Modeling Datalog Assertion and Retraction in Linear Logic

Iliano Cervesato iliano@cmu.edu
Edmund S. L. Lam sllam@qatar.cmu.edu

Carnegie Mellon University, Qatar
Department of Computer Science

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

Sept 21th, 2012

The Road Map

1. Introduction
2. Problem and Objective
3. Technical Hurdles
4. The IID^{Full} Interpretation
5. Discussions, Future works and Conclusion

Introduction

- Datalog, a Logic Programming Language
- Originally used for implementing deductive databases.
- Maintaining recursive views (logical consequences):
  $\mathcal{P}(\mathcal{B}) = \{p(\vec{t}) \mid \mathcal{P}, \mathcal{B} \vdash p(\vec{t})\}$
- Over past 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]
- Fact bases $\mathcal{B}$ of such domains are typically highly dynamic (i.e. **assertion** and **retraction**):

$$\mathcal{P}(\mathcal{B}) \overset{+a}{\Longrightarrow}_{\mathcal{P}} \mathcal{P}(\mathcal{B}, a) \quad (\mathit{Infer})$$

$$\mathcal{P}(\mathcal{B}, a) \overset{-a}{\Longrightarrow}_{\mathcal{P}} \mathcal{P}(\mathcal{B}) \quad (\mathit{Retract})$$

Introduction

- 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]
- But...
  - while it is well-known how to implement...
  - lacks logical specification, specifically retraction.

The Road Map

The Problem

- Consider the path relation, let E be *Edge* and P be *Path*:

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

- Computing inference: Base Facts $\supset$ Inferred Facts

$$\frac{\mathcal{P}, E(2,3), P(2,3) \vdash C}{\boxed{\mathcal{P}}, \boxed{E(2,3)} \vdash C}$$

- Forward-chaining application of inference rule, until *saturation*.

The Problem

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

- Assertion, e.g. adding of new base fact $E(3,4)$:

$$\dfrac{\dfrac{\mathcal{P}, E(2,3), P(2,3), E(3,4), P(3,4), P(2,4) \vdash C}{\boxed{\mathcal{P}, E(2,3)}, P(2,3), E(3,4), \boxed{P(3,4)} \vdash C}}{\boxed{\mathcal{P}}, E(2,3), P(2,3), \boxed{E(3,4)} \vdash C}$$

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

$$\dfrac{\dfrac{\mathcal{P}, E(3,4), P(3,4) \vdash C}{??}}{\mathcal{P}, E(2,3), P(2,3), E(3,4), P(3,4), P(2,4) \vdash C}$$

The Objective

- Define a logical specification of Datalog that supports **assertion** and **retraction** internally using linear logic.
- Linear logic because
  - Assumptions can grow or shrink as inference rules apply.
  - Facts are not permenent truths, but can be retracted (consumed)

The Objective

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

- $r_1 : \forall x, y.\ E(x, y) \supset P(x, y)$ interpreted as

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

- $r_2 : \forall x, y, z.\ E(x, y) \wedge P(y, z) \supset P(x, z)$ interpreted as

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

- 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:

$$[\![\mathcal{P}]\!]' = \forall x, y.\mathcal{I}_1^{(x,y)}, \forall x, y, z.\mathcal{I}_2^{(x,y,z)}$$

The Objective

- Two-sided intutionistic linear logic sequent calculus, LV^{obs}:
  $\Gamma ; \Delta \longrightarrow C$
- Assertion, e.g. adding of new base fact $E(4,5)$:

$$\dfrac{\dfrac{\llbracket \mathcal{P} \rrbracket', \mathcal{A}_{\mathcal{P}}; E(2,3), P(2,3), \mathcal{R}_1^{(2,3)}, \boxed{E(3,4)} \longrightarrow C}{\boxed{\llbracket \mathcal{P} \rrbracket'}, \mathcal{A}_{\mathcal{P}}; \boxed{E(2,3)}, P(2,3), \mathcal{R}_1^{(2,3)}, E(3,4), \boxed{P(3,4)}, \mathcal{R}_1^{(3,4)} \longrightarrow C}}{\llbracket \mathcal{P} \rrbracket', \mathcal{A}_{\mathcal{P}}; E(2,3), P(2,3), \mathcal{R}_1^{(2,3)}, E(3,4), P(3,4), \mathcal{R}_1^{(3,4)}, P(2,4), \mathcal{R}_2^{(2,3,4)} \longrightarrow C}$$

- 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

The Objective

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

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

The Road Map

Technical Hurdles

Looks like we are done...

Technical Hurdles

Looks like we are done... But wait!

Technical Hurdle 1: Trivial non-termination in assertions

$$\vdots$$

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

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

$$\boxed{\llbracket \mathcal{P} \rrbracket'}, \mathcal{A}_{\mathcal{P}}; \boxed{E(2,3)}, P(2,3), \mathcal{R}_1^{(2,3)}, E(3,4), \boxed{P(3,4)}, \mathcal{R}_1^{(3,4)} \longrightarrow C$$

$$\boxed{\llbracket \mathcal{P} \rrbracket'}, \mathcal{A}_{\mathcal{P}}; E(2,3), P(2,3), \mathcal{R}_1^{(2,3)}, \boxed{E(3,4)} \longrightarrow C$$

Technical Hurdle 2: Inexhaustive retraction

$$\llbracket \mathcal{P} \rrbracket', \mathcal{A}_{\mathcal{P}}; P(2,3), \mathcal{R}_1^{(2,3)}, E(3,4), P(3,4), \mathcal{R}_1^{(3,4)}, P(2,4), \mathcal{R}_2^{(2,3,4)} \longrightarrow C$$

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

Termination in Assertions

- We define an immediate interpretation of Datalog program $\mathcal{P}$, denoted $\lceil \mathcal{P} \rceil$
- Extend linear logic with *guarded implications*: $A \multimap_a B$
- Can be treated as *negation-by-absence* (i.e. $\neg a \otimes A \multimap B \otimes a$)
- Inference rules in $\lceil \mathcal{P} \rceil$ are guarded by "witnesses":

$$\begin{array}{lcl} \lceil \mathcal{P} \rceil & = & \forall x, y.\ \mathcal{I}_1^{(x,y)}, \forall x, y, z.\ \mathcal{I}_2^{(x,y,z)} \\ \mathcal{I}_1^{(x,y)} & = & E(x,y) \multimap_{\boxed{I_1^\sharp(x,y)}} P(x,y) \otimes E(x,y) \\ \mathcal{I}_2^{(x,y,z)} & = & E(x,y) \otimes P(y,z) \multimap_{\boxed{I_2^\sharp(x,y,z)}} P(x,z) \otimes E(x,y) \otimes P(y,z) \end{array}$$

- Witnesses $I_1^\sharp(x,y)$ and $I_2^\sharp(x,y,z)$ are just normal predicates.
- Derivations can reach *quiescence*: Inference rules can be applied at most once.

Termination in Assertions

Intuitionistic Linear Logic Sequent Calculus, LV^{obs-}

$$\frac{}{\Gamma; \Delta \longrightarrow \bigotimes \Delta}\ (\text{obs}) \qquad \frac{\Gamma, A; \Delta, A \longrightarrow C}{\Gamma, A; \Delta \longrightarrow C}\ (\text{clone})$$

$$\frac{\Gamma; \Delta \longrightarrow C}{\Gamma; \Delta, 1 \longrightarrow C}\ (1_L) \qquad \frac{\Gamma; \Delta, A_1, A_2 \longrightarrow C}{\Gamma; \Delta, A_1 \otimes A_2 \longrightarrow C}\ (\otimes_L)$$

$$\frac{\Gamma; \Delta, A_i \longrightarrow C}{\Gamma; \Delta, A_1 \& A_2 \longrightarrow C}\ (\&_{L_i}) \qquad \frac{t \in \Sigma \quad \Gamma; \Delta, [t/x]A \longrightarrow C}{\Gamma; \Delta, \forall x.A \longrightarrow C}\ (\forall_L)$$

$$\frac{\Gamma; \Delta, B \longrightarrow C}{\Gamma; \Delta, \Delta', (\bigotimes \Delta') \multimap B \longrightarrow C}\ (\multimap_L) \qquad \boxed{\frac{a \notin \Delta \qquad \Gamma; \Delta, B, a \longrightarrow C}{\Gamma; \Delta, \Delta', (\bigotimes \Delta') \multimap_a B \longrightarrow C}\ (\multimap_L^-)}$$

Exhaustive Retraction

Look at the problem again:

$$\frac{[\![\mathcal{P}]\!]', \mathcal{A}_{\mathcal{P}}; P(2,3), \mathcal{R}_1^{(2,3)}, E(3,4), P(3,4), \mathcal{R}_1^{(3,4)}, P(2,4), \mathcal{R}_2^{(2,3,4)} \longrightarrow C}{[\![\mathcal{P}]\!]', \boxed{\mathcal{A}_{\mathcal{P}}}; \begin{array}{l} \boxed{E(2,3)}, P(2,3), \mathcal{R}_1^{(2,3)}, E(3,4), P(3,4), \\ \mathcal{R}_1^{(3,4)}, P(2,4), \mathcal{R}_2^{(2,3,4)}, \boxed{\tilde{E}(2,3)} \end{array} \longrightarrow C}$$

- The culprit: **Absorption rule** for E was applied before retraction rules $\mathcal{R}_1^{(2,3)}$ and $\mathcal{R}_2^{(2,3,4)}$.
- We need to guarantee that **retraction rules** are exhaustively applied before **absorption rules**.

Exhaustive Retraction

The solution, use "witnesses" and guarded implications:

$$\begin{array}{lcl}\mathcal{P} & = & \boxed{I_1 : \forall x, y.\ E(x,y) \supset P(x,y)} \quad , \quad I_2 : \forall x, y, z.\ E(x,y) \wedge P(y,z) \supset P(x,z) \\ [\![\mathcal{P}]\!] & = & \boxed{\forall x, y.\ \mathcal{I}_1^{(x,y)}} \quad , \quad \forall x, y, z.\ \mathcal{I}_2^{(x,y,z)}\end{array}$$

$$\begin{array}{lcl}\mathcal{I}_1^{(x,y)} & = & E(x,y) \multimap_{I_1^\sharp(x,y)} P(x,y) \otimes E(x,y) \otimes \mathcal{R}_1^{(x,y)} \otimes \boxed{E^\sharp(x,y)} \\ \mathcal{R}_1^{(x,y)} & = & (\tilde{E}(x,y) \otimes I_1^\sharp(x,y) \otimes \boxed{E^\sharp(x,y)} \multimap \tilde{P}(x,y) \otimes \tilde{E}(x,y)) \\ \mathcal{A}_E^{(x,y)} & = & E(x,y) \otimes \tilde{E}(x,y) \multimap_{\boxed{E^\sharp(x,y)}} (E^\sharp(x,y) \multimap 1)\end{array}$$

- Witness $E^\sharp(x,y)$ is added when $\mathcal{I}_1^{(x,y)}$ is applied.
- Witness $E^\sharp(x,y)$ is consumed when $\mathcal{R}_1^{(x,y)}$ is applied.
- $\mathcal{A}_E^{(x,y)}$ cannot be applied as long as there is some $E^\sharp(x,y)$.
- Retraction rules involving $E^\sharp(x,y)$ must be applied before absorption rule $\mathcal{A}_E^{(x,y)}$.

Exhaustive Retraction

The solution, use "witnesses" and guarded implications:

$$\mathcal{P} = \boxed{I_1 : \forall x, y.\ E(x,y) \supset P(x,y)}\ ,\quad I_2 : \forall x, y, z.\ E(x,y) \wedge P(y,z) \supset P(x,z)$$

$$[\![\mathcal{P}]\!] = \boxed{\forall x, y.\ \mathcal{I}_1^{(x,y)}}\ ,\quad \forall x, y, z.\ \mathcal{I}_2^{(x,y,z)}$$

$$\mathcal{I}_1^{(x,y)} = E(x,y) \multimap_{I_1^\sharp(x,y)} P(x,y) \otimes E(x,y) \otimes \mathcal{R}_1^{(x,y)} \otimes E^\sharp(x,y)$$

$$\mathcal{R}_1^{(x,y)} = (\tilde{E}(x,y) \otimes I_1^\sharp(x,y) \otimes E^\sharp(x,y) \multimap \tilde{P}(x,y) \otimes \tilde{E}(x,y))$$

$$\mathcal{A}_E^{(x,y)} = E(x,y) \otimes \tilde{E}(x,y) \multimap_{E^\sharp(x,y)} \boxed{(E^\sharp(x,y) \multimap 1)}$$

$(E^\sharp(x,y) \multimap 1)$ eliminates the trailing $E^\sharp(x,y)$ produced by $\mathcal{A}_E^{(x,y)}$ itself.

$$\frac{a \notin \Delta \qquad \Gamma; \Delta, B, \boxed{a} \longrightarrow C}{\Gamma; \Delta, \Delta', (\bigotimes \Delta') \multimap_a B \longrightarrow C}\ (\multimap_L^-)$$

Exhaustive Retraction

The solution, use “witnesses” and guarded implications:

$$\mathcal{P} = \boxed{I_1 : \forall x, y.\ E(x,y) \supset P(x,y)} \quad , \quad I_2 : \forall x, y, z.\ E(x,y) \wedge P(y,z) \supset P(x,z)$$

$$[\![\mathcal{P}]\!] = \boxed{\forall x, y.\ \mathcal{I}_1^{(x,y)}} \quad , \quad \forall x, y, z.\ \mathcal{I}_2^{(x,y,z)}$$

$$\mathcal{I}_1^{(x,y)} = E(x,y) \multimap_{I_1^\sharp(x,y)} P(x,y) \otimes E(x,y) \otimes \mathcal{R}_1^{(x,y)} \otimes E^\sharp(x,y)$$

$$\mathcal{R}_1^{(x,y)} = (\tilde{E}(x,y) \otimes \boxed{I_1^\sharp(x,y)} \otimes E^\sharp(x,y) \multimap \tilde{P}(x,y) \otimes \tilde{E}(x,y))$$

$$\mathcal{A}_E^{(x,y)} = E(x,y) \otimes \tilde{E}(x,y) \multimap_{E^\sharp(x,y)} (E^\sharp(x,y) \multimap 1)$$

- $I_1^\sharp(x,y)$ is comsumed when retraction rule $\mathcal{R}_1^{(x,y)}$ is applied.
- This allows $E(x,y)$ to be re-asserted in the future.

The Road Map

The IID^{Full} Interpretation

$\llbracket \mathcal{P} \rrbracket$ is defined as follows:

$$
\begin{array}{rl}
\text{Fact} & \llbracket p(\vec{t}) \rrbracket = p(\vec{t}) \\
\text{Program} & \left\{ \begin{array}{l} \llbracket D,\ \mathcal{P} \rrbracket = \llbracket D \rrbracket, \llbracket \mathcal{P} \rrbracket \\ \llbracket . \rrbracket = . \end{array} \right. \\
\text{Conjunction} & \llbracket b \wedge B \rrbracket = \llbracket b \rrbracket \otimes \llbracket B \rrbracket \\
\text{Clause} & \left\{ \begin{array}{l} \llbracket r : \forall \vec{x}.\ B \supset h \rrbracket = \\ \qquad \forall \vec{x}.\ \llbracket B \rrbracket \multimap_{r^\sharp(\vec{x})} \llbracket h \rrbracket \otimes \llbracket B \rrbracket \otimes \llbracket B^\sharp \rrbracket \otimes Ret(B, B, r^\sharp(\vec{x}), h) \end{array} \right. \\
\text{Fact base} & \left\{ \begin{array}{l} \llbracket \mathcal{B}, a \rrbracket = \llbracket \mathcal{B} \rrbracket, \llbracket a \rrbracket \\ \llbracket \emptyset \rrbracket = . \end{array} \right.
\end{array}
$$

where

$$Ret(b \wedge B', B, r^\sharp(\vec{x}), h) = (\llbracket \tilde{b} \rrbracket \otimes r^\sharp(\vec{x}) \otimes \llbracket B^\sharp \rrbracket \multimap \llbracket \tilde{h} \rrbracket \otimes \llbracket \tilde{b} \rrbracket)\ \&\ Ret(B', B, r^\sharp(\vec{x}), h)$$

$$Ret(b, B, r^\sharp(\vec{x}), h) = (\llbracket \tilde{b} \rrbracket \otimes r^\sharp(\vec{x}) \otimes \llbracket B^\sharp \rrbracket \multimap \llbracket \tilde{h} \rrbracket \otimes \llbracket \tilde{b} \rrbracket)$$

The IID^{Full} Interpretation

- Dynamic (assertion and retraction) Datalog state transition system:

$$\frac{a \notin \Delta \quad [\![\mathcal{P}]\!], \mathcal{A}_{\mathcal{P}}; \Delta, a \longrightarrow \bigotimes \Delta' \qquad Quiescent(\Delta', ([\![\mathcal{P}]\!], \mathcal{A}_{\mathcal{P}}))}{\Delta \stackrel{+a}{\Longrightarrow}{}^{LL}_{[\![\mathcal{P}]\!]} \Delta'} \ (Infer)$$

$$\frac{a \in \Delta \quad [\![\mathcal{P}]\!], \mathcal{A}_{\mathcal{P}}; \Delta, \tilde{a} \longrightarrow \bigotimes \Delta' \qquad Quiescent(\Delta', ([\![\mathcal{P}]\!], \mathcal{A}_{\mathcal{P}}))}{\Delta \stackrel{-a}{\Longrightarrow}{}^{LL}_{[\![\mathcal{P}]\!]} \Delta'} \ (Retract)$$

- Correctness 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 $\Delta_1 = [\![\mathcal{P}(\mathcal{B}_1)]\!]$ and $\Delta_2 = [\![\mathcal{P}(\mathcal{B}_2)]\!]$, then we have the following:

$$(\Delta_1, \Delta_1^{\mathcal{R}}, \Delta_1^{\sharp}) \stackrel{\alpha}{\Longrightarrow}{}^{LL}_{[\![\mathcal{P}]\!]} (\Delta_2, \Delta_2^{\mathcal{R}}, \Delta_2^{\sharp}) \text{ iff } \mathcal{P}(\mathcal{B}_1) \stackrel{\alpha}{\Longrightarrow}_{\mathcal{P}} \mathcal{P}(\mathcal{B}_2)$$
where $\mathcal{P}(\mathcal{B}) = \{p(\vec{t}) \mid \mathcal{P}, \mathcal{B} \vdash p(\vec{t})\}$

Note if $\alpha = +a$ we have $\mathcal{B}_2 = \mathcal{B}_1, a$. If $\alpha = -a$ we have $\mathcal{B}_1 = \mathcal{B}_2, a$.

- See Tech-Report for details of the proofs.

The Road Map

Discussions

- **Cycles** are terminating, thanks to guarded implications:

$$\begin{aligned} \mathcal{P} &= I_1 : \forall x, y.\ P(x,y) \supset P(y,x) \\ [\![\mathcal{P}]\!] &= P(x,y) \multimap_{\boxed{I_1^\sharp(x,y)}} P(y,x) \otimes P(x,y) \otimes P^\sharp(x,y) \otimes .. \end{aligned}$$

- **Re-assertions** needed during retraction:
  - Retraction may be excessive.
  - Quiescence only reached after some reassertion of facts:
  - Cause: Multiple inference dependencies, e.g.

$$\begin{aligned} I_1 &: \forall x.\ A(x) \supset C(x) \\ I_2 &: \forall x.\ B(x) \supset C(x) \\ I_3 &: \forall x.\ C(x) \supset D(x) \end{aligned}$$

- See Tech-Report for more details.

Future works and conclusion

- We formalized a Linear Logic interpretation of Datalog:
  - Internally supports assertion and retraction (dynamic updates on base facts)
  - Soundness and completeness
- Future works:
  - Improving the IID^{Full} interpretation: No reassertion.
  - Implementation based on higher order multiset rewritings

A simplified example of our Linear Logic Interpretation

Datalog program $\mathcal{P}$:

$$\begin{array}{ll} P(x,y) \text{ :- } E(x,y). & - r_1 \\ P(x,z) \text{ :- } E(x,y), P(y,z). & - r_2 \end{array}$$

Linear logic interpretation $[\![\mathcal{P}]\!]$:

$$[\![\mathcal{P}]\!] = \forall x,y.\mathcal{I}_1^{(x,y)} \ , \ \forall x,y,z.\mathcal{I}_2^{(x,y,z)}$$

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

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

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

Example of Retraction, as Forward-Chaining Proof Search

Intutionistic Linear Logic Sequent Judgements: $\Gamma; \Delta \longrightarrow C$
Retraction, e.g. removal of fact $E(2,3)$:

$$\llbracket \mathcal{P} \rrbracket, \boxed{\mathcal{A}_\mathcal{P}}; E(3,4), P(3,4), \mathcal{R}_1^{(3,4)} \longrightarrow C$$

$$\llbracket \mathcal{P} \rrbracket, \boxed{\mathcal{A}_\mathcal{P}}; \boxed{E(2,3)}, E(3,4), P(3,4), \mathcal{R}_1^{(3,4)}, \boxed{\tilde{E}(2,3)} \longrightarrow C$$

$$\llbracket \mathcal{P} \rrbracket, \boxed{\mathcal{A}_\mathcal{P}}; E(2,3), \boxed{P(2,3)}, E(3,4), P(3,4), \mathcal{R}_1^{(3,4)}, \tilde{E}(2,3), \boxed{\tilde{P}(2,3)} \longrightarrow C$$

$$\llbracket \mathcal{P} \rrbracket, \mathcal{A}_\mathcal{P}; E(2,3), P(2,3), \boxed{\mathcal{R}_1^{(2,3)}}, E(3,4), P(3,4), \mathcal{R}_1^{(3,4)}, \boxed{\tilde{E}(2,3)} \longrightarrow C$$

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

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

M. P. Ashley-Rollman, P. Lee, S. C. Goldstein, P. Pillai, and J. D. Campbell.
A language for large ensembles of independently executing nodes.
In *Proc. of ICLP'09*, volume 5649/2009, pages 265–280. Springer-Verlag, 2009.

F. Cruz, M. P. Ashley-Rollman, S. C. Goldstein, Ricardo Rocha, and F. Pfenning.

Bottom-up logic programming for multicores.
In Vítor Santos Costa, editor, *Proc. of DAMP 2012 - Short Papers*. ACM Digital Library, January 2012.

I. Cervesato.
NEXCEL, a Deductive Spreadsheet.
The Knowledge Engineering Review, 22:221–236, 2007.

A. Gupta, I. S. Mumick, and V. S. Subrahmanian.
Maintaining views incrementally.
In *Proc. of SIGMOD'93*, pages 157–166. ACM, 1993.

S. Grumbach and F. Wang.
Netlog, a rule-based language for distributed programming.
In *Proc. of PADL'10*, volume 5937/2010, pages 88–103. Springer-Verlag, 2010.

B. T. Loo, T. Condie, M. Garofalakis, D. E. Gay, J. M. Hellerstein, P. Maniatis, R. Ramakrishnan, T. Roscoe, and I. Stoica.
Declarative networking: language, execution and optimization.
In *Proc. of SIGMOD '06*, pages 97–108. ACM, 2006.