Modeling Datalog Fact Assertion and Retraction in Linear Logic

Edmund S. L. Lam and Iliano Cervesato
Carnegie Mellon University, Qatar

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar
مؤسسة قطر
Qatar Foundation

1. Introducing Datalog and Deductive Databases

- A Logic Programming Language for *Deductive Databases*.
- An Example: Graph relation, let E be ***Edge*** and P be ***Path***,

$$\mathcal{P} = \begin{cases} r_1 : P(x,y) & :- \quad E(x,y) \\ r_2 : P(x,z) & :- \quad E(x,y), P(y,z) \end{cases}$$

- Assertion of new facts:

$$\begin{array}{ll} & E(2,3), P(2,3), \boxed{E(3,4)} \\ \Longrightarrow_{\mathcal{P}} & \mathcal{P}, E(2,3), P(2,3), E(3,4), \boxed{P(3,4)} \\ \Longrightarrow_{\mathcal{P}} & E(2,3), P(2,3), E(3,4), P(3,4), P(2,4) \end{array}$$

- Retraction of facts:

$$\begin{array}{ll} & \boxed{E(2,3)}, P(2,3), E(3,4), P(3,4), P(2,4) \\ \Longrightarrow_{\mathcal{P}} & \boxed{P(2,3)}, E(3,4), P(3,4), P(2,4) \\ \Longrightarrow_{\mathcal{P}} & E(3,4), P(3,4), \boxed{P(2,4)} \\ \Longrightarrow_{\mathcal{P}} & E(3,4), P(3,4) \end{array}$$

- Over recent ten years, Datalog has been applied to new domains, e.g.:
  - Implementing network protocols [GW10, LCG+06]
  - Distributed ensemble programming [ARLG+09]
  - Deductive spreadsheets [Cer07]
- Main challenge and focus so far:
  - Maintaining recursive views in presence of **assertion** and **retraction**.
  - Efficient algorithms and implementations are well-known [ARLG+09, CARG+12, GMS93, LCG+06]

2. Traditional Logical Interpretation of Datalog

- First order logic interpretation:

$$\mathcal{P} = \begin{cases} r_1 : \forall x,y.\ E(x,y) \supset P(x,y) \\ r_2 : \forall x,y,z.\ E(x,y) \wedge P(y,z) \supset P(x,z) \end{cases}$$

- Assertion = Forward chain application of implications, until *saturation*. e.g. adding of new base fact $E(3,4)$:

$$\begin{array}{c} \mathcal{P}, E(2,3), P(2,3), E(3,4), P(3,4), P(2,4) \vdash C \\ \hline \mathcal{P}, E(2,3), P(2,3), E(3,4), \boxed{P(3,4)} \vdash C \\ \hline \mathcal{P}, E(2,3), P(2,3), \boxed{E(3,4)} \vdash C \end{array}$$

- But what about *retraction*? E.g. removal of fact $E(2,3)$:

$$\begin{array}{c} \mathcal{P}, E(3,4), P(3,4) \vdash C \\ \hline ?? \\ \mathcal{P}, E(2,3), P(2,3), E(3,4), P(3,4), P(2,4) \vdash C \end{array}$$

3. Our Objective

- To define a logical specification of Datalog that supports **assertion** and **retraction** internally.
- Our Solution: Define a *Linear Logic* [Gir87] Interpretation of Datalog.
- Linear logic because
  - Assumptions can grow or shrink as inference rules apply.
  - Facts are not permanent truths, but can be retracted (consumed)

4. Linear Logic Interpretation of Datalog

Example: Linear logic interpretation (simplified) of the Graph program $\mathcal{P}$:

- $r_1 : P(x,y) \quad :- \quad E(x,y)$ is interpreted as

$$\begin{array}{l} \mathcal{I}_1^{(x,y)} = E(x,y) \multimap P(x,y) \otimes E(x,y) \otimes \mathcal{R}_1^{(x,y)} \\ \mathcal{R}_1^{(x,y)} = (\tilde{E}(x,y) \multimap \tilde{P}(x,y) \otimes \tilde{E}(x,y)) \end{array}$$

- $r_2 : P(x,z) \quad :- \quad E(x,y), P(y,z)$ is interpreted as

$$\begin{array}{l} \mathcal{I}_2^{(x,y,z)} = E(x,y) \otimes P(y,z) \multimap P(x,z) \otimes E(x,y) \otimes P(y,z) \otimes \mathcal{R}_2^{(x,y,z)} \\ \mathcal{R}_2^{(x,y,z)} = (\tilde{E}(x,y) \multimap \tilde{P}(x,z) \otimes \tilde{E}(x,y)) \,\&\, (\tilde{P}(y,z) \multimap \tilde{P}(x,z) \otimes \tilde{P}(y,z)) \end{array}$$

- Absorption rules:

$$\mathcal{A}_{\mathcal{P}} = \begin{cases} E(x,y) \otimes \tilde{E}(x,y) \multimap 1 \\ P(x,y) \otimes \tilde{P}(x,y) \multimap 1 \end{cases}$$

- Program interpretation denoted as:

$$\llbracket \mathcal{P} \rrbracket = \forall x,y.\mathcal{I}_1^{(x,y)}, \forall x,y,z.\mathcal{I}_2^{(x,y,z)}$$

5. Datalog Assertion in Linear Logic Interpretation

- Two-sided intutionistic linear logic sequent calculus, $\boldsymbol{LV^{obs}}$: $\Gamma; \Delta \longrightarrow C$
- Assertion, e.g. adding of new base fact $E(3,4)$:

$$\begin{array}{c} \llbracket \mathcal{P} \rrbracket, \mathcal{A}_{\mathcal{P}}; E(2,3), P(2,3), \mathcal{R}_1^{(2,3)}, E(3,4), P(3,4), \mathcal{R}_1^{(3,4)}, P(2,4), \mathcal{R}_2^{(2,3,4)} \longrightarrow C \\ \hline \boxed{\llbracket \mathcal{P} \rrbracket}, \mathcal{A}_{\mathcal{P}}; \boxed{E(2,3)}, P(2,3), \mathcal{R}_1^{(2,3)}, E(3,4), \boxed{P(3,4)}, \mathcal{R}_1^{(3,4)} \longrightarrow C \\ \hline \boxed{\llbracket \mathcal{P} \rrbracket}, \mathcal{A}_{\mathcal{P}}; E(2,3), P(2,3), \mathcal{R}_1^{(2,3)}, \boxed{E(3,4)} \longrightarrow C \end{array}$$

- Similar to traditional logic interpretation, Datalog assertions map to forward chaining fragment of Linear Logic proof search.
- Key difference: Inference of new facts leaves behind "bookkeeping" information:
  - Specifically retraction rules ($\mathcal{R}_1^{(2,3)}$, $\mathcal{R}_2^{(2,3,4)}$, etc..)
  - Act as "cookie crumbles" that guides retraction

6. Datalog Retraction in Linear Logic Interpretation

Retraction, e.g. removal of fact $E(2,3)$:

$$\begin{array}{c} \llbracket \mathcal{P} \rrbracket, \boxed{\mathcal{A}_{\mathcal{P}}}; E(3,4), P(3,4), \mathcal{R}_1^{(3,4)} \longrightarrow C \\ \hline \llbracket \mathcal{P} \rrbracket, \boxed{\mathcal{A}_{\mathcal{P}}}; \boxed{E(2,3)}, E(3,4), P(3,4), \mathcal{R}_1^{(3,4)}, \boxed{\tilde{E}(2,3)} \longrightarrow C \\ \hline \llbracket \mathcal{P} \rrbracket, \boxed{\mathcal{A}_{\mathcal{P}}}; E(2,3), \boxed{P(2,3)}, E(3,4), P(3,4), \mathcal{R}_1^{(3,4)}, \tilde{E}(2,3), \boxed{\tilde{P}(2,3)} \longrightarrow C \\ \hline \llbracket \mathcal{P} \rrbracket, \mathcal{A}_{\mathcal{P}}; E(2,3), P(2,3), \boxed{\mathcal{R}_1^{(2,3)}}, E(3,4), P(3,4), \mathcal{R}_1^{(3,4)}, \boxed{\tilde{E}(2,3)} \longrightarrow C \\ \hline \llbracket \mathcal{P} \rrbracket, \boxed{\mathcal{A}_{\mathcal{P}}}; \left(\begin{array}{l} E(2,3), P(2,3), \mathcal{R}_1^{(2,3)}, E(3,4), P(3,4), \\ \mathcal{R}_1^{(3,4)}, \boxed{P(2,4)}, \tilde{E}(2,3), \boxed{\tilde{P}(2,4)} \end{array} \right) \longrightarrow C \\ \hline \llbracket \mathcal{P} \rrbracket, \mathcal{A}_{\mathcal{P}}; \left(\begin{array}{l} E(2,3), P(2,3), \mathcal{R}_1^{(2,3)}, E(3,4), P(3,4), \\ \mathcal{R}_1^{(3,4)}, P(2,4), \boxed{\mathcal{R}_2^{(2,3,4)}}, \boxed{\tilde{E}(2,3)} \end{array} \right) \longrightarrow C \end{array}$$

Retraction can now be represented in forward chaining fragment of linear logic as well!!

7. Completeness and Soundness Results

- Define $\Delta \overset{\alpha}{\Longrightarrow}{}^{LL}_{\llbracket \mathcal{P} \rrbracket} \Delta'$ as an abstract state transition system that computes inference closures of Datalog states Δ.
- We define this, based on *linear logic proof search*:

$$\frac{a \notin \Delta \quad \llbracket \mathcal{P} \rrbracket, \mathcal{A}_{\mathcal{P}}; \Delta, a \longrightarrow \bigotimes \Delta' \quad \textit{Quiescent}(\Delta', (\llbracket \mathcal{P} \rrbracket, \mathcal{A}_{\mathcal{P}}))}{\Delta \overset{+a}{\Longrightarrow}{}^{LL}_{\llbracket \mathcal{P} \rrbracket} \Delta'} \ (\textit{Infer})$$

$$\frac{a \in \Delta \quad \llbracket \mathcal{P} \rrbracket, \mathcal{A}_{\mathcal{P}}; \Delta, \tilde{a} \longrightarrow \bigotimes \Delta' \quad \textit{Quiescent}(\Delta', (\llbracket \mathcal{P} \rrbracket, \mathcal{A}_{\mathcal{P}}))}{\Delta \overset{-a}{\Longrightarrow}{}^{LL}_{\llbracket \mathcal{P} \rrbracket} \Delta'} \ (\textit{Retract})$$

- Technical hurdles that we had to over-come to achieve this:
  - Trivial non-termination in assertions
  - In-exhaustive retraction
- *Correctness* and *Soundness* of assertion and retraction: Given a Datalog Program $\mathcal{P}$, for reachable states $\Delta_1, \Delta_1^{\mathcal{R}}, \Delta_1^{\sharp}$ and $\Delta_2, \Delta_2^{\mathcal{R}}, \Delta_2^{\sharp}$ such that $\Delta_1 = \llbracket \mathcal{P}(\mathcal{B}_1) \rrbracket$ and $\Delta_2 = \llbracket \mathcal{P}(\mathcal{B}_2) \rrbracket$, then we have the following:

$$(\Delta_1, \Delta_1^{\mathcal{R}}, \Delta_1^{\sharp}) \overset{\alpha}{\Longrightarrow}{}^{LL}_{\llbracket \mathcal{P} \rrbracket} (\Delta_2, \Delta_2^{\mathcal{R}}, \Delta_2^{\sharp}) \quad \text{iff} \quad \mathcal{P}(\mathcal{B}_1) \overset{\alpha}{\Longrightarrow}_{\mathcal{P}} \mathcal{P}(\mathcal{B}_2)$$

where $\mathcal{P}(\mathcal{B}) = \{p(\vec{t}) \mid \mathcal{P}, \mathcal{B} \vdash p(\vec{t})\}$ and α is either $+a$ or $-a$

- See our PPDP'12 paper or tech report (CMU-CS-12-126) for details.

8. Contributions and Future Works

- So why do we need a linear logic interpretation of Datalog?
- We've got a few reasons:
  - Provide a refined logical understanding of Datalog assertion and retraction, hence we can prove properties of Datalog programs via theorem provers (e.g. CLF)
  - Provide an operational semantics of Datalog style assertion and retraction based on higher order, forward chaining multiset rewrite rules.
  - Provide a cleaner and more theoretically well-founded way of implementing and reasoning about modern extensions of Datalog (e.g. Meld [ARLG+09], Dedalus [AMC+09], Distributed Datalog [NJLS11]).
- Future Works:
  - Implementation of Datalog based on higher order multiset rewritings.
  - Refine our linear logic interpretation.

* Funded by the Qatar National Research Fund as project NPRP 09-667-1-100 (Effective Programming for Large Distributed Ensembles)

* In proceedings of 14^{th} International Symposium on Principles and Practice of Declarative Programming (PPDP'12)

http://www.qatar.cmu.edu/~sllam/
sllam@qatar.cmu.edu