Weighted Timed MSO Logics

Karin Quaas

Institut für Informatik, Universität Leipzig
04009 Leipzig, Germany
quaas@informatik.uni-leipzig.de

Abstract. We aim to generalize Büchi's fundamental theorem on the coincidence of recognizable and MSO-definable languages to a weighted timed setting. For this, we investigate subclasses of weighted timed automata and show how we can extend existing timed MSO logics with weights. Here, we focus on the class of weighted event-recording automata and define a weighted extension of the full logic $\mathrm{MSO_{er}}(\Sigma)$ introduced by D'Souza. We show that every weighted event-recording automaton can effectively be transformed into a corresponding sentence of our logic and vice versa. The methods presented in the paper can be adopted to weighted versions of timed automata and Wilke's logic of relative distance. The results indicate the robustness of weighted timed automata models and may be used for specification purposes.

Introduction

Recently, the model of *weighted timed automata* has received much attention in the real-time community as it can be used to model continuous consumption of resources [2, 3, 5, 4, 11]. The goal of this paper is to generalize Büchi's and Elgot's fundamental theorems about the coincidence of languages *recognizable* by finite automata and languages *definable* by sentences in a monadic second-order (MSO) logic [6, 15] to weighted timed automata. For this, we introduce a weighted timed MSO logic, which may be used for specifying *quantitative* aspects of timed automata, e.g. *how often* a certain property is satisfied by the system.

In this paper, we focus on a weighted version of *event-recording automata*, a subclass of timed automata introduced by Alur et al. [1]. Recent results on event-recording automata include works on alternative characterizations using regular expressions [7] and MSO logic [14], real-time logics [25, 17], and inference/learning [16]. The main advantage of event-recording automata is that they - as opposed to timed automata - always can be determinized. This simplifies some of our constructions compared to the ones necessary for the class of weighted timed automata.

Our work is motivated by recent works on *weighted logics* by Droste and Gastin [8, 10]. The authors introduce a weighted MSO logic for characterizing the behaviour of weighted automata defined over a semiring. They extend classical MSO logic with formulas of the form k (for k an element of the semiring), which may be used to define the weight of a transition of a weighted automaton. They show that the behaviour of weighted automata coincides with the semantics of sentences of a fragment of the logic. Recently, this result has been generalized to weighted settings of infinite words [12], trees [13], pictures [20], traces [21], texts [18] and nested words [19].

Here, we aim to generalize the result to a weighted timed setting. The basis of our work is the MSO logic $\mathrm{MSO}_{\mathrm{er}}(\Sigma)$ introduced by D'Souza and used for the logical characterization of event-recording automata [14]. We extend it with two kinds of weighted formulas whose semantics correspond to the weights of edges and locations, respectively, in weighted event-recording automata. For proving a Büchi-type theorem we show that for every sentence φ in our logic there is a weighted event-recording automaton whose behaviour corresponds to the semantics of φ and vice versa.

For this, we use parts of the proofs presented by Droste and Gastin [10]. However, in the weighted timed setting we are faced with two new problems. First, due to the weights assigned to locations, the Hadamard product, which is used for defining the semantics of conjunction in our logic, does not preserve recognizability. Second, there are formulas φ such that there are no weighted event-recording automata whose behaviours correspond to the semantics of $\forall x.\varphi$ and $\forall X.\varphi$, respectively. To overcome these problems, we define a suitable fragment of our logic, for which, with the support of some new notions and techniques, we are able to show the result.

1 (Weighted) Event-Recording Automata

Let Σ, $\mathbb{N}$ and $\mathbb{R}_{\geq 0}$ denote an alphabet, the natural numbers and the positive reals, respectively. A *timed word* is a finite sequence $(a_1, t_1)...(a_k, t_k) \in (\Sigma \times \mathbb{R}_{\geq 0})^*$ such that the sequence $\bar{t} = t_1...t_k$ of timestamps is non-decreasing. Sometimes we denote a timed word as above by $(\bar{a}, \bar{t})$, where $\bar{a} \in \Sigma^*$. We write $T\Sigma^*$ for the set of timed words over Σ. A set $L \subseteq T\Sigma^*$ is called a *timed language*. With Σ we associate a set $C_\Sigma = \{x_a | a \in \Sigma\}$ of *event-recording clock variables* ranging over $\mathbb{R}_{\geq 0}$. The variable x_a measures the time distance between the current event in a timed word and the last occuring a. Formally, given a timed word $w = (a_1, t_1)...(a_k, t_k)$, we let $\mathrm{dom}(w)$ be the set $\{1, ..., k\}$ and define for every $i \in \mathrm{dom}(w)$ a clock valuation function $\gamma_i^w : C_\Sigma \to \mathbb{R}_{\geq 0} \cup \{\perp\}$ by

$$\gamma_i^w(x_a) = \begin{cases} t_i - t_j & \text{if there exists a } j \text{ such that } 1 \leq j < i \text{ and } a_j = a, \\ & \text{and for all } m \text{ with } j < m < i \text{, we have } a_m \neq a \\ \perp & \text{otherwise.} \end{cases}$$

We further use $|w|$ to denote the length of w. We define *clock constraints* ϕ over C_Σ to be conjunctions of formulas of the form $x = \perp$ or $x \sim c$, where $x \in C_\Sigma$, $c \in \mathbb{N}$, and $\sim \in \{<, \leq, =, \geq, >\}$. We use $\Phi(C_\Sigma)$ to denote the set of all clock constraints over C_Σ. A clock valuation γ_i^w satisfies ϕ, written $\gamma_i^w \models \phi$, if ϕ evaluates to true according to the values given by γ_i^w. An *event-recording automaton (ERA)* over Σ is a tuple $\mathcal{A} = (S, S_0, S_f, E)$, where

- S is a finite set of locations (states)
- $S_0 \subseteq S$ is a set of initial locations
- $S_f \subseteq S$ is a set of final locations
- $E \subseteq S \times \Sigma \times \Phi(C_\Sigma) \times S$ is a finite set of edges.

For w as above, we let a *run* of $\mathcal{A}$ on w be a finite sequence $s_0 \xrightarrow{e_1} s_1 \xrightarrow{e_2} \ldots \xrightarrow{e_k} s_k$ of edges $e_i = (s_{i-1}, a_i, \phi_i, s_i) \in E$ such that $\gamma_i^w \models \phi_i$ for all $1 \leq i \leq k$. We say that a run r is *successful* if $s_0 \in S_0$ and $s_k \in S_f$. We define the timed language $L(\mathcal{A}) = \{w \in T\Sigma^* |$ there is a successful run of $\mathcal{A}$ on $w\}$. We say that a timed language $L \subseteq T\Sigma^*$ is *recognizable over* Σ if there is an ERA $\mathcal{A}$ over Σ such that $L(\mathcal{A}) = L$.

Remark 1. The methods presented in this paper can easily be extended to event-clock automata additionally equipped with *event-predicting* clock variables [1].

An ERA $\mathcal{A}$ is *deterministic* if $|S_0| = 1$ and whenever (s, a, ϕ_1, s_1) and (s, a, ϕ_2, s_2) are two different edges in $\mathcal{A}$, then for all clock valuations γ we have $\gamma \not\models \phi_1 \wedge \phi_2$. A timed language is called *deterministically recognizable over* Σ if there is a deterministic ERA over Σ recognizing it.

Proposition 1. *[1] The class of recognizable timed languages is closed under boolean operations and equal to the class of deterministically recognizable timed languages.*

We extend ERA to be equipped with weights taken from a commutative semiring. For this, we let $\mathcal{K}$ be a *commutative semiring*, i.e., an algebraic structure $\mathcal{K} = (K, +, \cdot, 0, 1)$ such that $(K, +, 0)$ and $(K, \cdot, 1)$ are commutative monoids, multiplication distributes over addition and 0 is absorbing. As examples consider the semiring of natural numbers $(\mathbb{N}, +, \cdot, 0, 1)$, the Boolean semiring $(\{0, 1\}, \vee, \wedge, 0, 1)$ and the tropical semiring $(\mathbb{R}_{\geq 0} \cup \{\infty\}, \min, +, \infty, 0)$. Furthermore, we let $\mathcal{F}$ be a family of functions from $\mathbb{R}_{\geq 0}$ to $\mathcal{K}$. For instance, if $\mathcal{K}$ is the tropical semiring, $\mathcal{F}$ may be the family of linear functions of the form $\mu(\delta) = k \cdot \delta$ mapping every $\delta \in \mathbb{R}_{\geq 0}$ to a value $k \cdot \delta$ in K (for some $k \in \mathbb{R}_{\geq 0}$). Given $f_1, f_2 \in \mathcal{F}$, we define the pointwise product $f_1 \odot f_2$ of f_1 and f_2 by $(f_1 \odot f_2)(\delta) = f_1(\delta) \cdot f_2(\delta)$.

A *weighted event-recording automaton* (WERA) over Σ, $\mathcal{K}$ and $\mathcal{F}$ is a tuple $\mathcal{A} = (S, S_0, S_f, E, C)$ such that (S, S_0, S_f, E) is an ERA over Σ and $C = \{C_\mathcal{E}\} \cup \{C_s | s \in S\}$ is a cost function, where $C_\mathcal{E} : E \to K$ assigns a weight to each edge, and $C_s \in \mathcal{F}$ gives us the weight for staying in location s per time unit for each $s \in S$. A WERA $\mathcal{A}$ maps to each timed word $w \in T\Sigma^*$ a weight in K as follows: first, we define the *running weight* $rwt(r)$ of a run r as above to be $\prod_{i \in \mathrm{dom}(w)} C_{s_{i-1}}(t_i - t_{i-1}) \cdot C_\mathcal{E}(e_i)$, where $t_0 = 0$. The running weight of the empty run is defined to be $1 \in K$. Then, the *behaviour* $\|\mathcal{A}\| : T\Sigma^* \to K$ of $\mathcal{A}$ is given by $(\|\mathcal{A}\|, w) = \sum\{rwt(r) : r$ is a successful run of $\mathcal{A}$ on $w\}$. A function $\mathcal{T} : T\Sigma^* \to K$ is called a *timed series*. A timed series $\mathcal{T}$ is said to be *recognizable over* $\mathcal{K}$, Σ *and* $\mathcal{F}$ if there is a WERA $\mathcal{A}$ over $\mathcal{K}$, Σ and $\mathcal{F}$ such that $\|\mathcal{A}\| = \mathcal{T}$.

We define the function $\mathbb{1} : \mathbb{R}_{\geq 0} \to K$ by $\delta \mapsto 1$ for every $\delta \in \mathbb{R}_{\geq 0}$. In the following, we fix a commutative semiring $\mathcal{K}$ and a family $\mathcal{F}$ of cost functions from $\mathbb{R}_{\geq 0}$ to K containing $\mathbb{1}$.

For $L \subseteq T\Sigma^*$, the *characteristic series* 1_L is defined by $(1_L, w) = 1$ if $w \in L$, 0 otherwise. Notice that an ERA over Σ can be seen as a WERA over the Boolean semiring, Σ and the family of constant functions. The timed series recognized by such a WERA is the characteristic series $1_{L(\mathcal{A})}$. However, due to the determinizability of ERA, $1_{L(\mathcal{A})}$ can also be recognized over arbitrary semirings:

Lemma 1. *If $L \subseteq T\Sigma^*$ is recognizable over Σ, then 1_L is recognizable over $\mathcal{K}$, Σ and $\mathcal{F}$.*

Given timed series $\mathcal{T}, \mathcal{T}_1, \mathcal{T}_2$ and $k \in K$, we define the *sum* $\mathcal{T}_1 + \mathcal{T}_2$, the *Hadamard product* $\mathcal{T}_1 \odot \mathcal{T}_2$ and the *scalar products* $k \cdot \mathcal{T}$ and $\mathcal{T} \cdot k$ pointwise, i.e., by $(\mathcal{T}_1 + \mathcal{T}_2, w) = (\mathcal{T}_1, w) + (\mathcal{T}_2, w)$, $(\mathcal{T}_1 \odot \mathcal{T}_2, w) = (\mathcal{T}_1, w) \cdot (\mathcal{T}_2, w)$, $(k \cdot \mathcal{T}, w) = k \cdot (\mathcal{T}, w)$ and $(\mathcal{T} \cdot k, w) = (\mathcal{T}, w) \cdot k$ respectively. If $\mathcal{K}$ is the Boolean semiring, $+$ and $\odot$ correspond to the union and intersection of timed languages, respectively.

Later in the paper, we need closure properties of recognizable timed series under these operations. It can be shown in a straightforward manner that sum and scalar products preserve recognizability of timed series.

Lemma 2. *Recognizable timed series over $\mathcal{K}$, Σ and $\mathcal{F}$ are closed under $+$, $k\cdot$ and $\cdot k$ (for $k \in K$).*

In contrast to this, in general recognizable timed series are not closed under the Hadamard product. We illustrate this in the next example.

Example 1. Let $\mathcal{K} = (\mathbb{R}_{\geq 0} \cup \{\infty\}, \min, +, \infty, 0)$, $\Sigma = \{a\}$ and $\mathcal{F}$ be the family of linear functions $C : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. We define the WERA $\mathcal{A}^i$ over $\mathcal{K}$, Σ and $\mathcal{F}$ for each $i = 1, 2$ by $\mathcal{A}^i = (\{p^i, q^i\}, \{p^i\}, \{q^i\}, \{(p^i, a, true, q^i)\}, C^i)$ with $C^i_{\mathcal{E}}((p^i, a, true, q^i)) = 0$, $C^i_{q^i}(\delta)$ arbitrary, $C^1_{p^1}(\delta) = 2 \cdot \delta$ and $C^2_{p^2}(\delta) = 3 \cdot \delta$. Let $w \in T\Sigma^*$. If $w \neq (a, t)$ for some $t \in \mathbb{R}_{\geq 0}$, then $(\|\mathcal{A}^i\|, w) = 0$ for each $i = 1, 2$ and thus $(\|\mathcal{A}^1\| \odot \|\mathcal{A}^2\|, w) = 0$. So let $w = (a, t)$ for some $t \in \mathbb{R}_{\geq 0}$. Then we have $(\|\mathcal{A}^1\| \odot \|\mathcal{A}^2\|, w) = 2 \cdot t + 3 \cdot t = 5 \cdot t$. Clearly, this timed series is recognizable over the family of linear functions. If $\mathcal{K}$ and $\mathcal{F}$ are as above, for building a WERA recognizing the Hadamard product of the behaviours of two given WERA, we can use the usual product automaton construction together with defining a cost function such that the cost of each edge and location equals the pointwise product of the costs of the two corresponding edges and locations in the original WERA. This can be done since the pointwise product of each pair of linear functions is a linear function and thus in $\mathcal{F}$. However, this is not always the case. For instance, assume that $\mathcal{A}^i$ are WERA over the semiring $(\mathbb{R}_{\geq 0}, +, \cdot, 0, 1)$. Then, we have $(\|\mathcal{A}^1\| \odot \|\mathcal{A}^2\|, w) = 2 \cdot t \cdot 3 \cdot t = 6 \cdot t^2$. It can be easily seen that there is no WERA $\mathcal{A}$ over the family $\mathcal{F}$ of **linear** functions such that $\|\mathcal{A}\| = \|\mathcal{A}^1\| \odot \|\mathcal{A}^2\|$.

For this reason, we define the notion of *non-interfering* timed series. So for $i = 1, 2$, let $\mathcal{A}^i = (S^i, S^i_0, S^i_f, E^i, C^i)$ be two WERA. We say that $\mathcal{A}^1$ and $\mathcal{A}^2$ are *non-interfering* if for all pairs $(s_1, s_2) \in S^1 \times S^2$, whenever there is a run from (s_1, s_2) into $S^1_f \times S^2_f$, then $C^1_{s_1} = \mathbb{1}$ or $C^2_{s_2} = \mathbb{1}$. Observe that this implies

$C^1_{s_1} \odot C^2_{s_2} \in \mathcal{F}$. This enables us to use a product automaton construction for building a WERA recognizing $\|\mathcal{A}^1\| \odot \|\mathcal{A}^2\|$. Also notice that the premise of the condition is decidable for the whole class of weighted timed automata [2]. Two timed series $\mathcal{T}_1$ and $\mathcal{T}_2$ are non-interfering over $\mathcal{K}$, Σ and $\mathcal{F}$ if there are non-interfering WERA $\mathcal{A}^1$ and $\mathcal{A}^2$ over $\mathcal{K}$, Σ and $\mathcal{F}$ with $\|\mathcal{A}^i\| = \mathcal{T}_i$ for $i = 1, 2$.

Lemma 3. *1. If for all $f_1, f_2 \in \mathcal{F}$ we have $f_1 \odot f_2 \in \mathcal{F}$, then recognizable timed series over $\mathcal{K}$, Σ and $\mathcal{F}$ are closed under $\odot$.*

2. If $\mathcal{T}_1$ and $\mathcal{T}_2$ are non-interfering over $\mathcal{K}$, Σ and $\mathcal{F}$, then $\mathcal{T}_1 \odot \mathcal{T}_2$ is recognizable over $\mathcal{K}$, Σ and $\mathcal{F}$.

2 Weighted Timed MSO Logic

Next, we introduce a weighted timed MSO logic for specifying properties of timed series. Our logic is an extension of the logic $\mathrm{MSO}_{\mathrm{er}}(\Sigma)$ introduced by D'Souza, which we briefly recall here. Formulas of $\mathrm{MSO}_{\mathrm{er}}(\Sigma)$ are built inductively from atomic formulas $P_a(x)$, $x = y$, $x < y$, $x \in X$, $\lhd_a(x) \sim c$ using the connectives $\vee$, $\neg$, $\exists x.$ and $\exists X.$, where x, y are first-order variables, X is a second-order variable, $a \in \Sigma$, $c \in \mathbb{N}$ and $\sim \in \{<, \leq, =, \geq, >\}$ or $(\sim c) = (= \perp)$. As usual, we may also use $\rightarrow$, $\leftrightarrow$, $\wedge$, $\forall x.$ and $\forall X.$ as abbreviations. Formulas of $\mathrm{MSO}_{\mathrm{er}}(\Sigma)$ are interpreted over timed words. For this, we associate with $w = (a_1, t_1)...(a_k, t_k)$ the relational structure consisting of the domain $\mathrm{dom}(w)$ together with the unary relations $P_a = \{i \in \mathrm{dom}(w) | a_i = a\}$ and $\lhd_a(.) \sim c = \{i \in \mathrm{dom}(w) | \gamma^w_i(x_a) \sim c\}$ as well as the usual $<$ and $=$ relations on $\mathrm{dom}(w)$. Now, for $\varphi \in \mathrm{MSO}_{\mathrm{er}}(\Sigma)$, let $\mathrm{Free}(\varphi)$ be the set of free variables, $\mathcal{V} \supseteq \mathrm{Free}(\varphi)$ be a finite set of first- and second-order variables, and σ be a $(\mathcal{V}, w)$-assignment mapping first-order (second-order, resp.) variables to elements (subsets, resp.) of $\mathrm{dom}(w)$. For $i \in \mathrm{dom}(w)$, we let $\sigma[x \rightarrow i]$ be the assignment that maps x to i and agrees with σ on every variable $\mathcal{V} \backslash \{x\}$. Similarly, we define $\sigma[X \rightarrow I]$ for any $I \subseteq \mathrm{dom}(w)$. A timed word $(\bar{a}, \bar{t})$ and a $(\mathcal{V}, (\bar{a}, \bar{t}))$-assignment σ is encoded as timed word $((\bar{a}, \sigma), \bar{t})$ over the extended alphabet $\Sigma_{\mathcal{V}}$. A timed word over $\Sigma_{\mathcal{V}}$ is written as $((\bar{a}, \sigma), \bar{t})$, where $\bar{a}$ is the projection over Σ and σ is the projection over $\{0, 1\}^{\mathcal{V}}$. Then, σ represents a *valid* assignment over $\mathcal{V}$ if for each first-order variable $x \in \mathcal{V}$, the x-row of σ contains exactly one 1. In this case, σ is identified with the $(\mathcal{V}, (\bar{a}, \bar{t}))$-assignment such that for every first-order variable $x \in \mathcal{V}$, $\sigma(x)$ is the position of the 1 in the x-row, and for each second-order variable $X \in \mathcal{V}$, $\sigma(X)$ is the set of positions with a 1 in the X-row. We define $N_{\mathcal{V}} = \{((\bar{a}, \sigma), \bar{t}) \in (T\Sigma_{\mathcal{V}})^* | \sigma$ is a valid $(\mathcal{V}, (\bar{a}, \bar{t}))$-assignment$\}$. The definition that $((\bar{a}, \sigma), \bar{t})$ satisfies φ, written $((\bar{a}, \sigma), \bar{t}) \models \varphi$, is as usual. We let $L_{\mathcal{V}}(\varphi) = \{((\bar{a}, \sigma), \bar{t}) \in N_{\mathcal{V}} | ((\bar{a}, \sigma), \bar{t}) \models \varphi\}$. The formula φ *defines* the timed language $L(\varphi) = L_{\mathrm{Free}(\varphi)}(\varphi)$. A timed language $L \subseteq T\Sigma^*$ is *$MSO_{er}(\Sigma)$-definable* if there exists a sentence $\varphi \in \mathrm{MSO}_{\mathrm{er}}(\Sigma)$ such that $L(\varphi) = L$.

Theorem 1. *[14] A timed language $L \subseteq T\Sigma^*$ is $MSO_{er}(\Sigma)$-definable if and only if L is recognizable over Σ.*

Now, we turn to the logic $\mathrm{MSO}_{\mathrm{er}}(\mathcal{K}, \Sigma, \mathcal{F})$, defined inductively as follows. The *atomic* formulas are formulas of the form $P_a(x)$, $x = y$, $x < y$, $x \in X$, $\lhd_a(x) \sim c$ and their negations, where $x, y, X, a, c, \sim$ are as above. Atomic formulas and formulas of the form k and $C_\mu(x)$, where $k \in K$ and $\mu \in \mathcal{F}$, can be combined using the operators $\wedge$, $\vee$, $\exists x.$, $\forall x.$, $\exists X.$ and $\forall X$. Notice that we only allow to apply negation to basic formulas. Let $\varphi \in \mathrm{MSO}_{\mathrm{er}}(\mathcal{K}, \Sigma, \mathcal{F})$ and $\mathcal{V} \supseteq \mathrm{Free}(\varphi)$. The $\mathcal{V}$-semantics of φ is a timed series $[\![\varphi]\!]_{\mathcal{V}} : (T\Sigma_{\mathcal{V}})^* \to K$. Let $(\bar{a}, \bar{t}) \in T\Sigma^*$. If σ is a valid $(\mathcal{V}, (\bar{a}, \bar{t}))$-assignment, $[\![\varphi]\!]_{\mathcal{V}}((\bar{a}, \sigma), \bar{t}) \in K$ is defined inductively as follows:

$$
\begin{aligned}
[\![\varphi]\!]_{\mathcal{V}}((\bar{a},\sigma),\bar{t}) &= 1_{L_{\mathcal{V}}(\varphi)}((\bar{a},\sigma),\bar{t}) \text{ if } \varphi \text{ is atomic} \\
[\![k]\!]_{\mathcal{V}}((\bar{a},\sigma),\bar{t}) &= k \\
[\![C_\mu(x)]\!]_{\mathcal{V}}((\bar{a},\sigma),\bar{t}) &= \mu(t_{\sigma(x)} - t_{\sigma(x)-1}) \\
[\![\varphi \vee \varphi']\!]_{\mathcal{V}}((\bar{a},\sigma),\bar{t}) &= [\![\varphi]\!]_{\mathcal{V}}((\bar{a},\sigma),\bar{t}) + [\![\varphi']\!]_{\mathcal{V}}((\bar{a},\sigma),\bar{t}) \\
[\![\varphi \wedge \varphi']\!]_{\mathcal{V}}((\bar{a},\sigma),\bar{t}) &= [\![\varphi]\!]_{\mathcal{V}}((\bar{a},\sigma),\bar{t}) \cdot [\![\varphi']\!]_{\mathcal{V}}((\bar{a},\sigma),\bar{t}) \\
[\![\exists x.\varphi]\!]_{\mathcal{V}}((\bar{a},\sigma),\bar{t}) &= \sum_{i \in \mathrm{dom}((\bar{a},\bar{t}))} [\![\varphi]\!]_{\mathcal{V}\cup\{x\}}((\bar{a},\sigma[x \to i]),\bar{t}) \\
[\![\forall x.\varphi]\!]_{\mathcal{V}}((\bar{a},\sigma),\bar{t}) &= \prod_{i \in \mathrm{dom}((\bar{a},\bar{t}))} [\![\varphi]\!]_{\mathcal{V}\cup\{x\}}((\bar{a},\sigma[x \to i]),\bar{t}) \\
[\![\exists X.\varphi]\!]_{\mathcal{V}}((\bar{a},\sigma),\bar{t}) &= \sum_{I \subseteq \mathrm{dom}((\bar{a},\bar{t}))} [\![\varphi]\!]_{\mathcal{V}\cup\{X\}}((\bar{a},\sigma[X \to I]),\bar{t}) \\
[\![\forall X.\varphi]\!]_{\mathcal{V}}((\bar{a},\sigma),\bar{t}) &= \prod_{I \subseteq \mathrm{dom}((\bar{a},\bar{t}))} [\![\varphi]\!]_{\mathcal{V}\cup\{X\}}((\bar{a},\sigma[X \to I]),\bar{t})
\end{aligned}
$$

For σ not a valid $(\mathcal{V}, (\bar{a}, \bar{t}))$-assignment, we define $[\![\varphi]\!]_{\mathcal{V}}((\bar{a}, \sigma), \bar{t}) = 0$. We write $[\![\varphi]\!]$ for $[\![\varphi]\!]_{\mathrm{Free}(\varphi)}$.

Remark 2. If we let $\mathcal{K}$ be the Boolean semiring, then $\mathrm{MSO}_{\mathrm{er}}(\Sigma)$ corresponds to $\mathrm{MSO}_{\mathrm{er}}(\mathcal{K}, \Sigma, \mathcal{F})$ as every formula in $\mathrm{MSO}_{\mathrm{er}}(\Sigma)$ is equivalent to a formula where negation is applied to atomic subformulas only. Also, every such formula $\varphi \in \mathrm{MSO}_{\mathrm{er}}(\Sigma)$ can be seen to be a formula of our logic.

Example 2. Consider the formula $\varphi = \exists x. \lhd_a(x) < 2$ and let $w = (a, 1.7)(b, 3.0)(a, 3.6)(a, 6.0)$. If we interpret φ as an $\mathrm{MSO}_{\mathrm{er}}(\mathcal{K}, \Sigma, \mathcal{F})$-formula over the Boolean semiring or, equivalently, as an $\mathrm{MSO}_{\mathrm{er}}(\Sigma)$-formula, we have $[\![\varphi]\!](w) = 1$. If on the other hand, we let $\mathcal{K}$ be the semiring over the natural numbers with ordinary addition and multiplication, we have $[\![\varphi]\!](w) = 2$, i.e., we *count* the number of positions x in w where $\lhd_a(x) < 2$ is satisfied. Counting how often a certain property holds gives rise to interesting applications in the field of verification.

Let $\mathcal{L} \subseteq \mathrm{MSO}_{\mathrm{er}}(\mathcal{K}, \Sigma, \mathcal{F})$. A timed series $\mathcal{T} : T\Sigma^* \to \mathcal{K}$ is called $\mathcal{L}$-*definable* if there is a sentence $\varphi \in \mathcal{L}$ such that $[\![\varphi]\!] = \mathcal{T}$. The goal of this paper is to find a suitable fragment $\mathcal{L}$ of $\mathrm{MSO}_{\mathrm{er}}(\mathcal{K}, \Sigma, \mathcal{F})$ such that $\mathcal{L}$-definable timed series precisely correspond to recognizable timed series over $\mathcal{K}$, Σ and $\mathcal{F}$, i.e., we want to generalize Theorem 1 to the weighted setting. It is not surprising that $\mathrm{MSO}_{\mathrm{er}}(\mathcal{K}, \Sigma, \mathcal{F})$ does not constitute a suitable candidate for $\mathcal{L}$ since this is already not the case in the untimed setting [9]. In the next section, we explain the problems that occur when we do not restrict the logic and step by step develop solutions resulting in the logic $\mathrm{sRMSO}_{\mathrm{er}}(\mathcal{K}, \Sigma, \mathcal{F})$ for which we are able to give the following Büchi-type theorem.

Theorem 2. *A timed series $\mathcal{T} : T\Sigma^* \to K$ is recognizable over $\mathcal{K}$, Σ and $\mathcal{F}$ if and only if $\mathcal{T}$ is definable by some sentence in $sRMSO_{er}(\mathcal{K}, \Sigma, \mathcal{F})$. The respective transformations can be done effectively provided that the operations of $\mathcal{K}$ and $\mathcal{F}$ are given effectively.*

3 From Logic To Automata

In this section, we want to prove the direction from right to left in Theorem 2 and show that for every formula φ of our weighted timed MSO logic, $[\![\varphi]\!]$ is a recognizable timed series. We do this similarly to the corresponding proof for the classical setting [27], i.e., by induction over the structure of the logic.

For the induction base, we show that for every atomic formula φ in $\mathrm{MSO}_{\mathrm{er}}(\mathcal{K}, \Sigma, \mathcal{F})$ there is a WERA recognizing $[\![\varphi]\!]$. For φ of the form $P_a(x)$, $x = y$, $x < y$, $x \in X$ and its negations, this can be done as in the classical setting [27]. In Fig.2, we give the WERA recognizing the timed series $[\![\varphi]\!]$ for φ being one of $\triangleleft_a(x) \sim c$, k and $C_\mu(x)$.

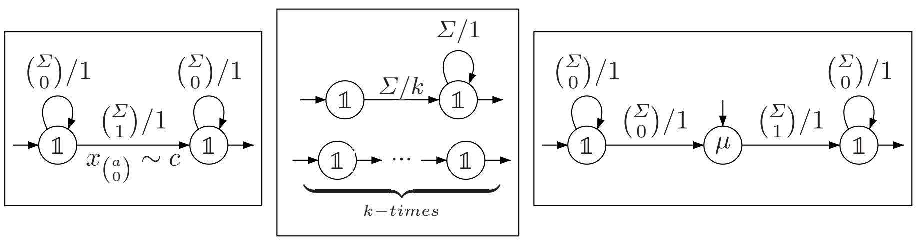

Figure 2. WERA with behaviours $[\![\triangleleft_a(x) \sim c]\!]$, $[\![k]\!]$ and $[\![C_\mu(x)]\!]$

For the induction step, we need to show closure properties of recognizable timed series under the constructs of the logic. For disjunction and existential quantification, we can give proofs very similar to the classical case (see Thomas for the case of formal languages [26] or Droste and Gastin for the case of (untimed) series [9, 10]). However, we will see that for the remaining operators of our logic, we cannot give easy extensions of the classical proofs.

First of all, in Sect.1 we have seen that recognizable timed series in general are not closed under the Hadamard product. Since the semantics of conjunction is defined using the Hadamard product, this means that we have to restrict the usage of conjunction. More precisely, we either have to require that $\mathcal{F}$ is such that for all $f_1, f_2 \in \mathcal{F}$ we have $f_1 \odot f_2 \in \mathcal{F}$, or we have to formulate a syntactical restriction implying that whenever two formulas φ_1 and φ_2 are combined by a conjunction, then $[\![\varphi_1]\!]$ and $[\![\varphi_2]\!]$ are non-interfering.

Lemma 4. *Let $\varphi_1, \varphi_2 \in MSO_{er}(\mathcal{K}, \Sigma, \mathcal{F})$ such that $[\![\varphi_1]\!]$ and $[\![\varphi_2]\!]$ are recognizable. Assume that whenever φ_1 contains the subformula $C_{\mu_1}(x_1)$ and φ_2 contains $C_{\mu_2}(x_2)$, then x_1, x_2 are free in both φ_1 and φ_2, and either φ_1 or φ_2 is of the form $\psi \wedge \neg(x_1 = x_2)$ for some $\psi \in MSO_{er}(\mathcal{K}, \Sigma, \mathcal{F})$. Then $[\![\varphi_1]\!]$ and $[\![\varphi_2]\!]$ are non-interfering.*

We give the intuition behind this lemma via an example. Consider the formula $C_{\mu_1}(x_1) \wedge C_{\mu_2}(x_2)$ and let $\mathcal{A}_i$ be a WERA such that $\|\mathcal{A}_i\| = [\![C_{\mu_i}(x_i)]\!]$ for each $i = 1, 2$ (see Fig.2). We use s_1 (s_2, resp.) to denote the location in $\mathcal{A}_1$ ($\mathcal{A}_2$, resp.) with cost function μ_1 (μ_2), resp.). We want to enforce that in the product automaton of $\mathcal{A}_1$ and $\mathcal{A}_2$, from the pair (s_1, s_2) there is no run to a final location. This is the case if from s_1 and s_2 no common letter can be read. Observe that from s_1 (s_2, resp.) every outgoing edge is labeled with (a, σ) such that $\sigma(x_1) = 1$ ($\sigma(x_2) = 1$, resp.) for every $a \in \Sigma$. Hence, in the product automaton every edge from (s_1, s_2) must be labeled with a letter of the form (a, σ) such that $\sigma(x_1) = \sigma(x_2) = 1$ for every $a \in \Sigma$. By requiring x_1 and x_2 to refer to different positions in a timed word, we can exclude that there is an edge from (s_1, s_2) labeled with a letter of this form. This is done by conjoining the formula above with $\neg(x_1 = x_2)$.

Second, examples [10] show that unrestricted application of $\forall x.$ and $\forall X.$ do not preserve recognizability. For instance, let $\mathcal{K} = (\mathbb{N}, +, \cdot, 0, 1)$ be the semiring of the natural numbers and $\mathcal{F}$ be the family of constant functions. We consider the formula $\varphi = \forall y. \exists x. C_{\mathbb{1}}(x)$. Then we have $[\![\varphi]\!](w) = |w|^{|w|}$. However, this cannot be recognized by any WERA as this timed series grows too fast (see [9] for a detailed proof which can also be applied to the timed setting). Similar examples can be given for $\forall X$. Hence, we need to restrict both the usage of $\forall x.$ and $\forall X.$ in our logic. We adopt the approach of Droste and Gastin [10].

For dealing with $\forall X.$, the idea is to restrict the application of $\forall X.$ to so-called *syntactically unambiguous* formulas. These are formulas $\varphi \in \mathrm{MSO}_{\mathrm{er}}(\Sigma)$ such that - even though interpreted over a semiring - the semantics $[\![\varphi]\!]$ of φ always equals 0 or 1[1]. We define the set of syntactically unambiguous formulas φ^+ and φ^- for $\varphi \in \mathrm{MSO}_{\mathrm{er}}(\Sigma)$ inductively as follows:

1. If φ is of the form $P_a(x)$, $x < y$, $x = y$, $x \in X$, $\triangleleft_a(x) \sim c$, then $\varphi^+ = \varphi$ and $\varphi^- = \neg\varphi$.
2. If $\varphi = \neg\psi$ then $\varphi^+ = \psi^-$ and $\varphi^- = \psi^+$.
3. If $\varphi = \psi \vee \zeta$ then $\varphi^+ = \psi^+ \vee (\psi^- \wedge \zeta^+)$ and $\varphi^- = \psi^- \wedge \zeta^-$
4. If $\varphi = \exists x.\psi$ then $\varphi^+ = \exists x.\psi^+ \wedge \forall y.(y < x \wedge \psi(y))^-$ and $\varphi^- = \forall x.\psi^-$
5. If $\varphi = \exists X.\psi$ then $\varphi^+ = \exists X.\psi^+ \wedge \forall Y.(Y < X \wedge \psi(Y))^-$ and $\varphi^- = \forall X.\psi^-$

where $X < Y = \exists y.y \in Y \wedge \neg(y \in X) \wedge \forall z.[z < y \longrightarrow (z \in X \longleftrightarrow z \in Y)]^+$. Notice that for each $\varphi \in \mathrm{MSO}_{\mathrm{er}}(\Sigma)$ we have $[\![\varphi^+]\!] = 1_{L(\varphi)}$ and $[\![\varphi^-]\!] = 1_{L(\neg\varphi)}$. Thus the semantics of syntactically unambiguous formulas are recognizable by Theorem 1 and Lemma 1. Moreover, if φ is syntactically unambiguous, one

[1] Recall that every $\mathrm{MSO}_{\mathrm{er}}(\Sigma)$-formula can also be seen as an $\mathrm{MSO}_{\mathrm{er}}(\mathcal{K}, \Sigma, \mathcal{F})$-formula and may have a semantics different from 0 or 1; see e.g. Ex.2

can easily see that also $\forall X.\varphi$ is syntactically unambiguous and thus $[\![\forall X.\varphi]\!]$ is recognizable.

Next, we explain how to deal with $\forall x$. The approach used by Droste and Gastin [10] is to restrict the subformula φ in $\forall x.\varphi$ to so-called *almost unambiguous* formulas. Formulas of this kind can be transformed into equivalent formulas of the form $\bigvee_{1\leq i\leq n} k_i \wedge \psi_i^+$ for some $n \in \mathbb{N}$, $k_i \in K$ and syntactically unambiguous formulas ψ_i^+ for each $i \in \{1, ..., n\}$. One can easily see that the series corresponding to the semantics of such a formula has a finite image. Moreover, closure properties of recognizable series under sum, Hadamard- and scalar products can be used to prove that the semantics of such a formula is recognizable by a weighted automaton. Finally, this particular form of the formula is the base of an efficient construction of a weighted automaton recognizing $[\![\forall x.\varphi]\!]$. Here, we use a very similar approach. However, we have to redefine the notion of almost unambiguous formulas a bit in order to include subformulas of the form $C_\mu(x)$.

Let x be a first-order variable. We say that a formula φ is *almost unambiguous over* x if it is in the disjunctive and conjunctive closure of syntactically unambiguous formulas, constants $k \in K$ and formulas $C_\mu(x)$ (for $\mu \in \mathcal{F}$), such that $C_\mu(x)$ may appear at most once in every subformula of φ of the form $\varphi_1 \wedge \varphi_2$. Using similar methods as in [10], one can show that every almost unambiguous formala can be transformed into an equivalent formula of the form $\bigvee_{1\leq i\leq n} C_{\mu_i}(x) \wedge k_i \wedge \psi_i^+$ for some $n \in \mathbb{N}$, $k_i \in K$, $(\mu_i \in \mathcal{F})$ and $\psi_i \in \mathrm{MSO_{er}}(\Sigma)$ for every $i \in \{1, ..., n\}$. Clearly, the semantics of formulas of this form is not guaranteed to have a finite image. As a counter example consider for instance the case where $\mathcal{F}$ is the family of linear functions. However, using Lemmas 2 and 3 as well as Theorem 1, one can prove that the semantics of every formula of this form (and thus of every almost unambiguous formula over x) is recognizable. So now assume that φ is almost unambiguous over x. The main challenge of this paper was to prove that $[\![\forall x.\varphi]\!]$ is recognizable. We were able to adapt the proof proposed by Droste and Gastin to the timed setting by applying an additional normalization technique to solve problems having their origin in formulas of the form $C_\mu(x)$. The proof is rather technical and omitted here; for the details see the full length version of this paper [23].

Finally, we define the fragment of $\mathrm{MSO_{er}}(\mathcal{K}, \Sigma, \mathcal{F})$ used in Theorem 2. A formula $\varphi \in \mathrm{MSO_{er}}(\mathcal{K}, \Sigma, \mathcal{F})$ is called *syntactically restricted* if it satisfies the following conditions:

1. Whenever φ contains a conjunction $\varphi_1 \wedge \varphi_2$ as subformula, φ_1 contains the subformula $C_{\mu_1}(x_1)$ and φ_2 contains $C_{\mu_2}(x_2)$, then x_1, x_2 are free in both φ_1 and φ_2, and either φ_1 or φ_2 is of the form $\psi \wedge \neg(x_1 = x_2)$ for some $\psi \in \mathrm{MSO_{er}}(\mathcal{K}, \Sigma, \mathcal{F})$.
2. Whenever φ contains $\forall x.\psi$ as a subformula, then ψ is an almost unambiguous formula over x.
3. Whenever φ contains $\forall X.\psi$ as a subformula, then ψ is a syntactically unambiguous formula.

We let $\mathrm{sRMSO_{er}}(\mathcal{K}, \Sigma, \mathcal{F})$ denote the set of all syntactically restricted formulas of $\mathrm{MSO_{er}}(\mathcal{K}, \Sigma, \mathcal{F})$. Notice that each of these conditions can be checked for in easy syntax tests. Hence, the logic $\mathrm{sRMSO_{er}}(\mathcal{K}, \Sigma, \mathcal{F})$ is a *decidable* fragment, i.e., for each formula in $\mathrm{MSO_{er}}(\mathcal{K}, \Sigma, \mathcal{F})$ we can decide whether it is syntactically restricted or not.

We want to give some final remarks on the correctness of the proof methods described above. Although not explicitly mentioned in the individual steps, we make use of renaming operations in the proofs for closure under the constructs of our logic. For instance, we adopt the classical proof method for showing that the application of $\exists x.$ preserves the recognizability of the semantics of a formula φ with $\mathrm{Free}(\varphi) = \mathcal{V}$ by using a renaming $\pi : \Sigma_{\mathcal{V}} \to \Sigma_{\mathcal{V}\setminus\{x\}}$ which erases the x-row (see e.g. [26, 10]). However, it is well-known that recognizable timed languages are not closed under renaming [1]. We solve this problem using an approach proposed by D'Souza [14] and consider so-called *quasi*-WERA. Timed languages recognizable by quasi-WERA share the same closure properties as recognizable timed languages, but additionally are closed under so-called *valid* renamings [14]. So, in the inductive proof described above, we actually show that the semantics of every formula in our logic is recognizable by a quasi-WERA rather than a WERA. Since quasi-WERA-recognizable timed series form a strict subclass of recognizable timed series, we get the final implication. For the sake of simplicity, we only mention this here; the correct proof can be found in [23].

4 From Automata To Logic

For the implication from left to right in Theorem 2, we extend the proof proposed by Droste and Gastin to the timed setting, briefly explained in the following. Let $\mathcal{A} = (S, S_0, S, E, C)$ be a WERA. We choose an enumeration $(e_1, ..., e_m)$ of E with $m = |E|$ and assume $e_i = (s_i, a_i, \phi_i, s_i')$. We define a syntactically unambiguous formula $\psi(X_1, ..., X_m)$ without any second-order quantifiers describing the successful runs of $\mathcal{A}$ (where for each $i \in \{1, ..., m\}$, X_i stands for the edge e_i). This can be done similarly to the classical setting [26]. The guards of the edges in E can be defined by a formula of the form $\forall x. \bigwedge\limits_{1\le i\le m} \big(x \in X_i \xrightarrow{+} \bigwedge\limits_{a\in\Sigma} (\bigwedge\limits_{(x_a\sim c)\in\phi_i} \triangleleft_a(x) \sim c)\big)$ where $\varphi \xrightarrow{+} \psi$ is an abbreviation for $\varphi^- \vee (\varphi^+ \wedge \psi^+)$. Then, for every non-empty timed word $(\bar{a}, \bar{t})$ and valid $(\{X_1, ..., X_m\}, (\bar{a}, \bar{t}))$-assignment σ, we have $[\![\psi(X_1, ..., X_m)]\!]((\bar{a}, \sigma), \bar{t}) = 1$, if there is a successful run of $\mathcal{A}$ on $(\bar{a}, \bar{t})$, and $[\![\psi(X_1, ..., X_m)]\!]((\bar{a}, \sigma), \bar{t}) = 0$, otherwise. Notice that we need to use syntactically unambiguous formulas here in order to avoid getting weights different from 1 or 0. Now, we "add weights" to ψ to obtain a formula ξ whose semantics corresponds to the running weight of a successful run of $\mathcal{A}$ on $(\bar{a}, \bar{t})$ as follows:

$$\xi = \psi \wedge \bigwedge_{e_i\in E} \forall x. \left(\neg(x \in X_i) \vee [x \in X_i \wedge C_{\mu_{s_i}}(x) \wedge C_{\mathcal{E}}(e_i)]\right).$$

For the empty timed word ε, we define a formula $\varphi = (\|\mathcal{A}\|, \varepsilon) \wedge \forall x. \neg(x \le x)$. Finally, we let $\zeta = \exists X_1 ... \exists X_m.(\xi \vee \varphi)$, and we obtain $[\![\zeta]\!] = \|\mathcal{A}\|$. Hence, we have shown the second implication, which finishes the proof of Theorem 2.

5 Conclusion

We have presented a weighted timed MSO logic, which is - at least to our knowledge - the first MSO logic allowing for the description of both timed and quantitative properties. On the one hand, we provide the real-time-community with a new tool, because sometimes it may be easier to specify properties in terms of logic rather than by automata devices. On the other hand, the coincidence between recognizable and definable timed series, together with a previous work on WERA concerning a Kleene-Schützenberger Theorem [22], shows the robustness of the notion of WERA-recognizable timed series, as they can equivalently be characterized in terms of automata, logics and rational operations. The same applies to timed series recognizable by weighted timed automata, for which we were successful in adapting the proofs presented in this paper using the relative distance logic $\mathcal{L}\overleftarrow{\mathrm{d}}$ introduced by Wilke and his results concerning timed languages with bounded variability [28, 24]. Notice that our result generalizes corresponding results on ERA-recognizable languages as well as formal power series [14, 10]. Also, we have stated conditions for closure of recognizable timed series under the Hadamard product, which corresponds to the intersection operation in the unweighted setting.

Acknowledgement

I would like to thank Manfred Droste and Christian Mathissen for their helpful discussions and comments.

References

1. R. Alur, L. Fix, and T. A. Henzinger. Event-clock automata: A determinizable class of timed automata. *Theor. Comput. Sci.*, 211(1-2):253–273, 1999.
2. R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. *Theoretical Computer Science*, 318:297–322, 2004.
3. G. Behrmann, A. Fehnker, T. Hune, K. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Minimum-cost reachability for priced timed automata. In *HSCC*, volume 2034 of *LNCS*, pages 147–161. Springer, 2001.
4. P. Bouyer, T. Brihaye, V. Bruyère, and J.-F. Raskin. On the optimal reachability problem on weighted timed automata. *Formal Methods in System Design*, 31(2):135–175, October 2007.
5. P. Bouyer, T. Brihaye, and N. Markey. Improved undecidability results on weighted timed automata. *Inf. Process. Lett.*, 98(5):188–194, 2006.
6. J. R. Büchi. On a decision method in restricted second order arithmetics. In E. Nagel et al., editor, *Proc. Intern. Congress on Logic, Methodology and Philosophy of Sciences*, pages 1–11, Stanford, 1960. Stanford University Press.
7. C. Dima. Kleene theorems for event-clock automata. In *FCT*, volume 1684 of *LNCS*, pages 215–225. Springer, 1999.
8. M. Droste and P. Gastin. Weighted automata and weighted logics. In *ICALP*, volume 3580 of *LNCS*, pages 513–525. Springer, 2005.

9. M. Droste and P. Gastin. Weighted automata and weighted logics. *Theor. Comput. Sci.*, 380(1-2):69–86, 2007.
10. M. Droste and P. Gastin. Weighted automata and weighted logics. In M. Droste, W. Kuich, and H. Vogler, editors, *Handbook of Weighted Automata*, pages 173–206. Springer, 2009. To appear.
11. M. Droste and K. Quaas. A Kleene-Schützenberger Theorem for Weighted Timed Automata. In *FoSSaCS*, volume 4962 of *LNCS*, pages 142–156. Springer, 2008.
12. M. Droste and G. Rahonis. Weighted automata and weighted logics on infinite words. In *Developments in Language Theory*, volume 4036 of *LNCS*, pages 49–58. Springer, 2006.
13. M. Droste and H. Vogler. Weighted tree automata and weighted logics. *Theor. Comput. Sci.*, 366(3):228–247, 2006.
14. D. D'Souza. A logical characterisation of event recording automata. In *FTRTFT*, volume 1926 of *LNCS*, pages 240–251. Springer, 2000.
15. C. C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Am. Math. Soc.*, 98:21–51, 1961.
16. O. Grinchtein, B. Jonsson, and P. Pettersson. Inference of event-recording automata using timed decision trees. In *CONCUR*, volume 4137 of *LNCS*, pages 435–449. Springer, 2006.
17. T. A. Henzinger, J.-F. Raskin, and P.-Y. Schobbens. The regular real-time languages. In *ICALP*, volume 1443 of *LNCS*, pages 580–591. Springer, 1998.
18. C. Mathissen. Definable transductions and weighted logics for texts. In *Developments in Language Theory*, volume 4588 of *LNCS*, pages 324–336. Springer, 2007.
19. C. Mathissen. Weighted logics for nested words and algebraic formal power series. In *ICALP (2)*, volume 5126 of *LNCS*, pages 221–232. Springer, 2008.
20. I. Mäurer. Weighted picture automata and weighted logics. In *STACS*, volume 3884 of *LNCS*, pages 313–324. Springer, 2006.
21. I. Meinecke. Weighted logics for traces. In *CSR*, volume 3967 of *LNCS*, pages 235–246. Springer, 2006.
22. K. Quaas. A Kleene-Schützenberger-Theorem for Weighted Event-Clock-Automata. http://www.informatik.uni-leipzig.de/~quaas/weca_ks.pdf, 2008.
23. K. Quaas. A logical characterization for weighted event-recording automata. http://www.informatik.uni-leipzig.de/~quaas/wera_logic.pdf, 2009.
24. K. Quaas. A logical characterization for weighted timed automata. http://www.informatik.uni-leipzig.de/~quaas/wta_logic.pdf, 2009.
25. J.-F. Raskin and P.-Y. Schobbens. State clock logic: A decidable real-time logic. In *HART*, volume 1201 of *LNCS*, pages 33–47. Springer, 1997.
26. W. Thomas. Languages, automata and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, pages 389–485. Springer, 1997.
27. Wolfgang Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 133–192. Elsevier and MIT Press, 1990.
28. T. Wilke. Specifying Timed State Sequences in Powerful Decidable Logics and Timed Automata. In *FTRTFT*, volume 863 of *LNCS*, pages 694–715, Lübeck, Germany, 1994. Springer-Verlag.