

# Adaptation of Semiring Provenance

Paul LANDRIER  
Supervised by Silviu Maniu

## 1 Existing framework - Provenance in databases

- Example
- Semirings and polynomials for provenance

## 2 Provenance for computation graphs

- Computation Graphs
- Abstraction of Computation Graphs
- Near-semirings

## 3 Future work

## 4 References

## 5 Failed attempt - Differential Geometry

## Existing framework - Provenance in databases

# Situation

This whole section is entirely based on [Green et al., 2007].

<i>ClientName</i> (A)	<i>Country</i> (B)	<i>Product</i> (C)
Alice	USA	tea
Bob	USA	coffee
Charlie	FR	coffee

A	C
Alice	tea
Alice	coffee
Bob	tea
Bob	coffee
Charlie	coffee

Figure: Databases  $D$  and  $q(D)$ .

$$q(R) \stackrel{\text{def}}{=} \pi_{AC}(\pi_{AB}R \bowtie \pi_{BC}R \cup \pi_{AC}R \bowtie \pi_{BC}R)$$

# Situation

A	B	C	
Alice	USA	tea	2
Bob	USA	coffee	5
Charlie	FR	coffee	1

A	B	C	
Alice	USA	tea	$b_1$
Bob	USA	coffee	$b_2$
Charlie	FR	coffee	$b_3$

A	C	
Alice	tea	$2 \cdot 2 + 2 \cdot 2 = 8$
Alice	coffee	$2 \cdot 5 = 10$
Bob	tea	$2 \cdot 5 = 10$
Bob	coffee	$5 \cdot 5 + 5 \cdot 5 + 5 \cdot 1 = 55$
Charlie	coffee	$1 \cdot 1 + 1 \cdot 1 + 5 \cdot 1 = 7$

A	C	
Alice	tea	$(b_1 \wedge b_1) \vee (b_1 \wedge b_1) = b_1$
Alice	coffee	$b_1 \wedge b_2$
Bob	tea	$b_1 \wedge b_2$
Bob	coffee	$(b_2 \wedge b_2) \vee (b_2 \wedge b_2) \vee (b_2 \wedge b_1)$
Charlie	coffee	$(b_3 \wedge b_3) \vee (b_3 \wedge b_3) \vee (b_2 \wedge b_3)$

# Polynomials for provenance

We abstract the two preceeding computations with variables  $p, r, s$  and abstract operations  $+$  and  $\cdot$ .

<i>A</i>	<i>B</i>	<i>C</i>	
<i>Alice</i>	<i>USA</i>	<i>tea</i>	$p$
<i>Bob</i>	<i>USA</i>	<i>coffee</i>	$r$
<i>Charlie</i>	<i>FR</i>	<i>coffee</i>	$s$

<i>A</i>	<i>C</i>	
<i>Alice</i>	<i>tea</i>	$2p^2$
<i>Alice</i>	<i>coffee</i>	$pr$
<i>Bob</i>	<i>tea</i>	$pr$
<i>Bob</i>	<i>coffee</i>	$2r^2 + rs$
<i>Charlie</i>	<i>coffee</i>	$2s^2 + rs$

# Positive Relational Algebra on $K$ -relations I

**empty relation** There is  $\emptyset : \begin{cases} U\text{-}\mathbf{Tuple} & \rightarrow K \\ t & \mapsto 0_K \end{cases}$ .

**union** Let  $R_1, R_2 : U\text{-}\mathbf{Tuple} \rightarrow K$  be two  $K$ -relations.

Then we define  $R_1 \cup R_2 : \begin{cases} U\text{-}\mathbf{Tuple} & \rightarrow K \\ t & \mapsto R_1(t) + R_2(t) \end{cases}$ .

**projection** Let  $R : U\text{-}\mathbf{Tuple} \rightarrow K$  be a  $K$ -relation and let  $V \subseteq U$  be a subdomain of  $U$ .

Then we define

$$\pi_V R : \begin{cases} U\text{-}\mathbf{Tuple} & \rightarrow K \\ t & \mapsto \sum_{t' \in \text{supp}(R) \text{ and } t'|_V = t} R(t') \end{cases}.$$

Where  $t'|_V \in V\text{-}\mathbf{Tuple}$  is the restriction of  $t' \in U\text{-}\mathbf{Tuple}$  seen as a function  $U \rightarrow \mathbb{D}$  to the subset  $V$ . Since  $\text{supp}(R)$  is finite, the sum is finite.

# Positive Relational Algebra on $K$ -relations II

**selection** Let  $R : U\text{-}\mathbf{Tup} \rightarrow K$  be a  $K$ -relation and  $\mathbf{P} : U\text{-}\mathbf{Tup} \rightarrow \{0_K, 1_K\}$  be a predicate.

Then we define  $\sigma_{\mathbf{P}}R : \begin{cases} U\text{-}\mathbf{Tup} & \rightarrow & K \\ t & \mapsto & R(t) \cdot \mathbf{P}(t) \end{cases}$ .

In the two remaining definitions, be aware that the set of attributes over which functions are defined changes.

**natural join** Let  $U_1, U_2$  be two finite sets of attributes. Let  $R_1 : U_1\text{-}\mathbf{Tup} \rightarrow K$  and  $R_2 : U_2\text{-}\mathbf{Tup} \rightarrow K$  be two  $K$  relations.

Then we define

$R_1 \bowtie R_2 : \begin{cases} (U_1 \cup U_2)\text{-}\mathbf{Tup} & \rightarrow & K \\ t & \mapsto & R_1(t|_{U_1}) \cdot R_2(t|_{U_2}) \end{cases}$ .

**renaming** Let  $R : U\text{-}\mathbf{Tup} \rightarrow K$  be a  $K$ -relation and  $\beta : U \rightarrow U'$  be a bijection.

Then we define  $\rho_{\beta}R : \begin{cases} U'\text{-}\mathbf{Tup} & \rightarrow & K \\ t & \mapsto & R(t \circ \beta) \end{cases}$ .



# Semirings for provenance

A very nice property :

Semirings are a good choice for provenance

The  $\mathcal{RA}^+$  equalities:

- Union is associative, commutative and has identity  $\emptyset$ ;
- Join is associative, commutative and distributive over union;
- Projections and selections commute with each other as well as with unions and joins;
- For all  $R$ ,  $\sigma_{\text{false}}(R) = \emptyset$  and  $\sigma_{\text{true}}(R) = R$

are satisfied if and only if  $(K, +, \cdot, 0_K, 1_K)$  is a commutative semiring.

## Provenance for computation graphs

# Definition

## Definition - Computation Graph

A computation graph  $(V, E, Fun, Op, F, Operator)$  over a set  $S$  is a directed acyclic graph  $(V, E)$  together with:

- two subsets  $F \subset (S \rightarrow S)$  and  $Operator \subset (S^{(\mathbb{N})} \rightarrow S)$ ,
- a function  $Fun : E \rightarrow F$ ,
- a function  $Op : V \rightarrow Operator$ .

# An Example

Goal : derive a computation for the function

$f(x) = e^{x^2}(x^2 + 2x + 2)$  using only the linear, square, "plus constant" and exponential functions, with  $+$  and  $\times$  operators.

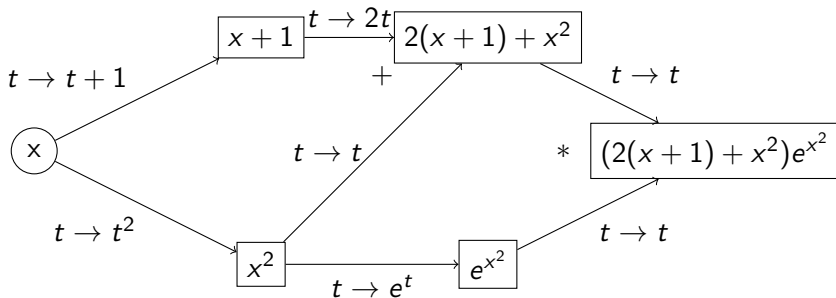


Figure: One possible computation graph of  $f$ .

# Links with Neural Networks

## Property

A Deep Neural Network can be represented by a computation graph.

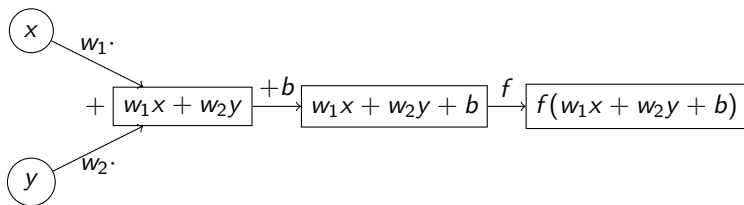


Figure: Computation graph of a perceptron.

# Abstract Computation Graph

## Definition - Computation Graph over a Set

A computation graph  $(V, E, Fun, Op, F, Operator)$  over a set  $S$  is a directed acyclic graph  $(V, E)$  together with :

- two subsets  $F \subset (S \rightarrow S)$  and  $Operator \subset (S^{(\mathbb{N})} \rightarrow S)$ ,
- a function  $Fun : E \rightarrow F$ ,
- a function  $Op : V \rightarrow Operator$ .

## Definition - Computation Graph over a General Algebraic Structure

A generalised computation graph  $(V, E, Fun, F)$  is a directed acyclic graph  $(V, E)$  together with :

- an algebraic structure  $(F, +, \circ, 0_F, id)$ ,
- a function  $Fun : E \rightarrow F$ .

# Interpretation of a Generalized Computation Graph

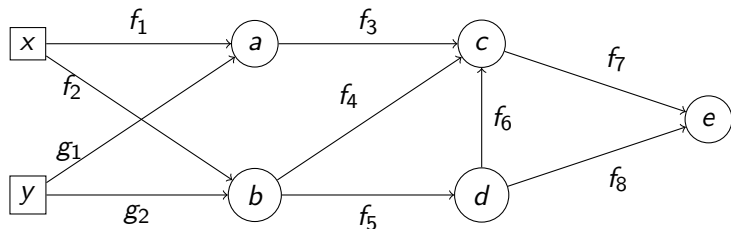


Figure: An abstract computation graph.

The expression represented by the figure above is

$$f_7 \circ [f_3 \circ (f_1 \circ x + g_1 \circ y) + f_4 \circ (f_2 \circ x + g_2 \circ y) + f_6 \circ f_5 \circ (f_2 \circ x + g_2 \circ y)] + f_8 \circ f_5 \circ (f_2 \circ x + g_2 \circ y)$$

# Identifying the Right Structure

- Since there is no order on the edges,  $+$  should be associative and commutative.
- For modularity,  $\circ$  should be associative.
- To model the concept of “0 weight”, there should be a neutral element for  $+$ , noted  $0_F$ , and it should satisfy  $\forall f \in Fun, 0_F \circ f = 0_F$
- To be able to transmit information in the graph, to initialize some algorithm, to model the one node graph, there should be a neutral element for  $\circ$  noted  $id$ .
- For modularity and theoretical reasons, we should have the equality:  $\forall (f, g, h) \in F^3, (f + g) \circ h = f \circ h + g \circ h$



# Near-semirings

## Definition - Near-semirings

An algebraic structure  $(F, +, \circ, 0_F, id)$  is a near-semiring if it satisfies:

- $(F, +, 0_F)$  is a commutative monoid,
- $(F, \cdot, id)$  is a monoid,
- $\forall (f, g, h) \in F^3, (f + g) \circ h = f \circ h + g \circ h,$
- $\forall f \in F, 0_F \circ f = 0_F.$

# Examples of Near-semirings

- 1 Every semiring is a near-semiring. The interpretation of elements of a semiring as functions can be made through the semi-module structure.
- 2 If  $(M, +, 0_M)$  is a monoid, the set of functions  $M \rightarrow M$  with the usual composition and identity and with the natural addition on function is a near-semiring.
- 3 Continuous functions on a topological monoid. (For instance  $(\mathbb{R}_+^n, +, 0)$ , that is stable by a linear function composed with ReLU.)
- 4 The  $\mathcal{C}^\infty$  functions over the real numbers.
- 5 The near-semiring generated by the linear functions, the translations ("plus constant") and an activation function.

# Mathematical Universal Object for Near-semirings

## Theorem - Universal Form of Near-semirings

It is equivalent to be given:

- 1 a near-semiring  $(F, +, \circ, 0, id)$
- 2 a monoid  $(M, +, 0)$  together with a set of functions  $M \rightarrow M$  containing  $0$  and  $id$  and stable by  $\circ$  and  $+$ .

Future work

# Instantiation of the Theory

We pass a first sanity check: we can model a neural network with a generalized computation graph using near-semiring. We can discuss other possible instantiation:

- Is layer-wise relevance propagation captured by the model?
- Is the work done by Antoine an instance of the model?
- Does the model allow to retrieve efficiently shapley values?
- Can the model be a theoretical basis for high-level circuits that explain neural networks, considering the near-semiring over the monoid  $\Sigma^*$ ?

# Further Theoretical Development

- Universal object for computer science. (The near-semiring generated by  $F$  is composed of elements that are either  $f \in F$  or  $f \circ (\sum_{i=1}^n M_i)$  with  $f \in F$  and  $M_i$  some element.)
- Investigate the differences between forward and backward modes.
- Investigate a potential theory of computation graphs with an algebraic derivation.
- Develop an analogous model for a coarse-grained provenance.

## References

# References



Green, T. J., Karvounarakis, G., and Tannen, V. (2007).  
Provenance semirings.

In *Proceedings of the twenty-sixth ACM  
SIGMOD-SIGACT-SIGART symposium on Principles of  
database systems*, pages 31–40, Beijing China. ACM.



Ramusat, Y. (2022).

*The semiring-based provenance framework for graph  
databases.*

phdthesis, Université Paris sciences et lettres.



Failed attempt - Differential Geometry



[Ramusat, 2022]