Provenance Analyzer: Exploring Provenance Semantics with Logic Rules

by Saumen Dey, Sean Riddle, and Bertram Ludäscher
Overview

- The Problem
  - The Open Provenance Model
  - Toward a Temporal Semantics
  - Provenance Analyzer prototype

- Approach
  - Deductive
  - Abduce Time
  - Abduce Partial Order
Open Provenance Model

- Graph model, $N = P$ (processes) $U A$ (artifacts)
- Edge types
  - $A$ genby $P$ (artifact $A$ was generated by process $P$)
  - $P_1$ informedBy $P_2$ (process $P_1$ influenced the execution of process $P_2$)
  - $A_1$ derivedFrom $A_2$ (artifact $A_1$ used $A_2$ in its creation)
  - $P$ used $A$ (process $P$ used artifact $A$)
- Time is optional annotation
- Legality
  - Each datum must have one immediate creating process
  - Broken use-generate-derive triangles [1] are not allowed
- Few completeness restrictions (a trace does not have to say very much – it is difficult to tell at a glance how detailed/restrictive a trace is)
Each artifact can only be precisely generated by one process

Each precise derived from edge is part of a use-generate-derive triangle
Example

- Indicates edge has a role, used in algorithm in [1], use-generate-derive triangles
Example

- ! Indicates edge has a role
- Forms a use-generate-derive triangle, which encodes additional semantics according to [1]

P used B to create A
Many Different Possible Worlds

(5 models elided)
Deductive Algorithm

- From Kwasnikowska et al.[1]
- Graph patterns $\rightarrow$ Temporal constraints

\[
\begin{align*}
(Ax\ 2) & \quad \text{begin}(P) \preceq \text{create}(A) \preceq \text{end}(P) \\
\end{align*}
\]

Figure 7: Characterization of temporal inference.
Implementation

- Implemented in ASP (answer set programming) system
  DLV: dlvsystem.com
- Instrumented/interconnected with Python
- Sets of rules run in different combinations:
  - Definitions of axioms in different terms (deductive, abductive over different structures)
  - Definitions of OPM legality
  - Rules to generate linear extensions of a partial order (i.e., total orders that obey the partial order)
ASP Example

- a :- not b.
  b :- not a.

- Evaluate bodies of rules wrt model; does it produce same model?

- Is \{a\} a model? Yes.
  - a :- true
  - b :- false

- Is \{b\} a model? Yes.
  - a :- false
  - b :- true

- Is \{a, b\} a model? No.
  - a :- false
  - b :- false
Implementation - Deductive

- Implemented axioms and temporal inference rules from [1]
- Deductive generation of constraints from graph structure

leq(beginP, createA) :- pGenBy(a, p).
leq(createA, endP) :- pGenBy(a, p).

Axiom 2 prototype implementation

\[
\begin{align*}
\text{AX 2} & \quad \frac{\text{begin}(P) \preceq \text{create}(A) \preceq \text{end}(P)}{}
\end{align*}
\]

Axiom 2 as given by [1]
Implementation – Abduce Time

- Abductive approach guesses mapping of events to time points and throws out violators.

Axiom 2 as given by [1]

\[
\begin{align*}
\text{begin}(P) & \leq \text{create}(A) \leq \text{end}(P) \\
\end{align*}
\]

Axiom 2 prototype implementation
Abductive approach guesses mapping of events to events and then throws out all that do not describe a partial order (or total order if desired).

Axiom 2 as given by [1]

\[
\text{begin}(P) \preceq \text{create}(A) \preceq \text{end}(P)
\]

\(\text{after} \) and \(\text{use} \) are used to model actions. Formally:

\[
\text{begin}(P) \preceq \text{create}(A) \preceq \text{end}(P)
\]

Axiom 2 prototype implementation

\[
\text{false} :-
\begin{align*}
    & \text{pGenBy}(a, p), \\
    & \text{after} \left( \text{begin}(P), \text{create}(A) \right).
\end{align*}
\]

\[
\text{false} :-
\begin{align*}
    & \text{pGenBy}(a, p), \\
    & \text{after} \left( \text{create}(A), \text{end}(P) \right).
\end{align*}
\]
Abduce Time vs. Abduce Partial Order

- Abduce Time faster on small examples, sensitive to definition of time points (too many can cause drastic slowdown)
- Abduce Time allows for constraints that depend on time, e.g., creation events can only occur with a certain maximum frequency
Design

- Inferences 1-9B & Extended Axioms
- Events Ev
- Axioms 1-3,8
- Axioms 4-7

- ComputePO
- GuessPO
- GuessTime

- One partial order
- All possible partial orders
- All possible total orders

- ComputeTO
- TOFilter
- TOFilter

- All valid total orders of events

- Deduce Partial Order
- Abduce Partial Order
- Abduce Time
Conclusions and Future Work

- Provides measure of constrained-ness (how many total orders of events correspond to this trace?)
- Can query what is true in some/all models (analogous to brave/cautious reasoning in LP)
- Allows experimentation with axioms
- Future work
  - Combinations of axioms that describe easily characterizable subsets of temporal constraints (eg., no-use inequalities)
  - Increase performance with subsumption
Provenance Analyzer

- Available at: http://code.google.com/p/provenance-analyzer

- Works Cited


- Acknowledgements

  - Work supported in part by NSF awards DGE-0841297, OCI-0830944, and IIS-1118088.