Regular Expressions for Provenance

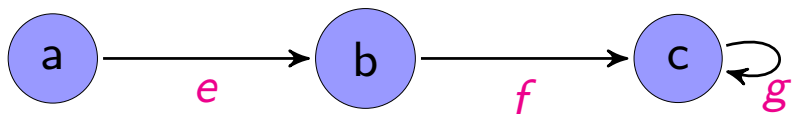
Maximilian Schlund

joint work with
Michael Luttenberger

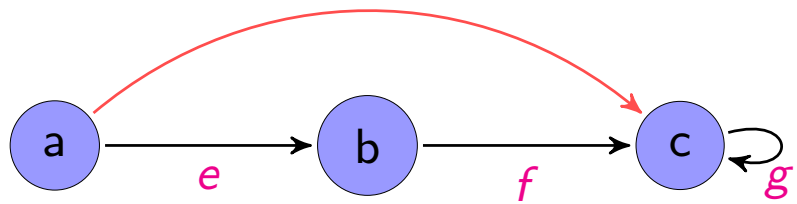
Department of Computer Science
TU München

June 12, 2014

Running Example: Transitive Closure of a Relation



Running Example: Transitive Closure of a Relation



Provenance = Solution of Polynomial Fixpoint Equations

(Green et al., PODS 2007)

Annotated facts

```
e E(a,b).
f E(b,c).
g E(c,c).
```

Query

```
T(X,Y) :- E(X,Y).
T(X,Y) :- T(X,Z),T(Z,Y).
```

Result

```
T(a,b)
T(a,c)
T(b,c)
T(c,c)
```

Provenance = Solution of Polynomial Fixpoint Equations

(Green et al., PODS 2007)

Annotated facts

```
e E(a,b).
f E(b,c).
g E(c,c).
```

Query

```
T(X,Y) :- E(X,Y).
T(X,Y) :- T(X,Z),T(Z,Y).
```

Result

```
T(a,b)
T(a,c)
T(b,c)
T(c,c)
```

Provenance equations

$$T_{a,c} = T_{a,b} \cdot T_{b,c} + T_{a,c} \cdot T_{c,c}$$

$$T_{a,b} = E_{a,b}$$

$$T_{b,c} = E_{b,c}$$

$$T_{c,c} = E_{c,c} + T_{c,c} \cdot T_{c,c}$$

$$E_{a,b} = e \quad E_{b,c} = f \quad E_{c,c} = g$$

Provenance = Solution of Polynomial Fixpoint Equations

(Green et al., PODS 2007)

Annotated facts

```
e E(a,b).
f E(b,c).
g E(c,c).
```

Query

```
T(X,Y) :- E(X,Y).
T(X,Y) :- T(X,Z),T(Z,Y).
```

Result

```
T(a,b)
T(a,c)
T(b,c)
T(c,c)
```

Provenance equations

$$T_{a,c} = T_{a,b} \cdot T_{b,c} + T_{a,c} \cdot T_{c,c}$$

$$T_{a,b} = E_{a,b}$$

$$T_{b,c} = E_{b,c}$$

$$T_{c,c} = E_{c,c} + T_{c,c} \cdot T_{c,c}$$

$$E_{a,b} = e \quad E_{b,c} = f \quad E_{c,c} = g$$

Semantics

What does "$+$" and "$\cdot$" mean?

"$+$" Alternative origin ($\cup$, π)

"$\cdot$" Joint origin ($\bowtie$)

Provenance = Solution of Polynomial Fixpoint Equations

(Green et al., PODS 2007)

Annotated facts

```
e E(a,b).
f E(b,c).
g E(c,c).
```

Query

```
T(X,Y) :- E(X,Y).
T(X,Y) :- T(X,Z),T(Z,Y).
```

Result

```
T(a,b)
T(a,c)
T(b,c)
T(c,c)
```

Provenance equations

$$T_{a,c} = T_{a,b} \cdot T_{b,c} + T_{a,c} \cdot T_{c,c}$$

$$T_{a,b} = E_{a,b}$$

$$T_{b,c} = E_{b,c}$$

$$T_{c,c} = E_{c,c} + T_{c,c} \cdot T_{c,c}$$

$$E_{a,b} = e \quad E_{b,c} = f \quad E_{c,c} = g$$

Semantics

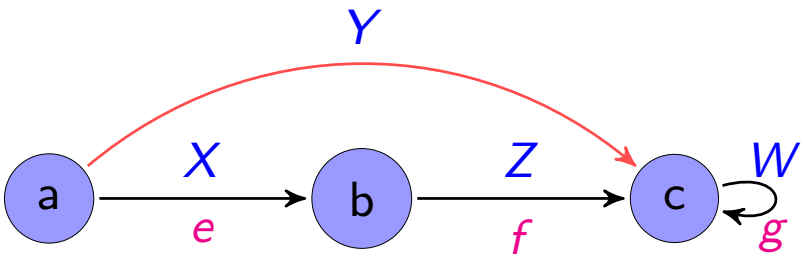
What does "$+$" and "$\cdot$" mean?

"$+$" Alternative origin ($\cup$, π)

"$\cdot$" Joint origin ($\bowtie$)

Natural algebraic structure to represent provenance: commutative semirings

Semiring Interpretations



$$Y = X \cdot Z + Y \cdot W$$

$$X = e$$

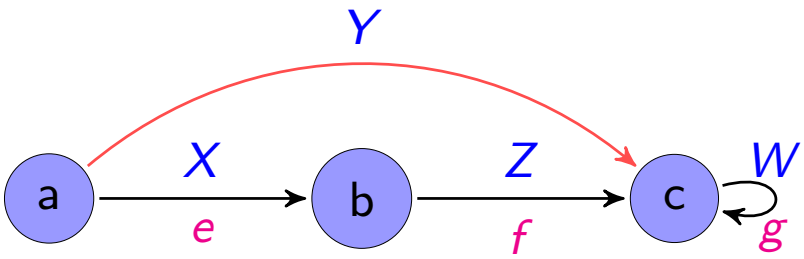
$$Z = f + Z \cdot W$$

$$W = g + W \cdot W$$

Why Provenance

Free interpretation of e, f, g, $+$, and $\cdot$ (modulo $1 + 1 = 1$ and $x^2 = x$).

Semiring Interpretations



$$Y = X \cdot Z + Y \cdot W$$

$$X = e$$

$$Z = f + Z \cdot W$$

$$W = g + W \cdot W$$

Why Provenance

Free interpretation of e, f, g, $+$, and $\cdot$ (modulo $1 + 1 = 1$ and $x^2 = x$).

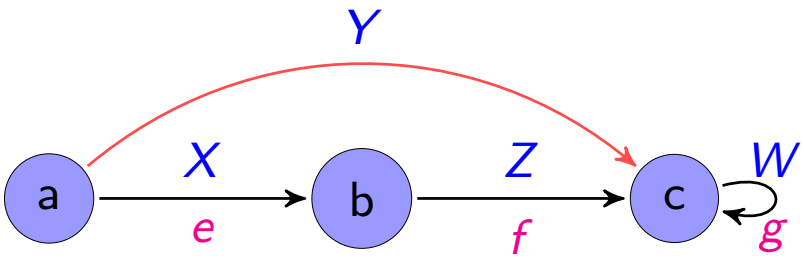
$$Y = e \cdot f + e \cdot f \cdot g$$

$$X = e$$

$$Z = f + f \cdot g$$

$$W = g$$

Semiring Interpretations



$$Y = X \cdot Z + Y \cdot W$$

$$X = e$$

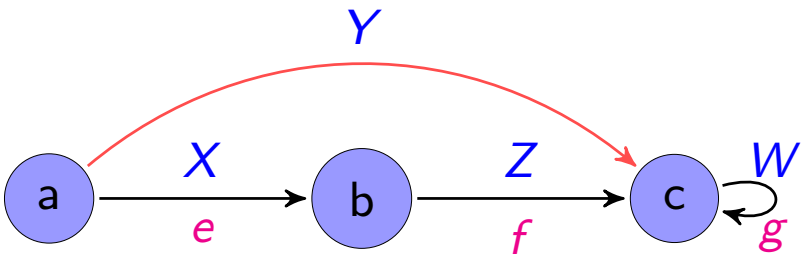
$$Z = f + Z \cdot W$$

$$W = g + W \cdot W$$

How Provenance

Free interpretation of e, f, g, $+$, and $\cdot$ ($\rightsquigarrow$ remember how many derivations are possible).

Semiring Interpretations



$$Y = X \cdot Z + Y \cdot W$$

$$X = e$$

$$Z = f + Z \cdot W$$

$$W = g + W \cdot W$$

How Provenance

Free interpretation of e, f, g, $+$, and $\cdot$ ($\rightsquigarrow$ remember how many derivations are possible).

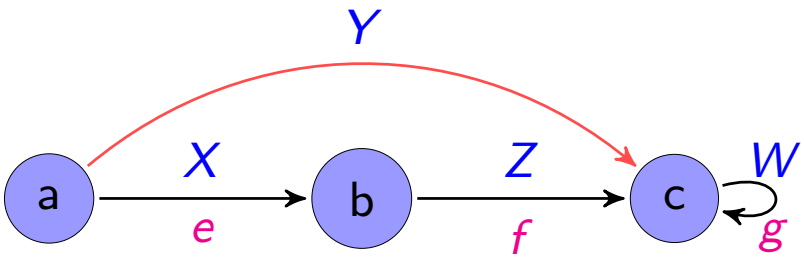
$$Y = \ldots$$

$$X = e$$

$$Z = f + f \cdot g + 2f \cdot g^2 + 5f \cdot g^3 \ldots$$

$$W = g + g^2 + 2g^3 + 5g^4 + \ldots$$

Semiring Interpretations



$$Y = X \cdot Z + Y \cdot W$$

$$X = e$$

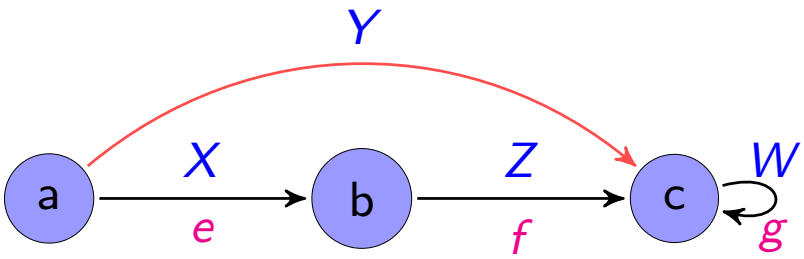
$$Z = f + Z \cdot W$$

$$W = g + W \cdot W$$

Weights (here: shortest paths)

Interpret e, f, g over the tropical semiring $(\overline{\mathbb{Z}}, \min, +, \infty, 0)$, e.g. $e = 4, f = -6, g = 1$.

Semiring Interpretations



$$Y = X \cdot Z + Y \cdot W$$

$$X = e$$

$$Z = f + Z \cdot W$$

$$W = g + W \cdot W$$

Weights (here: shortest paths)

Interpret e, f, g over the tropical semiring $(\overline{\mathbb{Z}}, \min, +, \infty, 0)$, e.g. $e = 4, f = -6, g = 1$.

$$Y = -2$$

$$X = 4$$

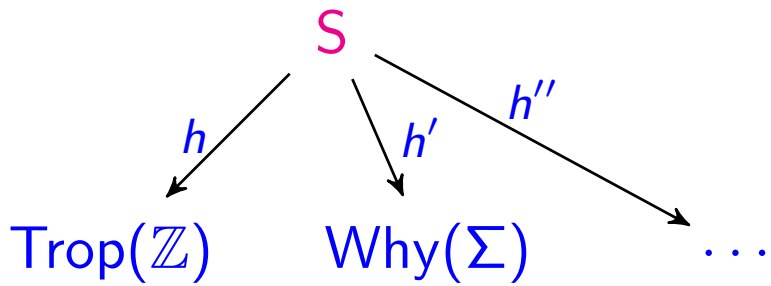
$$Z = -6$$

$$W = 1$$

Two Problems

Representing Solutions

Is there a general finitely representable semiring S that can be specialized to any semiring via a homomorphism?



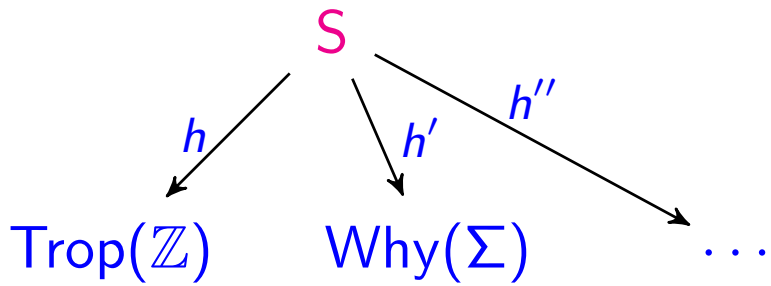
Theorem (Deutch et al., ICDT 2014)

It is not possible to use a finite provenance annotation that specializes to Why(Σ).

Two Problems

Representing Solutions

Is there a general finitely representable semiring S that can be specialized to any semiring via a homomorphism?



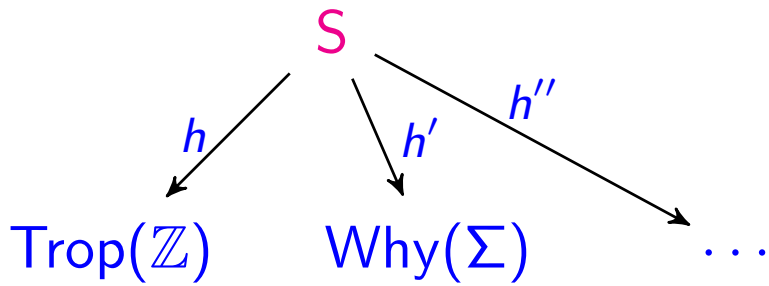
Theorem (Luttenberger, S., TaPP 2014)

We can find a finite representation that specializes to $\mathrm{Why}(\Sigma)$.

Two Problems

Representing Solutions

Is there a general finitely representable semiring S that can be specialized to any semiring via a homomorphism?



Theorem (Luttenberger, S., TaPP 2014)

We can find a finite representation that specializes to Why(Σ).

Computing Solutions

Can we compute a solution (in above representation) in finite time?

General Representation (Green et al. 2007)

Most general solution of $X = aX^2 + c$ (any commutative semiring):

$$X = c + ac^2 + 2a^2c^3 + 5a^3c^4 + 14a^4c^5 + \cdots = \sum_{n=0}^{\infty} \frac{1}{n+1} \binom{2n}{n} a^n c^{n+1}$$

$\rightsquigarrow$ Formal power series in commuting variables $\mathbb{N}_\infty \langle\langle \Sigma^\oplus \rangle\rangle$.

General Representation (Green et al. 2007)

Most general solution of $X = aX^2 + c$ (any commutative semiring):

$$X = c + ac^2 + 2a^2c^3 + 5a^3c^4 + 14a^4c^5 + \cdots = \sum_{n=0}^{\infty} \frac{1}{n+1} \binom{2n}{n} a^n c^{n+1}$$

$\rightsquigarrow$ Formal power series in commuting variables $\mathbb{N}_\infty \langle\langle \Sigma^\oplus \rangle\rangle$.

Problem

No finite representation for elements of $\mathbb{N}_\infty \langle\langle \Sigma^\oplus \rangle\rangle$!

General Representation (Green et al. 2007)

Most general solution of $X = aX^2 + c$ (any commutative semiring):

$$X = c + ac^2 + 2a^2c^3 + 5a^3c^4 + 14a^4c^5 + \cdots = \sum_{n=0}^{\infty} \frac{1}{n+1} \binom{2n}{n} a^n c^{n+1}$$

$\rightsquigarrow$ Formal power series in commuting variables $\mathbb{N}_\infty \langle\langle \Sigma^\oplus \rangle\rangle$.

Problem

No finite representation for elements of $\mathbb{N}_\infty \langle\langle \Sigma^\oplus \rangle\rangle$!

Our solution

Power series with coefficients in $\mathbb{N}_k = \{0, 1, \ldots, k\}$ can be finitely represented via regular expressions.

Example $X = aX^2 + c$ over $\mathbb{N}_1 \langle\langle \Sigma^\oplus \rangle\rangle$

$X = c + ac^2 + a^2c^3 + a^3c^4 + \cdots = \sum_{n=0}^{\infty} a^n c^{n+1} = c(ac)^*$

Concise Representations

Objection (?)

But . . . regular expressions can get **huge**!

Solution to linear equation in 2 dimensions:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} (a^*b(ca^*b+d)^*ca^* + a^*)e + a^*b(ca^*b+d)^*f \\ (ca^*b+d)^*ca^*e + (ca^*b+d)^*f \end{pmatrix}$$

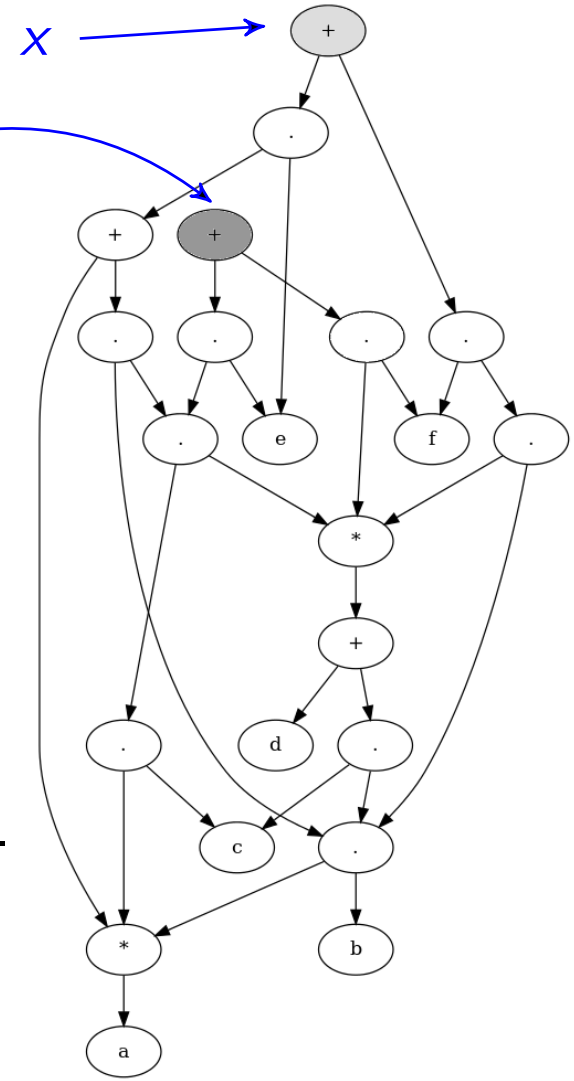
Concise Representations

Objection (?)

But . . . regular expressions can get **huge**!

Solution to linear equation in 2 dimensions:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$



Remedy

BDD-like shared representation/arithmetic circuit.

$\rightsquigarrow$ polynomial representation!

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} (a^*b(ca^*b+d)^*ca^* + a^*)e + a^*b(ca^*b+d)^*f \\ (ca^*b+d)^*ca^*e + (ca^*b+d)^*f \end{pmatrix}$$

Computing Solution of $X = F(X)$

Standard approach: fixpoint iteration (seminaive evaluation):

$$F(0),\ F(F(0)),\ F^3(0),\ \ldots,\ F^n(0).$$

Computing Solution of $X = F(X)$

Standard approach: fixpoint iteration (seminaive evaluation):

$$F(0),\ F(F(0)),\ F^3(0),\ \ldots,\ F^n(0).$$

Problem

Might not terminate, even for finitely presentable semirings!

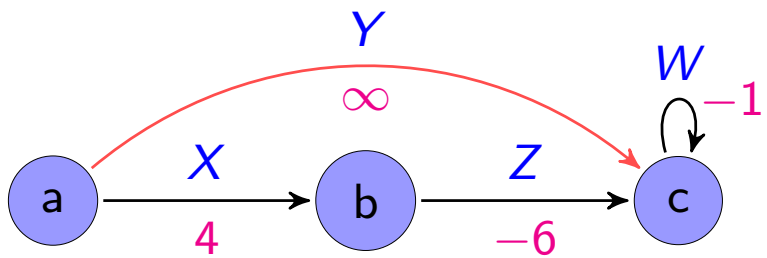
Computing Solution of $X = F(X)$

Standard approach: fixpoint iteration (seminaive evaluation):

$$F(0),\ F(F(0)),\ F^3(0),\ \ldots,\ F^n(0).$$

Problem

Might not terminate, even for finitely presentable semirings!



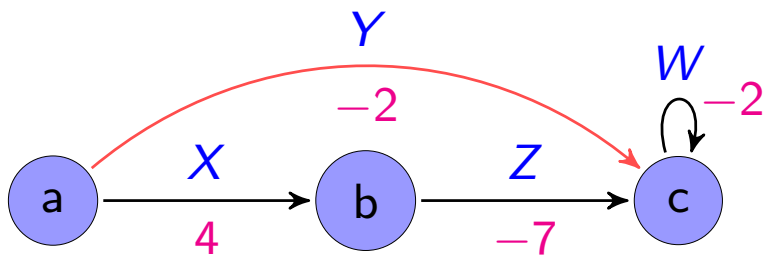
Computing Solution of $X = F(X)$

Standard approach: fixpoint iteration (seminaive evaluation):

$$F(0),\ F(F(0)),\ F^3(0),\ \ldots,\ F^n(0).$$

Problem

Might not terminate, even for finitely presentable semirings!



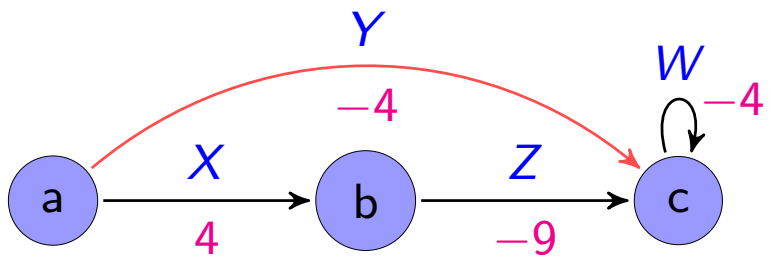
Computing Solution of $X = F(X)$

Standard approach: fixpoint iteration (seminaive evaluation):

$$F(0),\ F(F(0)),\ F^3(0),\ \ldots,\ F^n(0).$$

Problem

Might not terminate, even for finitely presentable semirings!



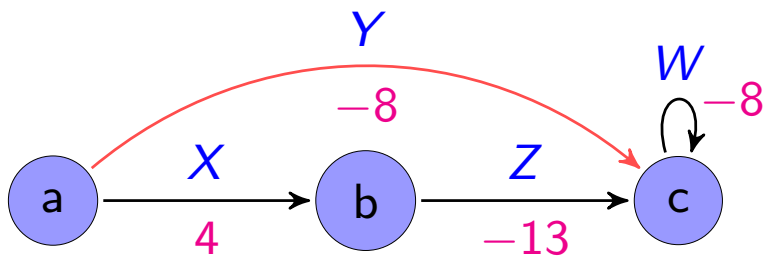
Computing Solution of $X = F(X)$

Standard approach: fixpoint iteration (seminaive evaluation):

$$F(0),\ F(F(0)),\ F^3(0),\ \ldots,\ F^n(0).$$

Problem

Might not terminate, even for finitely presentable semirings!



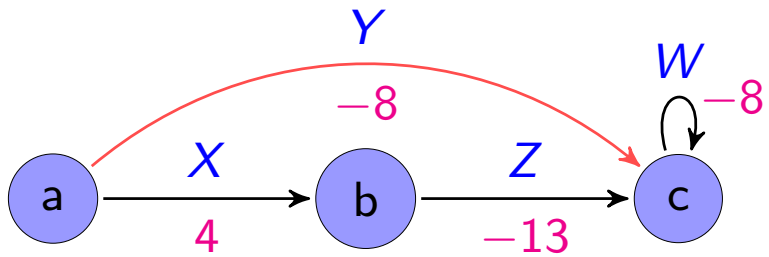
Computing Solution of $X = F(X)$

Standard approach: fixpoint iteration (seminaive evaluation):

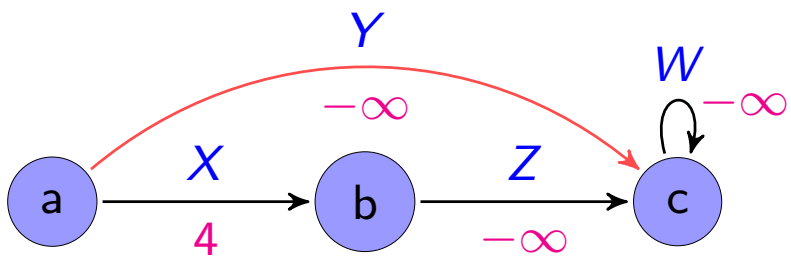
$$F(0),\ F(F(0)),\ F^3(0),\ \ldots,\ F^n(0).$$

Problem

Might not terminate, even for finitely presentable semirings!



$$\overrightarrow{n \to \infty}$$



Computing Solution of $X = F(X)$ (II)

Our solution

- For linear equations: accellerate fixpoint iteration via the Kleene-star. $X = aX + b \Longrightarrow X = a^* b$.

Computing Solution of $X = F(X)$ (II)

Our solution

- For linear equations: accellerate fixpoint iteration via the Kleene-star. $X = aX + b \Longrightarrow X = a^* b$.
- For non-linear equations: Linearize into sequence of linear systems (Newton's method for semirings, Esparza, Kiefer, Luttenberger, 2007).

Computing Solution of $X = F(X)$ (II)

Our solution

- For linear equations: accellerate fixpoint iteration via the Kleene-star. $X = aX + b \Longrightarrow X = a^* b$.
- For non-linear equations: Linearize into sequence of linear systems (Newton's method for semirings, Esparza, Kiefer, Luttenberger, 2007).

Theorem (Luttenberger, S., LATA 2013)

Over $N_k \langle\langle \Sigma^{\oplus} \rangle\rangle$ *at most* $n + 1 + \log\log k$ *linearizations needed.*

Summary

- Provenance analysis for recursive Datalog = solving polynomial equations.
- General representation for all commutative semirings (modulo generalized idempotence $k + 1 = k$): concise regular expressions.
- RegEx can be computed by Newton's method.
- Solution can be specialized to any semiring with computable Kleene-star (e.g. Why semiring).

Summary

- Provenance analysis for recursive Datalog = solving polynomial equations.
- General representation for all commutative semirings (modulo generalized idempotence $k + 1 = k$): concise regular expressions.
- RegEx can be computed by Newton's method.
- Solution can be specialized to any semiring with computable Kleene-star (e.g. Why semiring).

Get our tool/library FPsolve from `https://github.com/mschlund/FPsolve` if you want to play with Newton's method on semirings :-).

Summary

- Provenance analysis for recursive Datalog = solving polynomial equations.
- General representation for all commutative semirings (modulo generalized idempotence $k + 1 = k$): concise regular expressions.
- RegEx can be computed by Newton's method.
- Solution can be specialized to any semiring with computable Kleene-star (e.g. Why semiring).

Get our tool/library FPsolve from `https://github.com/mschlund/FPsolve` if you want to play with Newton's method on semirings :-).

Thank you!