An Existential Rule Framework for Computing Why-Provenance On-Demand for Datalog

Campanion

Ali Elhalawati, Markus Krötzsch, and Stephan Mennicke

Knowledge-Based Systems Group, TU Dresden, Germany
{fistname.lastname}@tu-dresden.de

Abstract. This short paper accompanies a submission of the paper with the same title as this document. It contains a holistic example of all the notions. We explain in detail how the different stages of our approach look like for the particular example given in the paper.

1 Databases and Datalog

Consider the database $\mathcal{D} = \{e(1,2), e(2,2), e(2,3), e(4,3)\}$ which may be interpreted as a graph with set of nodes $\{1,2,3,4\}$ and edge relation $\{(1,2),(2,2),(2,3),(4,3)\}$. Note, $\mathcal{D}$ encodes a directed graph. Furthermore, let Σ be the rule set containing the following rules:

$$\begin{aligned} \rho_1: & \quad e(x,y) \to t(x,y) \\ \rho_2: & \quad e(x,z) \wedge t(z,y) \to t(x,y) \end{aligned}$$

Σ is a classic transitive closure of the predicate e. Applying T_Σ iteratively on $\mathcal{D}$ yields additional t-atoms $t(1,2), t(1,3), t(2,2), t(2,3), t(4,3)$, such that $\Sigma(\mathcal{D}) = \mathcal{D} \cup \{t(1,2), t(1,3), t(2,2), t(2,3), t(4,3)\}$. For instance, $t(1,2)$ can be obtained from applying ρ_1 for $\sigma_1 = \{x \mapsto 1, y \mapsto 2\}$ (i.e., for rule instance $e(1,2) \to t(1,2)$). Another possibility to derive the same fact is by ρ_2 for match $\sigma_2 = \{x \mapsto 1, y \mapsto 2, z \mapsto 2\}$ (i.e., for rule instance $e(1,2) \wedge t(2,2) \to t(1,2)$). Of course, this application requires us to have also derived $t(2,2)$, which can be done by ρ_1 and $\sigma_3 = \{x \mapsto 2, y \mapsto 2\}$.

2 The Graph of Rule Instances

The set of all rule instances for rule set Σ and database $\mathcal{D}$ above is the following:

$$e(1,2) \to t(1,2) \tag{1}$$
$$e(2,2) \to t(2,2) \tag{2}$$
$$e(2,3) \to t(2,3) \tag{3}$$
$$e(4,3) \to t(4,3) \tag{4}$$
$$e(1,2) \wedge t(2,2) \to t(1,2) \tag{5}$$
$$e(1,2) \wedge t(2,3) \to t(1,3) \tag{6}$$
$$e(2,2) \wedge t(2,2) \to t(2,2) \tag{7}$$
$$e(2,2) \wedge t(2,3) \to t(2,3) \tag{8}$$

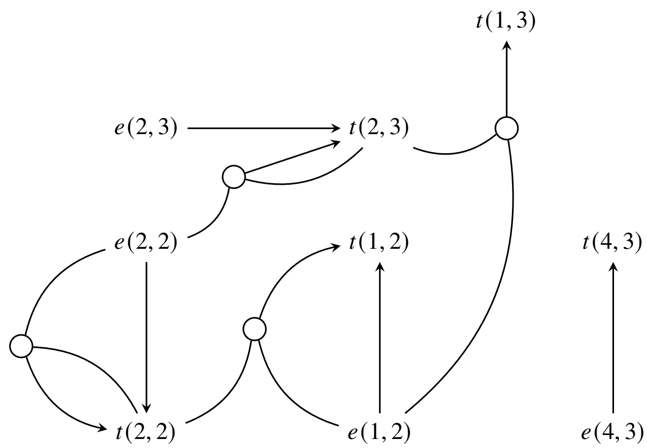

Fig. 1: The Graph of Rule Instances of Σ and $\mathcal{D}$

Each atom $A \in \Sigma(\mathcal{D})$ is a node in the set of all nodes of the graph of rule instances $\mathit{GRI}(\Sigma, \mathcal{D}) = (\Sigma(\mathcal{D}), \mathcal{E}, \mathsf{tip}, \mathsf{tail}, id_{\Sigma(\mathcal{D})})$. The hyperedges ($\mathcal{E}$) are determined by the eight rule instances above. For instance, $e = \{e(1,2), t(1,2)\} \in \mathcal{E}$ with $\mathsf{tip}(e) = t(1,2)$ and $\mathsf{tail}(e) = \{e(1,2)\}$ due to (1). Fig. 1 shows a depiction of the full graph, including all the edges. If an edge has a single tail-node, just like e above, we depict it as a directed arc, having the arrow tip pointing to the tip of the edge. Edges having more than one tail-node use a small circle to join the tail-nodes via undirected edges. For each such circle, there is exactly one directed edge, pointing to the tip of the edge.

3 Proof Trees and Why-Provenance

For atoms $t(1,3)$, $t(2,3)$, and $t(4,3)$ we have a single proof tree for each of them, all being depicted in Fig. 2. Note how the proof tree for $t(2,3)$ in Fig. 2 (a) is a subtree of the one in Fig. 2 (b). Let us denote the three proof trees by $T_{(a)}$, $T_{(b)}$, and $T_{(c)}$, each referring to the proof tree behind the respective graphical representation Fig. 2. Recall that α applied to a proof tree T returns the set of labels of its leaf nodes. Then $\alpha(T_{(a)}) = \{e(2,2)\}$, $\alpha(T_{(b)}) = \{e(1,2), e(2,3)\}$, and $\alpha(T_{(c)}) = \{e(4,3)\}$.

The proof trees of Fig. 2 are even proper subgraphs of the graph of rule instances depicted in Fig. 1. This is, however, not generally the case. Take, for instance, the atom $t(1,2)$, which can be directly derived via rule instance (1), depicted in Fig. 3 (a). Rule instance (5) is depicted in Fig. 3 (b). But there are more proof trees since $t(2,2)$ has alternative proof trees. In particular, $T_{(b)}$ uses rule instance (2) as a subtree while Fig. 3 (c) shows a proof tree incorporating (7) as a subtree. This last proof tree contains two nodes with label $t(2,2)$ and two nodes with label $e(2,2)$. Observe that the unfolding step performed between Fig. 3 (b) and (c) can be performed arbitrarily often, yielding an infinite set of proof trees for $t(1,2)$. However, there is not more information to gain when it comes to Why-provenance.

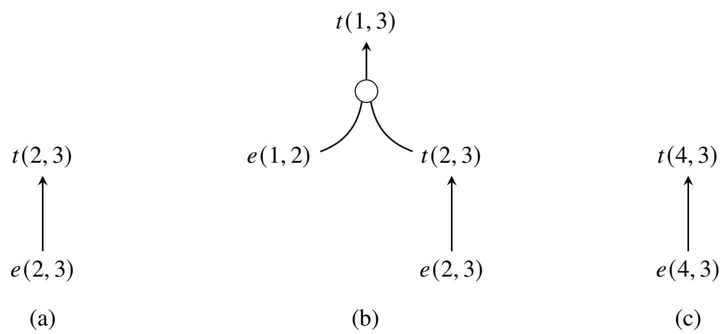

Fig. 2: Simple Proof Trees for (a) $t(2,3)$, (b) $t(1,3)$, and (c) $t(4,3)$

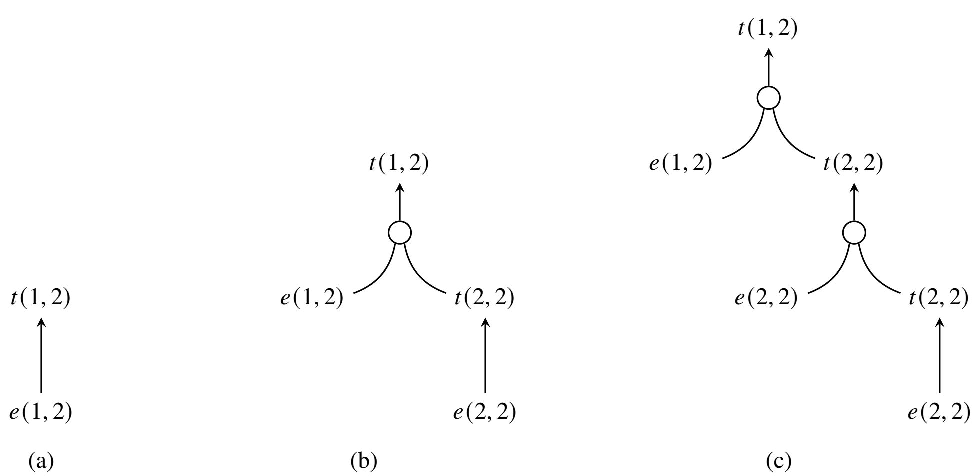

Fig. 3: A Selection of Proof Trees for $t(1,2)$

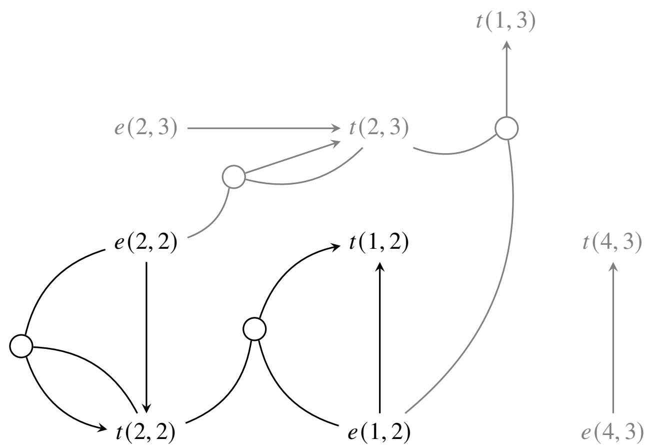

Fig. 4: The Downward-Closure $t(1,2)^{\downarrow}$

Note, $\alpha(T_{(a)}) = \{e(1,2)\}$ and $\alpha(T_{(b)}) = \{e(1,2), e(2,2)\} = \alpha(T_{(c)})$. Therefore, $w_1 = \{e(1,2)\}$ and $w_2 = \{e(1,2), e(2,2)\}$ are witnesses for $t(1,2)$. In fact, $\mathbf{Why}(t(1,2), \Sigma, \mathcal{D}) = \{w_1, w_2\}$ because every other proof tree just appends $e(2,2)$ to w_2 several times, yielding w_2 again.

4 Rule-Based Downward-Closure

We stick with atom $t(1,2) \in \Sigma(\mathcal{D})$ and prove the downward-closure $t(1,2)^{\downarrow}$ in Fig. 4 where all parts of the graph of rule instances that do not belong to the downward-closure are grayed-out.

The following rules construct the nodes of the graph of rule instances, collected as set Σ^r:

$$\begin{aligned} \rho_e :&\quad e(x,y) \to \exists v.\, \widehat{e}(x,y,v) \\ \rho_t :&\quad t(x,y) \to \exists v.\, \widehat{t}(x,y,v) \end{aligned}$$

The trigger rule for $t(1,2)$ is:

$$trig(t(1,2)) : \widehat{t}(1,2,v) \to G(v).$$

The remaining rules to construct the downward closure are collected in $\Sigma^{\diamond}$:

$$\begin{aligned} \widehat{\rho}_0 :&\quad \widehat{e}(x,y,v) \to \mathcal{E}_1(v) \\ \widehat{\rho}_1 :&\quad G(v) \wedge \widehat{t}(x,y,v) \wedge \widehat{e}(x,y,w) \to \mathcal{E}_2(v,w) \wedge G(w) \\ \widehat{\rho}_2 :&\quad G(v) \wedge \widehat{t}(x,y,u) \wedge \widehat{e}(x,z,v) \wedge \widehat{t}(z,y,w) \to \mathcal{E}_3(u,v,w) \wedge G(v) \wedge G(w) \end{aligned}$$

All the rules we just mentioned are collected in the rule set $\Sigma^{\star} = \Sigma \cup \Sigma^r \cup \Sigma^{\diamond} \cup \{trig(t(1,2))\}$, such that $\Sigma^{\star}(\mathcal{D}) = \Sigma(\mathcal{D}) \cup \mathcal{R}$ where

$$\mathcal{R} = \left\{ \begin{array}{l} G(v_{t(1,2)}), G(v_{e(1,2)}), G(v_{t(2,2)}), G(v_{e(2,2)}), \mathcal{E}_1(v_{e(1,2)}), \\ \mathcal{E}_1(v_{e(2,2)}), \mathcal{E}_2(v_{t(1,2)}, v_{e(1,2)}), \mathcal{E}_2(v_{t(2,2)}, v_{e(2,2)}), \\ \mathcal{E}_3(v_{t(1,2)}, v_{e(1,2)}, v_{t(2,2)}) \end{array} \right\}.$$

5 System of Equations On-Demand

By querying $\mathcal{R}$, we derive the following system of equations on-demand:

$$v_{e(1,2)} = \{\{\alpha(e(1,2))\}\} \tag{9}$$
$$v_{e(2,2)} = \{\{\alpha(e(2,2))\}\} \tag{10}$$
$$v_{t(1,2)} = v_{e(1,2)} \oplus (v_{e(1,2)} \otimes v_{t(2,2)}) \tag{11}$$
$$v_{t(2,2)} = v_{e(2,2)} \oplus (v_{e(2,2)} \otimes v_{t(2,2)}) \tag{12}$$

Assuming $\alpha(e(1,2)) = i$ and $\alpha(e(2,2)) = j$ and interpreting the system over the Why-semiring, we obtain the following solution

$$\beta = \left\{ \begin{array}{l} v_{e(1,2)} \mapsto \{\{i\}\} \\ v_{e(2,2)} \mapsto \{\{j\}\} \\ v_{t(1,2)} \mapsto \{\{i\}, \{i, j\}\} \\ v_{t(2,2)} \mapsto \{\{j\}\} \end{array} \right\}$$

We verify β for $v_{t(1,2)}$, meaning that $\beta(v_{t(1,2)}) = \beta(v_{e(1,2)}) \cup (\beta(v_{e(1,2)}) \Cup \beta(v_{t(2,2)})) = \{\{i\}\} \cup (\{\{i\}\} \Cup \{\{j\}\}) = \{\{i\}\} \cup \{\{i, j\}\}$.

6 Datalog(S) Realization

The only aspect of the original program Σ, our Datalog(S) program depends on is the maximal number of atoms in a single rule of Σ. Here, we have $k = 3$, meaning we use program $\Sigma^3_{\mathbf{Why}}$:

$$\begin{array}{lr} \epsilon_1: & \mathcal{E}_1(v) \rightarrow prov(v, \{v\}) \\ \epsilon_2: & \mathcal{E}_2(v, w) \wedge prov(w, X) \rightarrow prov(v, X) \\ \epsilon_3: & \mathcal{E}_3(u, v, w) \wedge prov(v, X) \wedge prov(w, Y) \rightarrow prov(u, X \bigcup Y) \end{array}$$

Considering $(\Sigma^3_{\mathbf{Why}} \cup \Sigma^\star)(\mathcal{D})$ we get $\Sigma(\mathcal{D}) \cup \mathcal{R} \cup \mathcal{P}$ where

$$\mathcal{P} = \left\{ \begin{array}{l} prov(v_{e(1,2)}, \{v_{e(1,2)}\}), prov(v_{e(2,2)}, \{v_{e(2,2)}\}), prov(v_{t(1,2)}, \{v_{e(1,2)}\}), \\ prov(v_{t(1,2)}, \{v_{e(1,2)}, v_{e(2,2)}\}), prov(v_{t(2,2)}, \{v_{e(2,2)}\}) \end{array} \right\}.$$

When replacing v_A inside the sets by $\alpha(A)$, we obtain

$$\mathcal{P}' = \left\{ \begin{array}{l} prov(v_{e(1,2)}, \{i\}), prov(v_{e(2,2)}, \{j\}), prov(v_{t(1,2)}, \{i\}), \\ prov(v_{t(1,2)}, \{i, j\}), prov(v_{t(2,2)}, \{j\}) \end{array} \right\},$$

which perfectly reflects on solution β above.