Weighted LARS for Quantitative Stream Reasoning

Thomas Eiter, Rafael Kiesel

Vienna University of Technology

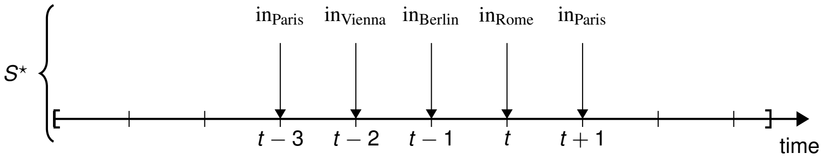
funded by FWF project W1255-N23

29th of August 2020

FWF Der Wissenschaftsfonds.

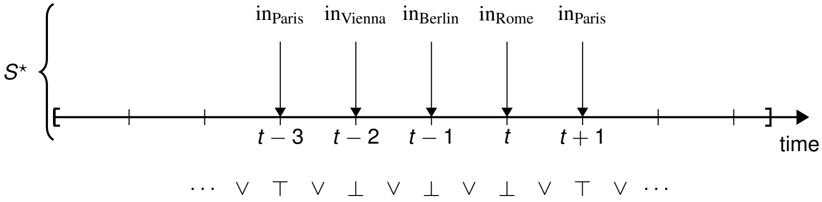
logics LOGICAL METHODS IN COMPUTER SCIENCE

(Qualitative) Stream Reasoning with LARS



- Does Peter visit Paris?

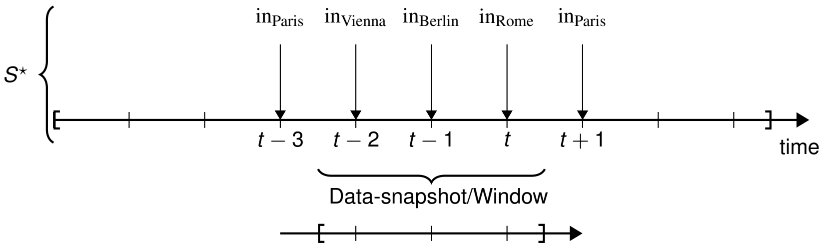
(Qualitative) Stream Reasoning with LARS



- Does Peter visit Paris?

$\rightarrow$ $\Diamond \text{in}_{\text{Paris}}$

(Qualitative) Stream Reasoning with LARS

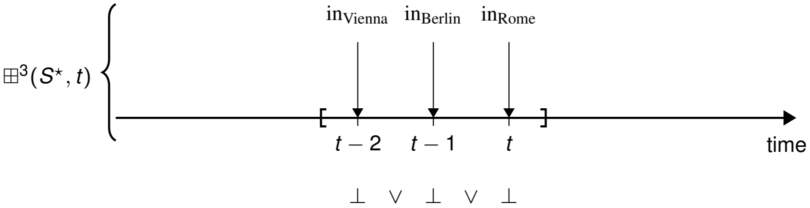


- Does Peter visit Paris?

$\rightarrow$ $\Diamond \mathrm{in}_{\mathrm{Paris}}$

- Did Peter visit Paris in the last three days?

(Qualitative) Stream Reasoning with LARS



- Does Peter visit Paris?

$\rightarrow$ $\Diamond \text{in}_{\text{Paris}}$

- Did Peter visit Paris in the last three days?

$\rightarrow$ $\boxplus^3 \Diamond \text{in}_{\text{Paris}}$

Quantitative?

- How often does Peter visit Paris?

Quantitative?

- How often does Peter visit Paris?

$\rightarrow$ Answer *quantitative* queries

Quantitative?

- How often does Peter visit Paris?

$\rightarrow$ Answer *quantitative* queries

- Which journey is optimal w.r.t. how often Paris is visited?

Quantitative?

- How often does Peter visit Paris?

$\rightarrow$ Answer *quantitative* queries

- Which journey is optimal w.r.t. how often Paris is visited?
- LARS Programs, consisting of rules $\alpha \leftarrow \beta$ with ASP-semantics give us answer streams

Quantitative?

- How often does Peter visit Paris?

$\rightarrow$ Answer *quantitative* queries

- Which journey is optimal w.r.t. how often Paris is visited?
- LARS Programs, consisting of rules $\alpha \leftarrow \beta$ with ASP-semantics give us answer streams
- Answer streams correspond to possible journeys

Quantitative?

- How often does Peter visit Paris?

$\rightarrow$ Answer *quantitative* queries

- Which journey is optimal w.r.t. how often Paris is visited?
- LARS Programs, consisting of rules $\alpha \leftarrow \beta$ with ASP-semantics give us answer streams
- Answer streams correspond to possible journeys

$\rightarrow$ Express *preferences* over answer streams

Quantitative?

- How often does Peter visit Paris?

$\rightarrow$ Answer *quantitative* queries

- Which journey is optimal w.r.t. how often Paris is visited?
- LARS Programs, consisting of rules $\alpha \leftarrow \beta$ with ASP-semantics give us answer streams
- Answer streams correspond to possible journeys

$\rightarrow$ Express *preferences* over answer streams

- Other quantitative questions (Probabilities, Weighted Model Counting, ...)

Quantitative!

Quantitative extension(s) needed!

- Ad Hoc?

Quantitative!

Quantitative extension(s) needed!

- Ad Hoc?
- Better: Framework! But, how? Need Computation

Quantitative!

Quantitative extension(s) needed!

- Ad Hoc?
- Better: Framework! But, how? Need Computation
- Approach: Use Semirings $(R, \oplus, \otimes, e_\oplus, e_\otimes)$, which are abstract models of computation

Quantitative!

Quantitative extension(s) needed!

- Ad Hoc?
- Better: Framework! But, how? Need Computation
- Approach: Use Semirings $(R, \oplus, \otimes, e_\oplus, e_\otimes)$, which are abstract models of computation

(Assoc) $(a \oplus b) \oplus c = a \oplus (b \oplus c) \quad (a \otimes b) \otimes c = a \otimes (b \otimes c)$

(Id) $e_\oplus \oplus a = a = a \oplus e_\oplus \quad e_\otimes \otimes a = a = a \otimes e_\otimes$

(Comm) $a \oplus b = b \oplus a$

(Distr.) $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
$(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$

(Ann) $e_\oplus \otimes a = e_\oplus = a \otimes e_\oplus$

Semiring Examples

Examples are

- $\mathbb{S} = (\mathbb{S}, +, \cdot, 0, 1)$, for $\mathbb{S} \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$, the semiring over the numbers in $\mathbb{S}$. It is typically used for arithmetic.
- $\mathcal{R}_{\max} = (\mathbb{Q} \cup \{-\infty, \infty\}, \max, +, -\infty, 0)$, the max tropical semiring. It is typically used in the context of optimisation.
- $\mathbb{B} = (\{0, 1\}, \vee, \wedge, 0, 1)$, the Boolean semiring. It is typically used for classical Boolean constraints.

Weighted LARS

- Evaluate formulas w.r.t. $(S^\star, S, t)$
- $S^\star$ starting stream, S current stream, t time

Weighted LARS

- Evaluate formulas w.r.t. $(S^\star, S, t)$
- $S^\star$ starting stream, S current stream, t time

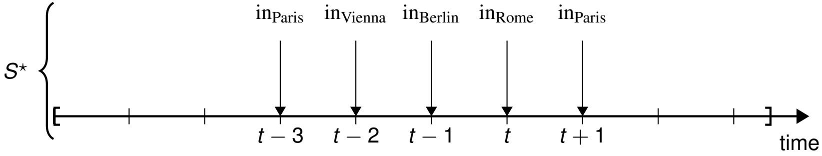
Formula	LARS
constant	$\bot, \top$
p	true, false
$\neg\alpha$	true $\leftrightarrow$ false
$\alpha \wedge \beta$	α and β
$\alpha \vee \beta$	α or β
$\Box\alpha$	for all t: α
$\Diamond\alpha$	exists t: α
$@_{t'}\alpha$	$(S^\star, S, t)$ changes to $(S^\star, S, t')$
$\boxplus^{w}\alpha$	$(S^\star, S, t)$ changes to $(S^\star, w(S, t), t)$
$\triangleright\alpha$	$(S^\star, S, t)$ changes to $(S^\star, S^\star, t)$

Weighted LARS

- Evaluate formulas w.r.t. $(S^\star, S, t)$
- $S^\star$ starting stream, S current stream, t time

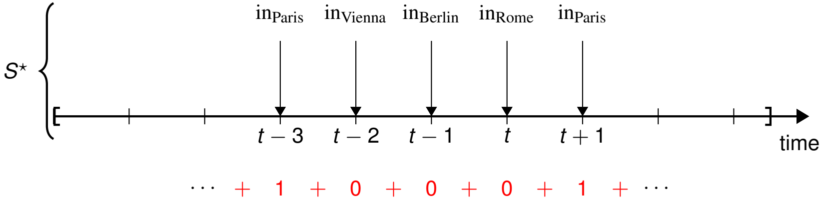
Formula	LARS	Weighted LARS
constant	$\bot, \top$	semiring value k
p	true, false	one, zero
$\neg\alpha$	true $\leftrightarrow$ false	zero $\rightarrow$ one, rest $\rightarrow$ zero
$\alpha \wedge \beta$	α and β	α times β
$\alpha \vee \beta$	α or β	α plus β
$\Box\alpha$	for all t: α	product of α over t
$\Diamond\alpha$	exists t: α	sum of α over t
$@_{t'}\alpha$	$(S^\star, S, t)$ changes to $(S^\star, S, t')$	
$\boxplus^w\alpha$	$(S^\star, S, t)$ changes to $(S^\star, w(S, t), t)$	
$\triangleright\alpha$	$(S^\star, S, t)$ changes to $(S^\star, S^\star, t)$	

Example I



- How often does Peter visit Paris?

Example I



- How often does Peter visit Paris?

$\rightarrow$ $\Diamond \text{in}_{\text{Paris}}$ over the natural number semiring $(\mathbb{N}, +, \cdot, 0, 1)$

LARS measure

- Goal: Assign answer streams a weight using α

LARS measure

- Goal: Assign answer streams a weight using α
- A *LARS Measure* μ is defined by a triple $\langle \Pi, \alpha, \mathcal{R} \rangle$, where
  - Π is a LARS program
  - α is a weighted LARS formula over $\mathcal{R}$
  - $\mathcal{R}$ is a semiring

LARS measure

- Goal: Assign answer streams a weight using α
- A *LARS Measure* μ is defined by a triple $\langle \Pi, \alpha, \mathcal{R} \rangle$, where
  - Π is a LARS program
  - α is a weighted LARS formula over $\mathcal{R}$
  - $\mathcal{R}$ is a semiring
- We set

$$\mu(S, t) = \begin{cases} [\![\alpha]\!]_{\mathcal{R}}(S, S, t) & \text{if } S \text{ is an answer stream}^{1} \text{of } \Pi \text{ at } t, \\ e_{\oplus} & \text{otherwise.} \end{cases}$$

[1] S is an answer stream of Π at t if (S, S, t) satisfies Π and (S, S, t) is a minimal model of the reduct $\Pi^{S,t} = \{\alpha \leftarrow \beta \in \Pi \mid (S, S, t) \text{ satisfies } \beta\}$

Expressivity Results I

LARS measures can be employed for . . .

- Preferential Reasoning, i.e.,
  choosing optimal answer streams w.r.t. some criteria
- Probabilistic Reasoning, i.e.,
  assigning answer streams a probability
- Weighted Model Counting, i.e.,
  aggregating the weights of all answer streams

Expressivity Results I

... and enable subsumption of corresponding ASP-extensions

- Preferential Reasoning
  Weak Constraints [Buccafurri *et al.*, 2000]
  (part.) *asprin* [Brewka *et al.*, 2015]
- Probabilistic Reasoning
  P-log [Baral *et al.*, 2009]
  LP^{MLN} [Lee and Yang, 2017]
  ProbLog [De Raedt *et al.*, 2007]
- Weighted Model Counting
  aProbLog [Kimmig *et al.*, 2011]

Expressivity Results II

- A plain fragment of LARS measures is expressively equivalent to
  - Weighted Automata (Finite State Machines with weighted transition function)
  - Rational Expressions (Regular Expressions with weights)

Expressivity Results II

- A plain fragment of LARS measures is expressively equivalent to
  - Weighted Automata (Finite State Machines with weighted transition function)
  - Rational Expressions (Regular Expressions with weights)
- Shows the expressiveness of LARS measures

Expressivity Results II

- A plain fragment of LARS measures is expressively equivalent to
  - Weighted Automata (Finite State Machines with weighted transition function)
  - Rational Expressions (Regular Expressions with weights)
- Shows the expressiveness of LARS measures
- Gives a rule-based alternative for specification via automata

Complexity Results

The evaluation of LARS measures

- is PSPACE-hard for any non-trivial semiring (LARS is already PSPACE-complete)
- possible in FPSPACE(poly) for under mild restrictions on the semiring and weighted formula

Complexity Results

The evaluation of LARS measures

- is PSPACE-hard for any non-trivial semiring (LARS is already PSPACE-complete)
- possible in FPSPACE(poly) for under mild restrictions on the semiring and weighted formula

Preferential Reasoning (over a restricted LARS measures)

- Preference Checking is Π_2^p-complete
- Brave Preferential Reasoning is Σ_3^p-complete

Conclusion

- LARS enables expressive stream reasoning
- Weighted LARS and LARS measures as a general underlying framework for quantitative stream reasoning
- Lift quantitative LP-extensions to the streaming context
- Restrictions on LARS measures can tame the complexity

Conclusion

- LARS enables expressive stream reasoning
- Weighted LARS and LARS measures as a general underlying framework for quantitative stream reasoning
- Lift quantitative LP-extensions to the streaming context
- Restrictions on LARS measures can tame the complexity
- ASP($\mathcal{AC}$) at ICLP 2020
  - Weighted formulas inside programs

References I

Chitta Baral, Michael Gelfond, and Nelson Rushton.
Probabilistic reasoning with answer sets.
Theory and Practice of Logic Programming, 9(1):57–144, 2009.

Gerhard Brewka, James Delgrande, Javier Romero, and Torsten Schaub.
asprin: Customizing answer set preferences without a headache.
In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

References II

Francesco Buccafurri, Nicola Leone, and Pasquale Rullo.
Enhancing disjunctive datalog by constraints.
IEEE Transactions on Knowledge and Data Engineering, 12(5):845–860, 2000.

Luc De Raedt, Angelika Kimmig, and Hannu Toivonen.
Problog: A probabilistic prolog and its application in link discovery.
In *IJCAI*, volume 7, pages 2462–2467. Hyderabad, 2007.

Angelika Kimmig, Guy Van den Broeck, and Luc De Raedt.
An algebraic prolog for reasoning about possible worlds.
In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

References III

Joohyung Lee and Zhun Yang.
Lpmln, weak constraints, and p-log.
In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.