# Logics for Weighted Timed Pushdown Automata

Manfred Droste and Vitaly Perevoshchikov*

Universität Leipzig, Institut für Informatik,
04109 Leipzig, Germany
{droste,perev}@informatik.uni-leipzig.de

**Abstract.** Weighted dense-timed pushdown automata with a timed stack were introduced by Abdulla, Atig and Stenman to model the behavior of real-time recursive systems. Motivated by the decidability of the optimal reachability problem for weighted timed pushdown automata and weighted logic of Droste and Gastin, we introduce a weighted MSO logic on timed words which is expressively equivalent to weighted timed pushdown automata. To show the expressive equivalence result, we prove a decomposition theorem which establishes a connection between weighted timed pushdown languages and visibly pushdown languages of Alur and Mudhusudan; then we apply their result about the logical characterization of visibly pushdown languages.

**Keywords:** timed automata, weighted automata, monadic second-order logic, weighted logic, formal power series, pushdown automata, timed stack

## 1 Introduction

Timed automata introduced by Alur and Dill [3] are a prominent model for the specification and analysis of real-time systems. Timed pushdown automata (TPDA) with a stack were studied in [7, 10, 16] in the context of the verification of real-time recursive systems. Recently, Abdulla, Atig and Stenman [1] proposed TPDA with a timed stack which keeps track of the age of its elements. In [2], they introduced weighted timed pushdown automata (WTPDA) as a model for quantitative properties of timed recursive systems and showed that the optimal reachability problem for WTPDA is decidable.

Since the seminal Büchi-Elgot-Trakhtenbrot theorem [9, 15, 22] about the expressive equivalence of finite automata and monadic second-order logic, a significant field of research investigates logical characterizations of language classes appearing from practically relevant automata models. The goal of this paper is to provide a logical characterization for weighted timed pushdown automata, i.e., to design a weighted logic on timed words which is expressively equivalent to WTPDA.

On the one hand, logic provides an intuitive way to describe the properties of systems. On the other hand, logical formulas can be translated into automata which may have interesting algorithmic properties. Furthermore, logic provides good insights into the understanding of the automata behaviors.

**Related work.** A logical characterization of unweighted TPDA was given in [14] where a *timed matching logic* is introduced. As in the logic of Lautemann, Schwentick

* Supported by DFG Research Training Group 1763 (QuantLA)

and Thérien [18], we handle the stack functionality by means of a binary *matching* predicate. As in the logic of Wilke [23], we use *relative distance formulas* to handle the functionality of clocks. Moreover, to handle the ages of stack elements, we lift the binary matchings to the timed setting, i.e., we can compare the time distance between matched positions with a constant.

On the other side, Droste and Gastin [11] introduced and investigated weighted MSO logic over semirings; this logic permits to describe quantitative properties of systems. In [11] it was shown that syntactically restricted weighted MSO logic is expressively equivalent to weighted automata over semirings (cf. [12] for surveys). In [21, 13] this result was extended to the setting of weighted timed automata. In [19] a logical characterization of algebraic formal power series was given.

**Contribution of this paper.** In this paper, we extend our result for unweighted TPDA to the setting of semiring-weighted TPDA. We introduce a *weighted timed matching logic* (wTML) and study its relation to WTPDA. As in [11], unrestricted weighted timed matching logic is more expressive than WTPDA. We study which formulas lead to unrecognizable weighted timed languages and introduce a reasonable syntactically restricted fragment of wTML which is expressively equivalent to WTPDA.

For the proof of our expressive equivalence result, we use the idea of [13] for weighted timed automata to separate weights from timed automata. In contrast to timed automata, TPDA as extensions of pushdown automata are not closed under intersection. Moreover, the storing weights in a timed stack are closely connected to the inner components of TPDA. Therefore, we have to modify the approach of [13]. We introduce a technique which establishes a connection between *weighted timed* pushdown automata and *visibly pushdown languages* of Alur and Madhusudan [4], a subclass of the classical context-free languages. In other words, our method permits to simultaneously separate weights and time from the discrete part of the model.

We show our expressive equivalence result as follows.

- We prove a Nivat-like decomposition theorem for WTPDA (cf. [20, 5]) which may be of independent interest; this theorem establishes a connection between weighted timed pushdown languages and unweighted untimed visibly pushdown languages of [4] by means of operations like renamings and intersections with simple weighted timed pushdown languages. This result extends our decomposition result for unweighted TPDA presented in [14].
- In a similar way, we separate the weighted timed part of wTML from the boolean part described by MSO logic with matchings over a visibly pushdown alphabet [4].
- Then we deduce our result from the logical characterization result of [4] for visibly pushdown languages.

Since our proof is constructive and the optimal reachability for WTPDA is decidable [2], we obtain the corresponding decidability result for restricted wTML.

**Outline.** For the clarity of presentation and convenience of the reader, we will concentrate on the simplified model of WTPDA without global clocks. In Sect. 2 we define WTPDA over timed semirings. In Sect. 3 we introduce our weighted timed matching logic and state our main result, namely Theorem 3.8. The proof of Theorem 3.8 will be given in Sect. 6. As a preparation for this proof, in Sect. 4 we recall some basic definitions about visibly pushdown languages and in Sect. 5 we prove our Nivat-like decomposition theorem. In Sect. 7 we explain how global clocks can be added to our main result. Our proof technique can easily recover this case.

**Acknowledgement.** Yuri Gurevich' survey article [17] was a source of inspiration for the first named author when he got interested in monadic second-order logic. Due to stimulating friendly contact since 1981, the authors would like to dedicate their paper to Yuri Gurevich.

## 2 Weighted Timed Pushdown Automata

*Timed pushdown automata* (TPDA) with a timed stack are introduced and investigated in [1]. These machines are nondeterministic automata equipped with *global clocks* (like timed automata [3]) and a stack (like pushdown automata). In contrast to untimed pushdown automata, in timed pushdown automata we push together with a letter a *local clock* which will measure the age of this letter in the stack. Then, we can pop this letter only if its age satisfies a given constraint. *Weighted timed pushdown automata* (WTPDA) of [2] extend TPDA by adding time-independent costs to the transitions of TPDA (like in the classical weighted automata [12]) and costs for storing a letter in the stack which depend on the age of this letter in the stack.

In this section, we introduce an algebraic model for WTPDA which is based on the classical model of *semiring-weighted* automata [12]. Moreover, we follow the idea of Quaas [21] to model time-dependent costs by means of functions of a real argument. Recall from the introduction that the goal of this paper is to give a logical characterization of WTPDA. Weighted timed automata with global clocks and without stack were studied in [21, 13] with respect to their logical characterization. The new feature of WTPDA is the quantitative timed stack, i.e., the timed stack equipped with time-dependent costs for storing stack letters. In order to concentrate on the significant details and for the clarity of presentation, we will omit global clocks in our considerations. However, in Sect. 7 we discuss how they can be added to our definitions and proofs.

An *alphabet* is a non-empty finite set. Let $\Sigma$ be a non-empty set (possibly infinite). A *finite word* over $\Sigma$ is a finite sequence $a_1...a_n$ where $n \geq 0$ and $a_1, ..., a_n \in \Sigma$. If $n = 0$, then $w$ is *empty* and we denote it by $\varepsilon$. Otherwise, we call $w$ *non-empty*. Let $\Sigma^*$ denote the set of all words and $\Sigma^+$ the set of all non-empty words over $\Sigma$. Let $\mathbb{R}_{\geq 0}$ be the set of all non-negative real numbers. A *timed word* over $\Sigma$ is a sequence $(a_1, t_1)(a_2, t_2)...(a_n, t_n) \in (\Sigma \times \mathbb{R}_{\geq 0})^*$ such that $t_1 \leq t_2 \leq ... \leq t_n$. Let $\mathbb{T}\Sigma^*$ denote the set of all timed words over $\Sigma$ and $\mathbb{T}\Sigma^+$ the set of all non-empty timed words over $\Sigma$. Any set $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ of non-empty timed words is called a *timed language*. Let $\mathcal{I}$ denote the class of all intervals of the form $[a, b]$, $(a, b]$, $[a, b)$, $(a, b)$, $[a, \infty)$ or $(a, \infty)$ where $a, b \in \mathbb{N}$. If $\Gamma$ is an alphabet, $u = (g_1, t_1)...(g_n, t_n) \in \mathbb{T}\Gamma^*$ and $t \in \mathbb{R}_{\geq 0}$, then let $u + t = (g_1, t_1 + t)...(g_n, t_n + t) \in \mathbb{T}\Gamma^*$.

A *timed semiring* is a tuple $\mathbb{S} = (S, \mathcal{F}, +, \cdot, \mathbb{0}, \mathbb{1})$ such that $(S, +, \cdot, \mathbb{0}, \mathbb{1})$ is a semiring and $\mathcal{F} \subseteq S^{\mathbb{R}_{\geq 0}}$ is a collection of functions containing the function $\overline{\mathbb{1}} \in S^{\mathbb{R}_{\geq 0}}$ defined for all $t \in \mathbb{R}_{\geq 0}$ by $\overline{\mathbb{1}}(t) = \mathbb{1}$.

*Example 2.1.* (a) The tropical semiring $(\mathbb{R}_{\geq 0} \cup \{\infty\}, \min, +, \infty, 0)$ together with the collection of linear functions $\mathbb{L} = \{t \mapsto c \cdot t \mid c \in \mathbb{R}_{\geq 0}\}$ forms a timed semiring which we denote by $\mathbb{L}$TROP. The model of [2] can be considered as WTPDA over $\mathbb{L}$TROP.

(b) The boolean semiring $(\{0,1\}, \vee, \wedge, 0, 1)$ together with $\mathcal{F} = \{\overline{\mathbb{1}}\}$ form a timed semiring which we denote by $\mathbb{1}$BOOL. Unweighted TPDA of [1] can be considered as WTPDA over $\mathbb{1}$BOOL.

(c) It could be also interesting to consider the case where the storing costs in the timed stack grow exponentially in time (cf., e.g., [8]). We augment the tropical semiring $(\mathbb{R}_{\geq 0} \cup \{\infty\}, \min, +, \infty, 0)$ with the collection of exponential functions $\mathcal{F} = \{t \mapsto e^{c \cdot t} \mid c \in \mathbb{R}_{\geq 0}\}$ and obtain a timed semiring which we denote by EXPTROP.

Let $\mathcal{S}(\Gamma) = \{\mathrm{push}(\gamma) \mid \gamma \in \Gamma\} \cup \{\#\} \cup \{\mathrm{pop}(\gamma, I) \mid \gamma \in \Gamma, I \in \mathcal{I}\}$ be the set of *stack commands* over $\Gamma$.

**Definition 2.2.** *Let $\Sigma$ be an alphabet and $\mathbb{S} = (S, \mathcal{F}, +, \cdot, \mathbb{0}, \mathbb{1})$ a timed semiring. A* weighted timed pushdown automaton *(WTPDA) over $\Sigma$ and $\mathbb{S}$ is a tuple $\mathcal{A} = (L, \Gamma, L_0, E, L_f, \mathrm{wt})$ where $L$ is a finite set of locations, $\Gamma$ is a finite stack alphabet, $L_0, L_f \subseteq L$ are sets of initial resp. final locations, $E \subseteq L \times \Sigma \times \mathcal{S}(\Gamma) \times L$ is a finite set of edges, and $\mathrm{wt} : E \cup \Gamma \to S \cup \mathcal{F}$ is a weight function with $\mathrm{wt}(E) \subseteq S$ and $\mathrm{wt}(\Gamma) \subseteq \mathcal{F}$.*

A stack command $\mathrm{push}(\gamma)$ means that we push the letter $\gamma$ into the timed stack with the initial age 0. The stack command $\#$ means that we do not perform any operations with the timed stack. A stack command $\mathrm{pop}(\gamma, I)$ means that we pop from the stack the letter $\gamma$ with the age lying in the interval $I$. The weights of the stack letters in WTPDA have the following meaning. Whenever we pop a letter $\gamma$ with the age $\tau$ from the stack, the storing cost $\mathrm{wt}(\gamma)(\tau)$ arises.

We will denote an edge $e = (\ell, a, \mathrm{st}, \ell') \in E$ by $\ell \xrightarrow{a,\mathrm{st}} \ell'$. We say that $a$ is the *label* of $e$ and denote in by $\mathrm{label}(e)$. We also let $\mathrm{stack}(e) = \mathrm{st}$, the stack command of $e$. Let $E^{\mathrm{push}} \subseteq E$ denote the set of all *push* edges $e$ with $\mathrm{stack}(e) = \mathrm{push}(\gamma)$ for some $\gamma \in \Gamma$. Similarly, let $E^{\#} = \{e \in E \mid \mathrm{stack}(e) = \#\}$ be the set of *local* edges and $E^{\mathrm{pop}} = \{e \in E \mid \mathrm{stack}(e) = \mathrm{pop}(\gamma, I)$ for some $\gamma \in \Gamma$ and $I \in \mathcal{I}\}$ the set of *pop* edges. Then, we have $E = E^{\mathrm{push}} \cup E^{\#} \cup E^{\mathrm{pop}}$.

A *configuration* $c$ of $\mathcal{A}$ is described by the present location and the stack which is a timed word over $\Gamma$. That is, $c$ is a pair $\langle \ell, u \rangle$ where $\ell \in L$ and $u \in \mathbb{T}\Gamma^*$. We say that $c$ is *initial* if $\ell \in L_0$ and $u = \varepsilon$. We say that $c$ is *final* if $\ell \in L_f$ and $u = \varepsilon$. Let $\mathcal{C}_{\mathcal{A}}$ denote the set of all configurations of $\mathcal{A}$, $\mathcal{C}_{\mathcal{A}}^0 \subseteq \mathcal{C}_{\mathcal{A}}$ the set of all initial configurations, and $\mathcal{C}_{\mathcal{A}}^f \subseteq \mathcal{C}_{\mathcal{A}}$ the set of all final configurations.

Consider configurations $c = \langle \ell, u \rangle$ and $c' = \langle \ell', u' \rangle$ with $u = (\gamma_1, t_1)...(\gamma_k, t_k)$ and let $e = \left(q \xrightarrow{a,\mathrm{st}} q'\right) \in E$ be an edge. We say that $c \vdash_e c'$ is a *switch transition* if $\ell = q$, $\ell' = q'$, and

- if $\mathrm{st} = \mathrm{push}(\gamma)$ for some $\gamma \in \Gamma$, then $u' = (\gamma, 0)u$;
- if $\mathrm{st} = \#$, then $u' = u$;
- if $\mathrm{st} = \mathrm{pop}(\gamma, I)$ with $\gamma \in \Gamma$ and $I \in \mathcal{I}$, then $k \geq 1$, $\gamma = \gamma_1$, $t_1 \in I$ and $u' = (\gamma_2, t_2)...(\gamma_k, t_k)$.

The weight of this switch transition is defined as follows. If $\mathrm{st} = \mathrm{push}(\gamma)$ or $\mathrm{st} = \#$, then we let $\mathrm{wt}(c \vdash_e c') = \mathrm{wt}(e)$. If $\mathrm{st} = \mathrm{pop}(\gamma, I)$, then we let $\mathrm{wt}(c \vdash_e c') = \mathrm{wt}(e) \cdot \mathrm{wt}(\gamma)(t_1)$.

For $t \in \mathbb{R}_{\geq 0}$, we say that $c \vdash_t c'$ is a *delay transition* if $\ell = \ell'$ and $u' = u + t$. For $t \in \mathbb{R}_{\geq 0}$ and $e \in E$, we write $c \vdash_{t,e} c'$ if there exists $c'' \in \mathcal{C}_{\mathcal{A}}$ with $c \vdash_t c''$ and $c'' \vdash_e c'$. Note that, for every $c \in \mathcal{C}_{\mathcal{A}}$ and $t \in \mathbb{R}_{\geq 0}$, there exists at most one $c'' \in \mathcal{C}_{\mathcal{A}}$ with $c \vdash_t c''$. Then, we let $\mathrm{wt}(c \vdash_{t,e} c') = \mathrm{wt}(c'' \vdash_e c')$. A *run* $\rho$ of $\mathcal{A}$ is an alternating sequence of delay and switch transitions which starts in an initial configuration and ends in a final configuration, formally, $\rho = c_0 \vdash_{t_1,e_1} c_1 \vdash_{t_2,e_2} ... \vdash_{t_n,e_n} c_n$ where $n \geq 1$, $c_0 \in \mathcal{C}^0_{\mathcal{A}}$ and $c_n \in \mathcal{C}^f_{\mathcal{A}}$. The *label* of $\rho$ is the timed word $\mathrm{label}(\rho) = (\mathrm{label}(e_1), t_1)(\mathrm{label}(e_2), t_1 + t_2)...(\mathrm{label}(e_n), \sum_{i=1}^n t_i) \in \mathbb{T}\Sigma^+$. The *weight* of $\rho$ is defined as $\mathrm{wt}(\rho) = \prod_{i=1}^n \mathrm{wt}(c_{i-1} \vdash_{t_i,e_i} c_i)$.

For each timed word $w \in \mathbb{T}\Sigma^+$, let $\mathrm{Run}_{\mathcal{A}}(w)$ denote the set of all runs $\rho$ of $\mathcal{A}$ such that $\mathrm{label}(\rho) = w$. The *behavior* of $\mathcal{A}$ is the mapping $[\![\mathcal{A}]\!] : \mathbb{T}\Sigma^+ \to S$ defined for every timed word $w \in \mathbb{T}\Sigma^+$ as $[\![\mathcal{A}]\!](w) = \sum \big( \mathrm{wt}(\rho) \mid \rho \in \mathrm{Run}_{\mathcal{A}}(w) \big)$. A mapping $\mathbb{L} : \mathbb{T}\Sigma^+ \to S$ is called a *weighted timed language* (WTL). We say $\mathbb{L}$ is *pushdown recognizable* over $\mathbb{S}$ if there exists a WTPDA $\mathcal{A}$ over $\Sigma$ and $\mathbb{S}$ such that $[\![\mathcal{A}]\!] = \mathbb{L}$.

*Remark 2.3.* Note that in our model of WTPDA without global clocks the first time stamp of a timed word is irrelevant for the behavior of a WTPDA. However, this is not the case if we add global clocks to this model.

*Example 2.4.* Let $\Sigma = \{[, ]\}$ be the set of brackets. Let $b = [$, $\overline{b} = ]$ and $I \in \mathcal{I}$ be an interval. We consider the timed language $\mathcal{D} \subseteq \mathbb{T}\Sigma^+$ of timed words $w = (a_1, t_1)...(a_n, t_n)$ where $a_1...a_n$ is a sequence of correctly nested brackets and, for all $i < j$ such that $a_i = b$ and $a_j = \overline{b}$ are two matching brackets, the time distance $t_j - t_i$ is in the interval $I$. Consider the weighted timed language $\mathbb{W}_{\mathcal{D}} : \mathbb{T}\Sigma^+ \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ such that $\mathbb{W}_{\mathcal{D}}(w) = \infty$ for all $w \notin \mathcal{D}$ and, for every $w \in \mathcal{D}$, $\mathbb{W}_{\mathcal{D}}(w)$ is the minimal time distance between matching brackets in $w$. Let $\mathbb{L}\textsc{Trop}$ be the timed semiring of Example 2.1 (a). We show that $\mathbb{W}_{\mathcal{D}}$ is pushdown recognizable over $\mathbb{L}\textsc{Trop}$. Let $\mathrm{id} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ with $\mathrm{id}(t) = t$ for all $t$. Consider the WTPDA $\mathcal{A}$ over $\Sigma$ and $\mathbb{L}\textsc{Trop}$ depicted in Fig. 1 with the stack alphabet $\Gamma = \{\gamma, \delta\}$, $\mathrm{wt}(\gamma) = \overline{\mathbb{1}}$, $\mathrm{wt}(\delta) = \mathrm{id}$, and $\mathrm{wt}(e) = 0$ for all edges $e$ of $\mathcal{A}$. Then $[\![\mathcal{A}]\!] = \mathbb{W}_{\mathcal{D}}$. For instance, let $I = [0, 3]$ and $w = (b, 0)(b, 1)(\overline{b}, 2)(\overline{b}, 3) \in \mathcal{D}$. Then there are two runs of $\mathcal{A}$ on $w$: the run $\rho$

$$\langle 1,\varepsilon\rangle \vdash_{0,\mathrm{push}(\gamma)} \langle 1,(\gamma,0)\rangle \vdash_{1,\mathrm{push}(\delta)} \langle 2,(\delta,0)(\gamma,1)\rangle \vdash_{1,\mathrm{pop}(\delta,I)} \langle 3,(\gamma,2)\rangle \vdash_{1,\mathrm{pop}(\gamma,I)} \langle 3,\varepsilon\rangle$$

with $\mathrm{wt}(\rho) = \mathrm{id}(1) = 1$ and the run $\rho'$

$$\langle 1,\varepsilon\rangle \vdash_{0,\mathrm{push}(\delta)} \langle 2,(\delta,0)\rangle \vdash_{1,\mathrm{push}(\gamma)} \langle 2,(\gamma,0)(\delta,1)\rangle \vdash_{1,\mathrm{pop}(\gamma,I)} \langle 2,(\delta,2)\rangle \vdash_{1,\mathrm{pop}(\delta,I)} \langle 3,\varepsilon\rangle$$

with $\mathrm{wt}(\rho') = \mathrm{id}(3) = 3$. Here, for simplicity, we write in $\vdash_{t,\mathrm{st}}$ a stack command st instead of an edge $e$. Then $[\![\mathcal{A}]\!](w) = \min\{\mathrm{wt}(\rho), \mathrm{wt}(\rho')\} = 1 = \mathbb{W}_{\mathcal{D}}(w)$.

## 3 Weighted Timed Matching Logic

The goal of this section is to develop a logical formalism which is expressively equivalent to WTPDA defined before. Our logic will use binary *matchings* introduced by

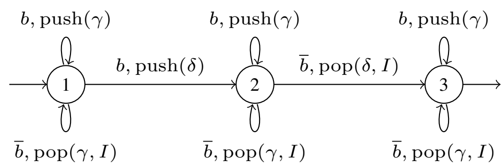

**Fig. 1.** The WTPDA $\mathcal{A}$ of Example 2.4

Lautemann, Schwentick and Thérien [18] for context-free languages as well as the approach of Droste and Gastin [11] to weighted logic over semirings. Moreover, we augment our logic with the possibility to measure the time distance between matched positions:

- we will be able to check whether this time distance belongs to a given interval (in order to model the timed stack);
- we will also be able to apply a function from a timed semiring to the time distance between matched positions (in order to model the storing costs in the timed stack).

As in [6], in order to describe easily boolean properties, we introduce two levels of formulas: boolean and weighted. We operate with the boolean formulas as in the usual logic. On the weighted level, we add weights and extend the logical operations by computations in a semiring.

Let $V_1$ and $V_2$ be countable pairwise disjoint sets of first-order and second-order variables. We also fix a *matching variable* $\mu \notin V_1 \cup V_2$. Let $V = V_1 \cup V_2 \cup \{\mu\}$.

Let $\Sigma$ be an alphabet and $\mathbb{S} = (S, \mathcal{F}, +, \cdot, \mathbb{0}, \mathbb{1})$ a timed semiring. The set $\mathbf{wTML}(\Sigma, \mathbb{S})$ of *weighted timed matching formulas* is the set of all formulas of the form $\bigoplus \mu.\varphi$ where $\varphi$ is produced by the grammar

$$\begin{aligned} \beta &::= P_a(x) \mid x \leq y \mid x \in X \mid \mu(x,y) \in I \mid \beta \vee \beta \mid \neg\beta \mid \exists x.\beta \mid \exists X.\beta \\ \varphi &::= \beta \mid s \mid f(\mu - x) \mid \varphi \oplus \varphi \mid \varphi \otimes \varphi \mid \bigoplus x.\varphi \mid \bigoplus X.\varphi \mid \bigotimes x.\varphi \end{aligned}$$

with $a \in \Sigma$, $s \in S$, $f \in \mathcal{F}$, $I \in \mathcal{I}$, $x, y \in V_1$ and $X \in V_2$. The formulas $\beta$ are called *boolean-valued* (or, simply, *boolean*) over $\Sigma$. Let $\mathbf{BOOL}(\Sigma)$ denote the set of all boolean formulas over $\Sigma$. The formulas of the form $\mu(x,y) \in I$ are called *distance matchings*. For a formula $\mu(x,y) \in [0,\infty)$ we will simply write $\mu(x,y)$. Using boolean formulas, we define the formulas $x < y$, $x = y$, $\beta_1 \wedge \beta_2$, $\forall x.\beta$, $\forall X.\beta$, $\beta_1 \rightarrow \beta_2$ and $\beta_1 \leftrightarrow \beta_2$ as usual.

The $\mathbf{wTML}(\Sigma, \mathbb{S})$-formulas are interpreted over timed words over $\Sigma$ and assignments of variables. Let $w \in \mathbb{T}\Sigma^+$ be a timed word and $\mathrm{dom}(w) = \{1, ..., |w|\}$, the *domain* of $w$. We say that a binary relation $M \subseteq \mathrm{dom}(w) \times \mathrm{dom}(w)$ is a *matching* on $w$ (cf. [18]) if:

- $M$ is compatible with $<$, i.e., whenever $(i,j) \in M$, we have $i < j$;
- each element $i \in \mathrm{dom}(w)$ belongs to at most one pair in $M$;
- $M$ is *noncrossing*, i.e., whenever $(i,j), (u,v) \in M$ with $i < u < j$, we have $i < v < j$.

| | | |
|---|---|---|
| $(w,\sigma) \models P_a(x)$ | $\Leftrightarrow$ | $a_{\sigma(x)} = a$ |
| $(w,\sigma) \models x \leq y$ | $\Leftrightarrow$ | $\sigma(x) \leq \sigma(y)$ |
| $(w,\sigma) \models x \in X$ | $\Leftrightarrow$ | $\sigma(x) \in \sigma(X)$ |
| $(w,\sigma) \models \mu(x,y) \in I$ | $\Leftrightarrow$ | $(\sigma(x),\sigma(y)) \in \sigma(\mu)$ and $t_{\sigma(y)} - t_{\sigma(x)} \in I$ |
| $(w,\sigma) \models \varphi_1 \vee \varphi_2$ | $\Leftrightarrow$ | $(w,\sigma) \models \varphi_1$ or $(w,\sigma) \models \varphi_2$ |
| $(w,\sigma) \models \neg\varphi$ | $\Leftrightarrow$ | $(w,\sigma) \models \varphi$ does not hold |
| $(w,\sigma) \models \exists x.\varphi$ | $\Leftrightarrow$ | $\exists j \in \mathrm{dom}(w) : (w,\sigma[x/j]) \models \varphi$ |
| $(w,\sigma) \models \exists X.\varphi$ | $\Leftrightarrow$ | $\exists J \subseteq \mathrm{dom}(w) : (w,\sigma[X/J]) \models \varphi$ |

**Table 1.** The satisfaction relation for boolean formulas

Let $\mathrm{Match}(w)$ denote the set of all matchings on $w$.

A $w$-*assignment* is a mapping $\sigma : V \to \mathrm{dom}(w) \cup 2^{\mathrm{dom}(w)} \cup \mathrm{Match}(w)$ such that $\sigma(V_1) \subseteq \mathrm{dom}(w)$, $\sigma(V_2) \subseteq 2^{\mathrm{dom}(w)}$ and $\sigma(\mu) \in \mathrm{Match}(w)$. Let $\mathbb{T}\Sigma_V^+$ denote the set of all pairs $(w,\sigma)$ where $w \in \mathbb{T}\Sigma^+$ and $\sigma$ is a $w$-assignment.

Let $\sigma$ be a $w$-assignment. For $x \in V_1$ and $j \in \mathrm{dom}(w)$, the *update* $\sigma[x/j]$ is the $w$-assignment defined by $\sigma[x/j](x) = j$ and $\sigma[x/j](y) = \sigma(y)$ for all $y \in V \setminus \{x\}$. Similarly, for $X \in V_2$ and $J \subseteq \mathrm{dom}(w)$, we define the update $\sigma[X/J]$ and, for $M \in \mathrm{Match}(w)$, the update $\sigma[\mu/M]$.

For a formula $\beta \in \mathbf{BOOL}(\Sigma)$, a timed word $w = (a_1,t_1)...(a_n,t_n) \in \mathbb{T}\Sigma^+$ and a $w$-assignment $\sigma$, we define the *satisfaction relation* $(w,\sigma) \models \beta$ inductively on the structure of $\beta$ as shown in Table 1. Now let $\psi \in \mathbf{wTML}(\Sigma,\mathbb{S})$. The *semantics* of $\psi$ is the mapping $[\![\psi]\!] : \mathbb{T}\Sigma^+ \to S$ defined inductively on the structure of $\psi$ as shown in Table 2.

Given a formula $\psi \in \mathbf{wTML}(\Sigma,\mathbb{S})$, the set $\mathrm{Free}(\psi)$ of *free variables* of $\psi$ is defined as usual. We say that $\psi$ is a *sentence* if $\mathrm{Free}(\psi) = \emptyset$. Clearly, the semantics of a sentence $\psi$ does not depend on a variable assignment. Then, we may consider the semantics of $\psi$ as the weighted timed language $[\![\psi]\!] : \mathbb{T}\Sigma^+ \to S$.

*Remark 3.1.* (a) Let $\mathbb{1}$Bool be the timed semiring of Example 2.1 (b). If in the definition of $\mathbf{wTML}(\Sigma, \mathbb{1}\text{Bool})$ we replace the quantitative formulas $\mu(x,y) \in I$ by the qualitative formulas $\mu(x,y)$, then the timed part of the timed words will be irrelevant and we obtain the matching logic of [18] for context-free languages.

(b) If we exclude the formulas $\mu(x,y) \in I$ and $f(\mu - x)$ from the definition of $\varphi$ and $\beta$ in $\mathbf{wTML}(\Sigma,\mathbb{S})$, then the formulas of the form $\bigoplus\mu.(\neg\exists x.\exists y.\mu(x,y) \otimes \varphi)$ correspond to the weighted MSO logic of Droste and Gastin [11].

*Example 3.2.* Let $\mathbb{W}_{\mathcal{D}} : \mathbb{T}\Sigma^+ \to \mathbb{R}_{\geq 0}$ be the WTL of Example 2.4. Recall that $b = [$ and $\overline{b} = ]$. Consider the $\mathbf{BOOL}(\Sigma)$-formula

$$\beta(\mu) = \forall x.([P_b(x) \to \exists y.\mu(x,y)] \wedge [P_{\overline{b}}(x) \to \exists y.\mu(y,x)])$$

which demands that for every opening bracket there is a matched closing bracket and vice versa. Then the WTL $\mathbb{W}_{\mathcal{D}}$ can be described by the following $\mathbf{wTML}(\Sigma, \mathbb{L}\text{Trop})$-sentence:

$$\psi = \bigoplus\mu.([\beta(\mu) \wedge \forall x.\forall y.(\mu(x,y) \to \mu(x,y) \in I)] \otimes \bigoplus x.(\exists y.\mu(x,y) \otimes \mathrm{id}(\mu - x)))$$

$$
\begin{array}{lcl}
[\![\beta]\!](w,\sigma) & = & \begin{cases} \mathbb{1}, & \text{if } (w,\sigma) \models \beta, \\ \mathbb{0}, & \text{otherwise} \end{cases} \\
[\![s]\!](w,\sigma) & = & s \\
[\![f(\mu - x)]\!](w,\sigma) & = & \begin{cases} f(t_j - t_{\sigma(x)}), & \text{if } (\sigma(x), j) \in \sigma(\mu) \text{ for some } j, \\ \mathbb{1}, & \text{otherwise} \end{cases} \\
[\![\varphi_1 \oplus \varphi_2]\!](w,\sigma) & = & [\![\varphi_1]\!](w,\sigma) + [\![\varphi_2]\!](w,\sigma) \\
[\![\varphi_1 \otimes \varphi_2]\!](w,\sigma) & = & [\![\varphi_1]\!](w,\sigma) \cdot [\![\varphi_2]\!](w,\sigma) \\
[\![\bigoplus x.\varphi]\!](w,\sigma) & = & \sum \big( [\![\varphi]\!](w,\sigma[x/i]) \mid i \in \mathrm{dom}(w) \big) \\
[\![\bigoplus X.\varphi]\!](w,\sigma) & = & \sum \big( [\![\varphi]\!](w,\sigma[X/I]) \mid I \subseteq \mathrm{dom}(w) \big) \\
[\![\bigotimes x.\varphi]\!](w,\sigma) & = & \prod \big( [\![\varphi]\!](w,\sigma[x/i]) \mid i \in \mathrm{dom}(w) \big) \\
[\![\bigoplus \mu.\varphi]\!](w,\sigma) & = & \sum \big( [\![\varphi]\!](w,\sigma[\mu/M]) \mid M \in \mathrm{Match}(w) \big)
\end{array}
$$

**Table 2.** The semantics of weighted timed matching formulas

where $\mathrm{id}$ is defined as in Example 2.4. Note that the boolean subformula of $\psi$ in the square brackets checks whether a timed word belongs to $\mathcal{D}$ (cf. Example 2.4). Then, the formula $\bigoplus x.(\exists y.\mu(x,y) \otimes \mathrm{id}(\mu - x))$ computes the minimal time distance between matching brackets.

Next we show that, as in [11], the logical $\otimes$- and $\bigotimes x$.-operators of $\mathbf{wTML}$ in general are not stable with respect to recognizability by WTPDA.

*Example 3.3.* Here we show that the use of a formula $f(\mu - x)$ in the scope of a quantifier $\bigotimes y$ with $y \neq x$ can lead to unrecognizability by WTPDA.

Let $\Sigma = \{a\}$ be a singleton alphabet and $\mathbb{L}\mathrm{TROP}$ the timed semiring of Example 2.1 (a). Let $\beta(\mu) \in \mathbf{BOOL}(\Sigma)$ denote the formula $\forall x.\forall y.(\mu(x,y) \leftrightarrow \forall z.(x \leq z \leq y))$. Consider the $\mathbf{wTML}(\Sigma, \mathbb{L}\mathrm{TROP})$-sentence

$$\psi = \bigoplus \mu.(\beta(\mu) \otimes \bigoplus x.\bigotimes y.[(x \leq y) \otimes \mathrm{id}(\mu - x)]).$$

For $n \in \mathbb{N} \setminus \{0,1\}$, let $w_n = (a,0)^{n-1}(a,n) \in \mathbb{T}\Sigma^+$. Then $[\![\psi]\!](w_n) = n^2$.

Suppose that there exists a WTPDA $\mathcal{A}$ over $\Sigma$ and $\mathbb{L}\mathrm{TROP}$ with $[\![\mathcal{A}]\!] = [\![\psi]\!]$. We may assume that $\mathcal{A}$ does not contain edges of the infinite weight. Let $M \in \mathbb{R}_{\geq 0}$ be the maximal value of all $m$ which are either weights of edges of $\mathcal{A}$ or appear in functions $t \mapsto m \cdot t$ which are the weights of the stack letters. Let $n \in \mathbb{N} \setminus \{0,1\}$ and $\rho \in \mathrm{Run}_{\mathcal{A}}(w_n)$. Then $\mathrm{wt}(\rho) \leq 2Mn$. Since $[\![\mathcal{A}]\!](w_n) = n^2 \neq \infty$, we have $n^2 = [\![\mathcal{A}]\!](w_n) \leq 2Mn$ which is false for big enough $n$. A contradiction. Hence the WTL $[\![\psi]\!]$ is not pushdown recognizable over $\mathbb{L}\mathrm{TROP}$.

*Example 3.4.* Here we show that the nested use of quantifiers $\bigotimes y$ can lead to unrecognizability by WTPDA. Consider the $\mathbf{wTML}(\Sigma, \mathbb{L}\mathrm{TROP})$-sentence

$$\psi' = \bigoplus \mu.(\beta(\mu) \otimes \bigotimes x.\bigotimes y.[(\exists z.z < x) \oplus ((\forall z.x \leq z) \otimes \mathrm{id}(\mu - x))])$$

where $\beta(\mu)$ is defined as in the previous example. Then $[\![\psi']\!] = [\![\psi]\!]$ where $\psi$ is defined as in the previous example. Hence the WTL $[\![\psi']\!]$ is not pushdown recognizable over $\mathbb{L}\mathrm{TROP}$.

*Example 3.5.* Here we show that the use of formulas $f(\mu-x)\otimes g(\mu-x)$ with $f, g \in \mathcal{F}$ can lead to the unrecognizability. Consider the timed semiring ExpTrop of Example 2.1 (c). Let $f_1 : t \mapsto e^t$ and $f_2 : t \mapsto e^{2t}$. Let $\Sigma = \{a\}$ be a singleton alphabet. Consider the $\mathbf{wTML}(\Sigma, \text{ExpTrop})$-sentence

$$\psi = \bigoplus \mu.(\beta(\mu) \otimes \bigotimes x.(f_1(\mu - x) \otimes f_2(\mu - x))$$

where $\beta(\mu) = \exists x.\exists y.(x < y \wedge \forall z.(z = x \vee z = y) \wedge \mu(x, y))$. For $t \in \mathbb{R}_{\geq 0}$, let $w_t = (a, 0)(a, t) \in \mathbb{T}\Sigma^+$. Then $[\![\psi]\!](w_t) = e^t + e^{2t}$. Suppose that there exists a WTPDA $\mathcal{A}$ over $\Sigma$ and ExpTrop with $[\![\mathcal{A}]\!] = [\![\psi]\!]$. Then, for every $t \in \mathbb{R}_{\geq 0}$, there exist $s(t), c(t) \in \mathbb{R}_{\geq 0}$ with $[\![\mathcal{A}]\!](w_t) = s(t) + e^{c(t)\cdot t}$. Moreover, the sets $\{s(t) \mid t \in \mathbb{R}_{\geq 0}\}$ and $\{c(t) \mid t \in \mathbb{R}_{\geq 0}\}$ are finite. Then for a big enough value $t_0 \in \mathbb{R}_{\geq 0}$ and all $t \geq t_0$ we have $c(t) = \frac{\ln(e^t + e^{2t} - s(t))}{t}$. So $c(t)$ has infinitely many values. A contradiction.

Motivated by these examples, we introduce a syntactically restricted fragment of $\mathbf{wTML}$ as follows. As in [11], we restrict the use of the $\bigotimes x$-quantifier to the *almost boolean formulas*. In contrast to [11], we have new formulas of the form $f(\mu - x)$ for which we have to take into account the situations described in Examples 3.3 and 3.4.

For $x \in V_1$, the set $\mathbf{aBOOL}(\Sigma, \mathbb{S}, x)$ of *almost boolean formulas* over $\Sigma$, $\mathbb{S}$ and $x$ is generated by the grammar

$$\gamma \ ::= \ \beta \mid s \otimes f(\mu - x) \mid \gamma \oplus \gamma \mid \beta \otimes \gamma$$

where $s \in S$, $f \in \mathcal{F}$ and $\beta \in \mathbf{BOOL}(\Sigma)$.

**Definition 3.6.** Restricted weighted timed matching logic $\mathbf{wTML}^{\text{res}}(\Sigma, \mathbb{S}) \subseteq \mathbf{wTML}(\Sigma, \mathbb{S})$ *is defined to be the set of all formulas of the form* $\bigoplus \mu.\varphi$ *where* $\varphi$ *is produced by the grammar*

$$\varphi \ ::= \ \beta \mid s \otimes f(\mu - x) \mid \varphi \oplus \varphi \mid \beta \otimes \varphi \mid \bigoplus x.\varphi \mid \bigoplus X.\varphi \mid \bigotimes x.\gamma$$

*with* $\beta \in \mathbf{BOOL}(\Sigma, \mathbb{S})$*,* $s \in S$*,* $f \in \mathcal{F}$*,* $x \in V_1$*,* $X \in V_2$ *and* $\gamma \in \mathbf{aBOOL}(\Sigma, \mathbb{S}, x)$*.*

*Remark 3.7.* Note that, in the logical fragments $\mathbf{aBOOL}(\Sigma, \mathbb{S}, x)$ and $\mathbf{wTML}^{\text{res}}(\Sigma, \mathbb{S})$, a constant $s \in S$ can be expressed by means of the formula $s \otimes \overline{\mathbb{1}}(\mu - x)$. Moreover, a formula $f(\mu - x)$ can be expressed by means of the formula $\mathbb{1} \otimes f(\mu - x)$.

Our main result is the following theorem.

**Theorem 3.8.** *Let* $\Sigma$ *be an alphabet,* $\mathbb{S} = (S, \mathcal{F}, +, \cdot, \mathbb{0}, \mathbb{1})$ *a timed semiring and* $\mathbb{W} : \mathbb{T}\Sigma^+ \to S$ *a WTL. Then* $\mathbb{W}$ *is pushdown recognizable over* $\mathbb{S}$ *iff* $\mathbb{W}$ *is* $\mathbf{wTML}^{\text{res}}(\Sigma, \mathbb{S})$*-definable.*

We will prove this theorem in Sect. 6.

## 4 Visibly Pushdown Languages

For the rest of the paper, we fix a special stack symbol $\bot$.

A *pushdown alphabet* is a triple $\tilde{\Sigma} = \langle \Sigma^{\mathrm{push}}, \Sigma^{\#}, \Sigma^{\mathrm{pop}} \rangle$ with pairwise disjoint sets $\Sigma^{\mathrm{push}}$, $\Sigma^{\#}$ and $\Sigma^{\mathrm{pop}}$ of *push*, *local* and *pop* letters, respectively. Let $\Sigma = \Sigma^{\mathrm{push}} \cup \Sigma^{\#} \cup \Sigma^{\mathrm{pop}}$. A *visibly pushdown automaton (VPA)* over $\tilde{\Sigma}$ is a tuple $\mathcal{A} = (Q, \Gamma, Q_0, T, Q_f)$ where $Q$ is a finite set of states, $Q_0, Q_f \subseteq Q$ are sets of initial resp. final states, $\Gamma$ is a stack alphabet with $\bot \notin \Gamma$, and $T = T^{\mathrm{push}} \cup T^{\#} \cup T^{\mathrm{pop}}$ is a set of transitions where $T^{\mathrm{push}} \subseteq Q \times \Sigma^{\mathrm{push}} \times \Gamma \times Q$ is a set of push transitions, $T^{\#} \subseteq Q \times \Sigma^{\#} \times Q$ is a set of local transitions and $T^{\mathrm{pop}} \subseteq Q \times \Sigma^{\mathrm{pop}} \times (\Gamma \cup \{\bot\}) \times Q$ is a set of pop transitions.

We define the label of a transition $\tau \in T$ depending on its sort as follows. If $\tau = (p, c, \gamma, p') \in T^{\mathrm{push}} \cup T^{\mathrm{pop}}$ or $\tau = (p, c, p') \in T^{\#}$, we let $\mathrm{label}(\tau) = c$, so $c \in \Sigma^{\mathrm{push}} \cup \Sigma^{\mathrm{pop}}$ resp. $c \in \Sigma^{\#}$.

A *configuration* of $\mathcal{A}$ is a pair $\langle q, u \rangle$ where $q \in Q$ and $u \in \Gamma^*$. Let $\tau \in T$ be a transition. Then, we define the transition relation $\vdash_\tau$ on configurations of $\mathcal{A}$ as follows. Let $c = \langle q, u \rangle$ and $c' = \langle q', u' \rangle$ be configurations of $\mathcal{A}$.

- If $\tau = (p, a, \gamma, p') \in T^{\mathrm{push}}$, then we put $c \vdash_\tau c'$ iff $p = q$, $p' = q'$ and $u' = \gamma u$.
- If $\tau = (p, a, p') \in T^{\#}$, then we put $c \vdash_\tau c'$ iff $p = q$, $p' = q'$ and $u' = u$,
- If $\tau = (p, a, \gamma, p') \in T^{\mathrm{pop}}$ with $\gamma \in \Gamma \cup \{\bot\}$, then we put $c \vdash_\tau c'$ iff $p = q$, $p' = q'$ and either $\gamma \neq \bot$ and $u = \gamma u'$, or $\gamma = \bot$ and $u' = u = \varepsilon$.

We say that $c = \langle q, u \rangle$ is an *initial* configuration if $q \in Q_0$ and $u = \varepsilon$. We call $c$ a *final* configuration if $q \in Q_f$. A *run* of $\mathcal{A}$ is a sequence $\rho = c_0 \vdash_{\tau_1} c_1 \vdash_{\tau_2} \ldots \vdash_{\tau_n} c_n$ where $c_0, c_1, \ldots, c_n$ are configurations of $\mathcal{A}$ such that $c_0$ is initial, $c_n$ is final and $\tau_1, \ldots, \tau_n \in T$. Let $\mathrm{label}(\rho) = \mathrm{label}(\tau_1) \ldots \mathrm{label}(\tau_n) \in \Sigma^+$, the *label* of $\rho$. Let $\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^+ \mid \text{there exists a run } \rho \text{ of } \mathcal{A} \text{ with } \mathrm{label}(\rho) = w\}$. We say that a language $\mathcal{L} \subseteq \Sigma^+$ is a *visibly pushdown language* over $\tilde{\Sigma}$ if there exists a VPA $\mathcal{A}$ over $\Sigma$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.

*Remark 4.1.* Note that we do not demand for final configurations that $u = \varepsilon$ and we can read a pop letter even if the stack is empty (using the special stack symbol $\bot$). This permits to consider the situations where some pop letters are not balanced by push letters and vice versa.

We say that a VPA $\mathcal{A} = (Q, \Gamma, Q_0, T, Q_f)$ is *deterministic* [4] if $|Q_0| = 1$ and for every $q \in Q$:

- for every $a \in \Sigma^{\mathrm{push}}$, there is at most one transition of the form $(q, a, \gamma, q') \in T$,
- for every $a \in \Sigma^{\#}$, there is at most one transition of the form $(q, a, q') \in T$, and
- for every $a \in \Sigma^{\mathrm{pop}}$ and $\gamma \in \Gamma$, there is at most one transition of the form $(q, a, \gamma, q') \in T$.

Note that in a deterministic VPA $\mathcal{A}$ for every word $w \in \Sigma^+$ there exists at most one run with label $w$.

**Theorem 4.2 (Alur, Madhusudan [4]).** *Let $\tilde{\Sigma}$ be a pushdown alphabet and $\mathcal{A}$ a VPA over $\tilde{\Sigma}$. Then there exists a deterministic VPA $\mathcal{A}'$ over $\tilde{\Sigma}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

We note that the visibly pushdown languages over $\tilde{\Sigma}$ form a proper subclass of the context-free languages over $\Sigma$, cf. [4] for further properties.

For any word $w = a_1...a_n \in \Sigma^+$, let $\textsc{Mask}(w) = b_1...b_n \in \{-1, 0, 1\}^+$ such that, for all $1 \leq i \leq n$, $b_i = 1$ if $a_i \in \Sigma^{\text{push}}$, $b_i = 0$ if $a_i \in \Sigma^{\#}$, and $b_i = -1$ otherwise. Let $\mathbb{L}(\tilde{\Sigma}) \subseteq \{-1, 0, 1\}^*$ be the language which contains $\varepsilon$ and all words $b_1...b_n \in \{-1, 0, 1\}^+$ such that $\sum_{j=1}^{n} b_j = 0$ and $\sum_{j=1}^{i} b_j \geq 0$ for all $i \in \{1, ..., n\}$. Here, we interpret $1$ as the left parenthesis, $-1$ as the right parenthesis and $0$ as an irrelevant symbol. Then, $\mathbb{L}(\tilde{\Sigma})$ is the set of all sequences with correctly nested parentheses.

Next, we turn to the logic $\mathbf{MSO}_{\mathbb{L}}(\tilde{\Sigma})$ over the pushdown alphabet $\tilde{\Sigma}$ which extends the classical MSO logic on finite words by the binary relation which checks whether a push letter and a pop letter are matching. This logic was shown in [4] to be equivalent to visibly pushdown automata. The logic $\mathbf{MSO}_{\mathbb{L}}(\tilde{\Sigma})$ is defined by the grammar

$$\varphi \ ::= \ P_a(x) \mid x \leq y \mid X(x) \mid \mathbb{L}(x, y) \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x.\varphi \mid \exists X.\varphi$$

where $a \in \Sigma$, $x, y \in V_1$ and $X \in V_2$. The formulas in $\mathbf{MSO}_{\mathbb{L}}(\tilde{\Sigma})$ are interpreted over a word $w = a_1...a_n \in \Sigma^+$ and a variable assignment $\sigma : V_1 \cup V_2 \to \text{dom}(w) \cup 2^{\text{dom}(w)}$. We will write $(w, \sigma) \models \mathbb{L}(x, y)$ iff $\sigma(x) < \sigma(y)$, $a_{\sigma(x)} \in \Sigma^{\text{push}}$, $a_{\sigma(y)} \in \Sigma^{\text{pop}}$ and $\textsc{Mask}(a_{\sigma(x)+1}...a_{\sigma(y)-1}) \in \mathbb{L}(\tilde{\Sigma})$. For other formulas, the satisfaction relation is defined as usual. If $\varphi$ is a sentence, then the satisfaction relation does not depend on a variable assignment and we can simply write $w \models \varphi$. For a sentence $\varphi \in \mathbf{MSO}_{\mathbb{L}}(\tilde{\Sigma})$, let $\mathcal{L}(\varphi) = \{w \in \Sigma^+ \mid w \models \varphi\}$. We say that a language $\mathcal{L} \subseteq \Sigma^+$ is $\mathbf{MSO}_{\mathbb{L}}(\tilde{\Sigma})$-*definable* if there exists a sentence $\varphi \in \mathbf{MSO}_{\mathbb{L}}(\tilde{\Sigma})$ such that $\mathcal{L}(\varphi) = \mathcal{L}$.

The following result states the expressive equivalence of visibly pushdown automata and $\mathbf{MSO}_{\mathbb{L}}$-logic.

**Theorem 4.3 (Alur, Madhusudan [4]).** *Let $\tilde{\Sigma} = \langle \Sigma^{\text{push}}, \Sigma^{\#}, \Sigma^{\text{pop}} \rangle$ be a pushdown alphabet, $\Sigma = \Sigma^{\text{push}} \cup \Sigma^{\#} \cup \Sigma^{\text{pop}}$, and $\mathcal{L} \subseteq \Sigma^+$ a language. Then, $\mathcal{L}$ is a visibly pushdown language over $\tilde{\Sigma}$ iff $\mathcal{L}$ is $\mathbf{MSO}_{\mathbb{L}}(\tilde{\Sigma})$-definable.*

## 5 Decomposition of Weighted Timed Pushdown Automata

In this section we prove a Nivat-like (cf. [20, 5]) decomposition theorem for WTPDA. This result establishes a connection between pushdown recognizable WTL and visibly pushdown languages of Alur and Madhusudan [4]. We will use this theorem for the proof of our Theorem 3.8. However, our result could be also of independent interest.

The key idea of our decomposition result is to consider a pushdown recognizable WTL as a renaming of a pushdown recognizable WTL over an extended alphabet which encodes the information about weights and a timed stack; on the level of this extended alphabet we can separate the setting of visibly pushdown languages from the weighted timed setting. Our separation technique appeals to the partitioning of $\mathbb{R}_{\geq 0}$ into finitely many intervals; this finite partition will be used for the construction of the desired extended alphabet.

A Nivat-like theorem for weighted timed automata without stack was given in [13] where a connection between recognizable weighted timed languages and unambiguously recognizable unweighted timed languages was established. This approach is not suitable for WTPDA since pushdown languages are not closed under intersection and the weights of the timed stack depend on the inner components of unweighted TPDA.

Let $\Sigma$ be an alphabet and $\mathbb{S} = (S, \mathcal{F}, +, \cdot, \mathbb{0}, \mathbb{1})$ a timed semiring.

Let $k \in \mathbb{N}$ and $\hat{S} \subseteq S$, $\hat{\mathcal{F}} \subseteq \mathcal{F}$ be finite non-empty sets. Let $\mathbb{P}(k) = \{[0,0], (0,1), [1,1], (1,2), ..., [k,k], (k,\infty)\} \subseteq 2^{\mathcal{I}}$, the *k-interval partition* of $\mathbb{R}_{\geq 0}$. Let $\Delta = \{\text{push}, \#, \text{pop}\}$ and $\Omega$ be an alphabet. For each $\delta \in \Delta$, let $\mathcal{R}^{\delta} = \Omega \times \mathbb{P}(k) \times \hat{S} \times \hat{\mathcal{F}} \times \{\delta\}$. For our decomposition result, we will use the *extended alphabet* $\mathcal{R} = \mathcal{R}^{\text{push}} \cup \mathcal{R}^{\#} \cup \mathcal{R}^{\text{pop}}$ and the *extended pushdown alphabet* $\tilde{\mathcal{R}} = \langle \mathcal{R}^{\text{push}}, \mathcal{R}^{\#}, \mathcal{R}^{\text{pop}} \rangle$. Note that $\mathcal{R}$ and $\tilde{\mathcal{R}}$ depend on the variables $k$, $\hat{S}$, $\hat{\mathcal{F}}$ and $\Omega$. However, we will not explicitly designate this dependence.

Let $\mathcal{T} \subseteq \mathbb{T}\mathcal{R}^+$ be the timed language defined as follows. Consider the timed word $w = (r_1, t_1)...(r_n, t_n) \in \mathbb{T}\mathcal{R}^+$ with $r_i = (\omega_i, I_i, s_i, f_i, \delta_i) \in \mathcal{R}$. Then we let $w \in \mathcal{T}$ iff the following hold:

- $\text{MASK}(r_1...r_n) \in \mathbb{L}(\tilde{\mathcal{R}})$;
- for all $i, i' \in \{1, ..., n\}$ with $i < i'$, $\delta_i = \text{push}$, $\delta_{i'} = \text{pop}$ and $\text{MASK}(r_{i+1}...r_{i'-1}) \in \mathbb{L}(\tilde{\mathcal{R}})$, we have $t_{i'} - t_i \in I_{i'}$.

Note that in the definition of $\mathcal{T}$ there are no restrictions on the components $\hat{S}$ and $\hat{\mathcal{F}}$ and that $\mathcal{T}$ is captured by a "simple" TPDA with a single state and a single stack letter; this TPDA processes the timed stack according to the information encoded in the additional components $\mathbb{P}(k)$ and $\Delta$ of the extended alphabet $\tilde{\mathcal{R}}$.

Let $\text{val}(\mathcal{T}) : \mathbb{T}\mathcal{R}^+ \to S$ denote the WTL defined as follows. For all $w \in \mathbb{T}\mathcal{R}^+ \setminus \mathcal{T}$, we let $\text{val}(\mathcal{T})(w) = \mathbb{0}$. For all $w = (r_1, t_1)...(r_n, t_n) \in \mathcal{T}$ with $r_j = (\omega_j, I_j, s_j, f_j, \delta_j)$, we let $\text{val}(\mathcal{T})(w) = \prod_{j=1}^{n} (s_j \cdot \phi_j)$ where $\phi_j = \mathbb{1}$ whenever $\delta_j \neq \text{pop}$ and $\phi_j = f_i(t_j - t_i)$ otherwise where $i < j$ and $\text{MASK}(r_i...r_j) \in \mathbb{L}(\tilde{\mathcal{R}})$.

We introduce the following operations.

- Let $\mathbb{W} : \mathbb{T}\mathcal{R}^+ \to S$ be a WTL and $\mathcal{L} \subseteq \mathcal{R}^+$ a language. Let $(\mathbb{W} \cap \mathcal{L}) : \mathbb{T}\mathcal{R}^+ \to S$ be the "restriction" of $\mathbb{W}$ to $\mathcal{L}$, i.e., for all $w = (r_1, t_1)...(r_n, t_n) \in \mathbb{T}\mathcal{R}^+$, we have: $(\mathbb{W} \cap \mathcal{L})(w) = \mathbb{W}(w)$ if $r_1...r_n \in \mathcal{L}$ and $(\mathbb{W} \cap \mathcal{L})(w) = \mathbb{0}$ otherwise.
- Let $\pi : \Omega \to \Sigma$ be a mapping called henceforth a *renaming*. For a letter $r = (\omega, I, s, f, \delta)$, let $h(r) = (h(\omega), I, s, f, \delta)$. For a word $w = (r_1, t_1)...(r_n, t_n) \in \mathbb{T}\mathcal{R}^+$, let $\pi(w) = (\pi(r_1), t_1)...(\pi(r_n), t_n) \in \mathbb{T}\Sigma^+$. For a WTL $\mathbb{W} : \mathbb{T}\mathcal{R}^+ \to S$, let $\pi(\mathbb{W}) : \mathbb{T}\Sigma^+ \to S$ be defined for all $w \in \mathbb{T}\Sigma^+$ as $\pi(\mathbb{W})(w) = \sum \big(\mathbb{W}(u) \mid u \in \mathbb{T}\mathcal{R}^+ \text{ and } \pi(u) = w\big)$.

Now we formulate our decomposition theorem.

**Theorem 5.1.** *Let $\Sigma$ be an alphabet, $\mathbb{S} = (S, \mathcal{F}, +, \cdot, \mathbb{0}, \mathbb{1})$ a timed semiring and $\mathbb{W} : \mathbb{T}\Sigma^+ \to S$ a WTL. Then the following are equivalent.*

*(a) $\mathbb{W}$ is pushdown recognizable over $\mathbb{S}$.*
*(b) There exist $k \in \mathbb{N}$, alphabets $\hat{S} \subseteq S$, $\hat{\mathcal{F}} \subseteq \mathcal{F}$ and $\Omega$, a visibly pushdown language $\mathcal{L} \subseteq \mathcal{R}^+$ over the pushdown alphabet $\tilde{\mathcal{R}}$, and a renaming $\pi : \Omega \to \Sigma$ such that $\mathbb{W} = \pi(\text{val}(\mathcal{T}) \cap \mathcal{L})$.*

First we show that (a) implies (b).

**Lemma 5.2.** *Let $\mathcal{A}$ be a WTPDA over $\Sigma$ and $\mathbb{S}$. Then there exist $k \in \mathbb{N}$, finite non-empty sets $\hat{S} \subseteq S$ and $\hat{\mathcal{F}} \subseteq \mathcal{F}$, a visibly pushdown language $\mathcal{L} \subseteq \mathcal{R}^+$ over the pushdown alphabet $\tilde{\mathcal{R}}$, and a renaming $\pi : \Omega \to \Sigma$ such that $[\![\mathcal{A}]\!] = \pi(\mathrm{val}(\mathcal{T}) \cap \mathcal{L})$.*

*Proof.* Let $\mathcal{A} = (L, \Gamma, L_0, E, L_f, \mathrm{wt})$. If $E^{\mathrm{pop}} = \emptyset$, then let $k = 0$. Otherwise, let $k \in \mathbb{N}$ be the maximal natural number which appears in $E$ (in the $\mathcal{S}(\Gamma)$-component of some edge in $E^{\mathrm{pop}}$). Let $\hat{S} = \mathrm{wt}(E)$, $\hat{\mathcal{F}} = \mathrm{wt}(\Gamma)$ and $\Omega = E$. Consider the visibly pushdown automaton $\mathcal{A}' = (L, \Gamma, L_0, T, L_f)$ over the pushdown alphabet $\tilde{\mathcal{R}}$ where the set $T = T^{\mathrm{push}} \cup T^{\#} \cup T^{\mathrm{pop}}$ is defined as follows. We simulate every edge $e = \left(\ell \xrightarrow{a,\mathrm{st}} \ell'\right) \in E$ with $\mathrm{wt}(e) = s$ by (possibly multiple) transitions in $T$ depending on the sort of $e$ as follows.

- Let $e \in E^{\mathrm{push}}$. Then we let $(\ell, r, \gamma, \ell') \in T^{\mathrm{push}}$ where $r = (e, [0,0], s, \overline{\mathbb{1}}, \mathrm{push})$.
- Let $e \in E^{\#}$. Then we let $(\ell, r, \ell') \in T^{\#}$ where $r = (e, [0,0], s, \overline{\mathbb{1}}, \#)$.
- Let $e \in E^{\mathrm{pop}}$ and $\mathrm{st} = (\mathrm{pop}, \gamma, I)$. Let $\mathrm{wt}(\gamma) = f$. Then we let $(\ell, r, \gamma, \ell') \in T^{\mathrm{pop}}$ for all $r = (a, \sigma, s, f, \mathrm{pop})$ with $\sigma \in \mathbb{P}(k)$ such that $\sigma \subseteq I$. Note that we do not have transitions in $T^{\mathrm{pop}}$ whose stack letter is $\perp$.

Note that although the emptiness of the stack at the end of run is not required by visibly pushdown automata, it is checked by intersection with the WTL $\mathrm{val}(\mathcal{T})$.

Let $\pi : \Omega \to \Sigma$ be defined as $\pi(e) = \mathrm{label}(e)$ for all $e \in \Omega$. Then $[\![\mathcal{A}]\!] = \pi(\mathrm{val}(\mathcal{T}) \cap \mathcal{L}(\mathcal{A}'))$. This can be shown using the intuition that the WTL $\mathrm{val}(\mathcal{T}) \cap \mathcal{L}(\mathcal{A}')$ checks whether a timed word over the extended alphabet $\mathcal{R}$ encodes a run of $\mathcal{A}$ and, if this is the case, computes the weight of this run; then the renaming $\pi$ removes the auxiliary components of the extended alphabet and computes the sum of the weights of all runs over a given timed word. $\square$

Now we turn to the converse direction of Theorem 5.1.

**Lemma 5.3.** *Let $k \in \mathbb{N}$, $\hat{S} \subseteq S$, $\hat{\mathcal{F}} \subseteq \mathcal{F}$ and $\Omega$ be alphabets, and $\mathcal{L} \subseteq \mathcal{R}^+$ a visibly pushdown language over the pushdown alphabet $\tilde{\mathcal{R}}$. Then there exists a WTPDA $\mathcal{A}'$ over $\mathcal{R}$ and $\mathbb{S}$ such that $[\![\mathcal{A}']\!] = \mathrm{val}(\mathcal{T}) \cap \mathcal{L}$.*

*Proof.* The main difficulty of the proof is to assign weights to the stack letters; note that they are encoded in the $\hat{\mathcal{F}}$-component of the extended alphabet $\mathcal{R}$. We proceed as follows. We take a deterministic VPA for $\mathcal{L}$. We mark the stack letters of $\mathcal{A}$ with a function from $\hat{\mathcal{F}}$ which will be the weight of this compound stack letter. Whenever we have to push a letter $\gamma$ into the stack of $\mathcal{A}$, we nondeterministically push into the stack of $\mathcal{A}'$ all letters $(\gamma, f)$ with $f \in \hat{\mathcal{F}}$. Whenever we pop a letter $\gamma$ from the stack of $\mathcal{A}$, in $\mathcal{A}'$ we can pop only the pair $(\gamma, f)$ where $f$ is the $\hat{\mathcal{F}}$-component of the input $\mathcal{R}^{\mathrm{pop}}$-letter. Note that this construction is unambiguous, i.e., for every input word $w$ there exists at most one run labeled by $w$.

By Theorem 4.2 there exists a deterministic visibly pushdown automaton $\mathcal{A} = (L, \Gamma, L_0, T, L_f)$ over the pushdown alphabet $\tilde{\mathcal{R}}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. Then we put $\mathcal{A}' = (L, \Gamma \times \hat{\mathcal{F}}, L_0, E, L_f, \mathrm{wt})$ where $E = E^{\mathrm{push}} \cup E^{\#} \cup E^{\mathrm{pop}}$ is defined as follows.

– For every push transition $t = (\ell, r, \gamma, \ell') \in T^{\mathrm{push}}$ with $r = (\omega, I, s, f, \mathrm{push})$ we let $e = \big(\ell \xrightarrow{r,(\mathrm{push},(\gamma,\varphi))} \ell'\big) \in E^{\mathrm{push}}$ for all $\varphi \in \hat{\mathcal{F}}$. Moreover, we put $\mathrm{wt}(e) = s$.
– For every local transition $t = (\ell, r, \ell') \in T^{\#}$ with $r = (\omega, I, s, f, \#)$ we let $e = \big(\ell \xrightarrow{r,\#} \ell'\big) \in E^{\#}$. Moreover, we put $\mathrm{wt}(e) = s$.
– For every pop transition $t = (\ell, r, \gamma, \ell') \in T^{\mathrm{pop}}$ with $r = (\omega, I, s, f, \mathrm{pop})$ we let $e = \big(\ell \xrightarrow{r,(\mathrm{pop},(\gamma,f),I)} \ell'\big) \in E^{\mathrm{pop}}$. Moreover, we put $\mathrm{wt}(e) = s$.

For every stack letter $(\gamma, f) \in \Gamma \times \hat{\mathcal{F}}$, we let $\mathrm{wt}(\gamma, f) = f$. Then $[\![\mathcal{A}']\!] = \mathrm{val}(\mathcal{T}) \cap \mathcal{L}$. □

**Lemma 5.4.** *Let $k \in \mathbb{N}$, $\hat{S} \subseteq S$, $\hat{\mathcal{F}} \subseteq \mathcal{F}$ and $\Omega$ be alphabets, $\mathbb{W} : \mathbb{T}\mathcal{R}^+ \to S$ be a WTL which is pushdown recognizable over $\mathbb{S}$, and $\pi : \Omega \to \Sigma$ a renaming. Then the WTL $\pi(\mathbb{W})$ is also pushdown recognizable over $\mathbb{S}$.*

*Proof.* Our construction is a slight modification of the standard renaming construction for semiring-weighted automata [12].

Let $\mathcal{A} = (L, \Gamma, L_0, E, L_f, \mathrm{wt})$ be a WTPDA over $\mathcal{R}$ and $\mathbb{S}$ with $[\![\mathcal{A}]\!] = \mathbb{W}$. We consider the WTPDA $\mathcal{A}' = (L, \Gamma, L_0, E', L_f, \mathrm{wt}')$ over $\mathcal{R}$ and $\mathbb{S}$ where $E'$ and $\mathrm{wt}'$ are defined as follows. For every edge $e = \big(\ell \xrightarrow{r,\mathrm{st}} \ell'\big) \in E$ with $r \in \mathcal{R}$, let $\pi(e) = \big(\ell \xrightarrow{\pi(r),\mathrm{st}} \ell'\big)$. Then we let $E' = \bigcup\{\pi(e) \mid e \in E\}$, $\mathrm{wt}'(e') = \sum \big(\mathrm{wt}(e) \mid e \in E \text{ and } \pi(e) = e'\big)$ for all $e' \in E'$, and $\mathrm{wt}'(\gamma) = \mathrm{wt}(\gamma)$ for all $\gamma \in \Gamma$. Then $[\![\mathcal{A}']\!] = \pi([\![\mathcal{A}]\!])$. □

Now Theorem 5.1 (b) implies (a) is immediate by Lemmas 5.3 and 5.4.

## 6 Definability Equals Recognizability

In this section, we give a proof of Theorem 3.8.

First we show that definability by $\mathbf{wTML}^{\mathrm{res}}(\Sigma, \mathbb{S})$-sentences implies recognizability by WTPDA. We follow a similar approach as in Theorem 6.6 of [13]. First, we transform a $\mathbf{wTML}^{\mathrm{res}}(\Sigma, \mathbb{S})$-formula $\psi$ into a canonical formula of the simpler form. Then, using Theorem 4.3, we establish for a canonical formula a decomposition of the form $\pi(\mathrm{val}(\mathcal{T}) \cap \mathcal{L})$ as stated in Theorem 5.1. Then, by Theorem 5.1, the WTL $[\![\psi]\!]$ is pushdown recognizable over $\mathbb{S}$.

For $\varphi \in \mathbf{wTML}(\Sigma, \mathbb{S})$ and a finite set $\mathcal{V} = \{\mathcal{X}_1, ..., \mathcal{X}_k\} \subseteq V_1 \cup V_2$ of pairwise distinct variables $\mathcal{X}_1, ..., \mathcal{X}_k$, let $\bigoplus\mathcal{V}.\varphi$ denote the formula $\bigoplus\mathcal{X}_1 ... \bigoplus\mathcal{X}_k.\varphi$. In particular, we let $\bigoplus\emptyset.\varphi = \varphi$. We say that a sentence $\psi \in \mathbf{wTML}^{\mathrm{res}}(\Sigma, \mathbb{S})$ is *canonical* if it is of the form $\psi = \bigoplus\mu.\bigoplus\mathcal{V}.\bigotimes x.\bigoplus_{i=1}^{l}(\beta_i \otimes s_i \otimes f_i(\mu - x))$ where $\mathcal{V} \subseteq V_1 \cup V_2$ is a finite set, $x \in V_1$, $l \geq 1$, $s_i \in S$, $f_i \in \mathcal{F}$ and $\beta_i \in \mathbf{BOOL}(\Sigma)$ such that, for every $(w, \sigma) \in \mathbb{T}\Sigma_V^+$, there exists exactly one $i \in \{1, ..., l\}$ with $(w, \sigma) \models \beta_i$. Let $\mathbf{wTML}^{\mathrm{can}}(\Sigma, \mathbb{S})$ denote the set of all canonical sentences. Clearly, $\mathbf{wTML}^{\mathrm{can}}(\Sigma, \mathbb{S}) \subseteq \mathbf{wTML}^{\mathrm{res}}(\Sigma, \mathbb{S})$.

**Lemma 6.1.** *The logical fragments $\mathbf{wTML}^{\mathrm{can}}(\Sigma, \mathbb{S})$ and $\mathbf{wTML}^{\mathrm{res}}(\Sigma, \mathbb{S})$ are expressively equivalent.*

*Proof.* Let $\psi \in \mathbf{wTML}^{\mathrm{res}}(\Sigma, \mathbb{S})$. We show that there exists a canonical sentence $\chi \in \mathbf{wTML}^{\mathrm{can}}(\Sigma, \mathbb{S})$ with $[\![\chi]\!] = [\![\psi]\!]$. Let $\psi = \bigoplus \mu.\psi'$.

We say that a formula $\zeta$ is *semi-canonical* if it is of the form $\bigoplus \mathcal{V}.\bigotimes x.\gamma$ with $\gamma \in \mathbf{aBOOL}(\Sigma, \mathbb{S}, x)$. Note that here we do not quantify over the matching variable $\mu$. First, we show by induction on the structure of a subformula $\xi$ of $\psi'$ that there exists a semi-canonical formula $\zeta(\xi)$ with $\mathrm{Free}(\zeta(\xi)) = \mathrm{Free}(\xi)$ and $[\![\zeta(\xi)]\!] = [\![\xi]\!]$.

- Let $x \in V_1$ and $\xi$ be a maximal $\mathbf{aBOOL}(\Sigma, \mathbb{S}, x)$-subformula of $\psi'$. Let $y \in V_1$ be a fresh variable. First, assume that $x \in \mathrm{Free}(\xi)$. Let $\xi[x/y] \in \mathbf{aBOOL}(\Sigma, \mathbb{S}, y)$ be obtained from $\xi$ by replacing $x$ by $y$. Then, we let $\zeta(\xi) = \bigoplus \emptyset.\bigotimes y.((x \neq y) \oplus (x = y) \otimes \xi[x/y])$.
  Now assume that $x \notin \mathrm{Free}(\xi)$. Let $\min(y)$ denote the formula $\forall z.(y \leq z)$. Then, we let $\zeta(\xi) = \bigoplus \emptyset.\bigotimes y.((\neg \min(y)) \oplus (\min(y) \otimes \xi))$.
- Let $\xi = \xi_1 \oplus \xi_2$ be not almost boolean. By simple manipulations with formulas such as renamings of variables and assignments of concrete values to useless variables (e.g., $V_1$-variables are assigned to the first position of a word and $V_2$-variables to the empty set), we may assume that $\zeta(\xi_1) = \bigoplus \mathcal{V}.\bigotimes x.\gamma_1$ and $\zeta(\xi_2) = \bigoplus \mathcal{V}.\bigotimes x.\gamma_2$ with $\gamma_1, \gamma_2 \in \mathbf{aBOOL}(\Sigma, \mathbb{S}, x)$. Let $X \in V_2$ be a fresh variable. Then, we let $\zeta(\xi) = \bigoplus (\mathcal{V} \cup \{X\}).\bigotimes x.((\Psi_1 \otimes \gamma_1) \oplus (\Psi_2 \otimes \gamma_2))$ where $\Psi_1, \Psi_2 \in \mathbf{BOOL}(\Sigma)$ determine two distinct concrete values for $X$, e.g., $\Psi_1 = \neg \exists z.X(z)$ and $\Psi_2 = \forall z.X(z)$.
- Let $\xi = \beta \otimes \xi'$ with $\beta \in \mathbf{BOOL}(\Sigma)$. Let $\zeta(\xi') = \bigoplus \mathcal{V}.\bigotimes x.\gamma$. We may assume that the variables from $\mathcal{V} \cup \{x\}$ do not appear in $\beta$. Then we let $\zeta(\xi) = \bigoplus \mathcal{V}.\bigotimes x.(\beta \otimes \gamma)$.
- Let $\xi = \bigoplus \mathcal{X}.\xi'$ where $\mathcal{X} \in V_1 \cup V_2$. Let $\zeta(\xi') = \bigoplus \mathcal{V}.\bigotimes x.\gamma$. We may assume that $\mathcal{X} \notin \mathcal{V}$. Then we let $\zeta(\xi) = \bigoplus (\mathcal{V} \cup \{\mathcal{X}\}).\bigotimes x.\gamma$.
- Let $\xi = \bigotimes x.\gamma$ with $\gamma \in \mathbf{aBOOL}(\Sigma, \mathbb{S}, x)$. Then we let $\zeta(\xi) = \bigoplus \emptyset.\bigotimes x.\gamma$.

Next we show that every formula $\gamma \in \mathbf{aBOOL}(\Sigma, \mathbb{S}, x)$ can be transformed into a formula of the form $\pi(\gamma) = \bigoplus_{i=1}^{l} \big(\beta_i \otimes \big[\bigoplus_{j=1}^{r} s_{i,j} \otimes f_{i,j}(\mu - x)\big]\big)$ where $l, r \geq 1$, $b_i \in \mathbf{BOOL}(\Sigma)$, $s_{i,j} \in S$, $f_{i,j} \in \mathcal{F}$ and, for every $(w, \sigma) \in \mathbb{T}\Sigma_V^+$, there exists exactly one $i \in \{1, ..., l\}$ with $(w, \sigma) \models \beta_i$. In other words, we show that $\mathrm{Free}(\pi(\gamma)) = \mathrm{Free}(\gamma)$ and $[\![\pi(\gamma)]\!] = [\![\gamma]\!]$. Again we proceed by induction on the structure of $\gamma$.

- Let $\gamma = \beta \in \mathbf{BOOL}(\Sigma)$. Then we define the formula $\pi(\gamma)$ as $\pi(\gamma) = (\beta \otimes [\mathbb{1} \otimes \overline{\mathbb{1}}(\mu - x)]) \oplus (\neg\beta \otimes [\mathbb{0} \otimes \overline{\mathbb{1}}(\mu - x)])$.
- Let $\gamma = s \otimes f(\mu - x)$. Then we let $\pi(\gamma) = \textsc{True} \otimes [s \otimes f(\mu - x)]$ where $\textsc{True} \in \mathbf{BOOL}(\Sigma)$ is a boolean sentence with $(w, \sigma) \models \textsc{True}$ for all $(w, \sigma) \in \mathbb{T}\Sigma_V^+$, e.g., $\textsc{True} = \forall x.(x \leq x)$.
- Let $\gamma = \gamma_1 \oplus \gamma_2$. Assume that $\pi(\gamma_1) = \bigoplus_{i=1}^{l} (\beta_i \otimes \kappa_i)$ such that $\kappa_i = \bigoplus_{j=1}^{r} [s_{i,j} \otimes f_{i,j}(\mu - x)]$. We assume also that $\pi(\gamma_2) = \bigoplus_{i'=1}^{l'} (\beta'_{i'} \otimes \kappa'_{i'})$ with $\kappa_{i'} = \bigoplus_{j'=1}^{r'} [s'_{i',j'} \otimes f'_{i',j'}(\mu - x)]$. Then we let
$$\pi(\gamma) = \bigoplus_{i=1}^{l} \bigoplus_{i'=1}^{l'} ([\beta_i \wedge \beta'_{i'}] \otimes [\kappa_i \oplus \kappa'_{i'}]).$$
- Let $\gamma = \beta \otimes \gamma'$. Assume that $\pi(\gamma') = \bigoplus_{i=1}^{l} (\beta_i \otimes \kappa_i)$ such that $\kappa_i = \bigoplus_{j=1}^{r} [s_{i,j} \otimes f_{i,j}(\mu - x)]$. Then, we let $\pi(\gamma) = \bigoplus_{i=1}^{l} ([\beta_i \wedge \beta] \otimes \kappa_i) \oplus (\neg\beta \otimes \kappa')$ with $\kappa' = \mathbb{0} \otimes \overline{\mathbb{1}}(\mu - x)$.

Our final goal is to resolve the sums $\bigoplus_{j=1}^{r} s_{i,j} \otimes f_{i,j}(\mu - x)$. They can be resolved using the fact that the formula $\bigotimes x.(\varphi_1 \oplus \varphi_2)$ is equivalent to the formula $\bigoplus X.\bigotimes x.([X(x) \otimes \varphi_1] \oplus [\neg X(x) \otimes \varphi_2])$.

By these transformations we obtain a canonical sentence $\chi \in \mathbf{wTML}^{\mathrm{can}}(\Sigma, \mathbb{S})$ with $[\![\chi]\!] = [\![\psi]\!]$. □

**Lemma 6.2.** *Let* $\psi \in \mathbf{wTML}^{\mathrm{can}}(\Sigma, \mathbb{S})$*. Then there exist a natural number* $k$*, alphabets* $\hat{S} \subseteq S$ *and* $\hat{\mathcal{F}} \subseteq \mathcal{F}$ *and* $\Omega$*, a sentence* $\varphi \in \mathbf{MSO}_{\mathbb{L}}(\tilde{\mathcal{R}})$*, and a renaming* $\pi : \Omega \to \Sigma$ *such that* $[\![\psi]\!] = \pi(\mathrm{val}(\mathcal{T}) \cap \mathcal{L}(\varphi))$*.*

*Proof.* Here we use a similar idea as in the proof of Theorem 6.6 of [13].

Let $\psi = \bigoplus \mu.\bigoplus \mathcal{V}.\bigotimes x.\bigoplus_{i=1}^{l}(\beta_i \otimes s_i \otimes f_i(\mu - x))$ be a canonical formula. Assume that $\mathcal{V} = \{\mathcal{X}_1, ..., \mathcal{X}_p\}$ where $\mathcal{X}_1, ..., \mathcal{X}_p \in V_1 \cup V_2$ are pairwise distinct variables. If $\psi$ does not contain any subformula of the form $\mu^I(x, y)$, then we let $k = 0$. Otherwise, let $k \in \mathbb{N}$ be the maximal natural number appearing as a lower or upper bound of some interval $I \in \mathcal{I}$ appearing in some subformula $\mu^I(x, y)$ of $\psi$. Let $\hat{S} = \{s_i \mid 1 \leq i \leq l\}$ and $\hat{\mathcal{F}} = \{f_i \mid 1 \leq i \leq l\}$. Let $\Omega = \Sigma \times 2^{\mathcal{V}}$. Let $\pi : \Omega \to \Sigma$ be the projection to the $\Sigma$-component. It remains to construct a sentence $\varphi \in \mathbf{MSO}_{\mathbb{L}}(\tilde{\mathcal{R}})$. For any formula $\beta \in \mathbf{BOOL}(\Sigma)$, let $\beta^* \in \mathbf{MSO}_{\mathbb{L}}(\tilde{\mathcal{R}})$ be obtained from $\beta$ by the following substitutions.

- If $P_a(x)$ with $a \in \Sigma$ is a subformula of $\beta$, then $P_a(x)$ is replaced by the formula $\bigvee \big(P_{(a,\kappa)}(x) \mid \kappa \in 2^{\mathcal{V}} \times \mathbb{P}(k) \times \hat{S} \times \hat{\mathcal{F}} \times \Delta\big)$.
- If $\mu^I(x, y)$ is a subformula of $\beta$ with $I \in \mathcal{I}$ such that either $I \subseteq [0, k]$ or $I = [k, \infty)$ or $I = (k, \infty)$, then $\mu^I(x, y)$ is replaced by the formula $\mathbb{L}(x, y) \wedge \bigvee \big(P_{(\omega,J,\kappa)}(y) \mid \omega \in \Omega, J \in \mathbb{P}(k) \text{ with } J \subseteq I, \kappa \in \hat{S} \times \hat{\mathcal{F}} \times \{\mathrm{pop}\}\big)$.

For a variable $\mathcal{Z} \in \mathcal{V}$ and $x \in V_1$, let $G_{\mathcal{Z}}(x)$ denote an $\mathbf{MSO}_{\mathbb{L}}(\tilde{\mathcal{R}})$-formula which demands that $\mathcal{Z}$ belongs to the $2^{\mathcal{V}}$-component of the letter at the position $x$. Using the standard Büchi encoding technique we construct the formula $\phi \in \mathbf{MSO}_{\mathbb{L}}(\tilde{\mathcal{R}})$ which encodes the values of $\mathcal{V}$-variables in the $2^{\mathcal{V}}$-component of a timed word. We let $\phi = \forall x.\big(\bigwedge_{z \in \mathcal{V} \cap V_1}(G_z(y) \leftrightarrow (x = z)) \wedge \bigwedge_{Z \in \mathcal{V} \cap V_2}(G_Z(x) \leftrightarrow (x \in Z))\big)$. For $s \in \hat{S}$ and $f \in \hat{\mathcal{F}}$ and $x \in V_1$, let $Q_{s,f}(x)$ denote the $\mathbf{MSO}_{\mathbb{L}}(\tilde{\mathcal{R}})$-formula $\bigvee \big(P_{(\kappa,s,f,\delta)}(x) \mid \kappa \in \Omega \times \mathbb{P}(k), \delta \in \Delta\big)$.

In addition, we need a formula to "fix" the $\mathbb{P}(k)$-component of the letters in $\mathcal{R}^{\mathrm{push}} \cup \mathcal{R}^{\#}$: $\zeta = \forall x.\big[\bigvee(P_r(x) \mid r \in \Omega \times \mathbb{P}(k) \times \hat{S} \times \hat{\mathcal{F}} \times \{\mathrm{push}, \#\}) \to \bigvee(P_{(\omega,[0,0],\kappa)}(x) \mid \omega \in \Omega, \kappa \in \hat{S} \times \hat{\mathcal{F}} \times \Delta)\big]$. Then the sentence $\varphi$ is defined as

$$\varphi = \exists \mathcal{X}_1 ... \exists \mathcal{X}_p.\left(\phi \wedge \zeta \wedge \forall x.\bigvee_{i=1}^{l}(\beta_i^* \wedge Q_{s_i,f_i}(x))\right).$$

Note that the $\mathbf{MSO}_{\mathbb{L}}(\tilde{\mathcal{R}})$-formulas $\beta_1^*, ..., \beta_l^*$ define a partition on the set of pairs $(w, \sigma)$ where $w \in \mathbb{T}\mathcal{R}^+$ and $\sigma$ is a variable assignment. Then, the $\hat{S}$- and $\hat{\mathcal{F}}$-components of the words satisfying $\varphi$ are uniquely determined by this partition.

Then $[\![\psi]\!] = \pi(\mathrm{val}(\mathcal{T}) \cap \mathcal{L}(\varphi))$. □

As a corollary from Lemmas 6.1 and 6.2 and Theorems 4.3 and 5.1 we obtain:

**Corollary 6.3.** *Let* $\psi \in \mathbf{wTML}^{\mathrm{res}}(\Sigma, \mathbb{S})$ *be a sentence. Then the WTL* $[\![\psi]\!]$ *is push-down recognizable over* $\mathbb{S}$.

Now we turn to the converse direction of Theorem 3.8.

**Lemma 6.4.** *Let* $\mathcal{A}$ *be a WTPDA over* $\Sigma$ *and* $\mathbb{S}$. *Then there exists a sentence* $\psi \in \mathbf{wTML}^{\mathrm{res}}(\Sigma, \mathbb{S})$ *with* $[\![\psi]\!] = [\![\mathcal{A}]\!]$.

*Proof.* We prove this theorem by a direct translation. Let $\mathcal{A} = (L, \Gamma, L_0, E, L_f, \mathrm{wt})$. Let $(e_i)_{1 \leq i \leq m}$ be an enumeration of $E$. We associate with every edge $e_i$ a fresh second-order variable $X_i$ which keeps track of positions where $e_i$ is taken along a run of $\mathcal{A}$. Let $\overline{X} = (X_1, ..., X_m)$. Let $\beta(\overline{X}) \in \mathbf{BOOL}(\Sigma)$ denote the formula which demands that values of the variables $X_1$, ..., $X_m$ form a partition of the domain of an input timed word, the successive edges of a run are connected via the same location, the labels of edges are compatible with the label of a run, a run starts in $L_0$ and ends in $L_f$.

Whenever $e_i \in E^{\mathrm{push}}$ assume that $e_i = \big(\ell_i \xrightarrow{a_i, \mathrm{push}(\gamma_i)} \ell_i'\big)$. Whenever $e_i \in E^{\mathrm{pop}}$ assume that $e_i = \big(\ell_i \xrightarrow{a_i, \mathrm{pop}(\gamma_i, I_i)} \ell_i'\big)$. For $e_i \in E^{\mathrm{pop}}$, let $\Phi_i = \mathrm{wt}(\gamma_i) \in \mathcal{F}$. For $e_i \in E^{\mathrm{push}} \cup E^{\#}$, let $\Phi_i = \overline{\mathbb{1}}$. Consider the $\mathbf{BOOL}(\Sigma)$-formula

$$\textsc{Stack}(\overline{X}, \mu) = \forall x. \forall y. \left( \mu(x,y) \rightarrow \bigvee_{\substack{1 \leq i,j \leq m, \\ e_i \in E^{\mathrm{push}}, \\ e_j \in E^{\mathrm{pop}}, \\ \gamma_i = \gamma_j}} (X_i(x) \wedge X_j(y) \wedge \mu^{I_j}(x,y)) \right) \wedge$$
$$\forall x. \left( \exists y. (\mu(x,y) \vee \mu(y,x)) \vee \bigvee_{\substack{1 \leq i \leq m, \\ e_i \in E^{\#}}} X_i(x) \right)$$

which describes the functionality of the timed stack. Finally, we construct a formula which takes care of the weights:

$$\textsc{Weighted}(\overline{X}, \mu) = \bigotimes x. \textstyle\bigoplus_{i=1}^{m} (X_i(x) \otimes \mathrm{wt}(e_i) \otimes \Phi_i(\mu - x)).$$

We let $\psi = \bigoplus \mu. \bigoplus X_1 ... \bigoplus X_m.((\beta(\overline{X}) \wedge \textsc{Stack}(\overline{X}, \mu)) \otimes \textsc{Weighted}(\overline{X}, \mu))$. Then $\psi \in \mathbf{wTML}^{\mathrm{res}}(\Sigma, \mathbb{S})$ and $[\![\psi]\!] = [\![\mathcal{A}]\!]$. □

*Proof of Theorem 3.8.* Immediate by Corollary 6.3 and Lemma 6.4.

## 7 Weighted Timed Pushdown Automata with Global Clocks

Let $\Sigma$ be an alphabet and $\mathbb{S} = (S, \mathcal{F}, +, \cdot, \mathbb{0}, \mathbb{1})$ a timed semiring. In this section, we augment the model of WTPDA of Sect. 2 with a finite set of global clocks as in the classical timed automata [3]. Note that this extended model was considered in [2].

A *weighted timed pushdown automaton with global clocks* (GWTPDA) over $\Sigma$ and $\mathbb{S}$ is a tuple $\mathcal{A} = (L, \Gamma, C, L_0, E, L_f, \mathrm{wt})$ where $L, \Gamma, L_0, L_f, \mathrm{wt}$ are defined as for

WTPDA, $C$ is a finite set of global clocks, and every edge $e : \ell \xrightarrow[\phi,\Lambda]{a,\mathrm{st}} \ell'$ of $E$ is augmented with two additional components: a constraint $\phi : C \to \mathcal{I}$ on global clocks (i.e., the edge $e$ can be taken only if the value of every global clock $c \in C$ is in the interval $\phi(c)$) and a subset $\Lambda \subseteq C$ of global clocks to be reset after taking $e$. Then, every configuration of $\mathcal{A}$ is a triple $c = (\ell, u, \nu)$ where $\ell \in L$ is a location, $u \in \mathbb{T}\Gamma^*$ is a timed stack and $\nu : C \to \mathbb{R}_{\geq 0}$ is a global clock valuation. In all other respects, the behavior of $\mathcal{A}$ is defined as in Sect. 2.

As it was shown in [23], the performance of a global clock $c \in C$ can be described by means of the so-called *relative distance formula* $\overleftarrow{\mathrm{d}}(D_c, x) \in I$ where $D_c \in V_2$, $x \in V_1$ and $I \in \mathcal{I}$. Here, the variable $D_c$ keeps track of all positions where the clock $c$ is reset. The relative distance $\overleftarrow{\mathrm{d}}(D_c, x)$ measures the time between the last reset of $c$ before the current position $x$, i.e., models the current value of the clock $c$. If such a second-order variable $D_c$ is allowed to be quantified only in the existential prefix of a logical formula, then we obtain a logical fragment which is expressively equivalent to timed automata [23].

In order to give a logical characterization of GWTPDA, we modify the fragment $\mathbf{wTML}^{\mathrm{res}}(\Sigma, \mathbb{S})$ as follows. On the level of $\mathbf{BOOL}(\Sigma)$ we add relative distance predicates of the form $\overleftarrow{\mathrm{d}}(D_c, x) \in I$ and, for such a variable $D_c$, allow only the existential quantification $\bigoplus D_c$ in the prefix of $\varphi$ (cf. Definition 3.6). So we obtain logic $\mathbf{wTML}^{\mathrm{res}}_{\mathrm{rd}}(\Sigma, \mathbb{S})$ (where rd stays for "relative distance").

**Theorem 7.1.** *Let $\Sigma$ be an alphabet, $\mathbb{S} = (S, \mathcal{F}, +, \cdot, \mathbb{0}, \mathbb{1})$ a timed semiring and $\mathbb{W} : \mathbb{T}\Sigma^+ \to S$ a WTL. Then, $\mathbb{W}$ is recognizable by a GWTPDA iff $\mathbb{W}$ is $\mathbf{wTML}^{\mathrm{res}}_{\mathrm{rd}}(\Sigma, \mathbb{S})$-definable.*

The proof of this theorem follows the same lines as the proof of Theorem 3.8 with several changes. The main difference is that we have to reflect the global clocks in the extended alphabet $\mathcal{R}$ and the extended pushdown alphabet $\tilde{\mathcal{R}}$. For every global clock $c$, we add the component $\mathbb{P}(k) \times \{0, 1\}$ where the $k$-interval partition $\mathbb{P}(k)$ takes care of the clock constraints and the $\{0, 1\}$-component indicates whether the clock was reset or not. Then, we correspondingly modify the definition of the timed language $\mathcal{T} \subseteq \mathbb{T}\mathcal{R}^+$.

Since the proof of Theorem 7.1 is constructive, as a corollary from our Theorem 7.1 and Theorem 1 of [2], we obtain:

**Corollary 7.2.** *Let $\mathbb{L}$TROP be the timed semiring of Example 2.1 (a). Then, it is decidable, given an alphabet $\Sigma$, a sentence $\psi \in \mathbf{wTML}^{\mathrm{res}}_{\mathrm{rd}}(\Sigma, \mathbb{S})$ and a threshold $\theta \in \mathbb{R}_{\geq 0}$, whether there exists a timed word $w \in \mathbb{T}\Sigma^+$ with $[\![\psi]\!](w) < \theta$.*

If we apply Theorem 7.1 to the timed semiring $\mathbb{1}$BOOL of Example 2.1 (b) and exclude redundant formulas, then we obtain a logical characterization result for unweighted TPDA stated in [14].

## 8 Conclusion

We introduced a weighted logic on timed words which is expressively equivalent to WTPDA. Since the proof of our expressive equivalence result is constructive, decidability results for WTPDA can be transferred into corresponding decidability results for our new logic.

For the proof of the main result we proved a decomposition theorem for WTPDA establishing a connection between WTPDA and visibly pushdown languages. We believe that this result can be helpful for the further study of WTPDA. In addition, our proof technique is robust against adding new components of the model (e.g., global clocks as in Sect. 7) and could be applied in different contexts.

It would be also interesting to investigate the model of WTPDA with time-dependent costs for staying in locations (as in the model of weighted timed automata without stack, cf. [21]) and its logical characterization.

## References


[1] Abdulla, P.A., Atig, M.F., Stenman, J.: Dense-timed pushdown automata. In: LICS 2012, pp. 35–44. IEEE Computer Society (2012).

[2] Abdulla, P.A., Atig, M.F., Stenman, J.: Computing optimal reachability costs in priced dense-timed pushdown automata. In: LATA 2014. LNCS, vol. 8370, pp. 62–75. Springer (2014).

[3] Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science 126(2). 183–235 (1994).

[4] Alur, R., Madhusudan, P.: Visibly pushdown languages. In: STOC 2004, pp. 202–211. ACM (2004).

[5] Berstel, J.: Transductions and Context-Free Languages. Teubner Studienbücher: Informatik. Teubner, Stuttgart (1979).

[6] Bollig, B., Gastin, P.: Weighted versus probabilistic logics. In: DLT 2009. LNCS, vol. 5583, pp. 18-38. Springer (2009).

[7] Bouajjani, A., Echahed, R., Robbana, R.: On the automatic verification of systems with continuous variables and unbounded discrete data structures. In: Hybrid Systems II. LNCS, vol. 999, pp. 64–85. Springer (1995).

[8] Bouyer, P., Fahrenberg, U., Larsen, K.G., Markey, N.: Timed automata with observers under energy constraints. In: HSCC 2010, pp. 61–70. ACM (2010).

[9] Büchi, J.R.: Weak second order arithmetic and finite automata. Zeitschrift für Mathematische Logik und Grundlagen der Informatik 6, 66–92 (1960).

[10] Dang, Z.: Pushdown timed automata: a binary reachability characterization and safety verification. Theoretical Computer Science 302, 93–121 (2003).

[11] Droste, M., Gastin, P.: Weighted automata and weighted logics. Theoret. Comp. Sci. 380(1-2), 69–86 (2007).

[12] Droste, M., Kuich, W., Vogler, H. (eds.): Handbook of Weighted Automata. EATCS Monographs on Theoretical Computer Science. Springer (2009).

[13] Droste, M., Perevoshchikov, V.: A Nivat theorem for weighted timed automata and relative distance logic. In: ICALP 2014. LNCS, pp. 171–182. Springer (2014).

[14] Droste, M., Perevoshchikov, V.: A logical characterization of timed pushdown languages. In: CSR 2015. To appear.

[15] Elgot, C.C.: Decision problems of finite automata design and related arithmetics. Trans. Amer. Math. Soc. 98, 21–51 (1961).

[16] Emmi, M., Majumdar, R.: Decision problems for the verification of real-time software. In: HSCC 2006. LNCS, vol. 3927, pp. 200–211. Springer (2006).

[17] Gurevich, Y.: Monadic theories. In: Model-Theoretic Logics, pp. 479–506. Springer (1985).

[18] Lautemann, C., Schwentick, T., Thérien, D.: Logics for context-free languages. In: CSL 1994. LNCS, vol. 933, pp. 205–216. Springer (1994).

[19] Mathissen, Ch.: Weighted logics for nested words and algebraic formal power series. Logical Methods in Computer Science, 6 (1:5), 1–34 (2010).

[20] Nivat, M.: Transductions des langages de Chomsky. Ann. de l'Inst. Fourier 18, 339–456 (1968).

[21] Quaas, K.: MSO logics for weighted timed automata. Formal Methods in System Design 38(3), 193–222 (2011).

[22] Trakhtenbrot, B.A.: Finite automata and logic of monadic predicates. Doklady Akademii Nauk SSSR, 140, 326–329 (1961). In Russian.

[23] Wilke, T.: Specifying timed state sequences in powerful decidable logics and timed automata. In: Formal Techniques in Real-Time and Fault-Tolerant Systems 1994. LNCS, vol. 863, pp. 694 – 715. Springer (1994).