Is provenance logical?

[Invited Keynote]

James Cheney
Laboratory for Foundations of Computer Science
University of Edinburgh
jchene@inf.ed.ac.uk

ABSTRACT

Research on provenance in databases (or other settings) sometimes has an arbitrary flavor. Once we abandon the classical semantics of queries there is a large design space for alternative semantics that could provide some useful provenance information, but there is little guidance for how to explore this space or justify or compare different proposals. Topics from mathematical or philosophical logic could be used as a way of inspiring, justifying or comparing different approaches to provenance in databases. This paper and invited talk will present several topics in logic that may be less familiar to database researchers and that could bear upon provenance techniques. These areas include nonclassical logics (e.g. relevance logic), algebraic logic (cylindric algebras), substructural logic (e.g. linear logic) and logics of knowledge, belief or causality.

1. INTRODUCTION

Logic is closely connected to databases, and to many other areas of computer science. For example, ideas from finite model theory, logic programming, modal logic, games, and connections between logics, automata and complexity classes have been applied extensively to problems in databases. However, some areas of logic, particularly areas of nonclassical and substructural logic, have not been widely explored in connection to databases. At the same time, in the last several years the topic of provenance has been studied extensively in databases. Essentially, provenance techniques for queries seek to provide auxiliary information that users or auditors can use to judge the correctness, quality, integrity or reliability of answers produced by a query. However, once we choose to abandon the traditional semantics for relational operations, there is a large design space and relatively little guidance as to which semantics is best. We have argued elsewhere [9] that provenance techniques need to be developed upon foundations that are just as solid as those underlying ordinary databases, programming languages, and logics.

The purpose of this invited talk is to argue that ideas from certain areas of logic that may be less familiar to database researchers but might provide justification or inspiration for different forms of provenance. Here are just a few possibilities which we shall explore:

1. Nonclassical logics such as intuitionistic [36] or relevance logic [1] are based on model theories that take the notion of evidence into account. In fact, why-provenance [6] seems to correspond closely to Urquhart’s classical semantics for relevance logic [38].

2. Algebraizations of logic, such as cylindric algebras as studied by Henkin, Monk and Tarski [22], could be used as an alternative class of annotations, analogous to the semiring-valued relational model introduced by Green et al. [18]. In fact, cylindric algebra applications to databases have been studied at least since Imielinski and Lipski [25], but the idea of using cylindric algebras, or other alternatives to semirings, for provenance does not appear to have been considered to date.

3. Substructural logics such as linear logic [17] have been studied extensively in programming languages for memory management [10], resource-bounded computation [23], and concurrency [29]. Some work on database query languages also employs this idea [35], but to our knowledge substructural logics have not been applied to related problems such as identity-tracking in provenance.

4. Logics of knowledge, belief and causality have been studied extensively in artificial intelligence [34, 20, 13, 19] and, recently, mathematical models of causality have been explored in connection to provenance [7, 31, 30, 32]. Some of these techniques are closely related to ideas from modal logic or incomplete information databases [24], which have also been used fruitfully for data integration and cleaning.

2. RELEVANCE LOGIC

In classical logic, and in much of classical database theory, an assertion is either true or false, and its truth value is not necessarily explained by evidence or proof. This view is embodied in Tarski’s semantics for first-order logic. During the great foundational investigations of the twentieth century, some logicians such as Brouwer rejected this classical approach to truth because it does not account for proof. Since then, intuitionist or constructive approaches to logic and mathematics have been developed (see e.g. [36]) which read formulas as statements about provability or constructibility.
and explicitly reject the law of excluded middle \( P \lor \neg P \) which can be read as “We either know a proof of \( P \) or we know a proof of \( \neg P \).

Philosophers and logicians have also explored many other nonclassical logics, including varieties of modal logic which have been widely studied in connection to databases and finite model theory, but also including logics, such as relevance logic, that explore different interpretations of logical connectives and have not (to my knowledge) been explored in connection to databases. Both nonclassical logics and provenance are often motivated by the need to provide evidence explaining why a given result is true, beyond the bare truth-value. So, there is some reason to believe that concepts from nonclassical logics could be useful for understanding provenance.

Some forms of provenance, particularly lineage \([12]\) or why-provenance \([6]\), are explicitly defined as providing a witness that gives sufficient information to explain why a given tuple is in the result of a query. Reading a positive query \( Q \) as a collection of conjunctive queries:

\[
A(x) \quad := \quad R_1(y), \ldots, R_n(z) \\
\vdots \quad \vdots \\
A(x) \quad := \quad R_1'(y'), \ldots, R_m'(z')
\]

one can think of lineage as a set including all of the input tuples that are used in some Prolog-style proof tree, or derivation, that proves that the result tuple holds. For a flat UQ, this simply means that for any output tuple \( A(\bar{a}) \), the lineage includes all subsets of input tuples that provide an instance of the right-hand sides.

\[
Lin(Q, I, \bar{a}) = \bigcup \{ \theta(B) \mid (A(\bar{a}) \quad := \quad B) \in Q, \theta(B) \subseteq I \}
\]

This definition is, however, syntactic: two queries may have the same extensional semantics while having different provenance behavior, because one may syntactically use an input tuple that is not actually necessary for the result to be derived. Cui et al. \([12]\) did show that for queries without self-joins, equivalent queries have equivalent lineage, but in general this is not the case \([8]\). For example, consider the two queries:

\[
A(x, y) \quad := \quad R(x, y). \\
B(x, y) \quad := \quad R(x, y), R(x, z).
\]

These two queries are equivalent, because the \( R(x, z) \) atom is redundant, and can be eliminated by minification. Nevertheless, the two queries have different lineage. For example, if \( R = \{(1, 2), (1, 3)\} \) then the lineage of \( A(1, 2) \) is \( \{R(1, 2)\} \) while the lineage of \( B \) is \( \{R(1, 2), R(1, 3)\} \).

Similarly, why-provenance can be thought of as a set of sets of input tuples that includes all of the syntactic witnesses to the output’s existence. We can define this as follows:

\[
Why(Q, I, \bar{a}) = \{ \theta(B) \mid (A(\bar{a}) \quad := \quad B) \in Q, \theta(B) \subseteq I \}
\]

Evidently, this is also a syntactic definition, and the same counterexample suffices to show that equivalent queries can have different why-provenance.

One natural response is to refine these definitions to remove the dependence on syntax. This has been explored in work on why-provenance by considering only minimal witnesses:

\[
MWit(Q, I, \bar{a}) = \{ J \subseteq I \mid J \text{ minimal}, A(\bar{a}) \in [Q]I \}
\]

Note that this definition is formulated in terms of the semantics of the query, not its syntax.

Another alternative, which I explore in the talk, is to adopt a different view of the semantics of a query that retains a connection to a logic, but the logic is relevance logic \([1, 38]\), not classical logic.

3. Algebraic Logic

There is an obvious tension between augmenting the semantics of relational queries to provide additional useful information, and the need to retain algebraic identities needed for optimization. One way to resolve this tension may be to identify compelling spaces of abstractions of query behavior, which still retain enough similarity to the usual semantics that existing optimization infrastructure can still be used.

Initial work on lineage, why- and where-provenance \([12, 6]\) sought to identify special cases where a proposed provenance semantics was compatible with classical optimization rules, and the more recent how-provenance or semiring-valued relational model developed by Green et al. \([18]\) pushed this idea much further. To me, the central contribution of the semiring-valued model is that it asks a fundamentally different question than previous work. Instead of taking the standard semantics of relational queries as given, defining an ad hoc provenance model, and then picking up the pieces when the model is incompatible with the usual semantics, it fixes an annotation-propagating semantics for relational algebra where the annotations are held abstract and then asks what structure and equational laws the annotations need to have in order to satisfy typical equational laws used to optimize relational queries. Specifically, the annotations need to form a commutative semiring, and if we further require the standard set-valued semantics the annotations need to form a distributive lattice.

In fact, the underlying idea of using an abstract structure such as a lattice, Boolean algebra, or semiring for the “truth values” has been explored already in a number of other settings. Udrea et al. \([37]\) actually used this technique for annotated RDF (the conference version of the paper actually appeared in 2006). Annotated semantics for logic programs was considered much earlier, already in the 1990s \([27]\). Indeed, the key idea of abstract truth values can be seen in Boolean-valued models of set theory \([2]\), originally developed as a generalization of Cohen’s famous proofs of the independence of the Axiom of Choice and Continuum Hypothesis from Zermelo-Fraenkel set theory. Of course, none of this prior work drew an explicit connection to provenance in databases, which has only become a concern recently.

Perhaps surprisingly, other provenance models including lineage and why-provenance, as well as other nonstandard interpretations of relational algebra, turn out to be instances of the semiring-valued model. This observation suggests a compelling class of models of provenance corresponding to semirings, including a “most general form of provenance”, corresponding to the free semiring \([\mathbb{K}[X]]\) generated by a set \( X \) \([18, 26]\). However, it is important to observe that while the semiring laws, and hence the existence of a most-general semiring-valued model, are forced by the need to validate laws of relational algebra, there is still a design choice element at the heart of this framework: specifically, the choice
of the semiring signature and the definition of the semantics of relational algebra over annotated relations.

In other words, while the free semiring is the most general instance of the semiring-annotated model, this does not imply that all forms of provenance are described by semirings. (Certain natural-seeming models, such as where-provenance, are provably not instances [8].) In other words, I disagree with a characterization of the semiring model as providing “the most general form of provenance” [18, 26]. For such a statement to even make sense, we first need a general definition of “models of provenance”. In this talk I won’t try to provide one (and it’s not clear that this is really even possible); instead, I will discuss alternatives to the semiring approach that draw on ideas from algebraic logic.

Geerts and Poggi [16] studied Bancilhon-Paredes completeness (BP-completeness) for annotated relations, concluding that for BP-completeness, annotated relations need to be extended to include a an annotation-subtraction operator (monus) as well as duplicate-elimination operators (scalars). I will discuss another alternative, inspired by algebraic structures called cylindric algebras [22], introduced by Tarski and studied previously in connection to relational databases by Imielinski and Lipski [25] among others.

4. IDENTITY AND LINEARITY

[This section discusses joint work in progress with Peter Buneman, Stijn Vansummeren and Jan van den Bussche.]

The copy-paste provenance model introduced by Buneman et al. [4, 5] was motivated in part by the idea of tracking the identities of objects in a database as they are propagated through a query or acted upon by an update. In particular, we can think of the annotations propagated by the copy-paste model initially as labels uniquely identifying parts of the input database. This model is also adopted in other annotation-propagating models of provenance, such as DBNotes [3] and the semiring-annotated relation model [18, 14].

For each of these systems, it is possible for identifiers that were unique in the input to appear in multiple places in the output. In the case of where-provenance for queries, as employed for example in DBNotes, repeated identifiers signify “copies” of parts of the input, and they must all have the same value. So, the annotations still uniquely identify a value even though they are not necessarily unique in the output. However, for updates, an identifier can appear multiple times in the output with different values, so the analogy with identities breaks down completely.

A notion of provenance that takes identity seriously could be useful in many settings, but it seems particularly important in settings where users or agents work concurrently on different parts of a logical whole, some of which may be distributed. For example, consider the process of applying for a visa.

1. Alice sends Bob an application, consisting of a passport, application form, fee, and supporting documents such as a diploma.
2. Bob cannot read the diploma because it is in Latin, and sends it to Charlie for translation, along with a fee.
3. Charlie returns a diploma attached with a notarized translation to Bob.
4. Bob checks the application and translation, archives them, and deposits the fee, and detaches the passport from the other materials.
5. Bob attaches the visa to Alice’s passport and sends it back to Alice, along with the diploma.

It is important to note that none of the operations discussed above copy an artifact, and in most cases, replacing an original artifact with a copy would not be acceptable. For example, visa or passport applications typically need to be accompanied by certain original documents, not photocopies, and obviously, if Alice sends a copy of her passport then she has missed the point of the whole exercise.

In the talk, I will present some initial steps towards a model of provenance that takes identity into account, and highlight ways in which ideas from substructural logics such as linear logic [17] could be used to design query or update languages that are well-behaved with respect to identity. Similar ideas have already been applied extensively in programming languages for safe memory management [10], for resource-bounded programming [23], concurrency [29], and for PTIME-bounded database query languages [35]. Ultimately, we hope to identify a semantic class of operations that preserve unique identities (analogous to the semantic characterization of where-provenance in previous work) and to design appropriate query and update languages that capture the full expressiveness of this class. Work on linear logic for concurrency (including [29] among many other papers on linear types and session types for concurrency calculi) may provide a smooth way of generalizing provenance models from query languages to concurrent, artifact-oriented workflows [11].

5. CAUSALITY, KNOWLEDGE AND BELIEF

Some forms of provenance, particularly in scientific workflow settings, are motivated in terms of explaining how input was produced from output, or how the input caused the output. There have been a number of influential studies of causality and explanation by philosophers, and modern treatments use mathematical models or logical semantics to try to make sense of these ideas. Lewis [28] proposed a definition based on ideas from modal logic and counterfactual reasoning. Roughly, A is an explanation of B if A and B both hold, and in all “plausible” possible worlds where A does not hold, B also does not hold. For example, if John is drunk and crashes his car, then John’s drunkenness is an explanation if in all scenarios where John isn’t drunk, he avoids crashing. If there are some plausible scenarios where John still crashes, e.g. because the car’s brakes are faulty, then John’s drunkenness alone may not be a sufficient explanation. However, this approach has been widely critiqued in part because it depends strongly on how one defines which worlds are similar enough to be considered plausible.

More recently, there has been renewed interest in mathematical models of causality, in particular based on structural equations that are also widely used in economics and social science; see for example Pearl [34] and more recent refinements such as Halpern and Pearl’s definitions of actual cause and causal explanation [20, 21]. While this approach has been most influential in artificial intelligence, it has also influenced some philosophers such as Woodward [39].

To a first approximation, we can think of a causal model as a directed acyclic graph (or a piece of straight-line code)
that represents some assumptions about the relationships between events in a system. These assumptions are reflected by the structure of the graph, and by functions that describe the input-output behavior of each node. Often, causal models have different kinds of parameters, including endogenous variables that reflect properties we can control and measure, and exogenous variables that reflect unknown influences that we cannot directly control or measure, or that are not captured by the model. Causal models are often probability by making the exogenous parameters independent random variables, in which case one can view the causal model as essentially a Bayesian network.

A distinctive feature of Pearl’s approach is the idea of intervention. If $M$ is a causal model, then $M_{X \rightarrow x}$ is the causal model obtained from $M$ by fixing the value of $X$ to a given value, and disconnecting $X$ from other factors that normally would influence it. Intervention is the basic operation on causal models that distinguishes them from ordinary, black-box functions. Halpern and Pearl [20, 21] use interventions to propose (rather complex) definitions of actual cause and explanation.

In a recent workshop paper [7] I considered how ideas from structural causal models can be used to make sense of provenance, in particular the graphical models of provenance popular in scientific workflows and the Open Provenance Model community [33]. Concurrently, several other researchers including Meliou, Gatterbauer, Stanciu, Halpern, and Koch [31, 30, 32] have been investigating causality in databases, including connections to provenance and techniques for quantifying the degree of responsibility of an input for an output.

Logics of knowledge, such as multimodal logic [13], may also provide useful insight into provenance. Multimodal logics have modalities $K_A(\phi)$ representing the knowledge states of principals; that is, $K_A(\phi)$ means that $\phi$ holds in all worlds considered possible by $A$. Other approaches, such as probabilistic or plausibility logics [19], could be useful for modeling degrees of uncertainty (naturally, probabilistic logic is closely related to probabilistic database semantics which is already known to be related to certain forms of provenance). Logics of knowledge and uncertainty may be useful for understanding provenance in settings with multiple agents or principals, particularly security settings where we might want to reason about what information principals cannot learn. These logics might also be connected to other kinds of annotation schemes [15].

6. CONCLUSIONS

So far, research on provenance has sometimes had an arbitrary flavor, since it appears essential to depart from some of the fundamental definitions of the semantics of query languages and it is not clear how to explore the search space systematically. Ultimately, such designs need to be validated empirically by showing that they can be implemented efficiently and provide value for users. However, exploring the design space by trial and error could be quite expensive.

In this invited talk and paper I have tried to identify some ways of leveraging ideas already explored in mathematical and philosophical logic as inspiration or justification for models of provenance. I suggest that relevance logic can provide a starting point for understanding why-provenance, and that algebras of logic such as cylindric algebras might provide a useful alternative to the semiring-valued model. I also outlined some more speculative directions, including the possibility of using ideas from linear logic (or other resource-sensitive logics) to give query and update languages a semantics that deals with identity, and mathematical models of causality, knowledge and belief for understanding forms of provenance that purport to explain causal relationships between parts of computations. It is left for future work to confirm whether these or other ideas from logic can indeed provide a solid foundation for provenance in databases.

7. REFERENCES

[15] Wolfgang Gatterbauer, Magdalena Balazinska, Nodira Khoussainova, and Dan Suciu. Believe it or not:


[http://eprints.ecs.soton.ac.uk/18332/]().


