

12 Werner Kuich. Semirings and formal power series: their relevance to formal languages and automata. In *Handbook of formal languages, Vol. 1*, pages 609–677. Springer, Berlin, 1997. `doi:10.1007/978-3-642-59136-5_9`.

13 Daniel J. Lehmann. Algebraic structures for transitive closure. *Theor. Comput. Sci.*, 4(1):59–76, 1977. `doi:10.1016/0304-3975(77)90056-1`.

14 Richard J. Lipton, Donald J. Rose, and Robert Endre Tarjan. Generalized nested dissection. *SIAM J. Numer. Anal.*, 16(2):346–358, 1979. `doi:10.1137/0716027`.

15 Richard J. Lipton and Robert Endre Tarjan. Applications of a planar separator theorem. *SIAM J. Comput.*, 9(3):615–627, 1980. `doi:10.1137/0209046`.

16 Flemming Nielson, Hanne Riis Nielson, and Chris Hankin. *Principles of program analysis*. Springer-Verlag, Berlin, 1999. `doi:10.1007/978-3-662-03811-6`.

17 Günter Rote. Path problems in graphs. In *Computational graph theory*, volume 7 of *Comput. Suppl.*, pages 155–189. Springer, Vienna, 1990. doi:10.1007/978-3-7091-9076-0_9.
18 Robert E. Tarjan. Graph theory and gaussian elimination, 1976. J.R. Bunch and D.J. Rose, eds.
19 Stephen Warshall. A theorem on boolean matrices. *J. ACM*, 9(1):11–12, 1962. doi:10.1145/321105.321107.
20 Robin J. Wilson. *Introduction to Graph Theory*. Prentice Hall/Pearson, New York, 2010.
21 U. Zimmermann. Linear and combinatorial optimization in ordered algebraic structures. *Ann. Discrete Math.*, 10:viii+380, 1981.

A Convergence of Naturally Ordered Semirings

▶ **Definition 23** (Natural Order). *In any (pre-)semiring $\boldsymbol{S}$, the relation $x \preceq_S y$ defined as $\exists z : x \oplus z = y$, is a* preorder*, which means that it is reflexive and transitive, but it is not anti-symmetric in general. When $\preceq_S$ is anti-symmetric, then it is a partial order, and is called the* natural order *on $\boldsymbol{S}$; in that case we say that $\boldsymbol{S}$ is* naturally ordered*.*

For simplicity, we may use $\leq$ in lieu of $\preceq$. We naturally extend the ordering to vectors and matrices: for two vectors $\boldsymbol{v}, \boldsymbol{w} \in \boldsymbol{S}^n$, we have $\boldsymbol{v} \leq \boldsymbol{w}$ iff $\boldsymbol{v}_i \leq \boldsymbol{w}_i$ for all $i \in [n]$. Similarly, for two matrices A and B (which includes vectors) $A \leq B$ means that componentwise each entry in A is at most the entry in B, that is for all i and j it is the case that $A_{i,j} \leq B_{i,j}$.

Here we take k to be the longest chain in the natural order.

▶ **Theorem 24.** *Every linear* Datalog$^\circ$ *program over a p-stable naturally-ordered commutative semiring with maximum chain size k converges in $O(kn)$ steps.*

▶ **Theorem 25.** *There are linear* Datalog$^\circ$ *programs over a p-stable naturally-ordered commutative semiring with maximum chain size k that require in $\Omega(kn)$ steps to converge.*

This section considers bounds in terms of the longest chain in the partial order of a naturally ordered semiring. Recall that natural ordering means the following: for two elements a and b $a \leq b$ if and only if there exists a c such that $a + c = b$. Let L be the length of the longest chain in this partial order. We seek to bound the convergence rate in terms of n and L.

▶ **Lemma 26.** *Consider a naturally ordered semiring where L is the length of the longest chain in the partial order. Let A be an $n \times n$ matrix. Convergence must occur within nL steps.*

Proof. Consider $A^{(k)}x$ as k increases for any fixed x. If there is a $k \leq nL$ such that $A^{(k)}x = A^{(k+1)}x$ then convergence has been reached within the desired number of steps. Otherwise when $A^{(k)}x \neq A^{(k+1)}x$ there exists an i such that dimension i in $A^{(k+1)}x$ is strictly greater than dimension i in $A^{(k)}x$. This can only occur L times for each i by definition of the partial order. Knowing that there are at most n dimensions in $A^{(k)}x$, the lemma follows. ◀

▶ **Lemma 27.** *There exists a naturally ordered semiring where L is the length of the longest chain in the partial order and a n by n matrix where convergence requires $\Omega(nL)$ steps.*

Proof. Consider the following semiring. The semiring is on the set of integers $0, 1, 2, \ldots, L$ and a special element $\mathcal{O}$. Here, the additive identity is $\mathcal{O}$ and the multiplicative identity is 0. Consider two elements a and b that are not $\mathcal{O}$. Define the addition and multiplication of a and b to be equal to $\min\{a + b, L\}$. Define a multiplied by $\mathcal{O}$ to be $\mathcal{O}$ for any a and a added to $\mathcal{O}$ to be a for any a. Intuitively, addition and multiplication act as standard addition capped at L, except for the special $\mathcal{O}$ element.

Consider the following graph corresponding to a $n \times n$ matrix A. There is a cycle on n nodes. Order the edges from 1 to n. Each edge is labeled 0 except the edge from 1 to 2, which is labeled as 1. Traversing the cycle k times and multiplying the labels of the edges returns $\min\{k, L\}$. It takes a walk of length $\Omega(nL)$ to reach the element L. ◀

B Missing Proof from Section 4.1

We show that the semiring we defined in Section 4.1 is indeed a (commutative) semiring.

Monoid $(S, +)$ with identity $\mathcal{O}$:

- $A + \mathcal{O} = \mathcal{O} + A = A$. This follows from the definition.
- $(A + B) + C = A + (B + C)$. When $A, B, C \neq \mathcal{O}$, this is equivalent to showing $\max\{\max\{A_i, B_i\}, C_i\} = \max\{A_i, \max\{B_i, C_i\}\}$ for all $i \in [m]$, which follows from max being commutative. If A, B, or C is $\mathcal{O}$, then it is easy to check that it holds true.

Monoid $(S, \cdot)$ with identity $\emptyset$:

- $A \cdot \emptyset = \emptyset \cdot A = A$. If $A \neq \mathcal{O}$, this is immediate from the definition. If $A = \mathcal{O}$, again by definition, $\mathcal{O} \cdot \emptyset = \mathcal{O} \cdot \emptyset = \mathcal{O}$.
- $(A \cdot B) \cdot C = A \cdot (B \cdot C)$. When $A, B, C \neq \mathcal{O}$, it is an easy exercise to see $((A \cdot B) \cdot C)_i = (A \cdot (B \cdot C))_i = \min\{A_i + B_i + C_i, p\}$. If A, B or C is $\mathcal{O}$, both sides become $\mathcal{O}$.

Commutative:

- $A + B = B + A$. If $A, B \neq \mathcal{O}$, we have $(A + B)_i = \max\{A_i, B_i\} = (B + A)_i$. Otherwise, it is immediate from the definition of $\mathcal{O}$.
- $A \cdot B = B \cdot A$. We also show that the multiplication is also commutative. If $A, B \neq \mathcal{O}$, we have $(A \cdot B)_i = \min\{A_i + B_i, p\} = \min\{B_i + A_i, p\} = (B \cdot A)_i$. Otherwise $A \cdot B = B \cdot A = \mathcal{O}$ from the definition.

$\mathcal{O}$ is an multiplicative annihilator: We have $\mathcal{O} \cdot A = A \cdot \mathcal{O} = \mathcal{O}$ for all $A \in S$ from the definition.

Distributive:

- $A \cdot (B + C) = A \cdot B + A \cdot C$. Assume that $A, B, C \neq \mathcal{O}$ since otherwise it is straightforward to see that it holds true. We then have,

$$\begin{aligned}(A \cdot (B + C))_i &= \min\{A_i + \max\{B_i, C_i\}, p\} = \min\{\max\{A_i + B_i, A_i + C_i\}, p\} \\ &= \max\{\min\{A_i + B_i, p\}, \min\{A_i + C_i, p\}\} = \max\{(A \cdot B)_i, (A \cdot C)_i\} \\ &= (A \cdot B + A \cdot C)_i\end{aligned}$$

- $(B + C) \cdot A = B \cdot A + C \cdot A$. The proof is symmetric.

C Warm-up for Proof of Theorem 7

In Section 5 we gave the full proof of Theorem 7, which gives an upper bound of $O(np \log L)$ on the convergence rate when the underlying semiring has a ground set of size at most L.

To convey intuition better of the analysis, we give two warm-up proofs. Our first warm-up is giving a looser bound on the convergence rate. The proof makes use of the fact that a sufficiently long walk must visit the same vertex many times with the same product value. Here, we think of a prefix of the walk as a product of edges on the prefix. This prefex evaluates to an element of the semiring.

▶ **Lemma 28.** *Let A be an n by n matrix over a p-stable semiring on a ground set S consisting of L elements. Then $A^{(k+1)} = A^{(k)}$, where $k = npL$.*

Proof. Fix $i, j \in [n]$. Consider any $W \in \mathcal{W}_{i,j}^{k+1}$. To show the lemma it suffices to show $\Phi(W) + A_{i,j}^{(k)} = A_{i,j}^{(k)}$. Let W_h be the prefix of W of length h. Let $v(W_h)$ denote the ending point of W_h. Consider all pairs $(\Phi(W_h), v(W_h))$, $h \in [k+1]$. Since these tuples are subsets of $S \times [n]$ and $|S| = L$, due to the pigeonhole principle, there must exist $H \subseteq [k+1]$ of size $p+1$ such that $(\Phi(W_h), v(W_h))$ is the same tuple for all $h \in H$. By renaming we can represent the prefixes as X_1, X_1X_2, $X_1X_2X_3$, ..., $X_1X_2X_3 \dots X_{p+1}$.

For some T (possibly empty), we have $W = X_1X_2X_3 \dots X_{p+1}T$. By definition $\Phi(X_1) = \Phi(X_1X_2) = \dots = \Phi(X_1X_2 \dots X_{p+1})$.

Thus, $\Phi(W) = \Phi(X_1X_2X_3 \dots X_{p+1}T) = \Phi(X_1X_2X_3 \dots X_pT) \dots = \Phi(X_1T)$, This uses the fact that X_1T, X_1X_2T, ..., $X_1X_2X_3 \dots X_pT$ all are walks from i to j since X_1, X_1X_2, ..., $X_1X_2 \dots X_{p+1}$ all end where T starts. Further, all walks X_1T, X_1X_2T, ..., $X_1X_2X_3 \dots X_pT$ are strictly shorter than W. This implies that $\Phi(W)$ appears at least p times in $A_{i,j}^{(k)}$. Using Proposition 10, we conclude $\Phi(W) + A_{i,j}^{(k)} = A_{i,j}^{(k)}$ as desired. ◀

Next we consider the special case of $p = 1$. In this case we give an exponential improvement over what we showed in the previous lemma. The key idea is the following. Previously we identified p disjoint cycles $X_2, X_3, \dots, X_{p+1}$ that share the same starting and ending vertex from a long walk W in $\mathcal{W}_{i,j}^{k+1}$. Then, by removing them sequentially we were able to obtain p copies of the same element that have already appeared; thus adding W (or more precisely $\Phi(W)$) doesn't change $A_{i,j}^{(k)}$. Now we would like to make the same argument with an exponentially smaller number of cycles. Roughly speaking, we will identify $\Theta(\lg L)$ such cycles and find $2^{\Theta(\lg L)}$ walks by combining subsets of them. That is, the key idea is that we find more walks with the same product from far fewer cycles.

▶ **Lemma 29.** *Let A be an n by n matrix both over a 1-stable semiring S with a ground set consisting of L elements. Then $A^{(k+1)} = A^{(k)}$, where $k = O(n \lg L)$.*

Proof. As before, fix $i, j \in [n]$. Consider any $k \geq \lceil 2 \lg L \rceil n$. For any $W \in \mathcal{W}_{i,j}^{k+1}$ we show $\Phi(W) + A_{i,j}^{(k)} = A_{i,j}^{(k)}$. Since there are n vertices, the walk must visit some vertex at least $\lceil 2 \lg L \rceil + 1$ times. Formally, we can decompose W into

$$W = TC_1C_2 \dots C_HT' \tag{13}$$

where T, TC_1, TC_1C_2, ..., $TC_1C_2 \dots C_H$ all end at the same vertex v, and $H = \lceil 2 \lg L \rceil$. Note that all the cycles (or closed walks) $C_1, C_2, \dots C_H$ start from v and end at the same vertex v. It is plausible that some of them are identical.

For a subset A of $[H]$ we let $\hat{\Phi}(A) := \prod_{h \in A} \Phi(C_h)$. Since there are $2^{|H|}$ subsets of $[H]$ and $2^H > L$, there must exist $A, B \subseteq [H]$ such that $A \neq B$ and $\hat{\Phi}(A) = \hat{\Phi}(B)$. Assume wlog that $B \setminus A \neq \emptyset$. Thus we know

$$\hat{\Phi}(A \cap B)\hat{\Phi}(A \setminus B) = \hat{\Phi}(A \cap B)\hat{\Phi}(B \setminus A) \tag{14}$$

We can then show,

$$\begin{aligned}
\Phi(W) &= \Phi(T)\hat{\Phi}([H])\Phi(T') \qquad \text{[Eqn. 13]} \\
&= \Phi(T)\hat{\Phi}(A \cap B)\hat{\Phi}(A \setminus B)\hat{\Phi}(B \setminus A)\hat{\Phi}([H] \setminus (A \cup B))\Phi(T') \\
&= \Phi(T)\hat{\Phi}(A \cap B)(\hat{\Phi}(B \setminus A))^2\hat{\Phi}([H] \setminus (A \cup B))\Phi(T') \qquad \text{[Eqn. 14]}
\end{aligned}$$

Consider a walk W' that starts with T, has C_h for each $h \in (A \cap B) \cup (B \setminus A) \cup ([H] \setminus (A \cup B)) = [H] \setminus (A \setminus B)$ and ends with T'. Similarly, consider a walk W'' that starts with T, has C_h for each $h \in (A \cap B) \cup ([H] \setminus (A \cup B))$ and ends with T'. Note that W and W' are different

since $B \setminus A \neq \emptyset$. Further they are walks from i to j since every cycle C_h, $h \in [H]$ starts from and ends at the same vertex v. Further, both walks are shorter than W, and therefore are in $\mathcal{W}_{i,j}^{(k)}$. Since we have $\Phi(W') = \Phi(T)\hat{\Phi}(A \cap B)\hat{\Phi}(B \setminus A)\hat{\Phi}([H] \setminus (A \cup B))\Phi(T')$ and $\Phi(W'') = \Phi(T)\hat{\Phi}(A \cap B)\hat{\Phi}([H] \setminus (A \cup B))\Phi(T')$. We will show using 1-stability of the semiring that $\Phi(W') + \Phi(W'') + \Phi(W) = \Phi(W') + \Phi(W'')$, implying $\Phi(W) + A_{i,j}^{(k)} = A_{i,j}^{(k)}$ as desired.

Thus, it suffices to show $\Phi(W') + \Phi(W'') + \Phi(W) = \Phi(W') + \Phi(W'')$. To see this:

$$
\begin{aligned}
&\Phi(W') + \Phi(W'') + \Phi(W) \\
&= \Phi(T)\hat{\Phi}(A \cap B)\hat{\Phi}(B \setminus A)\hat{\Phi}([H] \setminus (A \cup B))\Phi(T') + \Phi(T)\hat{\Phi}(A \cap B)\hat{\Phi}([H] \setminus (A \cup B))\Phi(T') \\
&\quad + \Phi(T)\hat{\Phi}(A \cap B)(\hat{\Phi}(B \setminus A))^2\hat{\Phi}([H] \setminus (A \cup B))\Phi(T') \\
&= \Phi(T)\hat{\Phi}(A \cap B)\Phi([H] \setminus (A \cup B))\Phi(T')(1 + \Phi(B \setminus A) + \Phi(B \setminus A)^2) \\
&\quad \text{[associative and 1 is the multiplicative identiy]} \\
&= \Phi(T)\hat{\Phi}(A \cap B)\Phi([H] \setminus (A \cup B))\Phi(T')(1 + \Phi(B \setminus A)) \qquad \text{[1-stable]} \\
&= \Phi(W') + \Phi(W'')
\end{aligned}
$$

◀