WEIGHTED LARS AND WEIGHTED AUTOMATA

Rafael Kiesel

Vienna University of Technology, Vienna, Austria

rafael.kiesel@tuwien.ac.at

funded by FWF project W1255-N23

Weighted LARS

Definition 1 (Weighted LARS Syntax and Semantics) *A* weighted LARS formula *over a semiring* $\mathcal{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$ *is of the form*

$$\alpha ::= k \mid p \mid \neg p \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Diamond\alpha \mid \Box\alpha \mid @_t\alpha \mid \boxplus^w\alpha,$$

where $k \in R$, $p \in \mathcal{P}$ *and* $w \in W$.
Given a weighted LARS formula α*, a pair* (S, t) *of a stream* $S = (T, v)$ *and a timepoint* t *and a semiring* $\mathcal{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$*, the semantics of* α *w.r.t.* (S, t) *and* $\mathcal{R}$ *are defined inductively by*

$$[\![k]\!]_\mathcal{R}(S,t) = k, \text{ for } k \in R \qquad [\![@_{t'}\alpha]\!]_\mathcal{R}(S,t) = [\![\alpha]\!]_\mathcal{R}(S,t')$$

$$[\![p]\!]_\mathcal{R}(S,t) = \begin{cases} e_\otimes, & \text{if } p \in v(t) \\ e_\oplus, & \text{otherwise.} \end{cases}, \text{ for } p \in \mathcal{P} \qquad [\![\Diamond\alpha]\!]_\mathcal{R}(S,t) = \bigoplus_{t' \in T}[\![\alpha]\!]_\mathcal{R}(S,t')$$

$$[\![\neg p]\!]_\mathcal{R}(S,t) = \begin{cases} e_\oplus, & \text{if } p \in v(t) \\ e_\otimes, & \text{otherwise.} \end{cases}, \text{ for } p \in \mathcal{P} \qquad [\![\Box\alpha]\!]_\mathcal{R}(S,t) = \bigotimes_{t' \in T}[\![\alpha]\!]_\mathcal{R}(S,t')$$

$$[\![\alpha \wedge \beta]\!]_\mathcal{R}(S,t) = [\![\alpha]\!]_\mathcal{R}(S,t) \otimes [\![\beta]\!]_\mathcal{R}(S,t) \qquad [\![\boxplus^w\alpha]\!]_\mathcal{R}(S,t) = [\![\alpha]\!]_\mathcal{R}(\boxplus^w(S,t),t)$$

$$[\![\alpha \vee \beta]\!]_\mathcal{R}(S,t) = [\![\alpha]\!]_\mathcal{R}(S,t) \oplus [\![\beta]\!]_\mathcal{R}(S,t)$$

Definition 2 (LARS Measure) *A* LARS measure $\langle \Pi, \alpha, \mathcal{R}\rangle$ *consists of a LARS program* Π*, a weighted LARS formula* α *and a semiring* $\mathcal{R}$*. The weight of a stream* S *for* Π *at time* t *is then defined by*

$$\mu(S,t) = [\![\alpha]\!]_\mathcal{R}(S,t).$$

We extend this definition to a data stream D *and time* t *via*

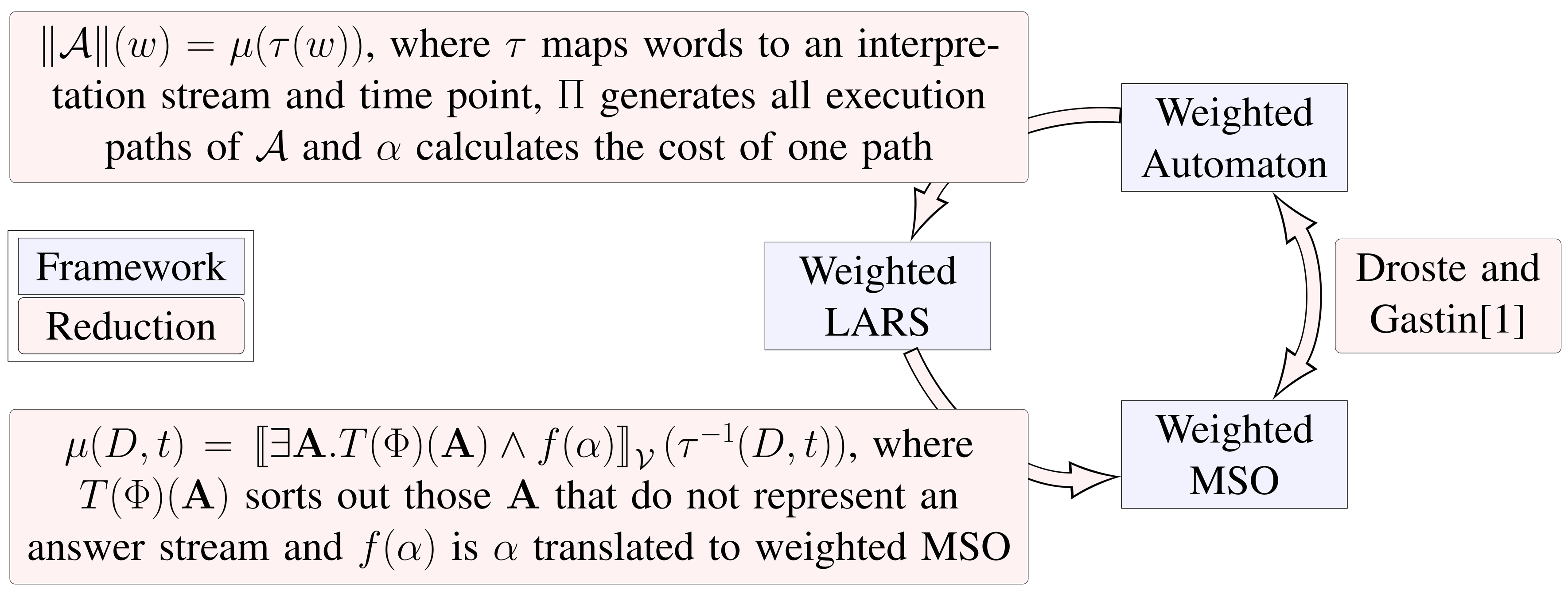
$$\mu(D,t) = \bigoplus_{S \in \mathcal{AS}(\Pi,D,t)} \mu(S,t).$$

Weighted Automaton

Definition 3 (Weighted Automaton[1]) *A* weighted automaton $\mathcal{A}$ *over a finite alphabet* A *is a tuple* $\langle Q, \lambda, \delta, \gamma\rangle$*, where* Q *is a finite set of states,* $\lambda, \gamma : Q \to R$ *and* $\delta : A^* \to R^{Q\times Q}$ *a monoid homomorphism, for some semiring* $\mathcal{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$*. Its* behaviour *is defined as*

$$\|\mathcal{A}\| : A^* \to R, w \mapsto \bigoplus_{q,q' \in Q} \lambda(q) \otimes \delta(w)_{q,q'} \otimes \gamma(q') = \lambda\delta(w)\gamma.$$

Proof Idea for Equivalence



Example: Travelling Problem

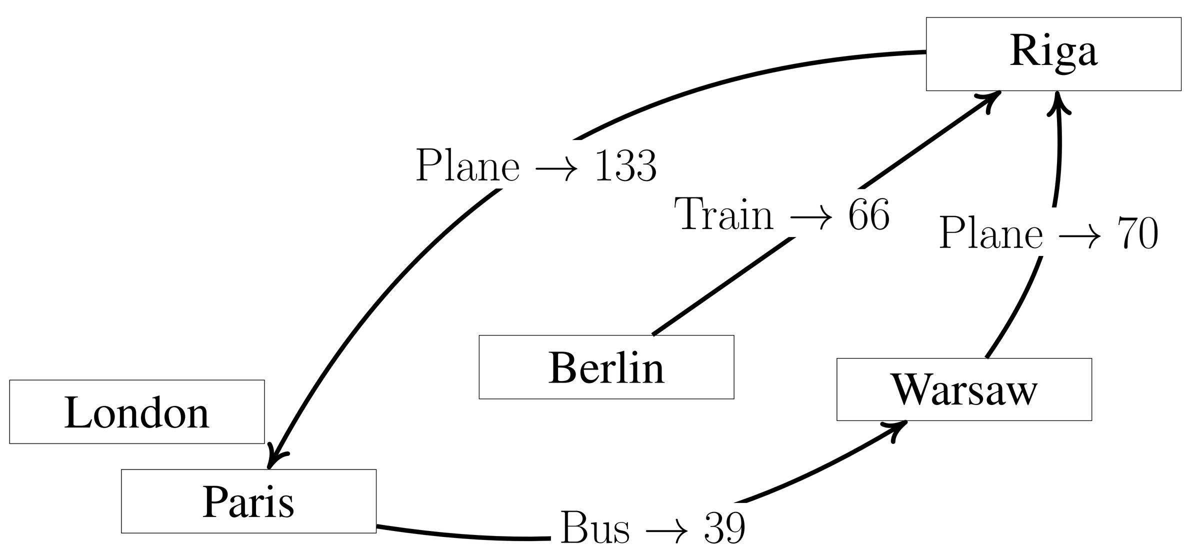
We want to arrange a journey which satisfies the following constraints

- Travel from S to E while staying within a set of cities Cities
- Only travel using transportation from the set Transports
- For the i^{th} trip we use t_i, the i^{th} element of a sequence $t \in \text{Transports}^n$
- Every trip changes the city

We want to find out the minimal cost of such a journey given the cost function

$$\text{Cost} : \text{Cities} \times \text{Transport} \times \text{Cities} \to \mathbb{R}, (\text{from}, \text{trans}, \text{to}) \mapsto \text{Cost}(\text{from}, \text{trans}, \text{to})$$

which is ∞ when we cannot travel between two cities using a mean of transportation



Possible journey, with costs, for

$$\text{Transports} = \{\text{Train}, \text{Bus}, \text{Plane}\}$$
$$\text{Cities} = \{\text{London}, \text{Paris}, \text{Berlin}, \text{Warsaw}, \text{Riga}\}$$

given that

$$S = \text{Berlin}, E = \text{Riga}$$

and

$$t = (\text{Train}, \text{Plane}, \text{Bus}, \text{Plane}).$$

Travelling Problem: Weighted LARS

We can solve the travelling problem using a LARS measure. We first construct a LARS program to generate possible journeys.

$$\Box \bigvee_{\text{city} \in \text{Cities}} \text{in}(\text{city}) \leftarrow \top, \qquad \bot \leftarrow \Diamond \bigvee_{\text{city1},\text{city2} \in \text{Cities}, \text{city1} \neq \text{city2}} \text{in}(\text{city1}) \wedge \text{in}(\text{city2})$$

The two above rules ensure that we are in exactly one city at each timepoint. Further we need to model the fact that one can not travel between some cities using some means of transport and that we always change city. Therefore we add the rules

$$\bot \leftarrow \bigvee_{\text{from},\text{to} \in \text{Cities}, \text{trans} \in \text{Transports}} @_T\text{in}(\text{from}) \wedge @_{T+1}\text{in}(\text{to}) \wedge \neg\text{possible}(\text{from}, \text{trans}, \text{to}) \wedge \text{travel}(\text{trans}),$$

$$\bot \leftarrow \bigvee_{\text{city} \in \text{Cities}} @_T\text{in}(\text{city}) \wedge @_{T+1}\text{in}(\text{city}).$$

In order to model the fact that we know the starting and end city we use

$$@_0\text{in}(S) \leftarrow \top, \qquad \bot \leftarrow \Diamond(\boxplus^{\text{next}}\Box\bot \wedge \text{in}(S)).$$

We further give a formula α which when evaluated over the semiring $\mathcal{R} = ([0,\infty], \min, +, \infty, 0)$ that gives us the minimum amount of money we need to spend. We choose

$$\alpha = \Box \bigvee_{\text{from},\text{to} \in \text{Cities}, \text{trans} \in \text{Transports}} \big(\text{travel}(\text{trans}) \wedge \text{in}(\text{from}) \wedge \boxplus^{\text{next}}\Diamond\text{in}(\text{to}) \wedge \text{Cost}(\text{from}, \text{trans}, \text{to})\big) \vee \boxplus^{\text{next}}\Box\bot.$$

For the LARS program Π containing the rules above, the LARS measure defined by $\langle \Pi, \alpha, \mathcal{R}\rangle$ gives us the minimal amount of money we need to spend, for a given data stream D, which encodes the sequence t of transports to be used.

Travelling Problem: Weighted Automaton

We can solve the travelling problem using a weighted automaton. We choose the alphabet $A = \text{Transports}$, and use the cities as states $Q = \text{Cities}$. We further define the transition function as $\delta(\text{trans})_{\text{from},\text{to}} = \text{Cost}(\text{from}, \text{trans}, \text{to})$ and set the value to ∞ when travel by trans is impossible or from $=$ to. We can now use the weighted automaton

$$\mathcal{A} = \langle Q, \lambda, \delta, \gamma\rangle$$

over alphabet A, where

$$\lambda(q) = \begin{Bmatrix} 0 & \text{if } q = S, \\ \infty & \text{otherwise} \end{Bmatrix}, \gamma(q) = \begin{Bmatrix} 0 & \text{if } q = E \\ \infty & \text{otherwise} \end{Bmatrix}$$

in order to model the problem of having a given sequence $t \in A^*$ of means of transportation that we want to use one after the other and finding the minimum price. In order to obtain the minimum we need to use the semiring

$$([0,\infty], \min, +, \infty, 0)$$

since then the costs for the transportation from one city to the next are added and then the minimum cost of a path from S to E is chosen. For example

$$\begin{aligned} \|\mathcal{A}\|((\text{bus}, \text{boat})) &= \bigoplus_{q,q' \in Q} \lambda(q) \otimes \delta((\text{bus}, \text{boat}))_{q,q'} \otimes \gamma(q') \\ &= \min_{q,q' \in Q} \lambda(q) + \delta((\text{bus}, \text{boat}))_{q,q'} + \gamma(q') \\ &= \lambda(S) + \min_{q^* \in Q} \delta(\text{bus})_{S,q^*} + \delta(\text{boat})_{q^*,E} + \gamma(E) \\ &= \min_{q^* \in Q} \delta(\text{bus})_{S,q^*} + \delta(\text{boat})_{q^*,E} \end{aligned}$$

would give us the minimal cost of going to E from S, when one first takes a bus and then a boat.

References

[1] Droste, M., Gastin, P.: Weighted automata and weighted logics. Theoretical Computer Science **380**(1), 69 (2007)

logics LOGICAL METHODS IN COMPUTER SCIENCE

FWF Der Wissenschaftsfonds.