Multi-weighted Automata Models and Quantitative Logics

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

D I S S E R T A T I O N

zur Erlangung des akademischen Grades

Doctor rerum naturalium
(Dr. rer. nat.)

im Fachgebiet
Informatik

Vorgelegt
von Vitaly Perevoshchikov
geboren am 06. April 1988 in Dschambul, Kasachstan.

Die Annahme der Dissertation wurde empfohlen von:

1. Prof. Dr. Manfred Droste (Leipzig)
2. Prof. Dr. Paul Gastin (Cachan, Frankreich)

Die Verleihung des akademischen Grades erfolgt mit Bestehen
der Verteidigung am 28.04.2015 mit dem Gesamtprädikat
summa cum laude

Acknowledgments

First of all, I would like to thank my supervisor Prof. Dr. Manfred Droste for his continuous support during the last four years, for giving me the possibility to research in his group and to live in Germany. He introduced me to the subject of automata theory and logic. He taught me a lot how to ask questions and find answers, how to write scientific papers and give scientific talks. I also would like to thank him for offering me a teaching position within the first two years of my research.

I am indebted to the *Deutsche Forschungsgemeinschaft* (DFG) for supporting me financially within the scope of the Graduiertenkolleg 1763 "Quantitative logics and automata".

I am grateful to Prof. Dr. Paul Gastin for the valuable discussions and interesting ideas concerning Chapter 3 of this thesis. His suggestion to restrict the use of almost Boolean formulas in the multi-weighted setting helped to improve the presentation of the results of Chapter 3.

I would like to thank Ingmar Meinecke, a former colleague and a good friend of mine, for his constant moral support. He helped me a lot at the beginning of my research.

I am grateful to Karin Quaas for reading the preliminary version of this thesis and for her helpful comments.

Finally, I would like to thank my parents for their love and encouragement and my darling Kießa for the faithful support during the last steps of my PhD studies. Thanks for being you!

Contents

Chapter 1

Introduction

Weighted automata introduced by Schützenberger [81] and timed automata introduced by Alur and Dill [3] are prominent models for quantitative aspects of systems like time, costs, probabilities and energy consumption. Weighted automata find applications in the areas of image processing [29], speech recognition [73] or verification of probabilistic systems [9], and timed automata are of great significance for the analysis of real-time systems [3]. Recently, compound automata frameworks combining the functionality of various automata models, for instance, clocks, multiple weight parameters and stack have received much attention for the quantitative analysis and modelling of systems. For instance, single-weighted timed automata [6, 7, 11, 23, 53, 54, 68] and *multi-weighted timed automata* [21, 22, 55, 69] extend the timed automata of [3] by featuring several weight parameters. This permits to compute objectives like the optimal ratio between rewards and costs [21, 22, 55], or the optimal consumption of several resources where more than one resource must be restricted [69]. Arising from the model of timed automata, the multi-weighted setting has also attracted much interest for classical non-deterministic automata on finite and infinite words [8, 10, 14, 52, 56]. *Timed pushdown automata* provide another example of a hybrid model combining timed automata with pushdown automata for context-free languages [12]; this model was studied in [1, 19, 31, 51] in the context of the verification of real-time recursive systems.

Since the seminal Büchi-Elgot theorem [25, 50] about the expressive equivalence of nondeterministic automata and monadic second-order logic, a significant field of research investigates logical characterizations of language classes appearing from practically relevant automata models, e.g., finite-state automata on trees [83], pictures [59], timed words [84], data words [20] as well as pushdown automata [70] and weighted automata [34]. The main goal of this thesis is to give logical characterizations for multi-weighted automata models on finite, infinite and timed words as well as for timed pushdown automata in the spirit of the classical Büchi-Elgot theorem.

The main problem of the logical characterization for multi-weighted automata is that these models involve several independent weight domains. For instance, in multi-weighted timed automata, transition weights can be tuples of values (corresponding to several price parameters), location weights can be tuples of functions (describing, e.g., the continuous consumption of several resources), whereas the behavior of multi-weighted timed automata takes on a single value. Therefore, the approach of Droste and Gastin [34] to semiring-weighted MSO logic (where all weights are taken from the same domain, namely, a semiring) cannot be easily extended to the multi-weighted setting. We solve this problem by splitting the semantics of formulas into two levels: the *auxiliary* semantics and the *proper* semantics. On the auxiliary level of the semantics, we deal with the weights appearing in the syntactic part of automata, e.g., tuples of values. On the proper level of the semantics, we operate with the weights appearing in the behaviors of automata, e.g., single values.

In addition, we study several algorithmic problems for multi-weighted automata which can be carried over to the developed logical formalisms. Another significant contribution of this thesis is a Nivat-like characterization for multi-weighted automata models as well as timed pushdown automata. Recall that Nivat's theorem [74] is one of the fundamental characterizations of rational transductions and establishes a connection between rational transductions and rational languages; versions of this theorem for semiring-weighted automata and weighted multioperator tree automata were given in [38, 82]. As an application of our Nivat-like decomposition theorems, inspired by the ideas of Kuske [67] for weighted rational expressions, we introduce a new proof technique for our Büchi-Elgot theorems. This technique permits to deduce our results in the quantitative setting from the corresponding results in the underlying qualitative setting.

Below we give a brief description of the contents of this thesis.

In *Chapter 2*, we develop a general model for multi-weighted automata on finite words which incorporates the following examples of single- and multi-weighted automata known from the literature: the reward-cost ratio automata [21, 56], the discounting automata with varying discounting factors [8, 45], multi-weighted automata with constraints on accumulated weights [69, 75] as well as weighted-automata over semirings [13, 36, 49, 64, 65, 80] and valuation monoids [39, 40] motivated by quantitative languages of [27, 28]. The existing concepts of semirings and valuation monoids do not cover the multi-weighted case since the weight constants in a multi-weighted automaton are tuples of weights (e.g., the reward-cost pair) whereas the behavior takes on a single value (e.g., the reward-cost ratio). Then, the weight of a run in a multi-weighted automaton cannot be defined by means of a binary operation (like in a semiring) or by means of a valuation function (like in a valuation monoid). In our framework, we process the transition weights in a general way, i.e., we take into account the history of weights and the nondeterminism before we evaluate the behavior on a given word. This means that we collect the strings of weights occurring along runs in a *multiset*. After that, we use an *aggregation function* which associates to such a multiset a single value. In Chapter 2 we also study several algorithmic problems for ratio automata on finite words. We show that the threshold problem for them is decidable in polynomial time (this result extends the result of [56]). We also show that the behavior of ratio automata on a given word is computable in polynomial time (like in the case of, e.g., the tropical semiring or the semiring of natural numbers).

In *Chapter 3*, we introduce a concept of *multi-weighted MSO logic* following the concept of *semiring-weighted MSO logic* of Droste and Gastin [34]. Here, the weights of constants could be tuples of weights. The semantics of formulas should be single values (not tuples of weights). In contrast to weighted MSO logics over semirings [34] and valuation monoids [40], this makes it impossible to define the semantics inductively on the structure of an MSO formula. Instead, for finite words, we introduce an intermediate semantics which maps each word to a finite multiset containing strings of tuples of weights. The semantics of a formula is then defined by applying the aggregation function to the multiset semantics. We show that our new approach to multi-weighted MSO logic extends the semiring-weighted MSO logic of [34]. We characterize multi-weighted automata by a fragment of the multi-weighted MSO logic. Here, in general, the fragment proposed by [34] for semirings is more expressive than multi-weighted automata. We put additional restrictions on the syntax and show the equivalence of this restricted multi-weighted MSO logic and multi-weighted automata. The proof of this result can be reduced to the case of semirings. However, we cannot apply directly the result of [34], since we must pay attention to the weight constants which appear in multi-weighted automata. Therefore, we revisit the proof of [34] with respect to the suitable constructions. Moreover, we show that if we add suitable properties to our algebraic weight structure, the fragment of [34] proposed for semirings also leads to recognizable quantitative languages (in the sense of multi-weighted automata). Semirings as well as various weight measures considered in literature satisfy this property.

In *Chapter 4*, we show that the double-step approach to the semantics of the multi-weighted MSO logic can be successfully applied to the definition of the semantics of *multi-weighted rational expressions*. First, we show that the semantics of our multi-weighted rational expressions over semirings coincides with the classical semantics of semiring-weighted rational expressions of [81]. As a main result of this chapter, we obtain expressive equivalence of multi-weighted rational expressions and multi-weighted automata; this result extends the classical Kleene-Schützenberger theorem for semiring-weighted automata [61, 81]. For the proof of this result, we show that every multi-weighted automaton (independently of its weight measure) can be characterized by a weighted automaton over the semiring of natural numbers. Then, we can deduce our result from the Kleene-Schützenberger theorem [81] for the semiring natural numbers. Note that, in [39], weighted automata over valuation monoids were characterized by weighted rational expressions. Since multi-weighted automata incorporate weighted automata over valuation monoids, our expressive equivalence result gives an alternative Kleene-Schützenberger characterization for valuation monoid weighted automata. Our result about the characterization of multi-weighted automata by weighted automata over natural numbers may be of independent interest since it could be helpful to transfer to the multi-weighted setting further results about weighted automata over the semiring natural numbers. We go further and study whether we can omit the multiplicities and consider classical finite automata as a basis of multi-weighted automata. We show that, in general, the multiplicities cannot be avoided. However, for structures like *idempotent semirings* and *non-idempotent infinite semifields*, the multiplicities are irrelevant.

In *Chapter 5*, we study multi-weighted automata on infinite words. We are motivated by the following examples of multi-weighted automata models considered in literature: the reward-cost ratio automata [21, 22], the discounting automata with transition-dependent discounting factors [8, 37, 45], and multi-weighted energy automata [52]. In these examples, multi-weighted automata are defined without taking into account an acceptance condition. In contrast, in this thesis we extend them by a *Büchi acceptance condition* since it is quite natural from the automata-theoretic point of view. In order to motivate this extension, we show the computability of the optimal value problem for multi-weighted Büchi automata with discounting. We introduce a general framework for multi-weighted Büchi automata; this framework extends the notion of ω-valuation monoids of [40]. In particular, as in [40], we evaluate ω-sequences of weights using a multi-weighted valuation function and resolve the nondeterminism on weights using an infinitary sum operation. We also give a Nivat-like decomposition theorem for multi-weighted Büchi automata; this result permits to separate the multi-weighted setting from the classical Büchi automata. As a corollary, we obtain a Nivat decomposition theorem for *unambiguous* multi-weighted Büchi automata where, for every input ω-word, there exists at most one accepting run. Unambiguous automata are also of considerable interest for automata theory as they can have better decidability properties. For instance, the equivalence problem for unambiguous max-plus automata is decidable [60] whereas, for nondeterministic max-plus automata, this problem is undecidable [63]. As a first application of our Nivat theorem, we show that as in the classical case, Büchi and Muller acceptance conditions are equivalent in the multi-weighted setting; this extends the result of [44] for totally complete semirings.

In *Chapter 6*, we give a logical characterization for multi-weighted Büchi automata. Recall that, for finite words, we used a double-step approach for the semantics of multi-weighted MSO formulas. The auxiliary semantics was defined as for the semiring-weighted MSO logic of Droste and Gastin [34] over the semiring of finite multisets $\mathbb{N}\langle M^* \rangle$ with the union and the Cauchy product. The proper semantics was defined by evaluating the finite multisets of the auxiliary semantics. However, if we want to extend this multi-weighted logic to infinite words, we face the problem that there is no natural concatenation operation on infinite strings. We introduce an alternative approach for infinite words which fits to the algebraic structure we use for multi-weighted Büchi automata. As a motivation, we consider, for instance, timed and picture automata whose logical characterizations [59, 84] were given by the sentences of the form $\psi =$

$\exists X_1. \ldots \exists X_n.\varphi$ where φ is constructed from logical operators which can be applied *without any restrictions*. Here, we develop a so-called *weight assignment logic* (WAL) on infinite words; this logic allows us to assign multi-weights to positions of an ω-word. We allow the use of the first-order and second-order existential quantifiers only in the prefix of a formula whereas, in the scope of this existential prefix, we can use Boolean formulas as well as weighted conjunction-like operators without any restrictions. Using our weight assignment logic, we can, for instance, express that whenever a position of an input word is labelled by a letter a, then the weight of this position is the reward-cost pair $(2, 1)$. As a weighted extension of the conjunction, we use the *merging* of partial mappings. In order to evaluate an infinite partially defined string of weights, we introduce the *default weight*, assign it to all undefined positions, and apply the ω-valuation function to the obtained totally defined infinite string. Finally, for the existential prefix, we use an infinitary sum operation. First, we show that the set of *unambiguous* sentences of weighted assignment logic, i.e., the set of all sentences without existential quantifiers, is expressively equivalent to unambiguous multi-weighted Büchi automata. A logical characterization of unambiguous semiring-weighted automata was given in [66] where, in general, the use of weighted conjunction and weighted universal first-order quantification is restricted, and the use of weighted universal second-order quantification is not allowed. In contrast, in our unambiguous WAL we can use these logical operators without any restrictions. Thereafter we show that our weight assignment logic is equally expressive as nondeterministic multi-weighted Büchi automata. The proofs of these results are based on our Nivat-like decomposition results for unambiguous and nondeterministic multi-weighted Büchi automata. Note that Droste and Meinecke [40] presented a weighted MSO logic over ω-valuation monoids. Our result gives an alternative logical characterization of recognizable quantitative ω-languages over ω-valuation monoids.

In *Chapter 7*, we consider the model of *multi-weighted timed automata* which are of much interest for the real-time community, since they can model continuous time-dependent consumption of resources (cf. linearly priced timed automata [7, 11, 68], multi-weighted timed automata with knapsack-problem objective [69] and the reward-cost ratio measure [21, 22] as well as single-weighted timed automata with discounting [6, 53, 54]). In [43, 77, 78], semiring-weighted timed automata were studied with respect to classical automata-theoretic questions. However, various models, e.g., multi-weighted timed automata with the ratio and knapsack measures as well as single-weighted timed automata with discounting do not fit into the framework of semirings. For the latter situations, only several algorithmic problems have been handled. But many questions whether the results known from the theories of timed and weighted automata also hold for multi-weighted timed automata remain open. Moreover, there is no unified framework for these automata. The main goal of this chapter is to build a bridge between the theories of multi-weighted timed automata and timed automata. First, we develop a general model of *timed valuation structures* for multi-weighted timed automata. Second, following the ideas of [38] and our ideas presented in Chapter 5, we give a Nivat-like characterization of quantitative timed languages recognizable by multi-weighted timed automata. The main difficulties here are that:

- multi-weighted timed automata have two sorts of weights and their behavior is computed using the time sequence of an input;
- timed automata are not determinizable; moreover, they are more expressive than unambiguous timed automata.

Nevertheless, for idempotent timed valuation structures, we do not need unambiguity. In this case, our Nivat theorem can be formulated as it was done for the case of ω-words. In the non-idempotent case, we give an example showing that this statement does not hold true. But in this case we can establish a connection between recognizable quantitative timed languages and sequentially, deterministically or unambiguously recognizable timed languages. Finally, using our Nivat theorem for multi-weighted timed automata, we study the connection via renamings between determinism and non-determinism in the multi-weighted timed setting. As we showed for multi-weighted Büchi automata, recognizable quantitative ω-languages are exactly the renamings of deterministically (and hence unambiguously) recognizable quantitative ω-languages. We can show that a similar connection holds between multi-weighted timed automata and unambiguous multi-weighted timed automata. However, interestingly, the renamings of the behaviors of deterministic multi-weighted timed automata form a proper subclass of recognizable quantitative timed languages.

In *Chapter 8*, we establish a Büchi-Elgot characterization for multi-weighted timed automata. We introduce a *timed weight assignment logic* which extends the weight assignment logic of Chapter 6 to the timed setting; the qualitative basis of this logic is Wilke's *relative distance logic* [84, 85]. We prove that our timed weight assignment logic is equally expressive as multi-weighted timed automata. To show this, we use the same proof technique as for weight assignment logic on ω-words, i.e., we apply our Nivat theorem for multi-weighted timed automata. The main difficulty here is that unambiguous timed automata are not equivalent to nondeterministic timed automata but that our Nivat theorem appeals to unambiguous timed automata. In order to overcome this difficulty, we present a logical characterization of unambiguous timed automata by means of a fragment of relative distance logic. Note that this unambiguous fragment will be used only in the proof of our main result but in our timed weight assignment logic we do not restrict the use of Boolean formulas. Also note that Quaas [77, 78] presented a weighted version of relative distance logic over semirings. Our result, applied to semirings, gives an alternative logical characterization of recognizable quantitative timed languages over semirings.

In *Chapter 9*, we provide a logical characterization for *dense-timed pushdown automata* proposed recently by [2] as a model for real-time recursive systems. These automata are equipped with a finite set of clocks as well as a stack which keeps track of the age of its elements. For our purpose, we introduce a so-called *timed matching logic*. As in the logic of Lautemann, Schwentick and Thérien

[70], we handle the stack functionality by means of a binary *matching* predicate. As in the logic of Wilke [84], we use *relative distance predicates* to handle the functionality of clocks. Moreover, for 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. The main result of this chapter is the expressive equivalence of timed pushdown automata and timed matching logic. Here, we face the following difficulties of the proof of our main result. The class of timed pushdown languages is most likely not closed under intersection and complement (as the class of context-free languages). Moreover, we cannot directly follow the approaches of [70] and [84], since the proof of [70] appeals to the logical characterization result for trees [83] (but, there is no suitable logical characterization for regular timed tree languages) and the proof of [84] appeals to the classical Büchi-Elgot result [25, 50] (and, this way does not permit to handle matchings). In our case, we appeal to the MSO-like characterization of *visibly pushdown languages* of Alur and Madhusudan [5]. We show our expressive equivalence result as follows.

- We prove a Nivat-like decomposition theorem for timed pushdown automata (cf. [74, 12]) which may be of independent interest; this theorem establishes a connection between timed pushdown languages and untimed visibly pushdown languages of [5] by means of operations like renamings and intersections with simple timed pushdown languages. So we can separate the continuous timed part of the model of timed pushdown automata from its discrete part. The main difficulty here is to encode the infinite time domain, namely $\mathbb{R}_{\geq 0}$, as a finite alphabet. We will show that it suffices to use several partitions of $\mathbb{R}_{\geq 0}$ into intervals to construct the desired extended alphabet. On the one hand, we interpret these intervals as components of the extended alphabet. On the other hand, we use them to contol the timed part of the model.
- In a similar way, we separate the quantitative timed part of timed matching logic from the qualitative part described by MSO logic with matchings over a visibly pushdown alphabet [5] by means of operations like renamings and intersections.
- Now we can deduce our result from the result of [5].

Since our proof is constructive and the reachability for timed pushdown automata is decidable [1], we can also decide the satisfiability for our timed matching logic.

Preliminary versions of the results of this thesis appeared in [41] and [42].

Chapter 2

Multi-weighted automata on finite words

Contents

The model of *multi-weighted* (or *multi-priced*) automata is an extension of the model of weighted automata over semirings [13, 36, 49, 64, 65, 80] and valuation monoids [39, 40] by featuring several weight parameters. In the literature, different situations of the behaviors of multi-weighted automata have been considered (cf. [10, 14, 21, 22, 52, 55, 56, 69]) to model the consumption of several resources. For instance, the model of multi-priced timed automata introduced in [21] permits to describe the optimal ratio between accumulated rewards and accumulated costs of transitions.

In this chapter, we introduce a general model to describe the behaviors of multi-weighted automata on finite words and study several algorithmic properties of multi-weighted automata with the reward-cost ratio measure. Interestingly, we can show that the evaluation problem for them is computable in polynomial time.

2.1 A general framework and examples

The motivation for a new algebraic framework for multi-weighted automata is the following. The existing concepts of semirings and valuation monoids do not cover the multi-weighted case since the weight constants in a multi-weighted automaton are tuples of weights (e.g., the reward-cost pair) whereas the behavior takes on a single value (e.g., the reward-cost ratio). Then, the weight of a run in a multi-weighted automaton cannot be defined by means of a binary product operation (like in a semiring) or by means of a valuation function (like in a valuation monoid).

In our framework, we process the transition weights in a general way, i.e., we take into account all the history of weights and the nondeterminism before we evaluate the behavior on a given word. This means that we collect the strings of weights occurred along runs in a *multiset*. After that, we use an *aggregation function* Φ which associates to such a multiset a single value. Now we turn to formal definitions.

An *alphabet* is a non-empty finite set. Let Σ be a non-empty set (not necessarily finite). A *finite word* over Σ is a finite sequence $w = a_1...a_n$ where $n \geq 0$ and $a_i \in \Sigma$ for all $i \in \{1, ..., n\}$. If $n = 0$, then we say that the word w is *empty* and denote it by ε. Otherwise, we say that w is *non-empty*. Let $|w| = n$, the *length* of w. We denote by Σ^* the set of all finite words over Σ. Let $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$, the set of all non-empty words over Σ. Any set $\mathcal{L} \subseteq \Sigma^+$ is called a *language* over Σ. Note that we eliminate the empty word ε when considering languages of finite words.

Let Σ be an alphabet. A *finite automaton* over Σ is a tuple $\mathcal{A} = (Q, I, T, F)$ where Q is a finite set of *states*, $I \subseteq Q$ is a set of *initial* states, $T \subseteq Q \times \Sigma \times Q$ is a set of *transitions*, and $F \subseteq Q$ is a set of *final* states.

A *run* of $\mathcal{A}$ is a sequence $\rho = \left(q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} ... \xrightarrow{a_n} q_n\right)$ such that $n \geq 1$, $q_0 \in I$, $t_i := (q_{i-1}, a_i, q_i) \in T$ for all $i \in \{1, ..., n\}$, and $q_n \in F$. The finite word $\mathrm{label}(\rho) := a_1...a_n \in \Sigma^+$ is called the *label* of ρ. Sometimes we will denote the run ρ as the word $\rho = t_1...t_n \in T^+$. Let $\mathrm{Run}_{\mathcal{A}}$ denote the set of all runs of $\mathcal{A}$. For any $w \in \Sigma^+$, let $\mathrm{Run}_{\mathcal{A}}(w) = \{\rho \in \mathrm{Run}_{\mathcal{A}} \mid \mathrm{label}(\rho) = w\}$. Let $\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^+ \mid \mathrm{Run}_{\mathcal{A}}(w) \neq \emptyset\}$, the language *recognized* by $\mathcal{A}$. A language $\mathcal{L} \subseteq \Sigma^+$ is called *recognizable* if $\mathcal{L} = \mathcal{L}(\mathcal{A})$ for some finite automaton over Σ.

We let $\mathbb{N} = \{0, 1, 2, ...\}$, the set of nonnegative integers. Let X be a set. A *multiset* over X is a mapping $\mu : X \to \mathbb{N}$. For any multiset μ over X, the *support* of μ is the set $\mathrm{supp}(\mu) = \{x \in X \mid \mu(x) \neq 0\}$. We say that μ is *finite* if $\mathrm{supp}(\mu)$ is a finite set. The set of all finite multisets over X is denoted by $\mathbb{N}\langle X \rangle$. Let Y be a set, $f : X \to Y$ a mapping and $X' \subseteq X$ a finite subset of X. We denote by $f[X'] \in \mathbb{N}\langle Y \rangle$ the multiset such that $f[X'](y) = |\{x \in X' \mid f(x) = y\}|$ for all $y \in Y$.

Now we introduce an algebraic structure for multi-weighed automata.

Definition 2.1. *An* evaluator *is a structure* $\mathbb{E} = (M, K, \Phi)$ *where* M, K *are non-empty sets and* $\Phi : \mathbb{N}\langle M^* \rangle \to K$ *is a mapping, called an* aggregation function.

Note that in Definition 2.1 we do not put any conditions on the aggregation function Φ.

Definition 2.2. *Let* Σ *be an alphabet and* $\mathbb{E} = (M, K, \Phi)$ *an evaluator. A* multi-weighted automaton *over* Σ *and* $\mathbb{E}$ *is a tuple* $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ *where* (Q, I, T, F) *is a finite automaton over* Σ *and* $\mathrm{wt} : T \to M$ *is a transition weight function.*

Note that the framework of evaluators also permits to handle the situations where the elements of M are not necessarily tuples of weights and the elements of K are not necessarily single values. Nevertheless, we will call all models of weighted automata which fit into our framework *multi-weighted*, since the multi-weighted setting is the original motivation of our framework.

Let $\mathcal{A}$ be a multi-weighted automaton over Σ and $\mathbb{E}$. We denote by $\textsc{Const}(\mathcal{A}) = \mathrm{wt}(T) \subseteq M$ the set of all weight constants of $\mathcal{A}$. We define the mapping $\mathrm{wt}^{\#}_{\mathcal{A}} : \mathrm{Run}_{\mathcal{A}} \to M^*$ as follows. For each run $\rho = t_1...t_n \in \mathrm{Run}_{\mathcal{A}}$ with $n \geq 1$ and $t_1, ..., t_n \in T$, we put $\mathrm{wt}^{\#}_{\mathcal{A}}(\rho) = \mathrm{wt}(t_1)... \mathrm{wt}(t_n) \in M^*$. Recall that, for any finite subset $X \subseteq \mathrm{Run}_{\mathcal{A}}$, we have $\mathrm{wt}^{\#}_{\mathcal{A}}[X] \in \mathbb{N}\langle M^*\rangle$. Then, the *behavior* of $\mathcal{A}$ is the mapping $[\![\mathcal{A}]\!] : \Sigma^+ \to K$ defined for all $w \in \Sigma^+$ by $[\![\mathcal{A}]\!](w) = \Phi(\mathrm{wt}^{\#}_{\mathcal{A}}[\mathrm{Run}_{\mathcal{A}}(w)])$. Intuitively, $\mathrm{wt}^{\#}_{\mathcal{A}}[\mathrm{Run}_{\mathcal{A}}(w)]$ is the multiset of all weight sequences of all accepting runs of $\mathcal{A}$ for w, and $[\![\mathcal{A}]\!](w)$ is the Φ-aggregation value of this multiset. Any mapping $\mathbb{L} : \Sigma^+ \to K$ is called a *quantitative language*. We say that $\mathbb{L}$ is *recognizable* over $\mathbb{E}$ if there exists a multi-weighted automaton $\mathcal{A}$ over Σ and $\mathbb{E}$ such that $[\![\mathcal{A}]\!] = \mathbb{L}$.

Since we ignore the empty word, it suffices to define Φ only on finite multisets containing only non-empty strings of the same length. However, the use of $\mathbb{N}\langle M^*\rangle$ will simplify our further considerations.

Now we consider several examples which show how to describe the behavior of single-weighted and multi-weighted automata known from the literature using evaluators.

Let $\mathbb{Q}$ denote the set of all rational numbers and $\mathbb{Q}_{\geq 0}$ the set of all non-negative rational numbers.

Example 2.3. *The model of double-weighted* reward-cost ratio *automata (cf. [14, 21, 22, 56, 57]) can be described by means of the evaluator* $\mathbb{E}^{\textsc{Ratio}} = (M, \mathbb{Q} \cup \{\infty\}, \Phi^{\textsc{Ratio}})$. *Here,* $M = \mathbb{Q} \times \mathbb{Q}_{\geq 0}$ *and* $\Phi^{\textsc{Ratio}}$ *is defined for every* $\mu \in \mathbb{N}\langle M^*\rangle$ *as*

$$\Phi^{\textsc{Ratio}}(\mu) = \min\left\{ \tfrac{r_1+...+r_k}{c_1+...+c_k} \mid (r_1, c_1)...(r_k, c_k) \in \mathrm{supp}(\mu)\right\}$$

where we put $\min \emptyset = \infty$, $\sum \emptyset = 0$ *and* $\frac{r}{0} = \infty$ *for all* $r \in \mathbb{Q}$. *Note that, for the empty word* ε, *the reward-cost ratio* $\frac{r_1+...+r_k}{c_1+...+c_k}$ *is equal to* $\frac{0}{0} = \infty$ *and does not influence the value of* $\Phi^{\textsc{Ratio}}(\mu)$. *In a multi-weighted automaton* $\mathcal{A}$ *over* Σ *and* $\mathbb{E}^{\textsc{Ratio}}$, *transitions have a reward and a cost, and* $[\![\mathcal{A}]\!](w)$ *is the minimal ratio between the total reward and the total cost of a run for* w.

Example 2.4. *Now we consider the model of double-priced automata with the optimal conditional reachability objective [69] (cf. also the multi-constraint routing problem [75]). Here, the first price parameter is called the* primary cost *and the second price parameter is called the* secondary cost. *The goal is to minimize the accumulated primary cost under some upper bound on the accumulated secondary cost. Since this objective is similar to the objective of the well known knapsack problem, we will call these automata* knapsack automata. *We define the evaluator for knapsack automata as follows. Let* $\eta \in \mathbb{Q}_{\geq 0}$ *be a secondary cost* *bound. Then, consider* $\mathbb{E}^{\textsc{Knap}(\eta)} = (M, \mathbb{Q} \cup \{\infty\}, \Phi^{\textsc{Knap}(\eta)})$ *where* $M = \mathbb{Q} \times \mathbb{Q}$ *and* $\Phi^{\textsc{Knap}(\eta)}$ *is defined for all* $\mu \in \mathbb{N}\langle M^* \rangle$ *by*

$$\Phi^{\textsc{Knap}(\eta)}(\mu) = \min \left\{ \textstyle\sum_{i=1}^{k} x_i \;\middle|\; (x_1, y_1)...(x_k, y_k) \in \mathrm{supp}(\mu) \wedge \sum_{i=1}^{k} y_i \leq \eta \right\},$$

with $\min \emptyset = \infty$.

Example 2.5. *Here, we consider multi-weighted automata with discounting. In this model, there are two weight parameters: the cost and the discounting factor (which is not fixed and depends on a transition). This situation was considered in [8] (cf. also the models of weighted automata [15, 27, 28, 37, 46] and weighted timed automata [53, 54] with the fixed discounting factor). A discounting automaton can be considered as a multi-weighted automaton over the evaluator* $\mathbb{E}^{\textsc{Disc}} = (M, \mathbb{Q} \cup \{\infty\}, \Phi^{\textsc{Disc}})$ *with* $M = \mathbb{Q} \times (\mathbb{Q} \cap (0, 1])$ *where* $\Phi^{\textsc{Disc}}$ *is defined by*

$$\Phi^{\textsc{Disc}}(\mu) = \min \left\{ \textstyle\sum_{i=1}^{k} c_i \cdot \prod_{j=1}^{i-1} d_j \;\middle|\; (c_1, d_1)...(c_k, d_k) \in \mathrm{supp}(\mu) \right\}$$

for all $\mu \in \mathbb{N}\langle M^* \rangle$. *Here,* $\sum \emptyset = \infty$ *and* $\prod \emptyset = 1$.

The following quantitative automaton model seems to be new.

Example 2.6. *In weighted automata over semirings [36] and valuation monoids [40], the behavior is defined by summing up the weights of accepting runs. However, it could be interesting to define the behavior by taking the average of the weights of all runs. This average measure could be useful in cases when we take into account not only the weights of runs but also how often these weights may occur. This average setting can be described by means of the evaluator* $\mathbb{E}^{\textsc{Avg}} = (\mathbb{Q}, \mathbb{Q} \cup \{\infty\}, \Phi^{\textsc{Avg}})$ *where the aggregation function* $\Phi^{\textsc{Avg}}$ *is defined for all* $\mu \in \mathbb{N}\langle \mathbb{Q}^* \rangle$ *by*

$$\Phi^{\textsc{Avg}}(\mu) = \frac{1}{|\mu|} \cdot \sum (\mu(u) \cdot (x_1 + ... + x_k) \mid u := x_1...x_k \in \mathrm{supp}(\mu))$$

where we put $\frac{r}{0} = \infty$ *for all* $r \in \mathbb{Q}$. *Here,* $|\mu| = \sum(\mu(u) \mid u \in \mathrm{supp}(\mu))$ *is the* size *of* μ.

The following example is, to the best of our knowledge, new in this context. However, a similar concept might have been considered in different settings. Let $\mathbb{R}$ denote the set of all real numbers and $\mathbb{R}_{\geq 0}$ the set of all non-negative real numbers.

Example 2.7. *Let* $M = \mathbb{R}^n$ *for some* $n \geq 1$ *and* $K = \mathbb{R}_{\geq 0} \cup \{\infty\}$. *Consider the evaluator* $\mathbb{E}^{\mathrm{Disp}(n)} = (M, K, \Phi^{\mathrm{Disp}(n)})$ *where* $\Phi^{\mathrm{Disp}(n)} : \mathbb{N}\langle M^* \rangle \to K$ *is defined as follows. For* $v_1, v_2 \in M$, *let* $(v_1 + v_2) \in M$ *be the componentwise sum of vectors. For a vector* $v = (v_1, ..., v_n) \in M$, *let* $||v|| = \sqrt{v_1^2 + ... + v_n^2}$, *the length of* v. *Then, for every* $\mu \in \mathbb{N}\langle M^* \rangle$, *we put*

$$\Phi^{\mathrm{Disp}(n)}(\mu) = \frac{1}{|\mu|} \cdot \sum \big(\mu(u) \cdot ||v_1 + ... + v_k|| \;\big|\; u := v_1...v_k \in \mathrm{supp}(\mu)\big)$$

where $\frac{r}{0} = \infty$ for all $r \in \mathbb{R}$. Suppose that $\mathcal{A}$ controls the movement of some object in $\mathbb{R}^n$ and each transition carries the coordinates of the displacement vector of this object. Then, the behavior of $\mathcal{A}$ is the value of the average displacement of the object after executing w.

Example 2.8. *Semiring-weighted automata (cf. [36] for surveys) also fit into the framework of evaluators. Given a semiring $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$, a weighted automaton over $\mathbb{S}$ can be considered as a multi-weighted automaton over the evaluator $\mathbb{E}^{\mathbb{S}} = (S, S, \Phi^{\mathbb{S}})$ where the aggregation function $\Phi^{\mathbb{S}}$ is defined as follows. For any multiset $\mu \in \mathbb{N}\langle S^* \rangle$, we put*

$$\Phi^{\mathbb{S}}(\mu) = \sum \left(\mu(u) \cdot \prod_{j=1}^{k} s_j \;\middle|\; u := s_1 ... s_k \in \mathrm{supp}(\mu) \right)$$

where, $\sum \emptyset = \mathbb{0}$, $\prod \emptyset = \mathbb{1}$ and $n \cdot s = s + ... + s$ (n summands) for $n \in \mathbb{N}$, $s \in S$.

Example 2.9. *A* valuation monoid *is a tuple $\mathbb{M} = (M, +, \mathrm{val}, \mathbb{0})$ where $(M, +, \mathbb{0})$ is a commutative monoid and* $\mathrm{val} : M^+ \to M$ *is a* valuation function *with $\mathrm{val}(m_1, ..., m_n) = \mathbb{0}$ whenever $m_i = \mathbb{0}$ for some $i \in \{1, ..., n\}$. Weighted automata over valuation monoids were considered in [40]. We can understand each weighted automaton over $\mathbb{M}$ as a multi-weighted automaton over $\mathbb{E}^{\mathbb{M}} = (M, M, \Phi^{\mathbb{M}})$ where the aggregation function $\Phi^{\mathbb{M}}$ is defined for each finite multiset $\mu \in \mathbb{N}\langle M^* \rangle$ by $\Phi^{\mathbb{M}}(\mu) = \sum(\mu(u) \cdot \mathrm{val}(m_1, ..., m_k) \mid u := m_1 ... m_k \in \mathrm{supp}(\mu))$. Here,* $\mathrm{val}(\varepsilon)$ *is defined arbitrarily.*

2.2 Algorithmic properties of ratio automata

In the next chapter, we will give a logical characterization of multi-weighted automata, i.e., we will develop a logical formalism for multi-weighted properties and an effective translation into multi-weighted automata. As a motivation for our new logic, we consider in the rest of this section several algorithmic problems for multi-weighted ratio automata (cf. Example 2.3) which can be carried over to the logic.

First, we show that the so-called *threshold problem* for reward-cost ratio automata is decidable.

Lemma 2.10. *Let $\mathbb{E}^{\mathrm{RATIO}}$ be the evaluator of Example 2.3, $\bowtie \in \{<, \leq\}$. Then, it is decidable in polynomial time, given an alphabet Σ, a multi-weighted automaton $\mathcal{A}$ over Σ and $\mathbb{E}^{\mathrm{RATIO}}$ and a threshold $\theta \in \mathbb{Q}$, whether there exists a word $w \in \Sigma^+$ such that $[\![\mathcal{A}]\!](w) \bowtie \theta$.*

Proof. It was shown in [56] that this problem is decidable in polynomial time if a given reward-cost ratio automaton has strictly positive costs, i.e., if the transition weights are in $\mathbb{Q} \times \mathbb{Q}_{>0}$ where $\mathbb{Q}_{>0} = \mathbb{Q}_{\geq 0} \setminus \{0\}$. Here we show that this problem is decidable for an arbitrary multi-weighted automaton $\mathcal{A}$ over

$\mathbb{E}^{\text{Ratio}}$. Let $\mathcal{A} = (Q, I, T, F, \text{wt})$ where $\text{wt} : T \to \mathbb{Q} \times \mathbb{Q}_{\geq 0}$. We will denote the behavior $[\![\mathcal{A}]\!]$ of such a multi-weighted automaton $\mathcal{A}$ over $\mathbb{E}^{\text{Ratio}}$ by $[\![\mathcal{A}]\!]^{\text{Ratio}}$. We proceed as follows:

(i) First, we construct a reward-cost ratio automaton $\mathcal{A}'$ such that $[\![\mathcal{A}']\!]^{\text{Ratio}} = [\![\mathcal{A}]\!]^{\text{Ratio}}$ and the accumulated cost of every run of $\mathcal{A}'$ is strictly positive. The idea is to label each state of $\mathcal{A}$ with the Boolean flag whose initial value is 0 and the value is switched to 1 after taking a transition whose cost is strictly positive. Then, $\mathcal{A}'$ will accept only such runs of $\mathcal{A}$ whose flags have been switched to 1.

(ii) Second, as in [56], we transform $\mathcal{A}'$ to the weighted automaton $\mathcal{A}_\theta = (Q', I', T', F', \text{wt}_\theta)$ over Σ and the tropical semiring $\text{Trop} = (\mathbb{Q} \cup \{\infty\}, \min, +, \infty, 0)$ as follows. For every $t \in T$ with $\text{wt}'(t) = (r, c)$, we put $\text{wt}_\theta(t) = r - \theta \cdot c$. Let $[\![\mathcal{A}_\theta]\!]^{\text{Trop}}$ denote the behavior of $\mathcal{A}_\theta$. Since the costs of runs of $\mathcal{A}'$ are strongly positive, we have: $[\![\mathcal{A}']\!]^{\text{Ratio}}(w) \bowtie \theta$ for some word w iff $[\![\mathcal{A}_\theta]\!]^{\text{Trop}}(w) \bowtie 0$ for some word w. As it was shown in [56], Theorem 3, $\bowtie$-threshold problems for weighted automata over the tropical semiring are decidable in polynomial time. Then, the claim follows.

□

Now we turn to the problem of *evaluation* of the behavior of ratio automata on an input word. Note that, for instance, for the tropical semiring, the behavior of a weighted automaton can be evaluated efficiently (in polynomial time) by matrix multiplications. The distributivity property of the tropical semiring is crucial for this method. In the case of ratio automata, we do not have distributivity and the method of matrix multiplications is not applicable. Note that, for a given word, there can be exponentially many runs. Hence, the naïve algorithm which computes the weights of all runs is an exponential time algorithm. Interestingly, we can still evaluate the behavior of ratio automata in polynomial time. In contrast, using a similar reduction as in [75], Theorem 1, it can be shown that for the evaluator $\mathbb{V}^{\text{Knap}(\eta)}$ of Example 2.4 the following problem is NP-complete: given an alphabet Σ, a secondary cost bound $\eta \in \mathbb{Q}_{\geq 0}$, a multi-weighted automaton $\mathcal{A}$ over Σ and $\mathbb{V}^{\text{Knap}(\eta)}$, a word $w \in \Sigma^+$ and a threshold $\theta \in \mathbb{Q}$, decide whether $[\![\mathcal{A}]\!](w) \leq \theta$.

Lemma 2.11. *Given an alphabet Σ, a ratio automaton $\mathcal{A}$ over Σ and a word $w \in \Sigma^+$, the value $[\![\mathcal{A}]\!]^{\text{Ratio}}(w)$ can be computed in polynomial time.*

Proof. Our algorithm will be based on the idea of the minimization of rational functions presented in [72].

Let Σ be an alphabet, $\mathcal{A} = (Q, I, T, F, \text{wt})$ a ratio automaton over Σ and $w \in \Sigma^+$. Assume that $\text{wt}(t) = (r_t, c_t)$ for all $t \in T$. For a run $\rho = t_1 ... t_n \in \text{Run}_\mathcal{A}$ with $t_1, ..., t_n \in T$, let $\text{Reward}(\rho) = r_{t_1} + ... + r_{t_n}$, the *reward* of ρ, and $\text{Cost}(\rho) = c_{t_1} + ... + c_{t_n}$, the *cost* of ρ. By part (i) of the previous lemma, we may assume without loss of generality that $\text{Cost}(\rho) > 0$ for all $\rho \in \text{Run}_\mathcal{A}(w)$. First, we can check in polynomial time whether $\mathrm{Run}_{\mathcal{A}}(w) = \emptyset$. In this case, $[\![\mathcal{A}]\!]^{\textsc{Ratio}}(w) = \infty$. Now assume that $\mathrm{Run}_{\mathcal{A}}(w) \neq \emptyset$. Then, for $\theta = [\![\mathcal{A}]\!]^{\textsc{Ratio}}(w) = \min\limits_{\rho \in \mathrm{Run}_{\mathcal{A}}(w)} \frac{\textsc{Reward}(\rho)}{\textsc{Cost}(\rho)}$, we have $\min\limits_{\rho \in \mathrm{Run}_{\mathcal{A}}(w)} \frac{\textsc{Reward}(\rho) - \theta \cdot \textsc{Cost}(\rho)}{\textsc{Cost}(\rho)} = 0$. Let $\varphi : \mathbb{Q} \to \mathbb{Q}$ be defined for all $x \in \mathbb{Q}$ by

$$\varphi(x) = \min_{\rho \in \mathrm{Run}_{\mathcal{A}}(w)} (\textsc{Reward}(\rho) - x \cdot \textsc{Cost}(\rho))$$

Note that the equation $\varphi(x) = 0$ has the unique solution θ. Then, our task of computing $[\![\mathcal{A}]\!]^{\textsc{Ratio}}(w)$ is equivalent to the task of finding this solution.

First, we mention an interesting property of the mapping φ. Let $x \in \mathbb{Q}$ be such that $\varphi(x) > 0$. Then, for any run $\rho \in \mathrm{Run}_{\mathcal{A}}(w)$, we have: $\frac{\textsc{Reward}(\rho)}{\textsc{Cost}(\rho)} > x$ and hence $\theta > x$. Now assume that $\varphi(x) < 0$. Then, there exists a run $\pi \in \mathrm{Run}_{\mathcal{A}}(w)$ such that $\textsc{Reward}(\pi) - x \cdot \textsc{Cost}(\pi) < 0$ which implies $\theta \leq \frac{\textsc{Reward}(\pi)}{\textsc{Cost}(\pi)} < x$. Hence, the following holds true:

$$\forall x \in \mathbb{Q} : ((\varphi(x) > 0 \to \theta > x) \wedge (\varphi(x) < 0 \to \theta < x)). \tag{2.1}$$

Let $\textsc{Trop} = (\mathbb{Q} \cup \{\infty\}, \min, +, \infty, 0)$, the *tropical semiring* of rational numbers. Let $x \in \mathbb{Q}$ be a parameter. We consider the semiring-weighted automaton $\mathcal{A}'(x) = (Q, I, T, F, \mathrm{wt}'(x))$ over Σ and $\textsc{Trop}$ where $\mathrm{wt}'(x) : T \to \mathbb{Q}$ is defined as follows. For any $t \in T$ with $\mathrm{wt}(t) = (r, c)$, we put $\mathrm{wt}'(x)(t) = r - x \cdot c$. Clearly, $[\![\mathcal{A}'(x)]\!]^{\textsc{Trop}}(w) = \varphi(x)$. Then, our task is to find $x \in \mathbb{Q}$ with $[\![\mathcal{A}'(x)]\!]^{\textsc{Trop}}(w) = 0$. We transform $\mathcal{A}'(x)$ to the matrix representation $(\gamma, \mu(x), \nu)$ where $\gamma \in \{0, \infty\}^{1 \times Q}$, $\mu(x) : \Sigma \to (\mathbb{Q} \cup \{\infty\})^{Q \times Q}$ and $\nu \in \{0, \infty\}^{Q \times 1}$. Let $w = a_1 \ldots a_n$. Then,

$$[\![\mathcal{A}'(x)]\!]^{\textsc{Trop}}(w) = \gamma \cdot \mu(x)(a_1) \cdot \ldots \cdot \mu(x)(a_n) \cdot \nu$$

where the product $\cdot$ of matrices is defined with respect to the semiring $\textsc{Trop}$. Now we will compute $[\![\mathcal{A}'(x)]\!]^{\textsc{Trop}}(w)$ as the product of matrices whose entries can depend on x as follows.

Let $M = \{\infty\} \cup \{a - b \cdot x \mid a, b \in \mathbb{Q}\}$. Let the sequence of row vectors $(u_i)_{1 \leq i \leq n}$ with $u_i \in M^{1 \times Q}$ be defined inductively as follows: $u_0 = \gamma$ and $u_{i+1} = u_i \cdot \mu(x)(a_{i+1})$. Then, $[\![\mathcal{A}'(x)]\!]^{\textsc{Trop}}(w) = u_n \cdot \nu$. We we will perform the operations with the parameter x in the semiring $\textsc{Trop}$ as follows. We assume that x belongs to some interval $I \subseteq (-\infty, \infty)$ which contains θ. At the beginning of computation, we put $I = (-\infty, \infty)$. However, during the computation we can restrict this interval.

- Let $a, b, c, d \in \mathbb{Q}$. Then: $(a - b \cdot x) + \infty = \infty + (a - b \cdot x) = \infty$ and $(a - b \cdot x) + (c - d \cdot x) = (a + c) - (b + d) \cdot x$.
- Let $a, b, c, d \in \mathbb{Q}$. Then: $\min\{a - b \cdot x, \infty\} = \min\{\infty, a - b \cdot x\} = \infty$ and $\min\{a - b \cdot x, a - b \cdot x\} = a - b \cdot x$. The most interesting case is to represent $\min\{a - b \cdot x, c - d \cdot x\}$ with $(a, b) \neq (c, d)$ in a parametric form. Here, we will use the property (2.1). Assume that the equation $a - b \cdot x = c - d \cdot x$ has a unique solution in the interval I (i.e., the two linear functions cross

in I). Let x_0 be this solution. We compute $\varphi(x_0) = [\![\mathcal{A}'(x_0)]\!]^{\textsc{Trop}}(w)$ (this can be done in polynomial time by matrix multiplications). The following situations are possible:

- $\varphi(x_0) = 0$. Then, $\theta = x_0$ and we terminate the computation.
- $\varphi(x_0) > 0$. Then, by (2.1), $\theta > x_0$ and we modify I by letting $I := I \cap (x_0, \infty)$. If $d < b$, then we put $\min\{a - b \cdot x, c - d \cdot x\} = a - b \cdot x$. Otherwise, we put $\min\{a - b \cdot x, c - d \cdot x\} = c - d \cdot x$
- $\varphi(x_0) < 0$. Then, by (2.1), $\theta < x_0$ and we modify I by letting $I := I \cap (-\infty, x_0)$. If $d < b$, then we put $\min\{a - b \cdot x, c - d \cdot x\} = c - d \cdot x$. Otherwise, we put $\min\{a - b \cdot x, c - d \cdot x\} = a - b \cdot x$.

If the equation $a - b \cdot x = c - d \cdot x$ does not have a solution in I, then we take an arbitrary point $x_0 \in I$ and compare the values $a - b \cdot x_0$ and $c - d \cdot x_0$. If $a - b \cdot x_0 < c - d \cdot x_0$, then $a - b \cdot x < c - d \cdot x$ for all $x \in I$, and we put $\min\{a - b \cdot x, c - d \cdot x\} = a - b \cdot x$. Otherwise, we put $\min\{a - b \cdot x, c - d \cdot x\} = c - d \cdot x$.

Since $\mathrm{Run}_{\mathcal{A}}(w) \neq \emptyset$, the result will be a linear function of the form $a - b \cdot x$ where $a \in \mathbb{Q}$ and $b \in \mathbb{Q}_{>0}$. Then, $\theta = \frac{a}{b}$.

Clearly, this algorithm has polynomial time complexity. $\square$

Chapter 3

Multi-weighted MSO logic on finite words

Contents

In [34], Droste and Gastin gave a logical characterization of semiring-weighted automata by means of *weighted MSO logic*; this result extends the classical Büchi-Elgot theorem [25, 50] to the quantitative setting. The goal of the present chapter is to expand this result of Droste and Gastin to the multi-weighted setting.

3.1 Multi-weighted MSO logic

In this section, we wish to develop a multi-weighted MSO logic where the weight constants are elements of a set M. Again, if weight constants are *pairs* of a reward and a cost, we want the semantics of formulas to be able to reflect the maximal reward-cost ratio setting, so the weights of formulas should be *single weights*. Note that one cannot define the semantics function inductively on the structure of a formula as in [34]. Therefore we proceed as follows. Given a formula, we associate to each word a multiset in $\mathbb{N}\langle M^* \rangle$. For disjunction and existential quantification we use the multiset union. For conjunction and universal quantification, we extend the concatenation of strings in M^* to the Cauchy product of multisets in $\mathbb{N}\langle M^* \rangle$. Then, we use an aggregation function $\Phi : \mathbb{N}\langle M^* \rangle \to K$ which associates to each multiset of elements a single value (e.g. the maximal reward-cost ratio of pairs contained in a multiset).

$(w,\sigma) \models P_a(x)$	iff	$a_{\sigma(x)} = a$
$(w,\sigma) \models x \leq y$	iff	$\sigma(x) \leq \sigma(y)$
$(w,\sigma) \models X(x)$	iff	$\sigma(x) \in \sigma(X)$
$(w,\sigma) \models \beta_1 \vee \beta_2$	iff	$(w,\sigma) \models \beta_1$ or $(w,\sigma) \models \beta_2$
$(w,\sigma) \models \neg\beta$	iff	$(w,\sigma) \models \beta$ does not hold
$(w,\sigma) \models \exists x.\beta$	iff	$(w,\sigma[x/i]) \models \beta$ for some $i \in \mathrm{dom}(w)$
$(w,\sigma) \models \exists X.\beta$	iff	$(w,\sigma[X/I]) \models \beta$ for some $I \subseteq \mathrm{dom}(w)$

Table 3.1: The satisfaction relation for Boolean formulas

For the rest of this section, we fix an alphabet Σ and an evaluator $\mathbb{E} = (M, K, \Phi)$ where $\Phi : \mathbb{N}\langle M^*\rangle \to K$ is an aggregation function. We fix countable and pairwise disjoint sets V_1 and V_2 of *first-order* resp. *second-order* variables. The first-order variables are denoted by lower-case letters, e.g., $x, y, z, ...$ whereas the second-order variables are denoted by upper-case letters, for instance, $X, Y, Z, ...$. Let $V = V_1 \cup V_2$.

The syntax of formulas of $\mathbf{mwMSO}(\Sigma, \mathbb{E})$, the *multi-weighted MSO logic* over Σ and $\mathbb{E}$, is given as in [16] by the grammar

$$\begin{aligned} \beta &::= P_a(x) \mid x \leq y \mid X(x) \mid \beta \vee \beta \mid \neg\beta \mid \exists x.\beta \mid \exists X.\varphi \\ \varphi &::= \beta \mid m \mid \varphi \oplus \varphi \mid \varphi \otimes \varphi \mid \bigoplus x.\varphi \mid \bigoplus X.\varphi \mid \bigotimes x.\varphi \end{aligned} \tag{3.1}$$

where $a \in \Sigma$, $m \in M$, $x, y \in V_1$ and $X \in V_2$. The formulas β are called *Boolean* formulas and the formulas φ are called *multi-weighted* formulas. Let $\mathbf{MSO}(\Sigma)$ denote the set of all Boolean formulas. For any formula $\varphi \in \mathbf{mwMSO}(\Sigma, \mathbb{E})$, let $\textsc{Const}(\varphi) \subseteq M$ be the set of all weight constants $m \in M$ occurring in φ.

For a formula $\varphi \in \mathbf{mwMSO}(\Sigma)$, we let $\mathrm{Free}(\varphi)$ be the set of all *free variables* of φ, i.e., the set of those variables in φ not bound by a quantifier. A formula $\varphi \in \mathbf{mwMSO}(\Sigma)$ is called a *sentence* if $\mathrm{Free}(\varphi) = \emptyset$. A word $w = a_1...a_n \in \Sigma^+$ is usually represented by the relational structure $(\mathrm{dom}(w), \leq, (R_a)_{a\in\Sigma})$ where $\mathrm{dom}(w) = \{1, ..., n\}$ is the *domain* of w and, for all letters $a \in \Sigma$, $R_a = \{i \in \mathrm{dom}(w) \mid a_i = a\}$. A *$w$-assignment* σ is a function mapping first-order variables in V_1 to elements of $\mathrm{dom}(w)$ and second-order variables in V_2 to subsets of $\mathrm{dom}(w)$. If $x \in V_1$ and $i \in \mathrm{dom}(w)$, then the *update* $\sigma[x/i]$ is the w-assignment with $\sigma[x/i](x) = i$ and $\sigma[x/i](y) = \sigma(y)$ for all $y \in V \setminus \{x\}$. Similarly, for $X \in V_2$ and $I \subseteq \mathrm{dom}(w)$, we define the update $\sigma[X/I]$ to be the w-assignment with $\sigma[X/I](X) = I$ and $\sigma[X/I](y) = \sigma(y)$ for all $y \in V \setminus \{X\}$. We denote by Σ_V^+ the set of all pairs (w, σ) where $w \in \Sigma^+$ and σ is a w-assignment. Let $(w, \sigma) \in \Sigma_V^+$ with $w = a_1...a_n \in \Sigma^+$ and $\beta \in \mathbf{MSO}(\Sigma)$ be a Boolean formula. The definition that (w, σ) satisfies β, denoted by $(w, \sigma) \models \beta$, is given inductively on the structure of β as shown in Table 3.1. Here, $a \in \Sigma$, $x, y \in V_1$ and $X \in V_2$.

Using Boolean formulas in $\mathbf{MSO}(\Sigma)$, we define formulas $\beta_1 \wedge \beta_2$, $\forall x.\beta$, $\forall X.\beta$, $\beta_1 \to \beta_2$ and $\beta_1 \leftrightarrow \beta_2$ as usual. Let $\textsc{True} \in \mathbf{MSO}(\Sigma)$ be an abbreviation for the sentence $\forall x.(x \leq x)$.

$$
\begin{aligned}
\langle\!\langle m\rangle\!\rangle(w,\sigma) &= [m] \\
\langle\!\langle \beta\rangle\!\rangle(w,\sigma) &= \begin{cases} [\varepsilon], & \text{if } (w,\sigma)\models\beta, \\ \emptyset, & \text{otherwise} \end{cases} \\
\langle\!\langle \varphi_1\oplus\varphi_2\rangle\!\rangle(w,\sigma) &= \langle\!\langle \varphi_1\rangle\!\rangle(w,\sigma) + \langle\!\langle \varphi_2\rangle\!\rangle(w,\sigma) \\
\langle\!\langle \varphi_1\otimes\varphi_2\rangle\!\rangle(w,\sigma) &= \langle\!\langle \varphi_1\rangle\!\rangle(w,\sigma) \cdot \langle\!\langle \varphi_2\rangle\!\rangle(w,\sigma) \\
\langle\!\langle \textstyle\bigoplus x.\varphi\rangle\!\rangle(w,\sigma) &= \textstyle\sum(\langle\!\langle \varphi\rangle\!\rangle(w,\sigma[x/i]) \mid i\in\mathrm{dom}(w)) \\
\langle\!\langle \textstyle\bigoplus X.\varphi\rangle\!\rangle(w,\sigma) &= \textstyle\sum(\langle\!\langle \varphi\rangle\!\rangle(w,\sigma[X/I]) \mid I\subseteq\mathrm{dom}(w)) \\
\langle\!\langle \textstyle\bigotimes x.\varphi\rangle\!\rangle(w,\sigma) &= \textstyle\prod(\langle\!\langle \varphi\rangle\!\rangle(w,\sigma[x/i]) \mid i\in\mathrm{dom}(w))
\end{aligned}
$$

Table 3.2: The auxiliary semantics of multi-weighted formulas

Let $\mu_1,\mu_2\in\mathbb{N}\langle M^*\rangle$ be multisets. The *union* (or *sum*) $(\mu_1+\mu_2)\in\mathbb{N}\langle M^*\rangle$ is defined by $(\mu_1+\mu_2)(u)=\mu_1(u)+\mu_2(u)$ for all $u\in M^*$. Clearly, the union $+$ is a commutative operation. The *Cauchy product* $(\mu_1\cdot\mu_2)\in\mathbb{N}\langle M^*\rangle$ is defined for all $u\in M^*$ as

$$(\mu_1\cdot\mu_2)(u)=\sum(\mu_1(u_1)\cdot\mu_2(u_2)\mid u_1\in\mathrm{supp}(\mu_1),u_2\in\mathrm{supp}(\mu_2),u=u_1u_2).$$

We use the Cauchy product for the semantics of multi-weighted formulas in order to reflect the concatenation of the sequences of weights. The *empty multiset* $\mu\in\mathbb{N}\langle M^*\rangle$ is defined as $\mu(u)=0$ for all $u\in M^*$. We will abuse the notation and denote the empty multiset by $\emptyset$. For $u\in M^*$, let $[u]\in\mathbb{N}\langle M^*\rangle$ denote the multiset such that $\mathrm{supp}([u])=\{u\}$ and $[u](u)=1$. The following proposition is a folklore result.

Proposition 3.1. $(\mathbb{N}\langle M^*\rangle,+,\cdot,\emptyset,[\varepsilon])$ *is a semiring.*

We will denote the semiring from Proposition 3.1 also by $\mathbb{N}\langle M^*\rangle$. Now let $\varphi\in\mathbf{mwMSO}(\Sigma,\mathbb{E})$ be a multi-weighted formula. We define the semantics of φ in two steps as follows:

- First, the *auxiliary semantics* $\langle\!\langle\varphi\rangle\!\rangle:\Sigma_V^+\to\mathbb{N}\langle M^*\rangle$ is defined for all $(w,\sigma)\in\Sigma_V^+$ inductively on the structure of φ as shown in Table 3.2. Here, $m\in M$, $\beta\in\mathbf{MSO}(\Sigma)$, $x\in V_1$ and $X\in V_2$. Note that $\langle\!\langle\varphi\rangle\!\rangle$ does not depend on K and Φ.
- Second, the *proper semantics* (or simply *semantics*) $[\![\varphi]\!]:\Sigma_V^+\to K$ is defined as $[\![\varphi]\!]=\Phi\circ\langle\!\langle\varphi\rangle\!\rangle$.

Sometimes, in order to emphasize that the semantics of a multi-weighted formula $\varphi\in\mathbf{mwMSO}(\Sigma,\mathbb{E})$ is defined with respect to $\mathbb{E}$, we will write $[\![\varphi]\!]^{\mathbb{E}}$ for $[\![\varphi]\!]$. Now let $\varphi\in\mathbf{mwMSO}(\Sigma,\mathbb{E})$ be a sentence. Then, for any $(w,\sigma)\in\Sigma_V^+$, the value $[\![\varphi]\!](w,\sigma)$ depends only on w. Then, ignoring the values of variables, we can consider $\langle\!\langle\varphi\rangle\!\rangle$ as the mapping $\langle\!\langle\varphi\rangle\!\rangle : \Sigma^+ \to \mathbb{N}\langle M^*\rangle$ and $[\![\varphi]\!]$ as the quantitative language $[\![\varphi]\!] : \Sigma^+ \to K$. Let $\Delta \subseteq \mathbf{mwMSO}(\Sigma, \mathbb{E})$ and $\mathbb{L} : \Sigma^+ \to K$ a quantitative language. We say that $\mathbb{L}$ is Δ-*definable* if there exists a sentence $\varphi \in \mathbf{mwMSO}(\Sigma, \mathbb{E})$ such that $[\![\varphi]\!] = \mathbb{L}$.

Let $\beta \in \mathbf{MSO}(\Sigma)$ and $\varphi_1, \varphi_2 \in \mathbf{mwMSO}(\Sigma, \mathbb{E})$. As in [18], we can define the weighted If-Then-Else formula $\beta\,?\,(\varphi_1 : \varphi_2)$ as an abbreviation for the formula $(\beta \otimes \varphi_1) \oplus ((\neg\beta) \otimes \varphi_2)$. Here, if the Boolean formula β holds, then we take the value of φ_1. Otherwise, we take the value of φ_2.

Example 3.2. *Let A be an object on the plane whose displacement is managed by two types of commands: $\leftrightarrow$ and $\updownarrow$. After receiving the command $\leftrightarrow$ the object moves one step to the left or to the right; after receiving $\updownarrow$ one step up or down. Consider the evaluator $\mathbb{E}^{\mathrm{Disp}(n)}$ of Example 2.7 for $n = 2$. Let $v_\leftarrow = (-1, 0)$, $v_\rightarrow = (1, 0)$, $v_\downarrow = (0, -1)$ and $v_\uparrow = (0, 1)$. Consider the following multi-weighted MSO sentence over the alphabet $\Sigma = \{\leftrightarrow, \updownarrow\}$ and the evaluator $\mathbb{E}^{\mathrm{Disp}(2)}$:*

$$\varphi = \bigotimes x.(P_\leftrightarrow(x)\,?\,(v_\leftarrow \oplus v_\rightarrow) : (v_\downarrow \oplus v_\uparrow)).$$

Then, for every sequence of commands $w \in \Sigma^+$, $[\![\varphi]\!](w)$ is the average displacement of the object after execution of all commands from w. For instance, let $w = \leftrightarrow\leftrightarrow$. Then, $\langle\!\langle\varphi\rangle\!\rangle(w) = [v_\leftarrow v_\leftarrow] + [v_\leftarrow v_\rightarrow] + [v_\rightarrow v_\leftarrow] + [v_\rightarrow v_\rightarrow]$ and $[\![\varphi]\!](w) = 1$. For $w = \leftrightarrow\updownarrow$, we have $[\![\varphi]\!](w) = \sqrt{2}$.

Next, we discuss how our new multi-weighted MSO logic is related to the semiring-weighted logic of Droste and Gastin [34]. Let $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ be a semiring. The syntax of *weighted MSO logic* $\mathbf{wMSO}(\Sigma, \mathbb{S})$ is given by the grammar (3.1) where we replace $m \in M$ by $s \in S$. As opposed to the multi-weighted case, the semantics $[\![\varphi]\!]^{\mathbb{S}} : \Sigma_V^+ \to S$ of a weighted MSO formula $\varphi \in \mathbf{wMSO}(\Sigma, \mathbb{S})$ is defined in one step using the weights $\mathbb{0}, \mathbb{1}$ for the Boolean values, the sum $+$ for $\oplus$, and the product $\cdot$ for $\otimes$. More precisely, the semantics can be defined as shown in Table 3.2 where we replace $\langle\!\langle ... \rangle\!\rangle$ by $[\![...]\!]^{\mathbb{S}}$, $m \in M$ by $s \in S$, $[m]$ by s, $\emptyset$ by $\mathbb{0}$, and $[\varepsilon]$ by $\mathbb{1}$.

As we showed in Example 2.8, a semiring $\mathbb{S}$ can be considered as the evaluator $\mathbb{E}^{\mathbb{S}} = (S, S, \Phi^{\mathbb{S}})$. The following lemma shows that our multi-weighted MSO logic extends the semiring-weighted logic of [34].

Lemma 3.3. *Let Σ be an alphabet, $\mathbb{S}$ a semiring and $\varphi \in \mathbf{wMSO}(\Sigma, \mathbb{S})$. Then, $\varphi \in \mathbf{mwMSO}(\Sigma, \mathbb{E}^{\mathbb{S}})$ and $[\![\varphi]\!]^{\mathbb{E}^{\mathbb{S}}} = [\![\varphi]\!]^{\mathbb{S}}$.*

The proof of this lemma follows from the following technical lemma.

Lemma 3.4. *Let $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ be a semiring. Then the mapping $\Phi^{\mathbb{S}} : (\mathbb{N}\langle S^*\rangle, +, \cdot, \emptyset, [\varepsilon]) \to (S, +, \cdot, \mathbb{0}, \mathbb{1})$ is a semiring morphism with $\Phi^{\mathbb{S}}([s]) = s$ for all $s \in S$.*

Proof. Clearly, $\Phi^{\mathbb{S}}(\emptyset) = \mathbb{0}$, $\Phi^{\mathbb{S}}([\varepsilon]) = \mathbb{1}$ and $\Phi^{\mathbb{S}}([s]) = s$ for all $s \in S$. Let $\mu_1, \mu_2 \in \mathbb{N}\langle S^*\rangle$. It can be easily shown that $\Phi^{\mathbb{S}}(\mu_1 + \mu_2) = \Phi^{\mathbb{S}}(\mu_1) + \Phi^{\mathbb{S}}(\mu_2)$.

We show explicitly that $\Phi^{\mathbb{S}}(\mu_1 \cdot \mu_2) = \Phi^{\mathbb{S}}(\mu_1) \cdot \Phi^{\mathbb{S}}(\mu_2)$. Let $g : (S^*, \cdot, \varepsilon) \to (S, \cdot, \mathbb{1})$ be the monoid morphism with $g(s) = s$ for all $s \in S$ and $\mu_1, \mu_2 \in \mathbb{N}\langle S^* \rangle$. Then:

$$\begin{aligned}
\Phi^{\mathbb{S}}(\mu_1) \cdot \Phi^{\mathbb{S}}(\mu_2) &= \Big(\sum_{u \in S^*} \mu_1(u) \cdot g(u) \Big) \cdot \Big(\sum_{v \in S^*} \mu_2(v) \cdot g(v) \Big) \\
&= \sum_{u,v \in S^*} (\mu_1(u) \cdot \mu_2(v)) \cdot (g(u)g(v)) = \sum_{w \in S^*} (\mu_1 \cdot \mu_2)(w) \cdot g(w) \\
&= \Phi^{\mathbb{S}}(\mu_1 \cdot \mu_2).
\end{aligned}$$

Since μ_1 and μ_2 are finite multisets, all infinite sums in the equations above have only finitely many non-zero summands. □

The following example illustrates a situation where the use of multi-weighed MSO logic is more convenient than the use of semiring-weighted MSO logic.

Example 3.5. *Let Σ be an alphabet and $a \in \Sigma$. Consider the quantitative language $\mathbb{L} : \Sigma^+ \to \mathbb{Q} \cup \{\infty\}$ defined for every $w \in \Sigma^+$ as $\mathbb{L}(w) = 2 \cdot |w|_a$ if $|w| \leq 1000$ and $\mathbb{L}(w) = \infty$ otherwise (here, $|w|_a$ is the number of a's in w). We can define $\mathbb{L}$ by means of the* $\mathbf{wMSO}(\Sigma, \textsc{Trop})$*-sentence $\beta \otimes \bigotimes x.(P_a(x) \,?\, (2:0))$ where $\textsc{Trop} = (\mathbb{Q} \cup \{\infty\}, \min, +, \infty, 0)$ is the tropical semiring of rational numbers and $\beta \in \mathbf{MSO}(\Sigma)$ describes the property $|w| \leq 1000$: $\beta = \exists x_1. \ldots \exists x_{1000}. \forall y. \bigvee_{i=1}^{1000} (x_i = y)$. Unfortunately, such a formula β is very long and the use of semiring-weighted MSO logic is not convenient. Here, it is easier to use our multi-weighted MSO logic. Let $\eta = 1000$. Consider the evaluator $\mathbb{E}^{\textsc{Knap}(\eta)} = (\mathbb{Q} \times \mathbb{Q}, \mathbb{Q} \cup \{\infty\}, \Phi^{\textsc{Knap}(\eta)})$ as defined in Example 2.4. Then, $\mathbb{L} = [\![\varphi]\!]$ where φ is the sentence $\varphi \in \mathbf{mwMSO}(\Sigma, \mathbb{E}^{\textsc{Knap}(\eta)})$ defined as $\varphi = \bigotimes x.(P_a(x) \,?\, (2,1) : (0,1))$.*

Next, we recall the Büchi-like result of Droste and Gastin [34] for semiring-weighted MSO logic. Since weighted MSO logic is more powerful than weighted automata (cf.[34]), we consider a restricted version of weighted MSO logic. Let $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ be a semiring. The set $\mathbf{aBOOL}(\Sigma, \mathbb{S})$ of *almost Boolean* formulas is generated by the grammar

$$\gamma \ ::= \ \beta \mid s \mid \gamma \oplus \gamma \mid \gamma \otimes \gamma$$

where $\beta \in \mathbf{MSO}(\Sigma)$ and $s \in S$. For a subset $X \subseteq S$, let $\mathrm{cl}(X)$ denote the minimal subset of S which contains $X \cup \{\mathbb{0}, \mathbb{1}\}$ and is closed under $+$ and $\cdot$. For subsets $X, Y \subseteq S$, we say that X and Y *commute elementwise*, if $x \cdot y = y \cdot x$ for all $x \in X$ and $y \in Y$. The set $\mathbf{wMSO}^{\mathrm{res}}(\Sigma, \mathbb{S}) \subseteq \mathbf{wMSO}(\Sigma, \mathbb{S})$ of *syntactically restricted* formulas is defined by the rules

$$\varphi \ ::= \ \gamma \mid \varphi \oplus \varphi \mid \varphi \otimes \varphi^{\,(!)} \mid \bigoplus x.\varphi \mid \bigoplus X.\varphi \mid \bigotimes x.\gamma$$

where $\gamma \in \mathbf{aBOOL}(\Sigma, \mathbb{S})$, $x \in V_1$ and $X \in V_2$; moreover, there is an additional restriction at the place (!): a formula $\varphi_1 \otimes \varphi_2$ belongs to $\mathbf{wMSO}^{\mathrm{res}}(\Sigma, \mathbb{S})$ iff the sets $\textsc{Const}(\varphi_1)$ and $\textsc{Const}(\varphi_2)$ commute elementwise.

Theorem 3.6 (Droste, Gastin [34]). *Let Σ be an alphabet, $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ a semiring, and $\mathbb{L} : \Sigma^+ \to S$ a quantitative language. Then, $\mathbb{L}$ is recognizable over $\mathbb{S}$ iff $\mathbb{L}$ is* $\mathbf{wMSO}^{\mathrm{res}}(\Sigma, \mathbb{S})$*-definable.*

3.2 An expressiveness equivalence result

In this section we will compare the expressive power of multi-weighted MSO logic and multi-weighted automata. Even for the case of a semiring, weighted MSO logic is more expressive than semiring-weighted automata (cf. [34]). As we will see in the next example, if we consider the restricted fragment $\mathbf{wMSO}^{\mathrm{res}}(\Sigma, \mathbb{S})$ for multi-weighted logic, it is, in general, more expressive than multi-weighted automata.

Given a non-empty set M and $n \geq 1$, let M^n denote the set of all finite words over M of the length n.

Example 3.7. *Here, we will consider examples of multi-weighted sentences which lead to unrecognizable quantitative languages. Let Σ be an arbitrary alphabet. Consider the evaluator $\mathbb{E} = (M, \mathbb{N}\langle M^* \rangle, \Phi)$ where M is an arbitrary non-empty set and $\Phi : \mathbb{N}\langle M^* \rangle \to \mathbb{N}\langle M^* \rangle$ is the identity mapping. Let $\mathbb{L} : \Sigma^+ \to \mathbb{N}\langle M^* \rangle$ be any quantitative language recognizable over $\mathbb{E}$. Then, for all $w \in \Sigma^+$, $\mathrm{supp}(\mathbb{L}(w)) \subseteq M^{|w|}$. Based on this property, we show the unrecognizability of the semantics of the following sentences:*

- *Let $\varphi = m$ where $m \in M$. Then, for all $w \in \Sigma^+$, $[\![\varphi]\!](w) = [m]$. Then, for all $w \in \Sigma^+$ with $|w| > 1$, we have: $\mathrm{supp}([\![\varphi]\!](w)) \cap M^{|w|} = \emptyset$. Hence, the quantitative language $[\![\varphi]\!]$ is not recognizable. In contrast, in the semiring-weighted logic of [34] the semantics of a constant is always recognizable by a semiring-weighted automaton.*
- *Let $\varphi = \mathrm{TRUE}$. Then, $[\![\varphi]\!](w) = [\varepsilon]$ for all $w \in \Sigma^+$. Clearly, $[\![\varphi]\!]$ is not recognizable over $\mathbb{E}$. In contrast, in the semiring-weighted logic of [34] the semantics of a Boolean sentence is always recognizable by a semiring-weighted automaton.*
- *Let $\varphi = \bigotimes x.m$ where $x \in V_1$. Then, $[\![\varphi]\!](w) = [m^{|w|}]$ for all $w \in \Sigma^+$. Clearly, $[\![\varphi]\!]$ is recognizable over $\mathbb{E}$. Note that, for all $w \in \Sigma^+$, $[\![\varphi \otimes \varphi]\!](w) = [m^{2|w|}]$. Then, $[\![\varphi \otimes \varphi]\!]$ is not recognizable over $\mathbb{E}$. Here, the situation is similar to the case of semirings, since, as it was shown in [35], the use of $\otimes$ for noncommutative semirings may not preserve recognizability.*
- *Let $\varphi = \bigotimes x.\bigotimes y.m$ where $x, y \in V_1$ and $m \in M$. Then, $[\![\varphi]\!](w) = [m^{|w|^2}]$ for all $w \in \Sigma^+$. Again, $[\![\varphi]\!]$ is not recognizable over $\mathbb{E}$. Note that in the case of semirings the nested use of the weighted first-order universal quantifier often leads to unrecognizability [34].*

$\square$

By the first two parts of Example 3.7, Theorem 3.6 cannot be easily extended to the multi-weighted setting. Our next task is to find a restricted fragment of

$\mathbf{mwMSO}(\Sigma, \mathbb{E})$ which is expressively equivalent to multi-weighted automata. First, we analyze Example 3.7. Here, besides the standard restrictions on $\otimes$ and $\bigotimes x$, we must pay attention to the length of the strings in the multisets of the auxiliary semantics: it must be equal to the length of an input word.

For multi-weighted MSO logic, instead of the almost Boolean fragment $\mathbf{aBOOL}(\Sigma, \mathbb{E})$, we consider the fragment $\mathbf{aBOOL}^{\mathrm{res}}(\Sigma, \mathbb{E})$[1] of *restricted almost Boolean formulas* which is defined by the grammar:

$$\gamma \ ::= \ m \mid \gamma \oplus \gamma \mid \beta \otimes \gamma$$

where $m \in M$ and $\beta \in \mathbf{MSO}(\Sigma)$. Note that, for each $(w, \sigma) \in \Sigma_V^+$, $\mathrm{supp}(\langle\!\langle \gamma \rangle\!\rangle(w, \sigma)) \subseteq M$. Then, we define the *strongly restricted* multi-weighted MSO logic $\mathbf{mwMSO}^{\mathrm{s.res}}(\Sigma, \mathbb{E}) \subseteq \mathbf{wMSO}(\Sigma, \mathbb{E})$ over Σ and $\mathbb{E}$ to be the set of all formulas generated by the grammar

$$\varphi \ ::= \ \bigotimes x.\gamma \mid \varphi \oplus \varphi \mid \beta \otimes \varphi \mid \bigoplus x.\varphi \mid \bigoplus X.\varphi$$

where $x \in V_1$, $X \in V_2$, $\beta \in \mathbf{MSO}(\Sigma)$ and $\gamma \in \mathbf{aBOOL}^{\mathrm{res}}(\Sigma, \mathbb{E})$. In relation to the fragment $\mathbf{wMSO}^{\mathrm{res}}(\Sigma, \mathbb{S})$ for a semiring $\mathbb{S}$, we restrict the use of constants, Boolean formulas and the conjunction-like operator $\otimes$. Note that the multi-weighted sentences from Examples 3.2 and 3.5 are strongly restricted. We call this fragment *strongly restricted* to avoid confusion with the definition of restricted semiring-weighted MSO logic.

For a semiring $\mathbb{S}$, let $\mathbf{aBOOL}^{\mathrm{res}}(\Sigma, \mathbb{S}) = \mathbf{aBOOL}^{\mathrm{res}}(\Sigma, \mathbb{E}^{\mathbb{S}})$ and $\mathbf{wMSO}^{\mathrm{s.res}}(\Sigma, \mathbb{S}) = \mathbf{mwMSO}^{\mathrm{s.res}}(\Sigma, \mathbb{E}^{\mathbb{S}})$

Now we state our main result about multi-weighted logic on finite words. We want to point out that here we do not put any restrictions on an evaluator $\mathbb{E}$ and that this result does not extend Theorem 3.6 to the multi-weighted case (because of the generality of our model, cf. Example 3.7). In Sect. 3.4, we consider evaluators with additional properties and show that multi-weighted automata over them can be characterized by the same logical fragment as in Theorem 3.6.

Theorem 3.8. *Let Σ be an alphabet, $\mathbb{E} = (M, K, \Phi)$ an evaluator and $\mathbb{L} : \Sigma^+ \to K$ a quantitative language. Then, the following are equivalent.*

(a) $\mathbb{L}$ is recognizable over $\mathbb{E}$.
(b) $\mathbb{L}$ is $\mathbf{mwMSO}^{\mathrm{s.res}}(\Sigma, \mathbb{E})$-definable.

The proof of this theorem will be given in the rest of this section. We start with the following remark.

Remark 3.9. *Recall that the semantics $[\![\varphi]\!]$ of a multi-weighted formula $\varphi \in \mathbf{mwMSO}(\Sigma, \mathbb{E})$ is defined as the composition $\Phi \circ \langle\!\langle \varphi \rangle\!\rangle$ where $\langle\!\langle \varphi \rangle\!\rangle$ is the auxiliary semantics of φ. The behavior $[\![\mathcal{A}]\!]$ of a multi-weighted automaton $\mathcal{A}$*

[1] I am grateful to Prof. Dr. Paul Gastin for his suggestion to restrict the use of conjuction in almost Boolean formulas. This idea helped to improve the preliminary results published in [42].

over Σ and $\mathbb{E}$ can be decomposed as $[\![\mathcal{A}]\!] = \Phi \circ \langle\!\langle \mathcal{A} \rangle\!\rangle$ where $\langle\!\langle \mathcal{A} \rangle\!\rangle : \Sigma^+ \to \mathbb{N}\langle M^* \rangle$ is defined for all $w \in \Sigma^+$ by $\langle\!\langle \mathcal{A} \rangle\!\rangle(w) = \mathrm{wt}^{\#}_{\mathcal{A}}[\mathrm{Run}_{\mathcal{A}}(w)]$. We call $\langle\!\langle \mathcal{A} \rangle\!\rangle$ the auxiliary behavior *of $\mathcal{A}$.*

Since we define the behavior of multi-weighted automata and the semantics of multi-weighted MSO logic by means of the same aggregation function Φ, by Remark 3.9 it suffices to show the equivalence of multi-weighted automata and logic with respect to the auxiliary behavior and the auxiliary semantics, respectively.

Recall that the codomain of the auxiliary behavior of multi-weighted automata and the codomain of the auxiliary semantics of multi-weighted MSO logic are $\mathbb{N}\langle M^* \rangle$, whereas the weight constants are taken from M.

Our further considerations will reduce the proof of Theorem 3.8 to the case of the semiring $\mathbb{N}\langle M^* \rangle$. Here we will use the idea that a weight constant $m \in M$ can be identified with the multiset $[m] \in \mathbb{N}\langle M^* \rangle$. Let $\mathrm{Mon}(M) = \{[m] \mid m \in M\} \subseteq \mathbb{N}\langle M^* \rangle$, the set of *monomials*.

Lemma 3.10. *Let $\mathbb{L} : \Sigma^+ \to \mathbb{N}\langle M^* \rangle$ be a mapping. Then the following are equivalent.*

(a) $\mathbb{L} = \langle\!\langle \mathcal{A} \rangle\!\rangle$ for some multi-weighted automaton $\mathcal{A}$ over Σ and $\mathbb{E}$.
(b) $\mathbb{L} = [\![\mathcal{A}']\!]^{\mathbb{N}\langle M^* \rangle}$ for some semiring-weighted automaton $\mathcal{A}'$ over Σ and $\mathbb{N}\langle M^* \rangle$ such that $\textsc{Const}(\mathcal{A}') \subseteq \mathrm{Mon}(M)$.

Proof. Given a multi-weighted automaton $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ over Σ and $\mathbb{E}$, we can define the semiring-weighted automaton $\mathcal{A}' = (Q, I, T, F, \mathrm{wt}')$ over Σ and $\mathbb{N}\langle M^* \rangle$ with $\mathrm{wt}'(t) = [\mathrm{wt}(t)]$ for all $t \in T$. Then, for all $w \in \Sigma^+$, we have:

$$[\![\mathcal{A}']\!]^{\mathbb{N}\langle M^* \rangle}(w) = \sum_{\rho \in \mathrm{Run}_{\mathcal{A}}(w)} [\mathrm{wt}^{\#}_{\mathcal{A}}(\rho)] = \mathrm{wt}^{\#}_{\mathcal{A}}[\mathrm{Run}_{\mathcal{A}}(w)] = \langle\!\langle \mathcal{A} \rangle\!\rangle(w)$$

and hence $[\![\mathcal{A}']\!]^{\mathbb{N}\langle M^* \rangle} = \langle\!\langle \mathcal{A} \rangle\!\rangle$.

Conversely, for each semiring-weighted automaton $\mathcal{B} = (Q, I, T, F, \mathrm{wt})$ over Σ and $\mathbb{N}\langle M^* \rangle$ with $\textsc{Const}(\mathcal{B}) \subseteq \mathrm{Mon}(M)$, we define a multi-weighted automaton $\mathcal{A} = (Q, I, T, F, \mathrm{wt}')$ over Σ and $\mathbb{E}$ such that $\mathrm{wt}(t) = [\mathrm{wt}'(t)]$ for all $t \in T$. Then, $\langle\!\langle \mathcal{A} \rangle\!\rangle = [\![\mathcal{B}]\!]^{\mathbb{N}\langle M^* \rangle}$. This proves the result. □

A similar equivalence holds for strongly restricted multi-weighted MSO logic:

Lemma 3.11. *Let $\mathbb{L} : \Sigma^+ \to \mathbb{N}\langle M^* \rangle$ be a mapping. Then, the following are equivalent.*

(a) $\mathbb{L} = \langle\!\langle \varphi \rangle\!\rangle$ for some multi-weighted sentence $\varphi \in \mathbf{mwMSO}^{\mathrm{s.res}}(\Sigma, \mathbb{E})$.
(b) $\mathbb{L} = [\![\varphi']\!]^{\mathbb{N}\langle M^* \rangle}$ for some semiring-weighted sentence $\varphi' \in \mathbf{wMSO}^{\mathrm{s.res}}(\Sigma, \mathbb{N}\langle M^* \rangle)$ with $\textsc{Const}(\varphi') \subseteq \mathrm{Mon}(M)$.

Proof. Given a sentence $\varphi \in \mathbf{mwMSO}^{\mathrm{s.res}}(\Sigma, \mathbb{E})$, we define the semiring-weighted sentence φ' over $\mathbb{N}\langle M^* \rangle$ by replacing every constant $m \in M$ occurring in φ by the finite multiset $[m]$. Then, $\textsc{Const}(\varphi') \subseteq \mathrm{Mon}(M)$ and $[\![\varphi']\!]^{\mathbb{N}\langle M^* \rangle} = \langle\!\langle \varphi \rangle\!\rangle$.

Conversely, if $\varphi' \in \mathbf{wMSO}^{\text{s.res}}(\Sigma, \mathbb{N}\langle M^*\rangle)$ is a sentence with $\textsc{Const}(\varphi') \subseteq \text{Mon}(M)$, we define the multi-weighted sentence φ from φ' by replacing every constant $[m]$ in φ' by m. Then, $\langle\!\langle \varphi \rangle\!\rangle = [\![\varphi']\!]^{\mathbb{N}\langle M^*\rangle}$. This proves the result. $\square$

To finish the proof of our Theorem 3.8, we show that part (b) of Lemma 3.10 is equivalent to part (b) of Lemma 3.11. To prove this, we cannot directly use the proof of [34] for Theorem 3.6 for the case $\mathbb{S} = \mathbb{N}\langle M^*\rangle$, since the translation from logic into automata presented in [34] employs some computations in the semiring $\mathbb{S}$ but the set $\text{Mon}(M)$ is not closed under the union $+$ and the Cauchy product $\cdot$.

Therefore, next we show for an arbitary semiring $\mathbb{S}$ a result stating that for the fragment $\mathbf{wMSO}^{\text{s.res}}(\Sigma, \mathbb{S})$ we have constant-preserving transformations from logic into automata and vice versa. This result could be also of the independent interest. For instance, if the operations in a semiring are not computable (consider, e.g., the addition and multiplication of real numbers), a constant-preserving transformation would be preferable.

Theorem 3.12. *Let Σ be an alphabet and $\mathbb{S}$ a semiring.*

(a) Let $\varphi \in \mathbf{wMSO}^{\text{s.res}}(\Sigma, \mathbb{S})$ be a sentence. Then, there exists a semiring-weighted automaton $\mathcal{A}$ over Σ and $\mathbb{S}$ such that $[\![\mathcal{A}]\!]^{\mathbb{S}} = [\![\varphi]\!]^{\mathbb{S}}$ and $\textsc{Const}(\mathcal{A}) = \textsc{Const}(\varphi)$.

(b) Let $\mathcal{A}$ be a semiring-weighted automaton over Σ and $\mathbb{S}$. Then, there exists a sentence $\varphi \in \mathbf{wMSO}^{\text{s.res}}(\Sigma, \mathbb{S})$ such that $[\![\varphi]\!]^{\mathbb{S}} = [\![\mathcal{A}]\!]^{\mathbb{S}}$ and $\textsc{Const}(\varphi) = \textsc{Const}(\mathcal{A})$.

We will present a proof of this theorem in the next section.

We finish the proof of Theorem 3.8 by summarizing the previous considerations:

Proof of Theorem 3.8. Immediate from Theorem 3.12 for $\mathbb{S} = \mathbb{N}\langle M^*\rangle$ and Lemmas 3.10 and 3.11. $\square$

3.3 Constant-preserving transformations

In this section, we present a proof of Theorem 3.12. Let $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ be a semiring.

We start with part (a). For a set $X \subseteq S$, let $\text{cl}_+(X) \subseteq S$ be the minimal set containing $X \cup \{\mathbb{0}\}$ which is closed under $+$.

Lemma 3.13. *Let $\varphi \in \mathbf{wMSO}^{\text{s.res}}(\Sigma, \mathbb{S})$ be a sentence. Then there exists a weighted automaton $\mathcal{A}$ over Σ and $\mathbb{S}$ such that $[\![\mathcal{A}]\!] = [\![\varphi]\!]$ and $\textsc{Const}(\mathcal{A}) \subseteq \text{cl}_+(\textsc{Const}(\varphi))$.*

Proof. We proceed by induction on φ. As in the proof presented in [34], we can restrict ourselves to a finite set $\mathcal{V} \supseteq \text{Free}(\varphi)$ of variables and encode values of variables as a word over the extended alphabet $\Sigma \times \{0,1\}^{\mathcal{V}}$. Next, we will omit the details of the proof which are analogous to the proof of [34].

- Let $\varphi = \bigotimes x.\gamma$ where $x \in V_1$ and $\gamma \in \mathbf{aBOOL}^{\mathrm{res}}(\Sigma, \mathbb{S})$. It can be easily shown by induction on the structure of a restricted almost Boolean formula $\gamma \in \mathbf{aBOOL}^{\mathrm{res}}(\Sigma, \mathbb{S})$ that $[\![\gamma]\!](\Sigma_V^+) \subseteq \mathrm{cl}_+(\textsc{Const}(\gamma))$ is a finite set. Then, we construct a weighted automaton $\mathcal{A}$ for φ as in [34], Lemma 4.4. Note that $\textsc{Const}(\mathcal{A}) \subseteq [\![\gamma]\!](\Sigma_V^+) \subseteq \mathrm{cl}_+(\textsc{Const}(\gamma))$.
- Let $\varphi = \varphi_1 \oplus \varphi_2$. In this case, we apply the standard disjoint union construction which preserves the set of weight constants of automata for φ_1 and φ_2.
- Let $\varphi = \beta \otimes \varphi'$ where $\beta \in \mathbf{MSO}(\Sigma)$. We proceed here like in the proof of [40], i.e., we take a product of a deterministic complete unweighted automaton for β and a weighted automaton for φ'. This construction preserves the set of weight constants of the weighted automaton for φ'.
- Let $\varphi = \exists \mathcal{X}.\varphi'$ with $\mathcal{X} \in V_1 \cup V_2$. Here, we apply the construction for the projection of [47], Lemma 1, which preserves the constants.

□

Now we transform the weighted automaton $\mathcal{A}$ from the previous lemma to a weighted automaton $\mathcal{A}'$ with $[\![\mathcal{A}']\!] = [\![\mathcal{A}]\!]$ and $\textsc{Const}(\mathcal{A}') = \textsc{Const}(\varphi)$.

Lemma 3.14. *Let $X \subseteq S$ be a finite set and $\mathcal{A}$ a weighted automaton over Σ and $\mathbb{S}$ such that $\textsc{Const}(\mathcal{A}) \subseteq \mathrm{cl}_+(X)$. Then, there exists a weighted automaton $\mathcal{A}'$ over Σ and $\mathbb{S}$ such that $[\![\mathcal{A}']\!] = [\![\mathcal{A}]\!]$ and $\textsc{Const}(\mathcal{A}') = X$.*

Proof. Let $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$. We may assume that $T \neq \emptyset$. For each $t \in T$, let $\mathrm{wt}(t) = s_{t,1} + ... + s_{t,n_t}$ where $n_t \geq 0$ and $s_{t,1}, ..., s_{t,n_t} \in X$. The key idea of our construction is to split each transition t into n_t transitions with the weights $s_{t,1}, ..., s_{t,n_t}$. Let $n = \max\{n_t \mid t \in T\}$. We let $\mathcal{A}' = (Q', I', T', F', \mathrm{wt}')$ where:

- $Q' = Q \times \{1, ..., n\}$, $I' = I \times \{1\}$, $F' = F \times \{1, ..., n\}$;
- T' consists of all transitions $t' := ((p, i), a, (q, j))$ where $t := (p, a, q) \in T$, $i \in \{1, ..., n\}$ and $j \in \{1, ..., n_t\}$. We define the weight of t' as $\mathrm{wt}'(t') = s_{t,j}$.

Clearly, $\textsc{Const}(\mathcal{A}') \subseteq X$. Using the distributivity property of the semiring $\mathbb{S}$, it can be easily shown that $[\![\mathcal{A}']\!] = [\![\mathcal{A}]\!]$.

If $\textsc{Const}(\mathcal{A}') \neq X$, then we add some idle transitions with the weights from $X \setminus \textsc{Const}(\mathcal{A}')$ to obtain a weighted automaton whose set of weight constants is exactly X. □

Cleary, Lemmas 3.13 and 3.14 imply Theorem 3.12 (a). Next, we show part (b) of Theorem 3.12.

Lemma 3.15. *Let $\mathcal{A}$ be a weighted automaton over Σ and $\mathbb{S}$. Then, there exists a sentence $\varphi \in \mathbf{wMSO}^{\mathrm{s.res}}(\Sigma, \mathbb{S})$ such that $\textsc{Const}(\varphi) = \textsc{Const}(\mathcal{A})$ and $[\![\varphi]\!] = [\![\mathcal{A}]\!]$.*

Proof. The proof of this lemma is a slight modification of the proof of Theorem 5.5 of [34]. Let $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$. As in [34], Theorem 5.5, we assign with every transition $t \in T$ a second-order variable X_t which will keep track of positions where this transition is taken. Let $\mathcal{V} = \{X_t\}_{t \in T}$. A run of $\mathcal{A}$ can be described using a formula $\beta \in \mathbf{MSO}(\Sigma)$ with $\mathrm{Free}(\beta) = \mathcal{V}$ which demands that the values of $\mathcal{V}$-variables form a partition of the domain of an input word, the transitions of a run are matching, the labels of transitions of a run are compatible with an input word, a run starts in I and ends in F. Then, the $\mathbf{wMSO}^{\mathrm{s.res}}(\Sigma, \mathbb{S})$-sentence φ is defined as

$$\varphi = \bigoplus \mathcal{V}. \left(\beta \otimes \bigotimes x. \bigoplus_{t \in T} (X_t(x) \otimes \mathrm{wt}(t))\right). \tag{3.2}$$

where $\bigoplus \mathcal{V}$ abbreviates $\bigoplus X_1. \; ... \; \bigoplus X_n$ for an enumeration $\mathcal{V} = \{X_1, ..., X_n\}$. Clearly, $\textsc{Const}(\varphi) = \mathrm{wt}(T) = \textsc{Const}(\mathcal{A})$. Moreover, $[\![\varphi]\!] = [\![\mathcal{A}]\!]$. $\square$

Proof of Theorem 3.12. Immediate from Lemmas 3.13, 3.14 and 3.15. $\square$

3.4 Evaluators with additional properties

As we already mentioned in Sect. 3.2, the concept of multi-weighted MSO logic extends the semiring-weighted MSO logic. However, our Büchi result for the multi-weighted setting (cf. Theorem 3.8) does not agree with Theorem 3.6 for semirings, since the logical fragment of Theorem 3.8 is more restricted than the logical fragment of Theorem 3.6. This restriction can be explained, e.g., by Example 3.7.

In order to complete the picture of the robustness of multi-weighted logic, we put additional conditions on the evaluator under which the logical fragment of Theorem 3.6 (considered in the multi-weighted setting) is equivalent to multi-weighted automata. First, we provide an informal description. As in the case of semiring-weighted automata, we will define the weights of runs (which are also in M) using a binary operation on M. After that, we collect the weights of runs in a multiset and evaluate this multiset using an aggregation function.

Let X, Y be sets, $f : X \to Y$ a mapping, and $\mu \in \mathbb{N}\langle X \rangle$ a finite multiset. Let $\mathbb{E} = (M, K, \Phi)$ be an evaluator and $\mathbb{M} = (M, \diamond, \mathbb{1})$ a monoid. Let $f_{\mathbb{M}} : (M^*, \cdot, \varepsilon) \to (M, \diamond, \mathbb{1})$ be the monoid morphism with $f_{\mathbb{M}}(m) = m$ for all $m \in M$. Let $F_{\mathbb{M}} : \mathbb{N}\langle M^* \rangle \to \mathbb{N}\langle M \rangle$ be defined for all $\mu \in \mathbb{N}\langle M^* \rangle$ and $m \in M$ by

$$F_{\mathbb{M}}(\mu)(m) = \sum (\mu(x) \mid x \in \mathrm{supp}(\mu), f_{\mathbb{M}}(x) = m)$$

where $\sum$ is the usual addition of natural numbers. Informally, the mapping $F_{\mathbb{M}}$ replaces each sequence $m_1 ... m_k \in M^*$ in a multiset by a single element $(m_1 \diamond ... \diamond m_k) \in M$, keeping multiplicities. A natural algebraic description of $F_{\mathbb{M}}$ will follow in Lemma 3.20. We will abuse the notation and understand a multiset $\mu \in \mathbb{N}\langle M \rangle$ as a multiset in $\mathbb{N}\langle M^* \rangle$ with $\mathrm{supp}(\mu) \subseteq M$.

Definition 3.16. *Let $\mathbb{E} = (M, K, \Phi)$ be an evaluator and $\mathbb{M} = (M, \diamond, \mathbb{1})$ a monoid. We say that $\mathbb{E}$ is an* $\mathbb{M}$-evaluator *if $\Phi \circ F_{\mathbb{M}} = \Phi$.*

Definition 3.16 means that the values of Φ of $\mathbb{N}\langle M \rangle$ completely determine its values on $\mathbb{N}\langle M^* \rangle$, and the diagram depicted in Fig. 3.1 commutes.

$$\begin{tikzcd} \mathbb{N}\langle M^*\rangle \arrow[rr, "F_{\mathbb{M}}"] \arrow[dr, "\Phi"'] & & \mathbb{N}\langle M\rangle \arrow[dl, "\Phi"] \\ & K & \end{tikzcd}$$

Figure 3.1: The diagram for $\mathbb{M}$-evaluators

Example 3.17. *(a) Let $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ be a semiring. Consider the monoid $\mathbb{M} = (S, \cdot, \mathbb{1})$. Then, $\mathbb{E}^{\mathbb{S}} = (S, S, \Phi^{\mathbb{S}})$ is an $\mathbb{M}$-evaluator.*

(b) Consider the evaluator $\mathbb{E}^{\text{RATIO}} = (M, K, \Phi^{\text{RATIO}})$ from Example 2.3 where $M = \mathbb{Q} \times \mathbb{Q}_{\geq 0}$ and $K = \mathbb{Q} \cup \{\infty\}$. Consider the monoid $\mathbb{M} = (M, +, (0, 0))$ where $+$ is the componentwise addition. Then, $\mathbb{E}^{\text{RATIO}}$ is an $\mathbb{M}$-evaluator.

(c) For a secondary cost bound $\eta \in \mathbb{Q}_{\geq 0}$, let $\mathbb{E}^{\text{KNAP}(\eta)} = (M, K, \Phi^{\text{KNAP}(\eta)})$ be the evaluator of Example 2.4 where $M = \mathbb{Q} \times \mathbb{Q}$ and $K = \mathbb{Q} \cup \{\infty\}$. Consider the monoid $\mathbb{M} = (M, +, (0, 0))$ where $+$ is the componentwise addition. Then, $\mathbb{E}^{\text{KNAP}(\eta)}$ is an $\mathbb{M}$-evaluator.

(d) Let $\mathbb{E}^{\text{DISC}} = (M, K, \Phi^{\text{DISC}})$ be the evaluator where $M = \mathbb{Q} \times (\mathbb{Q} \cap (0, 1])$, $K = \mathbb{Q} \cup \{\infty\}$ and Φ^{DISC} is defined as in Example 2.5. Consider the monoid $\mathbb{M} = (M, \diamond, (0, 1))$ where $\diamond$ is defined for all $(x_1, d_1), (x_2, d_2) \in M$ by $(x_1, d_1) \diamond (x_2, d_2) = (x_1 + d_1 \cdot x_2,\ d_1 \cdot d_2)$. Then, $\mathbb{E}^{\text{DISC}}$ is an $\mathbb{M}$-evaluator.

(e) Now we consider the evaluator from Example 3.7. This example was a witness why the restricted fragment of [34] (considered in the multi-weighted setting) is more expressive than multi-weighted automata over arbitrary evaluators. Let $\mathbb{E} = (M, \mathbb{N}\langle M^*\rangle, \Phi)$ where M is a non-empty set and Φ is the identity mapping. Let $\mathbb{M} = (M, \diamond, \mathbb{1})$ be any monoid. We show that $\mathbb{E}$ is not an $\mathbb{M}$-evaluator. Indeed, let $\mu = [\varepsilon] \in \mathbb{N}\langle M^*\rangle$. Then, $F_{\mathbb{M}}(\mu) = [\mathbb{1}] \neq \mu$ and hence $\Phi(F_{\mathbb{M}}(\mu)) = F_{\mathbb{M}}(\mu) \neq \mu = \Phi(\mu)$. Thus, there exists no monoid $\mathbb{M}$ such that $\mathbb{E}$ is an $\mathbb{M}$-evaluator.

We say that subsets M', M'' of M *$\diamond$-commute elementwise* if $m' \diamond m'' = m'' \diamond m'$ for all $m' \in M'$ and $m'' \in M''$.

Let the set $\mathbf{aBOOL}(\Sigma, \mathbb{E})$ be defined as in the case of semirings. The fragment $\mathbf{mwMSO}^{\text{res}}_{\mathbb{M}}(\Sigma, \mathbb{E}) \subseteq \mathbf{mwMSO}(\Sigma, \mathbb{E})$ is also defined as for semirings by the rules:

$$\varphi \ ::= \ \gamma \mid \varphi \oplus \varphi \mid \varphi \otimes \varphi^{\ (!)} \mid \bigoplus x.\varphi \mid \bigoplus X.\varphi \mid \bigotimes x.\gamma$$

where $\gamma \in \mathbf{aBOOL}(\Sigma, \mathbb{E})$, $x \in V_1$ and $X \in V_2$; moreover, there is an additional restriction at the place (!): a formula $\varphi_1 \otimes \varphi_2$ belongs to $\mathbf{mwMSO}^{\text{res}}_{\mathbb{M}}(\Sigma, \mathbb{E})$ iff the sets $\text{CONST}(\varphi_1)$ and $\text{CONST}(\varphi_2)$ $\diamond$-commute elementwise. Note that if $\mathbb{M}$ is a commutative monoid, then the use of $\varphi_1 \otimes \varphi_2$ is allowed without any restrictions. Clearly, $\mathbf{mwMSO}^{\text{s.res}}(\Sigma, \mathbb{E}) \subseteq \mathbf{mwMSO}^{\text{res}}_{\mathbb{M}}(\Sigma, \mathbb{E})$.

Remark 3.18. *Consider a semiring $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$, the corresponding evaluator $\mathbb{E}_{\mathbb{S}}$ and the monoid $\mathbb{M} = (S, \cdot, \mathbb{1})$. Then, $\mathbf{mwMSO}^{\text{res}}_{\mathbb{M}}(\Sigma, \mathbb{E}^{\mathbb{S}}) = \mathbf{wMSO}^{\text{res}}(\Sigma, \mathbb{S})$.*

The main result of this section is the following theorem.

Theorem 3.19. *Let Σ be an alphabet, $\mathbb{E} = (M, K, \Phi)$ an evaluator, $\mathbb{M} = (M, \diamond, \mathbb{1})$ a monoid such that $\mathbb{E}$ is an $\mathbb{M}$-evaluator, and $\mathbb{L} : \Sigma^+ \to K$ a quantitative language. Then, the following are equivalent.*

(a) *$\mathbb{L}$ is recognizable over $\mathbb{E}$.*
(b) *$\mathbb{L}$ is $\mathbf{mwMSO}_{\mathbb{M}}^{\mathrm{res}}(\Sigma, \mathbb{E})$-definable.*

If we apply this theorem to the evaluator $\mathbb{E} = \mathbb{E}^{\mathbb{S}}$ (where $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ is a semiring) and to the monoid $\mathbb{M} = (S, \cdot, \mathbb{1})$, then we obtain Theorem 3.6. Hence, Theorem 3.19 generalizes Theorem 3.6.

The rest of this section will be devoted to the proof of Theorem 3.19. Like in the proof of Theorem 3.8, we reduce the proof to the case of semirings. In contrast to the proof of Theorem 3.8, here we do not need to revisit the constructions for semiring-weighted formulas; we can apply the result of [34] as a "black box". Whereas in the proof of Theorem 3.8 we used the semiring $(\mathbb{N}\langle M^* \rangle, +, \cdot, \emptyset, [\varepsilon])$, here we will consider a different semiring. The domain of this semiring will be the set $\mathbb{N}\langle M \rangle$. We will consider the following operations. For all $\mu_1, \mu_2 \in \mathbb{N}\langle M \rangle$, the *union* $r_1 + r_2$ is defined as before. We extend $\diamond$ to finite multisets as follows. Let $\mu_1, \mu_2 \in \mathbb{N}\langle M \rangle$. Then, we define $(\mu_1 \diamond \mu_2) \in \mathbb{N}\langle M \rangle$ for all $m \in M$ by

$$(\mu_1 \diamond \mu_2)(m) = \sum \big(\mu_1(m_1) \cdot \mu_2(m_2) \mid m_1 \in \mathrm{supp}(\mu_1), m_2 \in \mathrm{supp}(\mu_2), \\ m = m_1 \diamond m_2\big)$$

It is well known that $(\mathbb{N}\langle M \rangle, +, \diamond, \emptyset, [\mathbb{1}])$ is a semiring. We will denote this semiring simply by $\mathbb{N}\langle M \rangle$.

Lemma 3.20. $F_{\mathbb{M}} : (\mathbb{N}\langle M^* \rangle, +, \cdot, \emptyset, [\varepsilon]) \to (\mathbb{N}\langle M \rangle, +, \diamond, \emptyset, [\mathbb{1}])$ *is the unique semiring morphism satisfying* $F_{\mathbb{M}}([m]) = [m]$ *for all* $m \in M$*.*

Proof. It is straightforward to see that $F_{\mathbb{M}}(\emptyset) = \emptyset$, $F_{\mathbb{M}}([\varepsilon]) = [\mathbb{1}]$, $F_{\mathbb{M}}([m]) = [m]$ for all $m \in M$ and $F_{\mathbb{M}}(\mu_1 + \mu_2) = F_{\mathbb{M}}(\mu_1) + F_{\mathbb{M}}(\mu_2)$ for all $\mu_1, \mu_2 \in \mathbb{N}\langle M^* \rangle$. We show that $F_{\mathbb{M}}(\mu_1 \cdot \mu_2) = F_{\mathbb{M}}(\mu_1) \diamond F_{\mathbb{M}}(\mu_2)$ for all $\mu_1, \mu_2 \in \mathbb{N}\langle M^* \rangle$. For the convenience, we will use the sum of infinite families of natural numbers under the assumption that only finitely many of them are non-zero. Let $m \in M$. On the one hand, we have:

$$\begin{aligned}(F_{\mathbb{M}}(\mu_1 \cdot \mu_2))(m) &= \sum ((\mu_1 \cdot \mu_2)(x) \mid x \in M^*, f_{\mathbb{M}}(x) = m) \\ &= \sum (\mu_1(x_1) \cdot \mu_2(x_2) \mid x_1, x_2 \in M^*, f_{\mathbb{M}}(x_1 x_2) = m) \\ &= \sum (\mu_1(x_1) \cdot \mu_2(x_2) \mid x_1, x_2 \in M^*, f_{\mathbb{M}}(x_1) \diamond f_{\mathbb{M}}(x_2) = m).\end{aligned}$$

One the other hand:

$$\begin{aligned}(F_{\mathbb{M}}(\mu_1) \diamond F_{\mathbb{M}}(\mu_2))(m) &= \sum (F_{\mathbb{M}}(\mu_1) \cdot F_{\mathbb{M}}(\mu_2) \mid m_1, m_2 \in M, m = m_1 \diamond m_2) \\ &= \sum (\mu_1(x_1) \cdot \mu_2(x_2) \mid x_1, x_2 \in M^*, f_{\mathbb{M}}(x_1) \diamond f_{\mathbb{M}}(x_2) = m).\end{aligned}$$

This shows that $F_{\mathbb{M}}(\mu_1 \cdot \mu_2) = F_{\mathbb{M}}(\mu_1) \diamond F_{\mathbb{M}}(\mu_2)$. It is also not difficult to show that whenever $\Psi : (\mathbb{N}\langle M^* , +, \cdot, \emptyset, [\varepsilon]) \to (\mathbb{N}\langle M\rangle, +, \diamond, \emptyset, [\mathbb{1}])$ is a semiring morphism with $\Psi([m]) = [m]$ for all $m \in M$, then for all $\mu \in \mathbb{N}\langle M^*\rangle$ and $m \in M$ we have

$$\Psi(\mu)(m) = \sum(\mu(x) \mid x \in \mathrm{supp}(\mu), f_{\mathbb{M}}(x) = m) = F_{\mathbb{M}}(\mu)(m).$$

Then, $F_{\mathbb{M}}$ is the unique semiring morphism preserving monomials. $\square$

Lemma 3.21. *Let $\varphi \in \mathbf{mwMSO}^{\mathrm{res}}_{\mathbb{M}}(\Sigma, \mathbb{E})$ be a formula. Then, there exists a formula $\varphi' \in \mathbf{wMSO}^{\mathrm{res}}(\Sigma, \mathbb{N}\langle M\rangle)$ such that* $\mathrm{Free}(\varphi') = \mathrm{Free}(\varphi)$ *and* $[\![\varphi']\!]^{\mathbb{N}\langle M\rangle} = F_{\mathbb{M}} \circ \langle\!\langle \varphi \rangle\!\rangle$.

Proof. Let $\varphi' \in \mathbf{wMSO}^{\mathrm{res}}(\Sigma, \mathbb{N}\langle M\rangle)$ be the formula obtained from φ by replacing each constant $m \in M$ occurring in φ by the multiset $[m] \in \mathbb{N}\langle M\rangle$. Let $m_1, m_2 \in M$ with $m_1 \diamond m_2 = m_2 \diamond m_1$. Then, $[m_1] \diamond [m_2] = [m_2] \diamond [m_1]$. Then, φ' satisfies the restrictions on the use of $\otimes$ in $\mathbf{wMSO}^{\mathrm{res}}(\Sigma, \mathbb{N}\langle M\rangle)$-formulas and hence $\varphi' \in \mathbf{wMSO}^{\mathrm{res}}(\Sigma, \mathbb{N}\langle M\rangle)$. The equality $[\![\varphi']\!]^{\mathbb{N}\langle M\rangle} = F_{\mathbb{M}} \circ \langle\!\langle \varphi \rangle\!\rangle$ can be easily shown inductively using Lemma 3.20. $\square$

Lemma 3.22. *Let $\mathcal{A}$ be a semiring-weighted automaton over Σ and $\mathbb{N}\langle M\rangle$. Then, there exists a multi-weighted automaton $\mathcal{A}'$ over Σ and $\mathbb{E}$ such that $F_{\mathbb{M}} \circ \langle\!\langle \mathcal{A}' \rangle\!\rangle = [\![\mathcal{A}]\!]^{\mathbb{N}\langle M\rangle}$.*

Proof. The proof is based on a similar construction as in the proof of Lemma 3.14. Here, for every transition t of $\mathcal{A}$, we represent its weight as $\mathrm{wt}(t) = [m_{t,1}] + ... + [m_{t,n_t}]$ where $n_t \geq 0$ and $m_{t_1}, ..., m_{t,n_t} \in M$. Then, as in Lemma 3.14, we split t into n_t transitions with the weights $m_{t,1}, ..., m_{t,n_t}$. $\square$

Proof of Theorem 3.19. (a) $\Rightarrow$ (b). Let $\mathcal{A}$ be a multi-weighted automaton over Σ and $\mathbb{E}$. Then, by Theorem 3.8, there exists a sentence $\varphi \in \mathbf{mwMSO}^{\mathrm{s.res}}(\Sigma, \mathbb{E})$ (and hence $\varphi \in \mathbf{mwMSO}^{\mathrm{res}}(\Sigma, \mathbb{E})$) with $[\![\varphi]\!] = [\![\mathcal{A}]\!]$.

(b) $\Rightarrow$ (a). Let $\varphi \in \mathbf{mwMSO}^{\mathrm{res}}_{\mathbb{M}}(\Sigma, \mathbb{E})$ be a sentence. By Lemma 3.21, there exists a sentence $\varphi' \in \mathbf{wMSO}^{\mathrm{res}}(\Sigma, \mathbb{N}\langle M\rangle)$ such that $[\![\varphi']\!]^{\mathbb{N}\langle M\rangle} = F_{\mathbb{M}} \circ \langle\!\langle \varphi \rangle\!\rangle$. By Theorem 3.6, there exists a semiring-weighted automaton $\mathcal{A}'$ over Σ and $\mathbb{N}\langle M\rangle$ such that $[\![\mathcal{A}']\!]^{\mathbb{N}\langle M\rangle} = [\![\varphi']\!]^{\mathbb{N}\langle M\rangle}$. By Lemma 3.22, there exists a multi-weighted automaton $\mathcal{A}$ over Σ and $\mathbb{E}$ such that $F_{\mathbb{M}} \circ \langle\!\langle \mathcal{A} \rangle\!\rangle = [\![\mathcal{A}']\!]^{\mathbb{N}\langle M\rangle}$. Then, we have:

$$\begin{aligned}[\![\varphi]\!]^{\mathbb{E}} &= \Phi \circ \langle\!\langle \varphi \rangle\!\rangle \stackrel{*}{=} \Phi \circ F_{\mathbb{M}} \circ \langle\!\langle \varphi \rangle\!\rangle = \Phi \circ [\![\varphi']\!]^{\mathbb{N}\langle M\rangle} = \Phi \circ [\![\mathcal{A}']\!]^{\mathbb{N}\langle M\rangle} \\ &= \Phi \circ F_{\mathbb{M}} \circ \langle\!\langle \mathcal{A} \rangle\!\rangle \stackrel{*}{=} \Phi \circ \langle\!\langle \mathcal{A} \rangle\!\rangle = [\![\mathcal{A}]\!]^{\mathbb{E}}\end{aligned}$$

where at $*$ we used that $\mathbb{E}$ is an $\mathbb{M}$-evaluator. $\square$

As a corollary from Lemma 2.10 and Theorem 3.19, we obtain:

Corollary 3.23. *Let the monoid $\mathbb{M}$ and the $\mathbb{M}$-evaluator $\mathbb{E}^{\mathrm{RATIO}}$ be defined as in Example 3.17 (b). Then, it is decidable, given an alphabet Σ, a sentence $\varphi \in \mathbf{mwMSO}^{\mathrm{res}}_{\mathbb{M}}(\Sigma, \mathbb{E}^{\mathrm{RATIO}})$, $\bowtie \in \{<, \leq\}$, and a threshold $\theta \in \mathbb{Q}$, whether $[\![\varphi]\!](w) \bowtie \theta$ for some word $w \in \Sigma^+$.*

Chapter 4

Multi-weighted rational expressions

Contents

The goal of this chapter is to extend the classical Kleene-Schützenberger theorem [81] about the equivalence of recognizable and rational formal power series to the multi-weighted setting. We introduce *multi-weighted rational expressions* and define their semantics following a similar double-step approach as for multi-weighted MSO logic. We show that these multi-weighted rational expressions are expressively equivalent to multi-weighted automata.

4.1 Preliminaries

In this section, we recall the classical Kleene-Schützenberger theorem [81] for semirings. Let Σ be an alphabet and $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ a semiring.

Let $\mathbb{L}_1, \mathbb{L}_2 : \Sigma^+ \to S$ be quantitative languages. The *addition* $\mathbb{L}_1 + \mathbb{L}_2 : \Sigma^+ \to S$ is defined for all $w \in \Sigma^+$ as $(\mathbb{L}_1 + \mathbb{L}_2)(w) = \mathbb{L}_1(w) + \mathbb{L}_2(w)$. The *Cauchy-product* $\mathbb{L}_1 \cdot \mathbb{L}_2 : \Sigma^+ \to S$ is defined for all $w \in \Sigma^+$ as

$$(\mathbb{L}_1 \cdot \mathbb{L}_2)(w) = \sum \big(\mathbb{L}_1(w_1) \cdot \mathbb{L}_2(w_2) \mid w_1, w_2 \in \Sigma^+, w_1 w_2 = w\big).$$

For $\mathbb{L} : \Sigma^+ \to S$, the *iteration* $\mathbb{L}^+ : \Sigma^+ \to S$ is defined for all $w \in \Sigma^+$ by

$$\mathbb{L}^+(w) = \sum \big(\mathbb{L}(w_1) \cdot ... \cdot \mathbb{L}(w_i) \mid 1 \leq i \leq |w|, w_1, ..., w_i \in \Sigma^+, w_1 ... w_i = w\big).$$

$$
\begin{aligned}
[\![\emptyset]\!] &= \mathbb{0}_{\Sigma^+} \\
[\![(s,a)]\!] &= sa \\
[\![e_1 + e_2]\!] &= [\![e_1]\!] + [\![e_2]\!] \\
[\![e_1 \cdot e_2]\!] &= [\![e_1]\!] \cdot [\![e_2]\!] \\
[\![e^+]\!] &= [\![e]\!]^+
\end{aligned}
$$

Table 4.1: The semantics of rational expressions over a semiring

For $s \in S$ and $a \in \Sigma$, let $sa : \Sigma^+ \to S$ denote the quantitative language such that $(sa)(a) = a$ and $(sa)(w) = \mathbb{0}$ for all $w \in \Sigma^+ \setminus \{a\}$. Let $\mathbb{0}_{\Sigma^+} : \Sigma^+ \to S$ denote the quantitative language with $\mathbb{0}_{\Sigma^+}(w) = \mathbb{0}$ for all $w \in \Sigma^+$.

The set $\text{RAT}(\Sigma, \mathbb{S})$ of *weighted rational expressions* over Σ and $\mathbb{S}$ is given by the grammar

$$E \ ::= \ \emptyset \mid (s,a) \mid E + E \mid E \cdot E \mid E^+ \tag{4.1}$$

where $s \in S$ and $a \in \Sigma$. For $e \in \text{RAT}(\Sigma, \mathbb{S})$, the semantics $[\![e]\!] : \Sigma^+ \to S$ is given inductively on the structure of e as shown in Table 4.1 where $s \in S$ and $a \in \Sigma$. In order to emphasize that the semantics of e is defined with respect to the semiring $\mathbb{S}$, we will sometimes write $[\![e]\!]^{\mathbb{S}}$ for $[\![e]\!]$. A quantitative language $\mathbb{L} : \Sigma^+ \to S$ is called *rational* over $\mathbb{S}$ if $\mathbb{L} = [\![e]\!]$ for some $e \in \text{RAT}(\Sigma, \mathbb{S})$.

Note that, for any $a \in \Sigma$, $[\![\emptyset]\!] = [\![(\mathbb{0}, a)]\!]$ and hence $\emptyset$ can be omitted. However, we will need $\emptyset$ in the multi-weighted setting and, for the convenience, we keep it here as well.

The following result is the Kleene-Schützenberger theorem for semiring-weighted automata.

Theorem 4.1 (Schützenberger [81]). *Let Σ be an alphabet, $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ a semiring, and $\mathbb{L} : \Sigma^+ \to S$ a quantitative language. Then, $\mathbb{L}$ is recognizable over $\mathbb{S}$ iff $\mathbb{L}$ is rational over $\mathbb{S}$.*

Note that here we consider the iteration $\mathbb{L}^+$ instead of the star operation $\mathbb{L}^*$. This is not a restriction, since we ignore the empty word (cf. [39]).

4.2 Multi-weighted rational expressions

Let Σ be an alphabet and $\mathbb{E} = (M, K, \Phi)$ an evaluator. The set $\text{RAT}(\Sigma, \mathbb{E})$ of *multi-weighted rational expressions* over Σ and $\mathbb{E}$ is given by the grammar (4.1) where $s \in M$ and $a \in \Sigma$.

As in the previous chapter, consider the semiring $(\mathbb{N}\langle M^* \rangle, +, \cdot, \emptyset, [\varepsilon])$ of finite multisets with the union and Cauchy-product; we will denote this semiring simply by $\mathbb{N}\langle M^* \rangle$.

Let $e \in \text{RAT}(\Sigma, \mathbb{E})$. We define the semantics of multi-weighted rational expressions following the double-step approach introduced in the previous chapter for multi-weighted MSO logic.

- The *auxiliary semantics* of e is the mapping $\langle\!\langle e\rangle\!\rangle : \Sigma^+ \to \mathbb{N}\langle M^*\rangle$ defined with respect to the semiring $\mathbb{N}\langle M^*\rangle$ as follows. Let $e' \in \textsc{Rat}(\Sigma, \mathbb{N}\langle M^*\rangle)$ be obtained from e by replacing every atomic expression (m, a) of e (with $m \in M$, $a \in \Sigma$) by $([m], a)$. Then, we put $\langle\!\langle e\rangle\!\rangle = [\![e']\!]^{\mathbb{N}\langle M^*\rangle}$.
- The *proper semantics* (or simply *semantics*) of e is defined as $[\![e]\!] = \Phi \circ \langle\!\langle e\rangle\!\rangle$. Note that $[\![e]\!] : \Sigma^+ \to K$.

In order to emphasize that e is defined with respect to the evaluator $\mathbb{E}$, we will sometimes write $[\![e]\!]^{\mathbb{E}}$ for the semantics $[\![e]\!]$. We say that a quantitative language $\mathbb{L} : \Sigma^+ \to K$ is *rational* over $\mathbb{E}$ if $\mathbb{L} = [\![e]\!]$ for some multi-weighted rational expression $e \in \textsc{Rat}(\Sigma, \mathbb{E})$.

Example 4.2. *Consider the alphabet $\Sigma = \{\leftrightarrow, \updownarrow\}$, the evaluator $\mathbb{E}^{\mathrm{Disp}(2)}$ as defined in Example 2.7, and the quantitative language $\mathbb{L}$ of Example 3.2. We construct the following multi-weighted rational expression e over Σ and $\mathbb{E}$:*

$$e = ((\leftrightarrow, v_{\leftarrow}) + (\leftrightarrow, v_{\rightarrow}) + (\updownarrow, v_{\downarrow}) + (\updownarrow, v^{\uparrow}))^+.$$

Then, $[\![e]\!] = \mathbb{L}$. For instance, let $w = \leftrightarrow\updownarrow$. Then,

$$\langle\!\langle e\rangle\!\rangle(w) = [v_{\leftarrow}v_{\downarrow}] \uplus [v_{\leftarrow}v_{\uparrow}] \uplus [v_{\rightarrow}v_{\downarrow}] \uplus [v_{\rightarrow}v_{\uparrow}]$$

and hence $[\![e]\!](w) = \sqrt{2}$.

Now let $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ be a semiring and $\mathbb{E}^{\mathbb{S}} = (S, S, \Phi^{\mathbb{S}})$ the evaluator generated by $\mathbb{S}$. Note that $\textsc{Rat}(\Sigma, \mathbb{S}) = \textsc{Rat}(\Sigma, \mathbb{E}^{\mathbb{S}})$, i.e., every formula in $\textsc{Rat}(\Sigma, \mathbb{E}^{\mathbb{S}})$ can be interpreted dually. We show that the use of our multi-weighted approach leads to the same semantics as for semirings.

Theorem 4.3. *Let $\varphi \in \textsc{Rat}(\Sigma, \mathbb{S})$. Then, $[\![\varphi]\!]^{\mathbb{S}} = [\![\varphi]\!]^{\mathbb{E}_{\mathbb{S}}}$.*

We prove this theorem in the rest of this section. First, we show the following technical lemma.

Lemma 4.4. *Let $r, r_1, r_2 : \Sigma^+ \to \mathbb{N}\langle S^*\rangle$. Then:*

(a) $\Phi^{\mathbb{S}} \circ (r_1 + r_2) = (\Phi^{\mathbb{S}} \circ r_1) + (\Phi^{\mathbb{S}} \circ r_2)$.
(b) $\Phi^{\mathbb{S}} \circ (r_1 \cdot r_2) = (\Phi^{\mathbb{S}} \circ r_1) \cdot (\Phi^{\mathbb{S}} \circ r_2)$.
(c) $\Phi^{\mathbb{S}} \circ (r^+) = (\Phi^{\mathbb{S}} \circ r)^+$.

Here, the operations $+, \cdot, ^+$ on the left hand side of the formulas are induced by the union and Cauchy-product of the multisets in $\mathbb{N}\langle S^*\rangle$, and the operations $+, \cdot, ^+$ on the right hand side of the formulas are induced by the operations of the semiring $\mathbb{S}$.

Proof. Part (a) is straightforward. Parts (b) and (c) can be shown in the same manner. We show explicitly (c).

Let $g : S^* \to S$ be defined as $g(\varepsilon) = \mathbb{1}$ and $g(s_1...s_n) = (s_1 \cdot ... \cdot s_n) \in S$ for all $n \geq 1$ and $s_1, ..., s_n \in S$. Then, $g(uu') = g(u) \cdot g(u')$ for all $u, u' \in S^*$. Let $w \in \Sigma^+$. On the one hand:

$$\begin{aligned}
(\Phi^{\mathbb{S}} \circ r^+)(w) &= \sum_{u \in S^*} (r^+(w))(u) \cdot g(u) \\
&= \sum_{u \in S^*} \left(\sum_{i=1}^{|w|} \sum_{w = w_1 ... w_i} r(w_1) \cdot ... r(w_i) \right)(u) \cdot g(u) \\
&= \sum_{u \in S^*} \sum_{i=1}^{|w|} \sum_{\substack{w = w_1 ... w_i \\ u = u_1 ... u_i}} r(w_1)(u_1) \cdot ... \cdot r(w_i)(u_i) \cdot g(u) \\
&= \sum_{i=1}^{|w|} \sum_{u_1, ..., u_i \in S^*} \sum_{w = w_1 ... w_i} r(w_1)(u_1) \cdot ... \cdot r(w_i)(u_i) \cdot g(u_1) \cdot ... \cdot g(u_i)
\end{aligned}$$

On the other hand:

$$\begin{aligned}
(\Phi^{\mathbb{S}} \circ r)^+(w) &= \sum_{i=1}^{|w|} \sum_{w_1 ... w_i = w} \Phi^{\mathbb{S}}(r(w_1)) \cdot ... \cdot \Phi^{\mathbb{S}}(r(w_i)) \\
&= \sum_{i=1}^{|w|} \sum_{w_1 ... w_i = w} \sum_{u_1, ..., u_i \in S^*} r(w_1)(u_1) \cdot ... \cdot r(w_i)(u_i) \cdot g(u_1) \cdot ... \cdot g(u_i).
\end{aligned}$$

Then, $\Phi^{\mathbb{S}} \circ r^+ = (\Phi^{\mathbb{S}} \circ r)^+$. $\square$

Proof of Theorem 4.3. We proceed by induction on the structure of a rational expression $e \in \mathrm{RAT}(\Sigma, \mathbb{S})$.

- The cases $e = \emptyset$ and $e = (s, a)$ with $s \in S$ and $a \in \Sigma$ are straightforward.
- Let $e = e_1 \diamond e_2$ where $\diamond \in \{+, \cdot\}$. Then:

$$\begin{aligned}
[\![e]\!]^{\mathbb{E}^{\mathbb{S}}} &= \Phi^{\mathbb{S}} \circ (\langle\!\langle e_1 \diamond e_2 \rangle\!\rangle) = \Phi^{\mathbb{S}} \circ (\langle\!\langle e_1 \rangle\!\rangle \diamond \langle\!\langle e_2 \rangle\!\rangle) \overset{(!)}{=} (\Phi^{\mathbb{S}} \circ \langle\!\langle e_1 \rangle\!\rangle) \diamond (\Phi^{\mathbb{S}} \circ \langle\!\langle e_2 \rangle\!\rangle) \\
&= [\![e_1]\!]^{\mathbb{E}^{\mathbb{S}}} \diamond [\![e_2]\!]^{\mathbb{E}^{\mathbb{S}}} \overset{(!!)}{=} [\![e_1]\!]^{\mathbb{S}} \diamond [\![e_2]\!]^{\mathbb{S}} = [\![e]\!]^{\mathbb{S}}.
\end{aligned}$$

  Here, at the place (!) we apply Lemma 4.4 (part (a) for $\diamond = +$ and part (b) for $\diamond = \cdot$), and at the place (!!) we apply induction hypothesis for e_1 and e_2.
- For $e = (e')^+$, the equality $[\![e]\!]^{\Phi^{\mathbb{S}}} = [\![e]\!]^{\mathbb{S}}$ can be shown similarly to the previous case using part (c) of Lemma 4.4.

$\square$

4.3 A Kleene-Schützenberger theorem

Our Kleene-Schützenberger theorem for multi-weighted automata is the following.

Theorem 4.5. *Let Σ be an alphabet, $\mathbb{E} = (M, K, \Phi)$ an evaluator, and $\mathbb{L} : \Sigma^+ \to K$ a quantitative language. Then, the following are equivalent.*

(a) $\mathbb{L}$ is recognizable over $\mathbb{E}$.
(b) $\mathbb{L}$ is rational over $\mathbb{E}$.

The proof of this theorem will be given in the rest of this section.

As in the case of multi-weighted MSO logic, we could try to reduce the proof to the case of the semiring $\mathbb{N}\langle M^* \rangle$. In particular, given a rational expression $e \in \textsc{Rat}(\Sigma, \mathbb{E})$, using Theorem 4.1 for $\mathbb{S} = \mathbb{N}\langle M^* \rangle$, we could construct a semiring-weighted automaton $\mathcal{A}$ over Σ and $\mathbb{N}\langle M^* \rangle$ with $[\![\mathcal{A}]\!]^{\mathbb{N}\langle M^* \rangle} = \langle\!\langle e \rangle\!\rangle$; note that the transition weights of $\mathcal{A}$ are multisets in $\mathbb{N}\langle M^* \rangle$. Then, as in the case of multi-weighted MSO logic, we could revisit the proof of Theorem 4.1 with respect to constant-preserving transformations.

Here, we introduce a different proof. We will reduce the proof of Theorem 4.5 to Kleene-Schützenberger Theorem 4.1 for the semiring $(\mathbb{N}, +, \cdot, 0, 1)$ of natural numbers (we will denote it simply by $\mathbb{N}$).

Let X be an alphabet, i.e., a non-empty finite set. For $w = a_1 ... a_n \in \Sigma^+$ and $u = m_1 ... m_k \in X^+$ with $n = k$, let $\langle x, u \rangle = (a_1, m_1) ... (a_n, m_n) \in (\Sigma \times X)^+$. For a quantitative language $\mathbb{L} : (\Sigma \times X)^+ \to \mathbb{N}$, let $\mathbb{L}^\bullet : \Sigma^+ \to \mathbb{N}\langle X^+ \rangle$ be defined for all $w \in \Sigma^+$ and $u \in X^+$ as:

$$\mathbb{L}^\bullet(w)(u) = \begin{cases} \mathbb{L}(\langle w, u \rangle), & \text{if } |w| = |u|, \\ 0, & \text{otherwise.} \end{cases}$$

Note that this definition is correct, since for each $w \in \Sigma^+$, we have $|\operatorname{supp}(\mathbb{L}^\bullet(w))| \leq |X|^{|w|}$.

The following theorem shows that multi-weighted automata can be characterized by means of semiring-weighted automata over $(\mathbb{N}, +, \cdot, \mathbb{0}, \mathbb{1})$.

Theorem 4.6. *Let Σ be an alphabet, $\mathbb{E} = (M, K, \Phi)$ an evaluator, and $\mathbb{L} : \Sigma^+ \to K$ a quantitative language. Then the following are equivalent.*

(a) $\mathbb{L}$ is recognizable over $\mathbb{E}$.
(b) There exist an alphabet $X \subseteq M$ and a quantitative language $\mathbb{M} : (\Sigma \times X)^+ \to \mathbb{N}$ such that $\mathbb{M}$ is recognizable over $\mathbb{N}$ and $\mathbb{L} = \Phi \circ \mathbb{M}^\bullet$.

Proof. First, we show that (a) implies (b). Let $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ be a multi-weighted automaton over Σ and $\mathbb{E}$. We may assume that $T \neq \emptyset$. Let $X = \textsc{Const}(\mathcal{A})$. Consider the semiring-weighted automaton $\mathcal{A} = (Q, I, T', F, \mathrm{wt}')$ over $\Sigma \times X$ and $\mathbb{N}$ where:

- $T' = \{(p, (a, x), q) \mid t := (p, a, q) \in T \text{ and } x = \mathrm{wt}(t)\}$;
- $\mathrm{wt}(t') = 1$ for all $t' \in T'$.

Then, for all $w = a_1...a_n \in \Sigma^+$ and $u = m_1...m_k \in X^+$:

$$[\![\mathcal{A}']\!]^\bullet(w)(u) = |\{\rho \in \mathrm{Run}_\mathcal{A}(w) \mid \mathrm{wt}^\#_\mathcal{A}(\rho) = u\}|.$$

Let $\mathbb{M} = [\![\mathcal{A}']\!]$. Then, for all $w \in \Sigma^+$, $\mathbb{M}^\bullet(w) = \mathrm{wt}^\#_\mathcal{A}[\mathrm{Run}_\mathcal{A}(w)]$ and hence $[\![\mathcal{A}]\!] = \Phi \circ \mathbb{M}^\bullet$.

Second, we show that (b) implies (a). Let $X \subseteq M$ be an alphabet and $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ a semiring-weighted automaton over $\Sigma \times X$ and $\mathbb{N}$. We may assume that $T \neq \emptyset$ and $n := \max\{\mathrm{wt}(t) \mid t \in T\} \geq 1$.

We define a multi-weighted automaton $\mathcal{A}' = (Q', I', T', F', \mathrm{wt}')$ over Σ and $\mathbb{E}$ with $[\![\mathcal{A}']\!] = \Phi \circ [\![\mathcal{A}]\!]^\bullet$ as follows:

- $Q' = Q \times X \times \{1, ..., n\}$, $I' = I \times \{m_0\} \times \{1\}$ where $m_0 \in X$ is fixed, $F' = F \times X \times \{1, ..., n\}$;
- T' consists of all transitions $t' := ((q, m, i), a, (q', m', i'))$ such that $t := (q, (a, m'), q') \in T$, $i \in \{1, ..., n\}$ and $i' \in \{1, ..., \mathrm{wt}(t)\}$. The weight of t' is defined as $\mathrm{wt}'(t') = m'$.

Then, whenever $\rho = t_1...t_k \in \mathrm{Run}_\mathcal{A}$ is a run such that, for all $i \in \{1, ..., k\}$, $\mathrm{label}(t_i) = (a_i, m_i) \in \Sigma \times X$ and $\mathrm{wt}(t_i) = j_i \in \mathbb{N}$, this run is simulated in $\mathcal{A}'$ by $j_1 \cdot ... \cdot j_k$ many runs $\rho \in \mathrm{Run}_{\mathcal{A}'}(a_1...a_k)$ with $\mathrm{wt}^\#_{\mathcal{A}'}(\rho) = m_1...m_k$. Then, the claim follows. □

Now we give a similar characterization of rational quantitative languages over $\mathbb{E}$.

Theorem 4.7. *Let Σ be an alphabet, $\mathbb{E} = (M, K, \Phi)$ an evaluator, and $\mathbb{L} : \Sigma^+ \to K$ a quantitative language. Then the following are equivalent.*

(a) *$\mathbb{L}$ is rational over $\mathbb{E}$.*
(b) *There exist an alphabet $X \subseteq M$ and a quantitative language $\mathbb{M} : (\Sigma \times X)^+ \to \mathbb{N}$ such that $\mathbb{M}$ is rational over $\mathbb{N}$ and $\mathbb{L} = \Phi \circ \mathbb{M}^\bullet$.*

The proof of this theorem will be based on the following technical lemma.

Lemma 4.8. *Let $X \subseteq M$ be an alphabet and $\mathbb{L}, \mathbb{L}_1, \mathbb{L}_2 : (\Sigma \times X)^+ \to \mathbb{N}$ quantitative languages. Then:*

(a) $(\mathbb{L}_1 + \mathbb{L}_2)^\bullet = \mathbb{L}_1^\bullet + \mathbb{L}_2^\bullet$.
(b) $(\mathbb{L}_1 \cdot \mathbb{L}_2)^\bullet = \mathbb{L}_1^\bullet \cdot \mathbb{L}_2^\bullet$.
(c) $(\mathbb{L}^+)^\bullet = (\mathbb{L}^\bullet)^+$.

Proof. Part (a) is straightforward and parts (b) and (c) can be shown in the same manner. We show explicitely the more difficult part (c).

First, we show that $(\mathbb{L}^+)^\bullet(w)(u) = (\mathbb{L}^\bullet)^+(w)(u)$ for all $w \in \Sigma^+$ and $u \in X^+$ with $|w| = |u|$. On the one hand:

$$(\mathbb{L}^+)^\bullet(w)(u) = \mathbb{L}^+(\langle w, u\rangle) = \sum_{i=1}^{|w|} \sum_{\langle w,u\rangle = \langle w_1,u_1\rangle ... \langle w_i,u_i\rangle} \mathbb{L}(\langle w_1, u_1\rangle) \cdot ... \cdot \mathbb{L}(\langle w_i, u_i\rangle).$$

On the other hand:

$$\begin{aligned}(\mathbb{L}^\bullet)^+(w)(u) &= \sum_{i=1}^{|w|} \sum_{\substack{w=w_1\dots w_i\\ u=u_1\dots u_i}} \mathbb{L}^\bullet(w_1)(u_1)\cdot\ldots\cdot\mathbb{L}^\bullet(w_i)(u_i)\\ &= \sum_{\langle w,u\rangle=\langle w_1,u_1\rangle\dots\langle w_i,u_i\rangle} \mathbb{L}(\langle w_1,u_1\rangle)\cdot\ldots\cdot\mathbb{L}(\langle w_i,u_i\rangle).\end{aligned} \tag{4.2}$$

Now let $w\in\Sigma^+$ and $u\in X^+$ with $|w|\neq|u|$. Then, clearly, $(\mathbb{L}^+)^\bullet(w)(u)=0$. We show that $(\mathbb{L}^\bullet)^+(w)(u)=0$. Let $i\in\{1,...,|w|\}$ and $w=w_1...w_i$ and $u=u_1...u_i$ as in Equation (4.2). Then, $|w_j|\neq|u_j|$ for some $j\in\{1,...,i\}$ and hence $\mathbb{L}^\bullet(w_j)(u_j)=0$. Then, as in is easy to see from (4.2): $(\mathbb{L}^\bullet)^+(w)(u)=0$. □

Now we turn to the proof of Theorem 4.7

Proof (of Theorem 4.7). First, we show that (a) implies (b). Let $e\in\textsc{Rat}(\Sigma,\mathbb{E})$ be a multi-weighted rational expression. Let X be the set of all constants $m\in M$ appearing in e. We may assume that $X\neq\emptyset$ (otherwise, all atomic expressions are $\emptyset$; we replace one of them by $\emptyset\cdot m_0$ with $m_0\in M$ and obtain the equivalent multi-weighted expression). Let $\varphi(e)\in\textsc{Rat}(\Sigma\times X,\mathbb{N})$ be obtained from e by replacing each atomic expression (m,a) of w (with $m\in M$ and $a\in\Sigma$) by $(1,(a,m))$. We show by induction on the structure of e that $\langle\!\langle e\rangle\!\rangle=(\llbracket\varphi(e)\rrbracket^{\mathbb{N}})^\bullet$.

- Let $e=\emptyset$. Then, for all $w\in\Sigma^+$, $\langle\!\langle e\rangle\!\rangle(w)=\emptyset$. Note that, for all $u\in X^+$ with $|u|=|w|$, we have $\llbracket\emptyset\rrbracket^{\mathbb{N}}(\langle w,u\rangle)=0$. Then, $(\llbracket\emptyset\rrbracket^{\mathbb{N}})^\bullet(w)=\emptyset$. Hence, $\langle\!\langle e\rangle\!\rangle=(\llbracket\varphi(e)\rrbracket^{\mathbb{N}})^\bullet$.
- Let $e=(m,a)$ with $m\in X$ and $a\in\Sigma$. Then, $\langle\!\langle e\rangle\!\rangle(a)=[m]$ and $\langle\!\langle e\rangle\!\rangle(w)=\emptyset$ for all $w\in\Sigma^+\setminus\{a\}$. Let $u\in X^+$ with $|u|=|w|$. Then,

$$\llbracket\varphi(e)\rrbracket^{\mathbb{N}}(\langle w,u\rangle)=\begin{cases}1, & \text{if } w=a \text{ and } u=m\\ 0, & \text{otherwise}\end{cases}$$

  and hence $\langle\!\langle e\rangle\!\rangle=(\llbracket\varphi(e)\rrbracket^{\mathbb{N}})^\bullet$.
- Let $e=e_1+e_2$. By induction hypothesis, $\langle\!\langle e_i\rangle\!\rangle=(\llbracket\varphi(e_i)\rrbracket^{\mathbb{N}})^\bullet$ for $i\in\{1,2\}$. Then,

$$\begin{aligned}(\llbracket\varphi(e)\rrbracket^{\mathbb{N}})^\bullet &= (\llbracket\varphi(e_1)\rrbracket^{\mathbb{N}}+\llbracket\varphi(e_1)\rrbracket^{\mathbb{N}})^\bullet \overset{(!)}{=} (\llbracket\varphi(e_1)\rrbracket^{\mathbb{N}})^\bullet+(\llbracket\varphi(e_1)\rrbracket^{\mathbb{N}})^\bullet\\ &= \langle\!\langle e_1\rangle\!\rangle+\langle\!\langle e_2\rangle\!\rangle=\langle\!\langle e_1+e_2\rangle\!\rangle.\end{aligned}$$

  Here, at the place (!), we apply Lemma 4.8 (a).
- The proofs for $e=e_1\cdot e_2$ and $e=(e')^+$ are similar to the previous case: for $e_1\cdot e_2$ we apply Lemma 4.8 (b) and for $(e')^+$ we apply Lemma 4.8 (c).

Let $\mathbb{L}=\llbracket\varphi(e)\rrbracket^{\mathbb{N}}$. Note that $\mathbb{L}$ is rational over $\mathbb{N}$. Then, $\llbracket e\rrbracket=\Phi\circ\mathbb{L}^\bullet$.

Now we show that (b) implies (a). Let $X\subseteq M$ be an alphabet and $e\in\textsc{Rat}(\Sigma\times X,\mathbb{N})$ such that $\mathbb{L}=\Phi\circ(\llbracket e\rrbracket^{\mathbb{N}})^\bullet$. We show that $\mathbb{L}$ is rational over $\mathbb{E}$. Note that, for all $u\in\Sigma\times X$, we have:

- $[\![(0,u)]\!]^{\mathbb{N}} = [\![\emptyset]\!]^{\mathbb{N}}$;
- $[\![(k,u)]\!]^{\mathbb{N}} = \underbrace{[\![(1,u)]\!]^{\mathbb{N}} + ... + [\![(1,u)]\!]^{\mathbb{N}}}_{k\text{-times}}$ for every $k \geq 1$.

Then, we may assume without loss of generality that, whenever $(k,(m,a))$ is an atomic subexpression of e (with $k \in \mathbb{N}$, $a \in \Sigma$ and $m \in X$), we have $k = 1$. Then, we replace all such $(1,(m,a))$ of e by (a,m) and obtain the multi-weighted rational expression $e' \in \mathrm{R}\scriptstyle{\mathrm{AT}}\textstyle(\Sigma, \mathbb{E})$. Based on Lemma 4.8, we can show as in the proof of the implication (a) $\Rightarrow$ (b) that $\langle\!\langle e' \rangle\!\rangle = ([\![e]\!]^{\mathbb{N}})^{\bullet}$. Then, $\mathbb{L} = [\![e']\!]^{\mathbb{E}}$ and hence $\mathbb{L}$ is rational over $\mathbb{E}$.

□

4.4 Qualitative evaluators

In Theorem 4.7, we showed that the basis of a multi-weighted automaton, without regard of its quantitative measure, can be described by a weighted automaton over the semiring of natural numbers. We can go further and ask whether the basis of a multi-weighted automaton is *qualitative*, i.e., can be described by a classical non-deterministic finite automaton. In this section, we will deal with this question.

Let Σ be an alphabet and $\mathbb{E} = (M, K, \Phi)$ an evaluator. For an alphabet X and a language $\mathcal{L} \subseteq (\Sigma \times X)^+$, let $\mathcal{L}^{\bullet} : \Sigma^+ \to \mathbb{N}\langle X^+ \rangle$ be defined for all $w \in \Sigma^+$ and $u \in X^+$ as

$$\mathcal{L}^{\bullet}(w)(u) = \begin{cases} 1, & \text{if } |w| = |u| \text{ and } \langle w, u \rangle \in \mathcal{L}, \\ 0, & \text{otherwise.} \end{cases}$$

We say that the evaluator $\mathbb{E}$ is *qualitative* if, for every alphabet Σ and every quantitative language $\mathbb{L} : \Sigma^+ \to K$, the following are equivalent:

(i) $\mathbb{L}$ is recognizable over $\mathbb{E}$;
(ii) there exist an alphabet $X \subseteq M$ and a recognizable language $\mathcal{L} \subseteq (\Sigma \times X)^+$ such that $\mathbb{L} = \Phi \circ \mathcal{L}^{\bullet}$.

The following lemma shows that the implication (ii) $\Rightarrow$ (i) holds for arbitrary evaluators.

Lemma 4.9. *Let $\mathbb{E} = (M, K, \Phi)$ be an evaluator, Σ and $X \subseteq M$ alphabets, and $\mathcal{L} \subseteq (\Sigma \times X)^+$ a recognizable language. Then, the quantitative language $\Phi \circ \mathcal{L}^{\bullet} : \Sigma^+ \to K$ is recognizable over $\mathbb{E}$.*

Proof. Let $\mathcal{A} = (Q, I, T, F)$ be a deterministic finite automaton over the alphabet $\Sigma \times X$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$ and let $\mathbb{L} = \Phi \circ \mathcal{L}^{\bullet}$. We show that $\mathbb{L}$ is recognizable over $\mathbb{E}$. For this, we construct a multi-weighted automaton $\mathcal{A}' = (Q', I', T', F', \mathrm{wt}')$ over Σ and $\mathbb{E}$ where:

- $Q' = Q \times X$, $I' = I \times \{x_0\}$ for some fixed $x_0 \in X$, $F' = F \times X$;

- T' consists of edges of the form $t' := ((q,x),a,(q',x'))$ where $(q,(a,x'),q') \in T$. For such an edge t', we let $\mathrm{wt}'(t') = x'$.

Since $\mathcal{A}$ is deterministic, we have $[\![\mathcal{A}']\!] = \mathbb{L}$. □

Unfortunately, not every evaluator is qualitative, since the implication (i) ⇒ (ii) does not always hold:

Lemma 4.10. *There exists an evaluator $\mathbb{E} = (M, K, \Phi)$ such that $\mathbb{E}$ is not qualitative.*

Proof. Let $M = \{m\}$ be a singleton set, $K = \mathbb{N}\langle M^*\rangle$ and $\Phi : \mathbb{N}\langle M^*\rangle \to K$ the identity mapping, i.e., $\Phi(\mu) = \mu$ for all μ. We show that $\mathbb{E}$ is not qualitative, i.e., that there exists an alphabet Σ and a quantitative language $\mathbb{L} : \Sigma^+ \to K$ for which the implication (i) ⇒ (ii) does not hold.

Indeed, let $\Sigma = \{a\}$ be a singleton alphabet and let $\mathbb{L} : \Sigma^+ \to \mathbb{N}\langle M^*\rangle$ be defined for all $w \in \Sigma^+$ as $\mathrm{supp}(\mathbb{L}(w)) = \{b^{|w|}\}$ and $\mathbb{L}(w)(b^{|w|}) = |w|$. Clearly, $\mathbb{L}$ is recognized by the multi-weighted automaton $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ over Σ and $\mathbb{E}$ where:

- $Q = \{1, 2\}$, $I = \{1\}$, $F = \{2\}$;
- $T = \{(1, a, 1), (1, a, 2), (2, a, 2)\}$ and $\mathrm{wt}(t) = m$ for all $t \in T$.

Now suppose that there exist an alphabet $X \subseteq M$ (and hence $X = M$) and a finite automaton $\mathcal{B}$ over $\Sigma \times X$ such that $\mathbb{L} = \Phi \circ (L(\mathcal{B}))^\bullet = (\mathcal{L}(\mathcal{B}))^\bullet$. Then, for all $w \in \Sigma^+$, either $\mathbb{L}(w) = \emptyset$ or $\mathbb{L}(w) = [b^{|w|}]$. A contradiction. □

We say that an evaluator $\mathbb{E} = (M, K, \Phi)$ is *idempotent* if, for all $\mu_1, \mu_2 \in \mathbb{N}\langle M^*\rangle$ with $\mathrm{supp}(\mu_1) = \mathrm{supp}(\mu_2)$, we have $\Phi(\mu_1) = \Phi(\mu_2)$.

Example 4.11. *(a) Let $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ be an* idempotent *semiring, i.e., $\mathbb{1} + \mathbb{1} = \mathbb{1}$. The tropical semiring* $\mathrm{Trop} = (\mathbb{Q} \cup \{\infty\}, \min, +, \infty, 0)$ *is an example of an idempotent semiring. Clearly, the evaluator $\mathbb{E}^{\mathbb{S}}$ generated by an idempotent semiring $\mathbb{S}$ is idempotent.*

(b) Let $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ be a non-idempotent *semiring, i.e., $\mathbb{1} + \mathbb{1} \neq \mathbb{1}$. The* probabilistic semiring $\mathrm{Prob} = (\mathbb{Q}_{\geq 0}, +, \cdot, 0, 1)$ *of non-negative real numbers is an example of a non-idempotent semiring. We show that the evaluator $\mathbb{E}^{\mathbb{S}} = (S, S, \Phi^{\mathbb{S}})$ generated by a non-idempotent semiring $\mathbb{S}$ is also non-idempotent. Indeed, consider the finite multisets $\mu_1, \mu_2 \in \mathbb{N}\langle S^*\rangle$ with* $\mathrm{supp}(\mu_1) = \mathrm{supp}(\mu_2) = \{\mathbb{1}\}$, $\mu_1(\mathbb{1}) = 1$ *and* $\mu_2(\mathbb{1}) = 2$*. Then, $\Phi^{\mathbb{S}}(\mu_1) = \mathbb{1} \neq \mathbb{1} + \mathbb{1} = \Phi^{\mathbb{S}}(\mu_2)$.*

(c) The evaluators Φ^{Ratio} of Example 2.3, $\Phi^{\mathrm{Knap}(\eta)}$ of Example 2.4, and Φ^{Disc} of Example 2.5 are idempotent.

(d) The evaluators Φ^{Avg} of Example 2.6 and $\Phi^{\mathrm{Disp}(n)}$ of Example 2.7 are non-idempotent.

(e) The evaluator Φ from the proof of Lemma 4.10 is non-idempotent.

Lemma 4.12. *Let $\mathbb{E} = (M, K, \Phi)$ be an idempotent evaluator. Then $\mathbb{E}$ is qualitative.*

Proof. Let Σ be an alphabet and $\mathbb{L} : \Sigma^+ \to K$ a quantitative language. Lemma 4.9 shows that (ii) implies (i). Then, it remains to show that (i) implies (ii). Let $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ be a multi-weighted automaton over Σ and $\mathbb{E}$ with $[\![\mathcal{A}]\!] = \mathbb{L}$. Let $X = \textsc{Const}(\mathcal{A})$ and $\mathcal{A}' = (Q, I, T', F)$ be the nondeterministic finite automaton over the alphabet $\Sigma \times X$ with

$$T' = \{(q, (a, m), q') \mid t := (q, a, q') \in T \text{ and } \mathrm{wt}(t) = m\}.$$

Since Φ is idempotent, we have $[\![\mathcal{A}]\!] = \Phi \circ (\mathcal{L}(\mathcal{A}'))^\bullet$. $\square$

In the rest of this section, we will answer the following question: is there a qualitative evaluator which is not idempotent? We show that the class of non-idempotent infinite *semifields* is contained in the class of qualitative evaluators. A *semifield* is a semiring $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ such that $(S \setminus \{\mathbb{0}\}, \cdot, \mathbb{1})$ is a commutative group. An example of a non-idempotent infinite semifield is the probabilistic semiring $\textsc{Prob} = (\mathbb{Q}_{\geq 0}, +, \cdot, 0, 1)$. Let $x, y \in S$ with $y \neq \mathbb{0}$. We will write $\frac{x}{y}$ for $x \cdot y^{-1}$.

Theorem 4.13. *Let $\mathbb{S}$ be an infinite semifield. Then, the evaluator $\mathbb{E}^{\mathbb{S}}$ generated by $\mathbb{S}$ is qualitative.*

In the rest of this section, we will give a proof of this theorem. Let $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$. By Lemma 4.9, it suffices to show that (i) implies (ii), i.e., we show the following.

Lemma 4.14. *Let Σ be an alphabet and $\mathbb{L} : \Sigma^+ \to S$ a quantitative language recognizable over $\mathbb{S}$. Then, there exists an alphabet $X \subseteq S$ and a recognizable language $\mathcal{L} \subseteq (\Sigma \times S)^+$ such that $\mathbb{L} = \Phi^{\mathbb{S}} \circ \mathcal{L}^\bullet$.*

The key idea of our proof is to transform a weighted automaton $\mathcal{A}$ over $\mathbb{S}$ to an equivalent weighted automaton where, for a given word w, the sequences of weights of runs on w are pairwise distinct. For our construction, we will use Theorem 4.1 to transform $\mathcal{A}$ to a rational expression. Then, we will construct the desired weighted automaton by induction on the structure of this weighted rational expression.

Let $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ be a weighted automaton over Σ and $\mathbb{S}$. We call $\mathcal{A}$ *normalized* if:

- $I = \{i\}$ for some $i \in Q$ and, for each $(p, a, q) \in T$, $q \neq i$;
- $F = \{f\}$ for some $f \in Q \setminus \{i\}$ and, for each $(p, a, q) \in T$, $p \neq f$;
- for all $p, q, q' \in Q$ and $a \in \Sigma$ with $(p, a, q), (p, a, q') \in T$ and $q \neq q'$, we have: $\mathrm{wt}(p, a, q) \neq \mathrm{wt}(p, a, q')$;
- $\mathrm{wt}(t) \neq \mathbb{0}$ for all $t \in T$.

Example 4.15. *Let $\Sigma = \{a, b, c\}$ and $\mathbb{S} = \textsc{Prob}$ be the probabilistic semiring. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be the weighted automata over Σ and $\mathbb{S}$ depicted in Figure 4.1. Then, $\mathcal{A}_1$ is normalized and $\mathcal{A}_2$ is not normalized, since the transitions $(2, b, 3)$ and $(2, b, 4)$ have the same weight.*

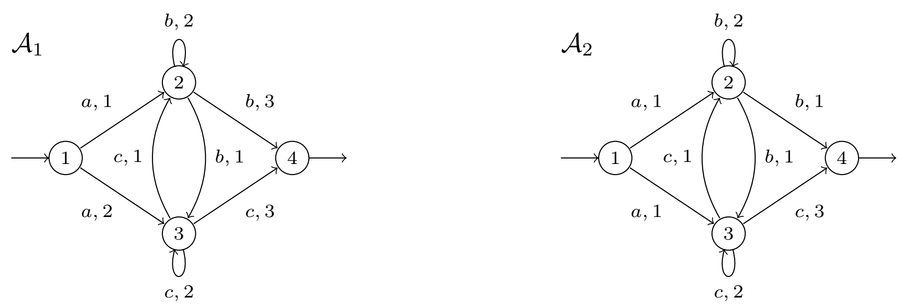

Figure 4.1: Weighted automata $\mathcal{A}_1$ and $\mathcal{A}_2$ from Example 4.15

For an alphabet Γ and a language $\mathcal{L} \subseteq \Gamma^+$, let $\mathrm{char}(\mathcal{L}) : \Sigma^+ \to \mathbb{N}$ be defined for all $w \in \Gamma^+$ as

$$\mathrm{char}(\mathcal{L})(w) = \begin{cases} 1, & \text{if } w \in \mathcal{L}, \\ 0, & \text{otherwise.} \end{cases}$$

Lemma 4.16. *Let $\mathcal{A}$ be a normalized weighted automaton over Σ and $\mathbb{S}$. Then, there exists a finite alphabet $X \subseteq S$ and a recognizable language $\mathcal{L} \subseteq (\Sigma \times X)^+$ such that $[\![\mathcal{A}]\!] = \Phi^{\mathbb{S}} \circ \mathcal{L}^\bullet$.*

Proof. Let $\mathcal{A} = (Q, \{i\}, T, \{f\}, \mathrm{wt})$ and $w = a_1 ... a_n \in \Sigma^+$. We show that $\mathrm{wt}^{\#}_{\mathcal{A}}|_{\mathrm{Run}_{\mathcal{A}}(w)}$ is an injective mapping, i.e., for all $\rho, \rho' \in \mathrm{Run}_{\mathcal{A}}(w)$ we have:

$$\rho \neq \rho' \;\Rightarrow\; \mathrm{wt}^{\#}_{\mathcal{A}}(\rho) \neq \mathrm{wt}^{\#}_{\mathcal{A}}(\rho').$$

Indeed, let $\rho = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} ... \xrightarrow{a_n} q_n$ and $\rho' = q'_0 \xrightarrow{a_1} q'_1 \xrightarrow{a_2} ... \xrightarrow{a_n} q'_n$ be such that $\mathrm{wt}^{\#}_{\mathcal{A}}(\rho) = s_1 ... s_n$ and $\mathrm{wt}^{\#}_{\mathcal{A}'}(\rho) = s'_1 ... s'_n$. Note that $q_0 = q'_0 = i$. Then, since $\rho \neq \rho'$, there exists $k \in \{0, ..., n-1\}$ such that $q_k = q'_k$ and $q_{k+1} \neq q'_{k+1}$. Then, since $\mathcal{A}$ is normalized and $(q_k, a_{k+1}, q_{k+1}), (q_k, a_{k+1}, q'_{k+1}) \in T$, we have: $s_{k+1} \neq s'_{k+1}$. Thus, $\mathrm{wt}^{\#}_{\mathcal{A}}(\rho) \neq \mathrm{wt}^{\#}_{\mathcal{A}}(\rho')$.

Let $X = \mathrm{wt}(T)$. Consider the finite automaton $\mathcal{B} = (Q, \{i\}, T', \{f\})$ over the alphabet $\Sigma \times X$ where $T' = \{(p, (a, s), q) \mid (p, a, q) \in T \text{ and } s = \mathrm{wt}(p, a, q)\}$. Let $\mathcal{L} = \mathcal{L}(\mathcal{A})$. Then, for all $w \in \Sigma^+$:

$$\mathcal{L}^\bullet(w) = \mathrm{char}(\mathrm{wt}^{\#}_{\mathcal{A}}(\mathrm{Run}_{\mathcal{A}}(w))) \overset{(!)}{=} \mathrm{wt}^{\#}_{\mathcal{A}}[\mathrm{Run}_{\mathcal{A}}(w)].$$

Here, at the place (!), we apply the fact that $\mathrm{wt}^{\#}_{\mathcal{A}}|_{\mathrm{Run}_{\mathcal{A}}(w)}$ is injective. Then, $\Phi \circ \mathcal{L}^\bullet = [\![\mathcal{A}]\!]$. □

Next, given a weighted automaton $\mathcal{A}$ over Σ and $\mathbb{S}$, we construct an equivalent normalized automaton. We proceed as follows. First, using the Kleene-Schützenberger Theorem 4.1 for $\mathbb{S}$, we can construct a weighted rational expression $e \in \textsc{Rat}(\Sigma, \mathbb{S})$ with $[\![e]\!] = [\![\mathcal{A}]\!]$. Then, we will construct a normalized weighted automaton for $[\![\mathcal{A}]\!]$ by induction on the structure of e.

Lemma 4.17. *(a) The quantitative language $\mathbb{0}_{\Sigma^+}$ is recognizable by a normalized weighted automaton over Σ and $\mathbb{S}$.*

(b) Let $s \in \mathbb{S} \setminus \{\mathbb{0}\}$ and $a \in \Sigma$. Then, the quantitative language $sa : \Sigma^+ \to \mathbb{S}$ is recognizable by a normalized weighted automaton over Σ and $\mathbb{S}$.

Proof. (a) Let $\mathcal{A} = (\{1,2\}, \{1\}, \emptyset, \{2\}, \emptyset)$. Then, $[\![\mathcal{A}]\!] = \mathbb{0}_{\Sigma^+}$.

(b) Let $\mathcal{A} = (\{1,2\}, \{1\}, \{(1,a,2)\}, \{2\}, \mathrm{wt})$ with $\mathrm{wt}(1,a,2) = s$. Then, $[\![\mathcal{A}]\!] = sa$.

□

Lemma 4.18. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be normalized weighted automata over Σ and $\mathbb{S}$. Then, there exists a normalized weighted automaton $\mathcal{A}$ over Σ and $\mathbb{S}$ such that $[\![\mathcal{A}]\!] = [\![\mathcal{A}_1]\!] + [\![\mathcal{A}_2]\!]$.*

Proof. Let $\mathcal{A}_1 = (Q_1, \{i\}, T_1, \{f\}, \mathrm{wt}_1)$ and $\mathcal{A}_2 = (Q_2, \{i\}, T_2, \{f\}, \mathrm{wt}_2)$. We assume that $Q_1 \cap Q_2 = \{i, f\}$. First, we construct the auxiliary weighted automaton $\mathcal{A}' = (Q', \{i\}, T', \{f\}, \mathrm{wt}')$ over Σ and $\mathbb{S}$ with $[\![\mathcal{A}']\!] = [\![\mathcal{A}_1]\!] + [\![\mathcal{A}_2]\!]$ and $\mathrm{wt}'(T') \subseteq S \setminus \{\mathbb{0}\}$ as follows. We let:

- $Q' = Q_1 \cup Q_2$;
- $T' = (T_1 \setminus T_2) \cup (T_2 \setminus T_1) \cup \{t \in T_1 \cap T_2 \mid \mathrm{wt}_1(t) + \mathrm{wt}_2(t) \neq \mathbb{0}\}$;
- for all $t \in T'$:

$$\mathrm{wt}'(t) = \begin{cases} \mathrm{wt}_1(t), & \text{if } t \in T_1 \setminus T_2, \\ \mathrm{wt}_2(t), & \text{if } t \in T_2 \setminus T_1, \\ \mathrm{wt}_1(t) + \mathrm{wt}_2(t), & \text{otherwise.} \end{cases}$$

Note that $\mathcal{A}'$ is not necessarily normalized, since there can exist $q, r \in Q'$ and $a \in \Sigma$ such that $q \neq r$, $(i,a,q), (i,a,r) \in T'$ and $\mathrm{wt}(i,a,q) = \mathrm{wt}(i,a,r)$.

Next, we transform $\mathcal{A}'$ into an equivalent normalized weighted automaton. Let $\tilde{Q} = Q' \setminus \{i, f\}$, $X = T' \cap (\{i\} \times \Sigma \times \tilde{Q})$ and $\Delta : X \to S \setminus \{\mathbb{0}\}$. Consider the weighted automaton $\mathcal{A}_\Delta = (Q, \{i\}, T, \{f\}, \mathrm{wt}_\Delta)$ where:

- $Q = \{i, f\} \cup (\tilde{Q} \times X)$;
- $T = T_1 \cup T_2 \cup T_3 \cup T_4$ where

$$\begin{aligned} T_1 &= \{(i, a, (q,t)) \mid t = (i,a,q) \in X\}, \\ T_2 &= \{((q,t), a, (q',t)) \mid (q,a,q') \in T' \text{ and } t \in X\}, \\ T_3 &= \{((q,t), a, f) \mid (q,a,f) \in T' \text{ and } t \in X\}, \\ T_4 &= T' \cap (\{i\} \times \Sigma \times \{f\}); \end{aligned}$$

- $\mathrm{wt}_\Delta : T \to S \setminus \{\mathbb{0}\}$ is defined as follows (depending on Δ):
  - for $(i, a, (q,t)) \in T_1$, we let $\mathrm{wt}_\Delta(i, a, (q,t)) = \Delta(t)$;
  - for $((q,t), a, (q',t)) \in T_2$, we let $\mathrm{wt}_\Delta((q,t), a, (q',t)) = \mathrm{wt}'(q,a,q')$;
  - for $((q,t), a, f) \in T_3$, let $\mathrm{wt}_\Delta((q,t), a, f) = \frac{\mathrm{wt}'(t) \cdot \mathrm{wt}'(q,a,f)}{\Delta(t)}$;

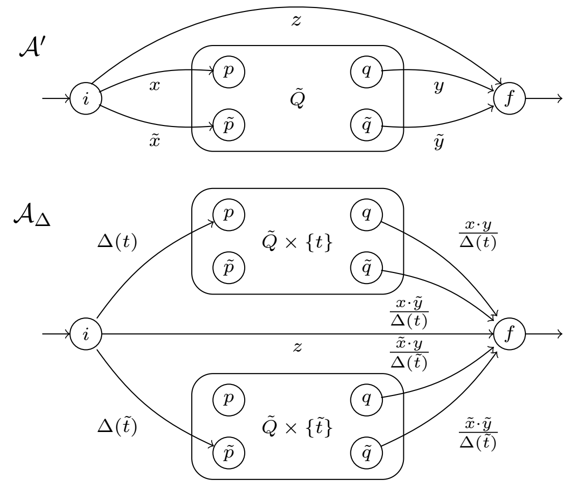

Figure 4.2: Construction of $\mathcal{A}_\Delta$ from $\mathcal{A}'$

– for $(i, a, f) \in T_4$, let $\mathrm{wt}_\Delta(i, a, f) = \mathrm{wt}'(i, a, f)$.

The idea of the construction is depicted in Figure 4.2. Here, we omit the labels of transitions.

Now we show that $[\![\mathcal{A}_\Delta]\!] = [\![\mathcal{A}']\!]$. Obviously, $[\![\mathcal{A}_\Delta]\!](a) = [\![\mathcal{A}']\!](a)$ for all $a \in \Sigma$. Now let $w = a_1 ... a_n \in \Sigma^+$ with $|w| \geq 2$. We construct a bijection $\varphi : \mathrm{Run}_{\mathcal{A}_\Delta}(w) \to \mathrm{Run}_{\mathcal{A}'}(w)$ as follows. Let $\rho \in \mathrm{Run}_{\mathcal{A}_\Delta}(w)$. Then, ρ has the form:

$$q_0 \xrightarrow{a_1} (q_1, t_\rho) \xrightarrow{a_2} (q_2, t_\rho) \xrightarrow{a_3} ... \xrightarrow{a_{n-1}} (q_{n-1}, t_\rho) \xrightarrow{a_n} q_n$$

where $q_0 = i$, $q_n = f$, $t_\rho = (i, a_1, q_1) \in X$, $q_1, ..., q_{n-1} \in \tilde{Q}$ and $t_i := (q_{i-1}, a_i, q_i) \in T'$ for all $1 \leq i \leq n$. Then, we define $\varphi(\rho)$ to be the run $t_1 ... t_n \in \mathrm{Run}_{\mathcal{A}'}(w)$. Since t_ρ is uniquely determined for all ρ, the mapping φ is bijective. Note that $t_\rho = t_1$. Since $\cdot$ is commutative, we have:

$$\begin{aligned}
\mathrm{wt}_{\mathcal{A}_\Delta}(\rho) &= \Delta(t_\rho) \cdot \mathrm{wt}'(t_2) \cdot ... \cdot \mathrm{wt}'(t_{n-1}) \cdot \frac{\mathrm{wt}'(t_\rho) \cdot \mathrm{wt}'(t_n)}{\Delta(t_\rho)} \\
&= \mathrm{wt}'(t_1) \cdot \mathrm{wt}'(t_2) \cdot ... \cdot \mathrm{wt}'(t_{n-1}) \cdot \mathrm{wt}'(t_n) = \mathrm{wt}_{\mathcal{A}'}(\varphi(\rho)).
\end{aligned}$$

Hence, φ is also weight-preserving. Then, $[\![\mathcal{A}_\Delta]\!](w) = [\![\mathcal{A}']\!](w)$.

Finally, we show that we can choose $\Delta : X \to S \setminus \{\mathbb{0}\}$ such that $\mathcal{A}_\Delta$ is normalized. Indeed, for $t \in X$, let

$$W_t = \bigcup_{\substack{q \in \tilde{Q}, a \in \Sigma, \\ (q,a,f) \in T'}} \left\{ \tfrac{\mathrm{wt}'(q,a,f) \cdot \mathrm{wt}'(t)}{\mathrm{wt}'(q,a,q')} \mid q' \in \tilde{Q} \text{ and } (q, a, q') \in T' \right\}$$

Let also $U = \{\mathrm{wt}'(i,a,f) \mid a \in \Sigma, (i,a,f) \in T'\}$ and $W = U \cup \bigcup_{t \in X} W_t$. Note that X and W are finite sets. Then, the set $S \setminus (W \cup \{\mathbb{0}\})$ is infinite and hence there exists an injective mapping $\tilde{\Delta} : X \to S \setminus (W \cup \{\mathbb{0}\})$. We show that, for all $p, q, r \in Q$ and all $a \in \Sigma$ with $q \neq r$ and $(p,a,q), (p,a,r) \in T$, we have: $\mathrm{wt}_{\tilde{\Delta}}(p,a,q) \neq \mathrm{wt}_{\tilde{\Delta}}(p,a,r)$. We prove it by contradiction. Assume that there exist $p, q, r \in Q$ and $a \in \Sigma$ with $q \neq r$, $(p,a,q), (p,a,r) \in T$ and $\mathrm{wt}_{\tilde{\Delta}}(p,a,q) = \mathrm{wt}_{\tilde{\Delta}}(p,a,r)$. We distinguish the following cases:

- $p = i$. Then, consider two subcases:
  - either $q = f$ or $r = f$. We may assume without loss of generality that $q = f$. Then, $r = (r', t') \in \tilde{Q} \times X$ where $t' = (p, a, r') \in X$, and $\mathrm{wt}_{\tilde{\Delta}}(p,a,q) = \mathrm{wt}'(i,a,f) \in U$. However, $\mathrm{wt}_{\tilde{\Delta}}(p,a,r) = \tilde{\Delta}(t') \notin U$. A contradiction.
  - $q \neq f$ and $r \neq f$. Then, $r = (r', t') \in \tilde{Q} \times X$ and $q = (q', t'') \in \tilde{Q} \times X$ where $t' = (p,a,r') \in X$ and $t'' = (p,a,q') \in X$. Note that $r' \neq q'$ (otherwise, $t' = (p,a,r') = (p,a,q') = t''$ and then $q = r$). Then, $t' \neq t''$. Since $\tilde{\Delta}$ is injective, we obtain $\tilde{\Delta}(t') \neq \tilde{\Delta}(t'')$. However,

    $$\tilde{\Delta}(t') = \mathrm{wt}_{\tilde{\Delta}}(p,a,r) = \mathrm{wt}_{\tilde{\Delta}}(p,a,q) = \tilde{\Delta}(t'').$$

    A contradiction.
- $p = (p', t') \in \tilde{Q} \times X$. Again, we consider two subcases:
  - either $q = f$ or $r = f$. We may assume without loss of generality that $q = f$. Then, $r = (r', t') \in \tilde{Q} \times X$ for some $r' \in \tilde{Q}$, $\mathrm{wt}_{\tilde{\Delta}}(p,a,q) = \frac{\mathrm{wt}'(t') \cdot \mathrm{wt}'(p',a,f)}{\tilde{\Delta}(t')}$ and $\mathrm{wt}_{\tilde{\Delta}}(p,a,r) = \mathrm{wt}'(p',a,r')$. Since $\mathrm{wt}_{\tilde{\Delta}}(p,a,q) = \mathrm{wt}_{\tilde{\Delta}}(p,a,r)$, we have:

    $$\tilde{\Delta}(t') = \frac{\mathrm{wt}'(p',a,f) \cdot \mathrm{wt}'(t')}{\mathrm{wt}'(p',a,r')} \in W_{t'} \subseteq W.$$

    However, $\tilde{\Delta}(t') \in S \setminus (W \cup \{\mathbb{0}\})$. A contradiction.
  - $q \neq f$ and $r \neq f$. Then, $r = (r', t') \in \tilde{Q} \times X$ and $q = (q', t') \in \tilde{Q} \times X$ where $(p',a,r'), (p',a,q') \in T'$ and $r' \neq q'$. Since $\mathrm{wt}_{\tilde{\Delta}}(p,a,q) = \mathrm{wt}_{\tilde{\Delta}}(p,a,r)$, we have $\mathrm{wt}'(p',a,q') = \mathrm{wt}'(p',a,r')$. Note that either $(p',a,q'), (p',a,r')$ are both in $T_1 \setminus T_2$ or $(p',a,q'), (p',a,r')$ are both in $T_2 \setminus T_1$. Since $\mathcal{A}_1$ and $\mathcal{A}_2$ are normalized, we obtain $\mathrm{wt}'(p',a,q') \neq \mathrm{wt}'(p',a,r')$. A contradiction.

□

Lemma 4.19. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be normalized weighted automata over Σ and $\mathbb{S}$. Then, the Cauchy product $[\![\mathcal{A}_1]\!] \cdot [\![\mathcal{A}_2]\!]$ is recognizable by a normalized weighted automaton $\mathcal{A}$ over Σ and $\mathbb{S}$.*

Proof. Let $\mathcal{A}_1 = (Q_1, \{i_1\}, T_1, \{f_1\}, \mathrm{wt}_1)$ and $\mathcal{A}_2 = (Q_2, \{i_2\}, T_2, \{f_2\}, \mathrm{wt}_2)$. Here, we use the usual construction for the Cauchy product. We assume that $f_1 = i_2$ and $Q_1 \cap Q_2 = \{f_1\}$. Since $\mathcal{A}_1$ and $\mathcal{A}_2$ are normalized, we have: $T_1 \cap T_2 = \emptyset$ and $f_1 \neq f_2$. We let $\mathcal{A} = (Q_1 \cup Q_2, \{i_1\}, T_1 \cup T_2, \{f_2\}, \mathrm{wt})$ where $\mathrm{wt}(t) = \mathrm{wt}_i(t)$ whenever $t \in T_i$ for $i = 1, 2$. Clearly, $\mathcal{A}$ is normalized. Moreover, $[\![\mathcal{A}]\!] = [\![\mathcal{A}_1]\!] \cdot [\![\mathcal{A}_2]\!]$. □

Lemma 4.20. *Let $\mathcal{A}$ be a normalized weighted automaton over Σ and $\mathbb{S}$. Then, there exists a normalized weighted automaton $\mathcal{A}^+$ over Σ and $\mathbb{S}$ such that $[\![\mathcal{A}^+]\!] = [\![\mathcal{A}]\!]^+$.*

Proof. Let $\mathcal{A} = (Q, \{i\}, T, \{f\}, \mathrm{wt})$. Let $\ell \notin Q$, $\tilde{Q} = Q \setminus \{i, f\}$ and $\Delta \in S \setminus \{\mathbb{0}, \mathbb{1}\}$. We construct the weighted automaton

$$\mathcal{A}^+_\Delta = (Q \cup \{\ell\}, \{i\}, T', \{f\}, \mathrm{wt}'_\Delta)$$

by letting:

- $T' = T \cup T_{\to\ell} \cup T_{\ell\circlearrowleft} \cup T_{\ell\to}$ where

$$\begin{aligned} T_{\to\ell} &= \{(q, a, \ell) \mid (q, a, f) \in T\}, \\ T_{\ell\circlearrowleft} &= \{(\ell, a, \ell) \mid (i, a, f) \in T\}, \\ T_{\ell\to} &= \{(\ell, a, q) \mid (i, a, q) \in T\}; \end{aligned}$$

- $\mathrm{wt}' : T' \to S$ is defined as follows:
  - for all $q \in \tilde{Q}$ and $a \in \Sigma$ with $(i, a, q) \in T$, we put $\mathrm{wt}'_\Delta(i, a, q) = \Delta \cdot \mathrm{wt}(i, a, q)$ and $\mathrm{wt}'_\Delta(\ell, a, q) = \mathrm{wt}(i, a, q)$;
  - for all $q \in \tilde{Q}$ and $a \in \Sigma$ with $(q, a, f) \in T$, we put $\mathrm{wt}'_\Delta(q, a, f) = \frac{\mathrm{wt}(q,a,f)}{\Delta}$ and $\mathrm{wt}'_\Delta(q, a, \ell) = \mathrm{wt}(q, a, f)$;
  - for all $p, q \in \tilde{Q}$ and $a \in \Sigma$ with $(p, a, q) \in T$, we put $\mathrm{wt}'_\Delta(p, a, q) = \mathrm{wt}(p, a, q)$;
  - for all $a \in \Sigma$ with $(i, a, f) \in T$, we put: $\mathrm{wt}'_\Delta(i, a, f) = \mathrm{wt}'_\Delta(\ell, a, \ell) = \mathrm{wt}(i, a, f)$, $\mathrm{wt}'_\Delta(i, a, \ell) = \Delta \cdot \mathrm{wt}(i, a, f)$ and $\mathrm{wt}'_\Delta(\ell, a, f) = \frac{\mathrm{wt}(i,a,f)}{\Delta}$.

The idea of the construction is depicted in Figure 4.3. Here, we omit the labels of transitions. It is not difficult to see that $[\![\mathcal{A}^+_\Delta]\!] = [\![\mathcal{A}]\!]$. It remains to show that we can choose $\Delta \in S \setminus \{\mathbb{0}, \mathbb{1}\}$ such that $\mathcal{A}^+_\Delta$ is normalized. Let

$$W = \bigcup_{\substack{q \in \tilde{Q} \cup \{i\}, a \in \Sigma, \\ (q,a,f) \in T}} \left\{ \tfrac{\mathrm{wt}(q,a,f)}{\mathrm{wt}(q,a,q')} \mid q' \in \tilde{Q}, (q, a, q') \in T \right\}.$$

Since W is finite, there exists $\tilde{\Delta} \in S \setminus (W \cup \{\mathbb{0}, \mathbb{1}\})$. We show that, for all $p, q, r \in Q \cup \{\ell\}$ and $a \in \Sigma$ with $(p, a, q), (p, a, r) \in T'$ and $q \neq r$, we have: $\mathrm{wt}'_{\tilde{\Delta}}(p, a, q) \neq \mathrm{wt}'_{\tilde{\Delta}}(p, a, r)$. Suppose that this does not hold. Then, there exist $p, q, r \in Q \cup \{\ell\}$ and $a \in \Sigma$ such that $(p, a, q), (p, a, r) \in T'$, $q \neq r$ and $\mathrm{wt}'_{\tilde{\Delta}}(p, a, q) = \mathrm{wt}'_{\tilde{\Delta}}(p, a, r)$. We distinguish the following cases:

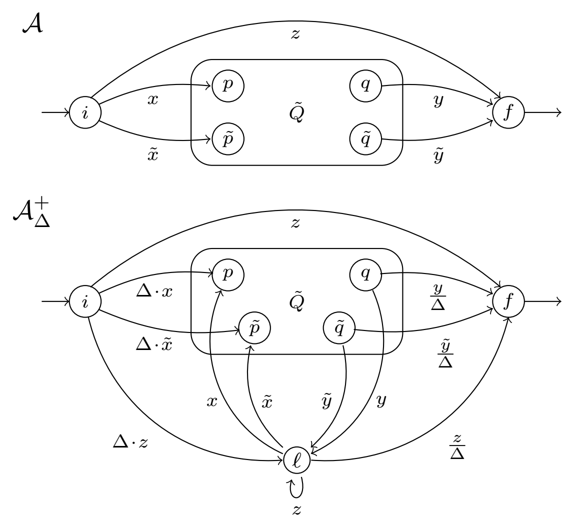

Figure 4.3: Construction of $\mathcal{A}^{+}_{\Delta}$ from $\mathcal{A}$

- $p = i$. Suppose that $q, r \in \tilde{Q} \cup \{\ell\}$. Then, for $h \in \{q, r\}$, we have:

$$\mathrm{wt}'_{\tilde{\Delta}}(i, a, h) = \begin{cases} \tilde{\Delta} \cdot \mathrm{wt}(i, a, h), & \text{if } h \neq \ell, \\ \tilde{\Delta} \cdot \mathrm{wt}(i, a, f), & \text{otherwise.} \end{cases}$$

  Then, the equality $\mathrm{wt}'_{\tilde{\Delta}}(i, a, q) = \mathrm{wt}'_{\tilde{\Delta}}(i, a, r)$ implies the equality $\mathrm{wt}(i, a, q') = \mathrm{wt}(i, a, r')$ for some $q', r' \in Q$ with $q' \neq r'$. Then, $\mathcal{A}$ is not normalized. A contradiction. Then, either $q = f$ or $r = f$. We may assume that $r = f$. Since $q \neq r$, there are two possibilities for q:

  - $q = \ell$. Then, $\mathrm{wt}(i, a, f) = \tilde{\Delta} \cdot \mathrm{wt}(i, a, f)$ and hence $\tilde{\Delta} = \mathbb{1}$ which is impossible.
  - $q \in \tilde{Q}$. Then, $\mathrm{wt}(i, a, f) = \tilde{\Delta} \cdot \mathrm{wt}(i, a, q)$ and hence $\tilde{\Delta} = \frac{\mathrm{wt}(i,a,f)}{\mathrm{wt}(i,a,q)} \in W$ which is also impossible.

  Thus, the case $p = i$ is impossible.
- $p = \ell$. Again, suppose that $q, r \in \tilde{Q} \cup \{\ell\}$. Then, for $h \in \{q, r\}$:

$$\mathrm{wt}'_{\tilde{\Delta}}(\ell, a, h) = \begin{cases} \mathrm{wt}(i, a, h), & \text{if } h \in \tilde{Q}, \\ \mathrm{wt}(i, a, f), & \text{if } h = \ell. \end{cases}$$

  Hence, there exist $q', r' \in Q$ with $q' \neq r'$ and $\mathrm{wt}(i, a, q') = \mathrm{wt}(i, a, r')$ which is impossible since $\mathcal{A}$ is normalized. A contradiction. Then, either $q = f$ or $r = f$. Again, we may assume that let $r = f$. Then, there are two possibilities for q:

  - $q = \ell$. Then, $\mathrm{wt}(i, a, f) = \frac{\mathrm{wt}(i,a,f)}{\tilde{\Delta}}$ and hence $\tilde{\Delta} = \mathbb{1}$ which is impossible.
  - $q \in \tilde{Q}$. Then, $\mathrm{wt}(i, a, q) = \frac{\mathrm{wt}(i,a,f)}{\tilde{\Delta}}$ and hence $\tilde{\Delta} = \frac{\mathrm{wt}(i,a,f)}{\mathrm{wt}(i,a,q)} \in W$ which is also not possible.

  This, the case $p = \ell$ is also impossible.
- $p \in \tilde{Q}$. Again, since $\mathcal{A}$ is normalized, it is not possible that $q, r \in \tilde{Q} \cup \{\ell\}$. Assume w.l.o.g. that $r = f$. Then, there are two possibilities for q:
  - $q = \ell$. Then, $\mathrm{wt}(p, a, f) = \frac{\mathrm{wt}(p,a,f)}{\tilde{\Delta}}$ and hence $\tilde{\Delta} = \mathbb{1}$ which is impossible.
  - $q \in \tilde{Q}$. Then, $\mathrm{wt}(p, a, q) = \frac{\mathrm{wt}(p,a,f)}{\tilde{\Delta}}$ and hence $\tilde{\Delta} = \frac{\mathrm{wt}(p,a,f)}{\mathrm{wt}(p,a,q)} \in W$. A contradiction.

  Thus, the case $p \in \tilde{Q}$ is impossible.
- $p = f$. Clearly, this case is also impossible, since f does not have outgoing transitions.

Then, we put $\mathcal{A}^+ = \mathcal{A}^+_{\tilde{\Delta}}$. □

Summarizing Lemmas 4.17, 4.18, 4.19, 4.20 and Theorem 4.1, we obtain:

Lemma 4.21. *Let $\mathcal{A}$ be a weighted automaton over Σ and $\mathbb{S}$. Then, there exists a normalized weighted automaton $\mathcal{A}'$ over Σ and $\mathbb{S}$ with $[\![\mathcal{A}']\!] = [\![\mathcal{A}]\!]$.*

Proof of Theorem 4.13. Immediate by Lemmas 4.9, 4.16 and 4.21. □

Chapter 5

Multi-weighted ω-automata

Contents

The goal of this chapter is to study multi-weighted automata on infinite words. We introduce a general framework for multi-weighted Büchi automata and give a Nivat-like decomposition theorem for them. As a corollary, we obtain a Nivat decomposition theorem for *unambiguous* multi-weighted Büchi automata. As a first application of our Nivat theorem, we show that multi-weighted Büchi automata and multi-weighted Muller automata are expressively equivalent. Finally, we motivate our new model of multi-weighted Büchi automata by the computability of the *discount-optimal value problem.*

5.1 A general framework and examples

Let Γ be a set (possibly infinite). An ω-*word* over Γ is an infinite sequence $(\gamma_i)_{i\in\mathbb{N}}$ where $\gamma_i \in \Gamma$ for all $i \in \mathbb{N}$. Let Γ^ω denote the set of all ω-words over Γ. Any set $\mathcal{L} \subseteq \Gamma^\omega$ is called an ω-*language* over Γ.

Let Σ be an alphabet. A *Büchi automaton* over Σ is a tuple $\mathcal{A} = (Q, I, T, F)$ where Q is a finite set of states, $I, F \subseteq Q$ are sets of initial resp. accepting states, and $T \subseteq Q \times \Sigma \times Q$ is a transition relation.

A *run* $\rho = (t_i)_{i\in\mathbb{N}}$ of $\mathcal{A}$ is defined as an infinite sequence of matching transitions, say $t_i = (q_i, a_i, q_{i+1})$ for each $i \in \mathbb{N}$, such that $q_0 \in I$ and $\{q \in Q \mid q = q_i \text{ for infinitely many } i \in \mathbb{N}\} \cap F \neq \emptyset$. Let $\mathrm{label}(\rho) := (a_i)_{i\in\mathbb{N}} \in \Sigma^\omega$, the *label* of ρ. As in the case of finite words, we denote by $\mathrm{Run}_\mathcal{A}$ the set of all runs of $\mathcal{A}$ and, for each $w \in \Sigma^\omega$, we denote by $\mathrm{Run}_\mathcal{A}(w)$ the set of all runs ρ of $\mathcal{A}$ with $\mathrm{label}(\rho) = w$. Let $\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^\omega \mid \mathrm{Run}_\mathcal{A}(w) \neq \emptyset\}$, the ω-language *accepted* by $\mathcal{A}$. We say that an ω-language $\mathcal{L} \subseteq \Sigma^\omega$ is *recognizable* if there exists a Büchi automaton $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.

Next, we will consider *multi-weighted Büchi automata*. Again, the weights of transitions can be tuples of weights (e.g., reward-cost pairs), and the values of the behavior should be single values. Our algebraic structure for multi-weighted Büchi automata will be defined in the spirit of *ω-valuation monoids* of Droste and Meinecke [40].

We say that a monoid $\mathbb{K} = (K, +, \mathbb{0})$ is *complete* (cf. [35]) if it is equipped with infinitary sum operations $\sum_I : K^I \to K$, for any index set I, such that, for all I and all families $(k_i)_{i \in I}$ of elements of K, the following hold:

- $\sum_{i \in \emptyset} k_i = \mathbb{0}$, $\sum_{i \in \{j\}} k_i = k_j$, $\sum_{i \in \{p,q\}} k_i = k_p + k_q$ for $p \neq q$;
- $\sum_{j \in J}(\sum_{i \in I_j} k_i) = \sum_{i \in I} k_i$, if $\bigcup_{j \in J} I_j = I$ and $I_j \cap I_{j'} = \emptyset$ for $j \neq j'$.

Definition 5.1. *An* ω-valuation structure $\mathbb{V} = (M, \mathbb{K}, \mathrm{val}^\omega)$ *consists of a non-empty set M, a complete monoid $\mathbb{K} = (K, +, \mathbb{0})$, and a mapping* $\mathrm{val}^\omega : M^\omega \to K$ *called an* ω-valuation function.

Note that, in contrast to an *ω-valuation monoid* of [40], the domain and the codomain of the ω-valuation function val^ω of the *ω-valuation structure* $\mathbb{V}$ do not necessarily coincide; moreover, val^ω is not equipped with additional properties as in [40]. Then, ω-valuation structures are more general than ω-valuation monoids.

Definition 5.2. *Let Σ be an alphabet and $\mathbb{V} = (M, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ an ω-valuation structure. A* multi-weighted Büchi automaton *(MWBA) over Σ and $\mathbb{V}$ is a tuple $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ where (Q, I, T, F) is a Büchi automaton over Σ and $\mathrm{wt} : T \to M$ is a transition weight function.*

The behavior of MWBA is defined using a similar approach as in [40]. First, given a run ρ of this automata, we evaluate the ω-sequence of transition weights of ρ (which is in M^ω) using the ω-valuation function val^ω and then resolve the nondeterminism on the weights of runs using the complete monoid $\mathbb{K}$. Formally, let $\rho = (t_i)_{i \in \mathbb{N}} \in \mathrm{Run}_\mathcal{A}$ be a run of $\mathcal{A}$ where $t_i \in T$ for all $i \in \mathbb{N}$. Then, the *weight* of ρ is defined as $\mathrm{wt}_\mathcal{A}(\rho) = \mathrm{val}^\omega((\mathrm{wt}(t_i))_{i \in \mathbb{N}}) \in K$. The *behavior* of $\mathcal{A}$ is a mapping $[\![\mathcal{A}]\!] : \Sigma^\omega \to K$ defined for all $w \in \Sigma^\omega$ by

$$[\![\mathcal{A}]\!](w) = \sum(\mathrm{wt}_\mathcal{A}(\rho) \mid \rho \in \mathrm{Run}_\mathcal{A}(w)).$$

Note that the sum in the equation above can be infinite. Therefore we consider a complete monoid $(K, +, \mathbb{0})$.

A mapping $\mathbb{L} : \Sigma^\omega \to K$ is called a *quantitative ω-language*. We say that $\mathbb{L}$ is *recognizable* over $\mathbb{V}$ if there exists a MWBA $\mathcal{A}$ over Σ and $\mathbb{E}$ such that $[\![\mathcal{A}]\!] = \mathbb{L}$.

Remark 5.3. *As in the case of multi-weighted automata on finite words, we could consider a more general algebraic structure given by an* ω-aggregation function *which evaluates infinite multisets of ω-sequences. However, this approach has the following disadvantages:*

- *This approach is not motivated by examples of MWBA known from literature.*
- *The use of this abstract structure would diminish the clarity of presentation (since we have to consider cardinal arithmetic for infinite multisets).*
- *The use of finite multisets of finite strings was interesting for finite words, since we defined the Cauchy-product on them and obtained a semiring. In contrast, there is no natural concatenation operation on infinite sequences.*

Therefore, we will use the less general ω-valuation structure to describe the behavior of MWBA.

Now we show that various models of multi-weighted automata on infinite words can be described using ω-valuation structures.

Example 5.4. *(a) Here, we consider the reward-cost ratio setting for infinite words (cf. [21, 22]). We can handle this situation using the ω-valuation structure $\mathbb{V}^{\omega\text{Ratio}} = (M, \mathbb{K}, \mathrm{val}^{\omega\text{Ratio}})$ where $M = \mathbb{Q} \times \mathbb{Q}_{\geq 0}$ models the reward-cost pairs, $\mathbb{K} = (\mathbb{R} \cup \{-\infty, \infty\}, \sup, -\infty)$, and $\mathrm{val}^{\omega\text{Ratio}} : M^\omega \to K$ is defined as follows. For a sequence $u = (m_i)_{i \in \mathbb{N}} \in M^\omega$ with $m_i = (r_i, c_i)$, we let*

$$\mathrm{val}^{\omega\text{Ratio}}(u) = \limsup_{n\to\infty} \frac{r_1 + ... + r_n}{c_1 + ... + c_n}$$

with $\frac{r}{0} = \infty$.

(b) Here, we consider the transition-dependent discounting measure (cf. [8, 37]). In order to fit this weight measure into the setting of multi-weighted Büchi automata, we consider the ω-valuation structure $\mathbb{V}^{\omega\text{Disc}} = (M, \mathbb{K}, \mathrm{val}^{\omega\text{Disc}})$ where $M = \mathbb{Q}_{\geq 0} \times ((0,1] \cap \mathbb{Q})$ models the pairs of a cost and a discounting factor, $\mathbb{K} = (\mathbb{R}_{\geq 0} \cup \{\infty\}, \inf, \infty)$, and $\mathrm{val}^{\omega\text{Disc}}$ is defined for all $u = (m_i)_{i \in \mathbb{N}} \in M^\omega$ with $m_i = (c_i, d_i)$ as

$$\mathrm{val}^{\omega\text{Disc}}(u) = \sum_{i=0}^{\infty} c_i \cdot \prod_{j=0}^{i-1} d_j$$

where $\prod_{j\in\emptyset} d_j = 1$.

(c) Here, we present ω-evaluators for the model of multi-weighted automata which correspond to one-player energy games considered in [52]. Let $n \geq 1$, $s_1, ..., s_n$ be energy storages and $E^1_{\max}, ..., E^n_{\max} \in \mathbb{N}$ their maximal capacities. We start with empty storages and, along a run, the energy level of each storage s_j can be increased (if we regain energy) or decreased (if we consume energy). If the energy level of a storage s_j exceeds its maximal capacity $E^j_{\max}$, then we trim it to $E^j_{\max}$. The goal is to keep the energy level of every energy storage not less that zero. Consider the sequence $u = (u_i)_{i\in\mathbb{N}}$ where, for all $i \in \mathbb{N}$, $u_i = (u^1_i, ..., u^n_i) \in \mathbb{Z}^n$ is a vector of the energy level changes for each storage. We transform this sequence to the sequence $\tilde{u} = (\tilde{u}_i)_{i\in\mathbb{N}}$ of the absolute energy levels $\tilde{u}_i = (\tilde{u}^1_i, ..., \tilde{u}^n_i) \in \mathbb{Z}^n$ defined inductively on $i \geq 0$ as follows. For $i = 0$ and $j \in \{1, ..., n\}$, let $\tilde{u}^j_i = \min\{u^j_i, E^j_{\max}\}$. Then, for $i \geq 1$ and

$j \in \{1, ..., n\}$, we let $\tilde{u}_i^j = \min\{\tilde{u}_{i-1}^j + u_i^j, E_{\max}^j\}$. Note that, for all $i \in \mathbb{N}$ and $j \in \{1, ..., n\}$, $\tilde{u}_i^j \in (-\infty, E_{\max}^j)$. We say that u is correct *if $\tilde{u}_i^j \geq 0$ for all $i \in \mathbb{N}$ and $j \in \{1, ..., n\}$. Then, for this situation we consider the ω-valuation structure $\mathbb{V}^{\omega\text{Energy}} = (M, \mathbb{K}, \mathrm{val}^{\omega\text{Energy}})$ where $M = \mathbb{Z}^n$, $\mathbb{K} = (\{0,1\}, \vee, 0)$, and $\mathrm{val}^{\omega\text{Energy}}$ is defined for all $u \in M^\omega$ as*

$$\mathrm{val}^{\omega\text{Energy}}(u) = \begin{cases} 1, & \textit{if } u \textit{ is correct}, \\ 0, & \textit{otherwise.} \end{cases}$$

(d) *Since an ω-valuation monoid $(K, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ of Droste and Meinecke [40] is a special case of ω-valuation structures, all examples considered there also fit into our new framework.* □

5.2 A Nivat-like characterization

Nivat's theorem [74] (see also [12], Theorem 4.1) is one of the fundamental characterizations of rational transductions and establishes a connection between rational transductions and rational languages. A version for semiring-weighted automata was given in [38]; this shows a connection between recognizable quantitative and qualitative languages. A version for weighted multioperator tree automata was given in [82]. In this chapter, we prove a Nivat-like decomposition theorem for recognizable *quantitative* ω-languages, i.e., we show how they are related to *qualitative* ω-regular languages. As a corollary from the proof this theorem, we give a Nivat-like characterization of *unambiguously* recognizable quantitative ω-languages.

Let Σ be an alphabet and $\mathbb{V} = (M, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ an ω-valuation structure. For a (possibly different from Σ) alphabet Γ, we introduce the following operations:

- Let Δ be an arbitrary non-empty set and $h : \Gamma \to \Delta$ a mapping, called henceforth a *renaming*. For any ω-word $u = (\gamma_i)_{i \in \mathbb{N}} \in \Gamma^\omega$, we let $h(u) = (h(\gamma_i))_{i \in \mathbb{N}} \in \Delta^\omega$.
- Let $h : \Gamma \to \Sigma$ be a renaming and $\mathbb{L} : \Gamma^\omega \to K$ a quantitative ω-language. We define the *renaming* $h(\mathbb{L}) : \Sigma^\omega \to K$ for all $w \in \Sigma^\omega$ by

$$h(\mathbb{L})(w) = \sum (\mathbb{L}(u) \mid u \in \Gamma^\omega \text{ and } h(u) = w).$$

  Note that the sum in the equation above can be infinite.
- Let $g : \Gamma \to M$ be a renaming. The *composition* $\mathrm{val}^\omega \circ g : \Gamma^\omega \to K$ is defined for all $u \in \Gamma^\omega$ as $(\mathrm{val}^\omega \circ g)(u) = \mathrm{val}^\omega(g(u))$.
- Let $\mathbb{L} : \Gamma^\omega \to K$ be a quantitative ω-language and $\mathcal{L} \subseteq \Gamma^\omega$ an ω-language. Then, the *intersection* $(\mathbb{L} \cap \mathcal{L}) : \Gamma^\omega \to K$ is defined for all $u \in \Gamma^\omega$ as

$$(\mathbb{L} \cap \mathcal{L})(u) = \begin{cases} \mathbb{L}(u), & \text{if } u \in \mathcal{L}, \\ \mathbb{0}, & \text{otherwise.} \end{cases}$$

We say that a Büchi automaton $\mathcal{A}$ over Γ is *unambiguous* if, for every ω-word $u \in \Gamma^\omega$, we have $|\operatorname{Run}_{\mathcal{A}}(u)| \leq 1$.

Theorem 5.5 (Carton, Michel [26]). *Let Γ be an alphabet and $\mathcal{L} \subseteq \Gamma^\omega$ an ω-language. Then, $\mathcal{L}$ is recognizable iff there exists an unambiguous Büchi automaton $\mathcal{A}$ over Γ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.*

Now we state our Nivat theorem for MWBA .

Theorem 5.6. *Let Σ be an alphabet, $\mathbb{V} = (M, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ an ω-valuation structure, and $\mathbb{L} : \Sigma^\omega \to K$ a quantitative ω-language. Then, the following are equivalent.*

(a) *$\mathbb{L}$ is recognizable over $\mathbb{V}$.*
(b) *There exist an alphabet Γ, renamings $h : \Gamma \to \Sigma$ and $g : \Gamma \to M$, and a recognizable ω-language $\mathcal{L} \subseteq \Gamma^\omega$ such that $\mathbb{L} = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$.*

We start the proof of Theorem 5.6 with closure properties for recognizable quantitative ω-languages.

Lemma 5.7. *Let Γ be an alphabet.*

(a) *Let $g : \Gamma \to M$ be a renaming. Then, the composition $(\mathrm{val}^\omega \circ g) : \Gamma^\omega \to K$ is recognizable over $\mathbb{V}$.*
(b) *Let $\mathbb{L} : \Gamma^\omega \to K$ be a quantitative ω-language recognizable over $\mathbb{V}$ and $\mathcal{L} \subseteq \Gamma^\omega$ a recognizable ω-language. Then, the intersection $(\mathbb{L} \cap \mathcal{L}) : \Gamma^\omega \to K$ is recognizable over $\mathbb{V}$.*
(c) *Let $h : \Gamma \to \Sigma$ be a renaming and $\mathbb{L} : \Gamma^\omega \to K$ a quantitative ω-language recognizable over $\mathbb{V}$. Then, the quantitative ω-language $h(\mathbb{L}) : \Sigma^\omega \to K$ is recognizable over $\mathbb{V}$.*

Proof. (a) We define a MWBA $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ over Γ and $\mathbb{V}$ as follows:

- $Q = I = F = \{1\}$ and $T = \{1\} \times \Gamma \times \{1\}$;
- for every $\gamma \in \Gamma$, $\mathrm{wt}(1, \gamma, 1) = g(\gamma)$.

Then $[\![\mathcal{A}]\!] = \mathrm{val}^\omega \circ g$.

(b) We use Theorem 5.5 and apply the standard product construction for Büchi automata. Let $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ be a MWBA over Γ and $\mathbb{V}$ and let $\mathcal{A}' = (Q', I', T', F')$ be an unambiguous Büchi automaton over Γ. We construct a MWBA $\mathcal{A}'' = (Q'', I'', T'', F'', \mathrm{wt}'')$ over Γ and $\mathbb{V}$ as follows:

- $Q'' = Q \times Q' \times \{1, 2\}$, $I'' = I \times I' \times \{1\}$, $F'' = \{(q, q', 2) \mid q' \in F'\}$;
- T'' consists of all transitions $t = ((p, p', i), a, (q, q', f(p, p', i)))$ such that $(p, a, q) \in T$, $(p', a, q') \in T'$, $i \in \{1, 2\}$, and $f(p, p', i) \in \{1, 2\}$ is defined as follows. If $i = 1$ and $p \in F$, then $f(p, p', i) = 2$. If $i = 1$ and $p \notin F$, then $f(p, p', i) = 1$. If $i = 2$ and $p' \in F'$, then $f(p, p', i) = 1$. If $i = 2$ and $p' \notin F$, then $f(p, p', i) = 2$. For such a transition t, we let $\mathrm{wt}''(t) = \mathrm{wt}(p, a, q)$.

Then $[\![\mathcal{A}'']\!] = [\![\mathcal{A}]\!] \cap \mathcal{L}(\mathcal{A}')$.

(c) The proof uses the construction of Droste and Vogler [47] for the renaming. Let $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ be a MWBA over Γ and $\mathbb{V}$. We define a MWBA $\mathcal{A}' = (Q', I', T', F', \mathrm{wt}')$ over Σ and $\mathbb{V}$ as follows:

- $Q' = Q \times \Gamma$, $I' = I \times \{\gamma_0\}$ for some $\gamma_0 \in \Gamma$; $F' = F \times \Gamma$;
- T' consists of all transitions $t = ((p, \gamma), a, (p', \gamma')) \in Q' \times \Sigma \times Q'$ such that $(p, \gamma', p') \in T$ and $h(\gamma') = a$. For such a transition t, we let $\mathrm{wt}'(t) = \mathrm{wt}(p, \gamma', p')$.

Then $h([\![\mathcal{A}']\!]) = [\![\mathcal{A}]\!]$.

□

By successive application of parts (a), (b), (c) of Lemma 5.7, we obtain:

Corollary 5.8. *Let Γ be an alphabet, $h : \Gamma \to \Sigma$ and $g : \Gamma \to M$ renamings, and $\mathcal{L} \subseteq \Gamma^\omega$ a recognizable ω-language. Then, the quantitative ω-language $h((\mathrm{val}^\omega \circ g) \cap \mathcal{L}) : \Sigma^\omega \to K$ is recognizable over $\mathbb{V}$.*

Now we show the implication (a) $\Rightarrow$ (b) of Theorem 5.6.

Lemma 5.9. *Let $\mathcal{A}$ be a MWBA over Σ and $\mathbb{V}$. Then, there exist an alphabet Γ, renamings $h : \Gamma \to \Sigma$, $g : \Gamma \to M$, and a recognizable ω-language $\mathcal{L} \subseteq \Gamma^\omega$ such that $[\![\mathcal{A}]\!] = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$.*

Proof. Let $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$. Let $\Gamma = T$, $h : T \to \Sigma$ be defined for all $t = (p, a, q) \in T$ as $h(t) = a$, and $g : T \to M$ be defined for all $t \in T$ as $g(t) = \mathrm{wt}(t)$. Let $\mathcal{L} = \{\rho = (t_i)_{i \in \mathbb{N}} \mid \rho \text{ is a run of } \mathcal{A}\}$.

Consider the Büchi automaton $\mathcal{A}' = (Q, I, T', F)$ over Γ where $T' = (p, (p, a, q), q) \mid (p, a, q) \in T\}$. Then, $\mathcal{L}(\mathcal{A}') = \mathcal{L}$ and hence $\mathcal{L}$ is recognizable. We show that $[\![\mathcal{A}]\!] = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$.

Let $w \in \Sigma^\omega$. Then:

$$h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})(w) = \sum_{\substack{u \in \mathcal{L}, \\ h(u) = w}} \mathrm{val}^\omega(g(u)) = \sum_{\rho \in \mathrm{Run}_{\mathcal{A}}(w)} \mathrm{wt}_{\mathcal{A}}(\rho) = [\![\mathcal{A}]\!](w).$$

□

Proof of Theorem 5.6. Immediate by Lemma 5.9 and Corollary 5.8. □

We say that a MWBA $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ over Σ and $\mathbb{V}$ is *unambiguous* if the underlying Büchi automaton (Q, I, T, F) is unambiguous. In the rest of this section, we show a Nivat theorem for *unambiguous* MWBA.

Let $\mathcal{L} \subseteq \Gamma^\omega$ be an ω-language and $h : \Gamma \to \Sigma$ a renaming. We say that $\mathcal{L}$ is *h-unambiguous* if, for all $w \in \Sigma^\omega$, there exists at most one $u \in \mathcal{L}$ with $h(u) = w$.

As a corollary of the proof of Theorem 5.6, we obtain:

Corollary 5.10. *Let Σ be an alphabet, $\mathbb{V} = (M, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ an ω-valuation structure, and $\mathbb{L} : \Sigma^\omega \to K$ a quantitative ω-language. Then, the following are equivalent.*

(a) $\mathbb{L} = [\![\mathcal{A}]\!]$ *for some unambiguous MWBA* $\mathcal{A}$ *over* Σ *and* $\mathbb{V}$.

(b) There exist an alphabet Γ*, renamings* $h : \Gamma \to \Sigma$ *and* $g : \Gamma \to M$*, and a recognizable and* h*-unambiguous* ω*-language* $\mathcal{L} \subseteq \Gamma^\omega$ *such that* $\mathbb{L} = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$.

Proof. The proof relies on the same constructions as the proof of Theorem 5.6. Using the constructions of Lemma 5.7 (a), (b), we can construct an unambiguous MWBA for $\mathbb{L} := (\mathrm{val}^\omega \circ g) \cap \mathcal{L}$. In general, the construction of Lemma 5.7 (c) for the renaming leads to an ambiguous MWBA. However, the h-unambiguity of $\mathcal{L}$ guarantees that the multi-weighted automaton for $h(\mathbb{L})$ is unambiguous.

Notice also that, given an unambiguous automaton $\mathcal{A}$, the ω-language $\mathcal{L} \subseteq \Gamma^\omega$ defined in Lemma 5.9 is h-unambiguous.

□

We say that a MWBA $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ over an alphabet Σ and an ω-valuation structure $\mathbb{V}$ is *deterministic* if the underlying finite automaton (Q, I, T, F) is deterministic, i.e., $|I| = 1$ and, for all $(p, a) \in Q \times \Sigma$, there exists at most one $q \in Q$ with $(p, a, q) \in T$. We can use the proof of Theorem 5.6 to show that the recognizable quantitative ω-languages are exactly the renamings of deterministically recognizable quantitative ω-languages.

Corollary 5.11. *Let* Σ *be an alphabet,* $\mathbb{V} = (M, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ *an* ω*-valuation structure, and* $\mathbb{L} : \Sigma^\omega \to K$ *a quantitative* ω*-language. Then, the following are equivalent.*

(a) $\mathbb{L}$ *is recognizable over* $\mathbb{V}$.

(b) There exist an alphabet Γ*, a renaming* $h : \Gamma \to \Sigma$*, and a quantitative* ω*-language* $\mathbb{L}' : \Gamma^\omega \to K$ *recognizable by a deterministic MWBA over* $\mathbb{V}$*, such that* $\mathbb{L} = h(\mathbb{L}')$.

Proof. First, we show that (a) implies (b). Let $\mathbb{L}$ be recognizable over $\mathbb{V}$. By Lemma 5.9, there exist an alphabet Γ, renamings $h : \Gamma \to \Sigma$ and $g : \Gamma \to M$, and a recognizable ω-language $\mathcal{L} \subseteq \Gamma^\omega$ such that $\mathbb{L} = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$. Let $\mathbb{L}' = (\mathrm{val}^\omega \circ g) \cap \mathcal{L}$. It remains to show that the quantitative ω-language $\mathbb{L}'$ is deterministically recognizable over Γ and $\mathbb{V}$.

Note that in the proof of Lemma 5.9, the Büchi automaton $\mathcal{A}'$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}$ has a deterministic transition relation but possibly multiple initial states. Note that here we can replace them by a new single initial state; this does not harm the deterministic transition table since each label of a transition contains information about an outgoing state. Then, $\mathcal{L}$ is deterministically recognizable. Following the constructions of Lemma 5.7, we can construct a deterministic MWTA $\mathcal{B}$ over Γ and $\mathbb{V}$ with $[\![\mathcal{B}]\!] = \mathrm{val}^\omega \circ g$ and, using $\mathcal{B}$ and the deterministic automaton for $\mathcal{L}$, a deterministic MWTA $\mathcal{C}$ over Γ and $\mathbb{V}$ with $[\![\mathcal{C}]\!] = \mathbb{L}'$.

The implication (b) $\Rightarrow$ (a) follows from Lemma 5.7 (c). □

5.3 Multi-weighted Muller automata

In this section, we consider the model of multi-weighted automata on infinite words with a *Muller acceptance condition*. It is well known that, in the Boolean setting, Büchi and Muller automata are expressively equivalent. In [44], this result was extended to totally complete semirings.

As an application of our Nivat theorem for MWBA, we show the expressive equivalence of MWBA and multi-weighted Muller automata.

Let Σ be an alphabet. A *Muller automaton* over Σ is a tuple $\mathcal{M} = (Q, I, T, \mathcal{F}, \mathrm{wt})$ where Q is a finite set of states, $I \subseteq Q$ is a set of initial states, $T \subseteq Q \times \Sigma \times Q$ is a set of transition and $\mathcal{F} \subseteq 2^Q$ is a Muller acceptance condition.

A run $\rho = (t_i)_{i \in \mathbb{N}}$ of $\mathcal{M}$ is defined as an infinite sequence of matching transitions, say $t_i = (q_i, a_i, q_{i+1})$ for all $i \in \mathbb{N}$, such that $q_0 \in I$ and $\{q \in Q \mid q = q_i \text{ for infinitely many } i \in \mathbb{N}\} \in \mathcal{F}$. Let $\mathrm{label}(\rho) = (a_i)_{i \in \mathbb{N}}$, the *label* of ρ. Let $\mathrm{Run}_{\mathcal{M}}$ be the set of all runs of $\mathcal{M}$ and, for every $w \in \Sigma^\omega$, let $\mathrm{Run}_{\mathcal{M}}(w)$ be the set of all runs ρ of $\mathcal{M}$ with $\mathrm{label}(\rho) = w$. Let $\mathcal{L}(\mathcal{M}) = \{w \in \Sigma^\omega \mid \mathrm{Run}_{\mathcal{M}}(w) \neq \emptyset\}$, the ω-language *accepted* by $\mathcal{M}$.

The following proposition is a folklore result.

Proposition 5.12. *Let Σ be an alphabet and $\mathcal{L} \subseteq \Sigma^\omega$ an ω-language. Then, the following are equivalent.*

(a) $\mathcal{L} = \mathcal{L}(\mathcal{A})$ for some Büchi automaton $\mathcal{A}$ over Σ.
(b) $\mathcal{L} = \mathcal{L}(\mathcal{M})$ for some Muller automaton $\mathcal{M}$ over Σ.

Let $\mathbb{V} = (M, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ be an ω-valuation structure. A *multi-weighted Muller automaton* over Σ and $\mathbb{V}$ is a tuple $\mathcal{M} = (Q, I, T, \mathcal{F}, \mathrm{wt})$ where $(Q, I, T, \mathcal{F})$ is a Muller automaton over Σ and $\mathrm{wt} : T \to M$ is a transition weight function.

As for multi-weighted Büchi automata, given a run $\rho = (t_i)_{i \in \mathbb{N}} \in \mathrm{Run}_{\mathcal{M}}$ with $t_i \in T$ for all $i \in \mathbb{N}$, the *weight* of ρ is defined as $\mathrm{wt}_{\mathcal{M}}(\rho) = \mathrm{val}^\omega((\mathrm{wt}(t_i))_{i \in \mathbb{N}})$. Then, the *behavior* $[\![\mathcal{M}]\!] : \Sigma^\omega \to K$ of $\mathcal{M}$ is defined for all $w \in \Sigma^\omega$ as $[\![\mathcal{M}]\!](w) = \sum(\mathrm{wt}_{\mathcal{M}}(\rho) \mid \rho \in \mathrm{Run}_{\mathcal{M}}(\rho))$.

The following theorem states the equivalence of multi-weighted Muller automata and multi-weighted Büchi automata.

Theorem 5.13. *Let Σ be an alphabet, $\mathbb{V} = (M, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ an ω-valuation structure, and $\mathbb{L}$ a quantitative ω-language. Then, the following are equivalent.*

(a) $\mathbb{L} = [\![\mathcal{A}]\!]$ for some multi-weighted Büchi automaton $\mathcal{A}$ over Σ and $\mathbb{V}$.
(b) $\mathbb{L} = [\![\mathcal{M}]\!]$ for some multi-weighted Muller automaton $\mathcal{M}$ over Σ and $\mathbb{V}$.

This theorem extends the expressive equivalence result of weighted Büchi automata and weighted Muller automata over *totally complete semirings* established in [44]. Whereas the proof of [44] was given by direct non-trivial transformations of automata, here we give a simple proof based on Proposition 5.12 and Theorem 5.6. It suffices to show that Muller-recognizable quantitative ω-languages permit the same decompositions as in Theorem 5.6 for Büchi-recognizable quantitative ω-languages.

Theorem 5.14. *Let Σ be an alphabet, $\mathbb{V} = (M, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ an ω-valuation structure, and $\mathbb{L} : \Sigma^\omega \to K$ a quantitative ω-language. Then, the following are equivalent.*

(a) $\mathbb{L} = [\![\mathcal{M}]\!]$ *for some multi-weighted Muller automaton $\mathcal{M}$ over Σ and $\mathbb{V}$.*
(b) There exist an alphabet Γ, renamings $h : \Gamma \to \Sigma$ and $g : \Gamma \to M$, and a recognizable ω-language $\mathcal{L} \subseteq \Gamma^\omega$ such that $\mathbb{L} = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$.

The proof of this theorem is much the same as the proof of Theorem 5.6. We only have to replace $F \subseteq Q$ by $\mathcal{F} \subseteq 2^Q$ in the proof of Lemma 5.9 and slightly modify the constructions of Lemma 5.7.

- We construct a multi-weighted Muller automaton for $\mathrm{val}^\omega \circ g$ as in Lemma 5.7 (a) where we replace $F = \{1\}$ by $\mathcal{F} = \{\{1\}\}$.
- For the intersection $\mathbb{L} \cap \mathcal{L}$, we apply Proposition 5.12 to construct a Muller automaton for $\mathcal{L}$, determinize it and use the standard product construction for Muller automata.
- For the renaming $h(\mathbb{L})$, we proceed as in Lemma 5.7 (c) where we replace F' by the Muller acceptance condition $\mathcal{F}'$ which consists of all sets $\{(q_1, \gamma_1), ..., (q_k, \gamma_k)\} \subseteq Q'$ such that $\{q_1, ..., q_k\} \in \mathcal{F}$ (a similar idea was used in [40]).

Proof of Theorem 5.13. Immediate by Theorems 5.6 and 5.14. □

5.4 The discount-optimal value problem

Let Σ be an alphabet and $\mathbb{V} = (M, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ an ω-valuation structure. For a quantitative ω-language $\mathbb{L} : \Sigma^\omega \to K$, let $\mathbb{L}^\sharp = \sum(\mathbb{L}(w) \mid w \in \Sigma^\omega) \in K$, the *optimal value* of $\mathbb{L}$ (this notion is motivated by examples with $+ = \inf$ or $+ = \sup$). The *optimal weight problem* for $\mathbb{V}$ is, given an alphabet Σ and a multi-weighted Büchi automaton $\mathcal{A}$ over Σ and $\mathbb{V}$, to compute $[\![\mathcal{A}]\!]^\sharp$.

The *discount-optimal value problem* is the optimal value-problem for the ω-valuation structure $\mathbb{V}^{\omega\mathrm{Disc}}$ as defined in Example 5.4 (b).

We call a MWBA $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ over Σ and $\mathbb{V}^{\omega\mathrm{Disc}}$ *simple* if $Q = F$, i.e., the Büchi acceptance condition is irrelevant for $\mathcal{A}$. The discount-optimal problem for simple MWBA was studied in [8] (for the case when the discounting factors of transitions are less that 1) and in [53] (where the discounting factor 1 is allowed):

Theorem 5.15 ([8], [53])**.** *Given a singleton alphabet Δ and a simple MWBA $\mathcal{A}$ over Δ and $\mathbb{V}^{\omega\mathrm{Disc}}$, the value $[\![\mathcal{A}]\!]^\sharp$ is computable.*

In this section, we show that this computability result holds for arbitrary MWBA.

Theorem 5.16. *The discount-optimal value problem is computable.*

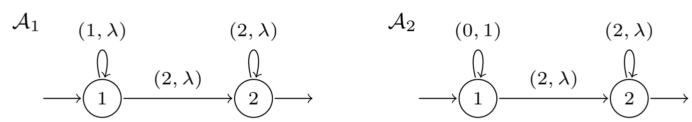

Figure 5.1: MWBA $\mathcal{A}_1$ and $\mathcal{A}_2$ of Example 5.17

In the rest of this section, we give a proof of this theorem. Our proof will use Theorem 5.15 for simple MWBA. First of all, we consider an example which will provide an intuition for our constructions.

Example 5.17. *Consider a singleton alphabet Δ and a MWBA $\mathcal{A}_1$ over Δ and $\mathbb{V}^{\omega\text{Disc}}$ depicted on Fig. 5.1 (the labels of transitions are omitted). Here, $\lambda \in (0,1) \cap \mathbb{Q}$ is a discounting factor. Note that every run ρ of $\mathcal{A}_1$ takes $k \geq 0$ times the self-loop of the state 1, jumps to the state 2 and takes the self-loop of the state 2 infinitely often. Then, $[\![\mathcal{A}_1]\!]^\sharp = \inf_{k \in \mathbb{N}} \left(\frac{1-\lambda^k}{1-\lambda} + \frac{2\lambda^k}{1-\lambda} \right) = \frac{1}{1-\lambda}$. Note that $\frac{1}{1-\lambda}$ is the weight of the non-accepting run which always stays in 1. This can be explained by the fact that we can stay in the beneficial self-loop of the state 1 as many times as we want (this makes the product of discounting factors arbitrarily small) and jump to state 2 in order to fulfill the Büchi acceptance condition. Then, in order to compute $[\![\mathcal{A}]\!]^\sharp$, we can make the state 1 accepting and apply the algorithm of Theorem 5.15 for the obtained simple MWBA. So, the solution in this case is to keep only those states of $\mathcal{A}_1$ which are visited by some accepting run of $\mathcal{A}_1$ and to make all these states accepting.*

However, this approach does not always work if we have a transition with the discounting factor 1. Consider, for instance, the MWBA $\mathcal{A}_2$ of Figure 5.1 where $[\![\mathcal{A}_2]\!]^\sharp = \frac{2}{1-\lambda}$. If we apply to $\mathcal{A}_2$ the method considered for $\mathcal{A}_1$, then we obtain the value 0 which is not correct.

So, the situation with the discounting factor 1 is the main difficulty of the proof. However, the discounting factors 1 appeared in the corner-point abstraction (which is a multi-weighted automaton with discounting factors $\lambda \in (0,1) \cap \mathbb{Q}$ and 1) for weighted timed automata with discounting considered in [53]. Therefore, we do not exclude this case.

We will show that we can apply the method of Example 5.17 to MWBA with the discounting factor 1 if we:

- remove each cycle θ such that θ does not contain an accepting state and all transitions of θ have the cost 0 and the discounting factor 1;
- remove every strongly connected component U such that all transitions of U have the discounting factor 1.

Let Σ be an alphabet and $\mathcal{A} = (Q, I, T, F, \text{wt})$ a MWBA over Σ and $\mathbb{V}^{\omega\text{Disc}}$. For every transition $t \in T$ with $\text{wt}(t) = (c', d')$ for some $c' \in \mathbb{Q}_{\geq 0}$ and $d' \in (0,1] \cap \mathbb{Q}$, let $\text{wt}_c(t) = c'$, the *cost* of t, and let $\text{wt}_d(t) = d'$, the *discounting factor* of t. A *run fragment* of $\mathcal{A}$ is a sequence $\pi = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} q_n$ where

$n \geq 0$ and $t_i := (q_{i-1}, a_i, q_i) \in T$ for all $i \in \{1, ..., n\}$. Let $C(\pi) = \sum_{i=1}^{n} \mathrm{wt}_c(t_i)$, the *cost* of π, and $D(\pi) = \prod_{i=1}^{n} \mathrm{wt}_d(t_i)$, the *discounting factor* of π. Let $\mathcal{Q}_\pi^\mathcal{A} = \{q_0, ..., q_n\}$. We say that π is a *cycle* if $q_0 = q_n$. We say that π is a *simple cycle* if π is a cycle and the states $q_0, ..., q_{n-1}$ are pairwise distinct.

Let θ be a *cycle* of $\mathcal{A}$. We say that θ is *accepting* if $\mathcal{Q}_\theta^\mathcal{A} \cap F \neq \emptyset$. We say that θ is *singular* if θ is simple, not accepting, $C(\theta) = 0$ and $D(\theta) = 1$. Let $\mathcal{S}(\mathcal{A})$ be the number of singular cycles of $\mathcal{A}$. We say that $\mathcal{A}$ is *non-singular* if $\mathcal{S}(\mathcal{A}) = 0$.

Lemma 5.18. *Let $\mathcal{A}$ be a MWBA over an alphabet Σ and $\mathbb{V}^{\omega\mathrm{Disc}}$. Then, there exist an alphabet Γ and a non-singular MWBA $\mathcal{B}$ over Γ and $\mathbb{V}^{\omega\mathrm{Disc}}$ such that $[\![\mathcal{B}]\!]^\sharp = [\![\mathcal{A}]\!]^\sharp$.*

Proof. The idea of our construction is the following. We replace a singular cycle θ by a single state v^θ. Since the transitions of θ do not influence the weight of a run, we augment v^θ with a self-loop for each discounting factor appearing in the transitions between the states of θ and label it with the minimal cost for this discounting factor. Correspondingly, we connect the other states with v^θ. In order to avoid an automaton with the multiple occurrences of the same transitions, we have to extend the alphabet in order to label them by distinct letters.

Let $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ such that $\mathcal{S}(\mathcal{A}) > 0$. Let $\Lambda = \{\mathrm{wt}_d(t) \mid t \in T\}$ and $\Gamma = \Sigma \times \Lambda$. Let $\mathcal{B}_0 = (Q, I, T', F, \mathrm{wt}')$ be a MWTA over Γ and $\mathbb{V}^{\omega\mathrm{Disc}}$ such that $T' = \{(q, (a, \lambda), q') \mid t := (q, a, q') \in T \text{ and } \mathrm{wt}_d(t) = \lambda\}$ and, for all $t = (q, (a, \lambda), q') \in T'$, $\mathrm{wt}'(t) = \mathrm{wt}(q, a, q')$. Clearly, $\mathcal{S}(\mathcal{B}_0) = \mathcal{S}(\mathcal{A})$ and $[\![\mathcal{B}_0]\!]^\sharp = [\![\mathcal{A}]\!]^\sharp$.

Assume that, for some $i \geq 0$, a MWTA $\mathcal{B}_i$ over Γ and $\mathbb{V}^{\omega\mathrm{Disc}}$ is defined and contains a singular cycle θ. Let $U = Q_i \setminus \mathcal{Q}_\theta^{\mathcal{B}_i}$. Then, we define $\mathcal{B}_{i+1} = (Q_{i+1}, I_{i+1}, T_{i+1}, F_{i+1}, \mathrm{wt}_{i+1})$ from $\mathcal{B}_i$ as follows:

- $Q_{i+1} = U \cup \{v^\theta\}$ where $v^\theta \notin U$ is a new state which will simulate θ;
- $I_{i+1} = \begin{cases} (I_i \cap U) \cup \{v^\theta\}, & \text{if } I_i \cap \mathcal{Q}_\theta^{\mathcal{B}_i} \neq \emptyset, \\ I_i \cap U, & \text{otherwise;} \end{cases}$
- $T_{i+1} = (T_i \cap (U \times \Gamma \times U)) \cup T_{\rightarrow\theta} \cup T_{\circlearrowleft\theta} \cup T_{\theta\rightarrow}$ such that:
  - $T_{\rightarrow\theta} = \{(q, \gamma, v^\theta) \mid q \in U \text{ and } (q, \gamma, q') \in T_i \text{ for some } q' \in \mathcal{Q}_\theta^{\mathcal{B}_i}\}$;
  - $T_{\circlearrowleft\theta} = \{(v^\theta, \gamma, v^\theta) \mid \gamma \in \Gamma \setminus (\Sigma \times \{1\}), \exists q, q' \in \mathcal{Q}_\theta^{\mathcal{B}_i} : (q, \gamma, q') \in T_i\}$;
  - $T_{\theta\rightarrow} = \{(v^\theta, \gamma, q') \mid q' \in U \text{ and } (q, \gamma, q') \in T_i \text{ for some } q \in \mathcal{Q}_\theta^{\mathcal{B}_i}\}$;
- $F_{i+1} = F_i$ (note that $\mathcal{Q}_\theta^{\mathcal{B}_i} \cap F = \emptyset$);
- for all $t \in T_{i+1}$, $\mathrm{wt}_{i+1}(t)$ is defined as follows:
  - if $t \in T_i \cap (U \times \Gamma \times U)$, then $\mathrm{wt}_{i+1}(t) = \mathrm{wt}_i(t)$;
  - if $t = (v^\theta, \gamma, q') \in T_{\theta\rightarrow}$ with $\gamma = (a, \lambda) \in \Gamma$, then $(\mathrm{wt}_{i+1})_d(t) = \lambda$ and $(\mathrm{wt}_{i+1})_c(t) = \min\{(\mathrm{wt}_i)_c(q, \gamma, q') \mid q \in \mathcal{Q}_\theta^{\mathcal{B}_i} \text{ and } (q, \gamma, q') \in T_i\}$;
  - if $t = (v^\theta, \gamma, v^\theta) \in T_{\circlearrowleft\theta}$ with $\gamma = (a, \lambda)$, then $(\mathrm{wt}_{i+1})_d(t) = \lambda$ and $(\mathrm{wt}_{i+1})_c(t) = \min\{(\mathrm{wt}_i)_c(q, \gamma, q') \mid q, q' \in \mathcal{Q}_\theta^{\mathcal{B}_i} \text{ and } (q, \gamma, q') \in T_i\}$;

– if $t = (q, \gamma, v^\theta) \in T_{\to\theta}$ with $\gamma = (a, \lambda) \in \Gamma$, then $(\mathrm{wt}_{i+1})_d(t) = \lambda$ and $(\mathrm{wt}_{i+1})_c(t) = \min\{(\mathrm{wt}_i)_c(q, \delta, q') \mid q' \in \mathcal{Q}_\theta^{\mathcal{B}_i}$ and $(q, \delta, q') \in T_i\}$;

Thus, we eliminated in $\mathcal{B}_i$ at least one singular cycle θ of $\mathcal{B}_{i-1}$ and we did not add any new singular cycle. Then, $\mathcal{S}(\mathcal{B}_i) < \mathcal{S}(\mathcal{B}_{i-1})$. Moreover, we simulated all potentially optimal runs of $\mathcal{A}$ and excluded some of runs which do not influence the value $[\![\mathcal{B}_{i-1}]\!]^\sharp$. Then, $[\![\mathcal{B}_i]\!]^\sharp = [\![\mathcal{B}_{i-1}]\!]^\sharp$. Following this process, in finitely many steps we obtain a MWBA $\mathcal{B}_k$ over Γ and $\mathbb{V}^{\omega\textsc{Disc}}$ such that $\mathcal{S}(\mathcal{B}_k) = 0$ and $[\![\mathcal{B}_k]\!]^\sharp = [\![\mathcal{A}]\!]^\sharp$. Then, we let $\mathcal{B} = \mathcal{B}_k$. □

We fix a singleton alphabet $\Delta = \{\delta\}$.

Lemma 5.19. *Let $\mathcal{A}$ be a non-singular MWBA over an alphabet Σ and $\mathbb{V}^{\omega\textsc{Disc}}$. Then, there exists a non-singular MWBA $\mathcal{B}$ over Δ and $\mathbb{V}^{\omega\textsc{Disc}}$ such that $[\![\mathcal{B}]\!]^\sharp = [\![\mathcal{A}]\!]^\sharp$.*

Proof. Let $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$. Consider the renaming $h : \Sigma \to \Delta$. Then, $h([\![\mathcal{A}]\!])^\sharp = [\![\mathcal{A}]\!]^\sharp$. By Lemma 5.7 (c), $h([\![\mathcal{A}]\!])$ is recognized by a MWBA $\mathcal{B}$ over Δ and $\mathbb{V}^{\omega\textsc{Disc}}$ such that $[\![\mathcal{B}]\!] = h([\![\mathcal{A}]\!])$. Let $\mathcal{B}$ be constructed as in the proof of Lemma 5.7. We show that $\mathcal{B}$ is non-singular. Suppose that $\mathcal{B}$ contains a singular cycle $\theta = (q_0, a_0) \xrightarrow{\delta} (q_1, a_1) \xrightarrow{\delta} \ldots \xrightarrow{\delta} (q_n, a_n)$ where $q_0, ..., q_n \in Q$, $a_0, ..., a_n \in \Sigma$, $(q_0, a_0) = (q_n, a_n)$, $C(\theta) = 0$ and $D(\theta) = 1$. Then, we can construct the non-accepting cycle $\eta = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} q_n$ of $\mathcal{A}$ such that $C(\eta) = C(\theta) = 0$ and $D(\eta) = D(\theta) = 1$. Then, η contains a trivial subcycle η' (note that $q_0, ..., q_{n-1}$ are not necessarily pairwise distinct). Hence, $\mathcal{A}$ is non-singular. A contradiction. □

Then, it suffices to show that $[\![\mathcal{A}]\!]^\sharp$ is computable for a non-singular MWBA $\mathcal{A} = (Q, I, T, F, \mathrm{wt})$ over the singleton alphabet Δ. Since the labels of transitions in T are irrelevant, we will slightly abuse the notation and assume that $T \subseteq Q \times Q$. So, we will consider the runs of $\mathcal{A}$ as sequences $\rho = (q_i)_{i \in \mathbb{N}} \in Q^\omega$ such that $q_0 \in I$, $(q_i, q_{i+1}) \in T$ for all $i \in \mathbb{N}$, and there exists $f \in F$ such that $q_i = f$ for infinitely many $i \in \mathbb{N}$. For each $i \geq 0$, let $D_i(\rho) = \prod_{j=1}^{i-1} \mathrm{wt}_d(q_j, q_{j+1})$. We will also represent a run fragment θ of $\mathcal{A}$ as a finite sequence $v_0 ... v_n \in Q^+$. Moreover, we may assume without loss of generality that $I = \{\iota\}$ is a singleton set (otherwise, we can add a new unique initial state ι and connect it with the states in I by transitions t with $\mathrm{wt}(t) = (0, 1)$. Clearly, this does not harm the non-singularity of a MWBA).

Let $\xrightarrow{+}_{\mathcal{A}} \subseteq Q \times Q$ be the transitive closure of T and $\xrightarrow{*}_{\mathcal{A}} \subseteq Q \times Q$ the reflexive and transitive closure of T. For $q \in Q$ and $U \subseteq Q$, we write $q \xrightarrow{*}_{\mathcal{A}} U$ if $q \xrightarrow{*}_{\mathcal{A}} u$ for some $u \in U$. We say that a subset $U \subseteq Q$ is *strongly connected* if $u \xrightarrow{+}_{\mathcal{A}} u'$ for all $u, u' \in U$. We say that U is a *strongly connected component* (SCC) if U is strongly connected and every subset $V \subseteq Q$ with $V \supsetneq U$ is not strongly connected. A SCC $U \subseteq Q$ is called *accepting* if $U \cap F \neq \emptyset$. A SCC U of $\mathcal{A}$ is called *reachable* if $\iota \xrightarrow{*}_{\mathcal{A}} U$. A SCC U of $\mathcal{A}$ is called *trivial* if $\mathrm{wt}_d(t) = 1$ for all $t \in T \cap (U \times U)$. Let $\mathcal{T}(\mathcal{A})$ be the number of SCC of $\mathcal{A}$ which are reachable and trivial. We say that $\mathcal{A}$ is *non-trivial* if $\mathcal{T}(\mathcal{A}) = 0$.

Lemma 5.20. *Let $\mathcal{A}$ be a non-singular MWBA over Δ and $\mathbb{V}^{\omega\mathrm{Disc}}$ such that $\mathcal{T}(\mathcal{A}) > 0$. Then, there exists a non-singular MWBA $\mathcal{B}$ over Δ and $\mathbb{V}^{\omega\mathrm{Disc}}$ such that $\mathcal{T}(\mathcal{B}) = \mathcal{T}(\mathcal{A}) - 1$ and $[\![\mathcal{B}]\!]^\sharp = [\![\mathcal{A}]\!]^\sharp$.*

Proof. Let $\mathcal{A} = (Q, \{\iota\}, T, F, \mathrm{wt})$ and $U \subseteq Q$ a trivial and reachable SCC of $\mathcal{A}$. We construct $\mathcal{B}$ by eliminating U. The construction is based on the following ideas.

- If $U \cap F = \emptyset$, then stay in U is not an optimal solution (since it will not decrease the global discounting factor and will probably increase the cost) and we have to leave it with the minimal cost.
- If there exists $f \in U \cap F$ and there exists a simple cycle θ of $\mathcal{A}$ such that $f \in \mathcal{Q}_\theta$ and $C(\theta) = 0$, then the eventual recurring of θ could be a potential optimal solution. Then, we retain the state f, reach it with the minimal cost and equip it with the self-looping transition with the cost 0 and an arbitrary discounting factor $\lambda < 1$ (in order to avoid a new trivial SCC).
- If there exists $f \in U \cap F$ and there exists no simple cycle θ of $\mathcal{A}$ such that $f \in \mathcal{Q}_\theta$ and $C(\theta) = 0$, then the visit of f infinitely many times is not an optimal solution, since we obtain an accepting run of the weight ∞.

We may assume that $\iota \notin U$. Let

$$^{\rightarrow}U = \{(q, u) \in T \mid q \in Q \setminus U \text{ and } u \in U\},$$

the set of exterior ingoing transitions of U, and

$$U^{\rightarrow} = \{(u, q) \in T \mid u \in U \text{ and } q \in Q \setminus U\},$$

the set of exterior outgoing transitions of U. Let F_U be the set of all $f \in U \cap F$ such that there exists a simple cycle θ of $\mathcal{A}$ with $f \in \mathcal{Q}_\theta$ and $C(\theta) = 0$. For $u, u' \in U$, let $C_{\min}(u, u') = \min\{C(\theta) \mid \theta = v_0 ... v_n \in Q^+ \text{ is a path fragment with } v_0 = u, v_n = u'\}$.

The idea of our construction is the following. Assume that $p, p' \in U$ and $q, q' \in Q \setminus U$ such that $(q, p) \in T$ and $(p', q') \in T$. Then, we add the states $s_1 = \langle q, p, p', q', 1\rangle$ and $s_2 = \langle q, p, p', q', 2\rangle$. Then transition (q, p) will be modelled by (q, s_1), and the transition (p', q') will be modelled by (s_2, q'). Moreover, we add the transition (s_1, s_2) with the discounting factor 1 and the minimal cost of a run fragment between p and p'.

We let $\mathcal{B} = (Q', \{\iota\}, T', F', \mathrm{wt}')$ such that:

- $Q' = (Q \setminus U) \cup Q'' \cup F_U \cup F''$ where
  - $Q'' = \{(v, v', i) \mid v \in {}^{\rightarrow}U, v' \in U^{\rightarrow}, i \in \{1, 2\}\}$;
  - $F'' = \{(v, f) \mid v \in {}^{\rightarrow}U \text{ and } f \in F_U\}$;
- $T' = (T \cap (Q \setminus U)^2) \cup T_1 \cup T_{12} \cup T_2 \cup T_{F'',1} \cup T_{F'',2} \cup T_{F_U}$ where:
  - $T_1 = \{(q, ((q, u), v', 1)) \mid (q, u) \in {}^{\rightarrow}U \text{ and } v' \in U^{\rightarrow}\}$;
  - $T_{12} = \{((v, v', 1), (v, v', 2)) \mid v \in {}^{\rightarrow}U \text{ and } v' \in U^{\rightarrow}\}$;

- $T_2 = \{((v, (u', q'), 2), q') \mid (u', q') \in U^{\rightarrow}\}$;
- $T_{F'',1} = \{(q, ((q, u), f)) \mid (q, u) \in {}^{\rightarrow}U, f \in F_U\}$;
- $T_{F'',2} = \{(((q, u), f), f) \mid (q, u) \in {}^{\rightarrow}U, f \in F_U\}$;
- $T_{F_U} = \{(f, f) \mid f \in F_U\}$.

- wt' is defined as follows:
  - for $t \in (T \cap (Q \setminus U)^2)$: $\mathrm{wt}'(t) = \mathrm{wt}(t)$;
  - for $t = (q, ((q, u), v', 1)) \in T_1$: $\mathrm{wt}'(t) = \mathrm{wt}(q, u)$;
  - for $t = ((v, v', 1), (v, v', 2)) \in T_{12}$ with $v = (q, u)$ and $v' = (u', q')$: $\mathrm{wt}'(t) = (C_{\min}(u, u'), 1)$;
  - for $t = ((v, (u', q'), 2), q') \in T_2$: $\mathrm{wt}'(t) = \mathrm{wt}(u', q')$;
  - for $t = (q, ((q, u), f)) \in T_{F'',1}$: $\mathrm{wt}'(t) = \mathrm{wt}(q, u)$;
  - for $t = (((q, u), f), f) \in T_{F'',2}$: $\mathrm{wt}'(t) = (C_{\min}(u, f), 1)$;
  - for $t = (f, f) \in T_{F_U}$, let $\mathrm{wt}'(t) = (0, \lambda)$ where $\lambda \in \mathbb{Q} \cap (0, 1)$ is fixed.

Note that the set $Q' \setminus (Q \setminus U)$ does not contain any SCC. Then, $\mathcal{T}(\mathcal{B}) = \mathcal{T}(\mathcal{A}) - 1$. Moreover, $\mathcal{B}$ simulates all potential optimal runs of $\mathcal{A}$ and excludes some runs which does not influence the value $[\![\mathcal{A}]\!]^\sharp$. Then, $[\![\mathcal{B}]\!]^\sharp = [\![\mathcal{A}]\!]^\sharp$. Moreover, our U-elimination process did not add any singular cycle to $\mathcal{A}$. Then, $\mathcal{B}$ is non-singular. $\square$

As a corollary from Lemma 5.20, we obtain:

Corollary 5.21. *Let $\mathcal{A}$ be a non-singular MWBA over Δ and $\mathbb{V}^{\omega\mathrm{Disc}}$. Then, there exists a MWBA $\mathcal{B}$ over Δ and $\mathbb{V}^{\omega\mathrm{Disc}}$ such that $\mathcal{B}$ is non-singular and non-trivial, and $[\![\mathcal{B}]\!]^\sharp = [\![\mathcal{A}]\!]^\sharp$.*

The following lemma shows that we can decide whether $[\![\mathcal{A}]\!]^\sharp = \infty$ for a non-trivial MWTA $\mathcal{A}$.

Lemma 5.22. *Let $\mathcal{A} = (Q, \{\iota\}, T, F, \mathrm{wt})$ be a MWBA over Δ and $\mathbb{V}^{\omega\mathrm{Disc}}$ such that $\mathcal{A}$ is non-trivial. Then, the following are equivalent.*

(a) $[\![\mathcal{A}]\!]^\sharp < \infty$;

(b) There exists $f \in F$ such that $\iota \xrightarrow{*}_{\mathcal{A}} f \xrightarrow{+}_{\mathcal{A}} f$.

Proof. The implication (a) $\Rightarrow$ (b) is trivial.

We show that (b) implies (a). Since $\iota \xrightarrow{*} f$, there exists a run fragment $u_0 ... u_k \in Q^+$ of $\mathcal{A}$ such that $1 \le k \le |Q|$, $u_0 = \iota$, $u_k = f$ and $(u_i, u_{i+1}) \in T$ for all $i \in \{0, ..., k-1\}$. Since $f \xrightarrow{+}_{\mathcal{A}} f$ and $\mathcal{A}$ is non-trivial, the SCC P containing f is non-trivial, i.e., there exist $p, p' \in P$ such that $(p, p') \in T$ and $\lambda := \mathrm{wt}_d(p, p') < 1$. Since $f \xrightarrow{*} p$ and $p' \xrightarrow{*} f$, we can construct a cycle $\theta = v_0 ... v_l \in Q^+$ with $v_0 = v_m = f$, $1 \le l < 2 \cdot |Q|$ and $(v_i, v_{i+1}) = (p, p')$ for some $i \in \{0, ..., l-1\}$. Then, we construct the run $\rho = u_0 ... u_{k-1}(f v_1 ... v_{l-1})^\omega$. Clearly,

$\rho \in \mathrm{Run}_{\mathcal{A}}$. We show that $\mathrm{wt}_{\mathcal{A}}(\rho) < \infty$. Indeed, let $M = \max\{\mathrm{wt}_c(t) \mid t \in T\}$. Then:

$$\begin{aligned}\mathrm{wt}_{\mathcal{A}}(\rho) &\leq M \cdot k + \sum_{i \geq 0} M \cdot l \cdot \lambda^i < M \cdot |Q| + 2 \cdot M \cdot |Q| \cdot \sum_{i \geq 0} \lambda^i \\ &= M \cdot |Q| \cdot \frac{3 - \lambda}{1 - \lambda} < \infty.\end{aligned}$$

Hence, $[\![\mathcal{A}]\!]^\sharp < \mathrm{wt}_{\mathcal{A}}(\rho) < \infty$. $\square$

Lemma 5.23. *Let $\mathcal{A}$ be a non-trivial, non-singular MWBA over Δ and $\mathbb{V}^{\omega\mathrm{Disc}}$ such that $[\![\mathcal{A}]\!]^\sharp < \infty$. Then, there exists a simple MWBA $\mathcal{B}$ over Σ and $\mathbb{V}^{\omega\mathrm{Disc}}$ such that $[\![\mathcal{B}]\!]^\sharp = [\![\mathcal{A}]\!]^\sharp$.*

Proof. Let $U = \bigcup\{P \mid P$ is an accepting and reachable SCC of $\mathcal{A}\}$. Since $[\![\mathcal{A}]\!]^\sharp < \infty$, by Lemma 5.22, $U \neq \emptyset$. Let $M = \max\{\gamma_c(t) \mid t \in T\} + 1 > 0$ and $\lambda = \max\{\mathrm{wt}_d(t) \mid t \in T$ and $\mathrm{wt}_d(t) < 1\}$. Since $\mathcal{A}$ is non-trivial, non-singular and $[\![\mathcal{A}]\!]^\sharp < \infty$, then $0 < \lambda < 1$.

We construct $\mathcal{B} = (Q', \{\iota\}, T', Q', \mathrm{wt}')$ as follows:

- $Q' = U \cup \{q \in Q \mid \iota \xrightarrow{*}_{\mathcal{A}} q \xrightarrow{*}_{\mathcal{A}} U\}$;
- $T' = T \cap (Q' \times Q')$;
- $\mathrm{wt}' = \mathrm{wt}\,|_{T'}$.

Clearly, since $\mathcal{A}$ is non-singular, $\mathcal{B}$ is also non-singular.

First, we prove that $[\![\mathcal{B}]\!]^\sharp \leq [\![\mathcal{A}]\!]^\sharp$. For this, we show $\mathrm{Run}_{\mathcal{A}} \subseteq \mathrm{Run}_{\mathcal{B}}$. Indeed, let $\rho = (q_i)_{i \in \mathbb{N}} \in \mathrm{Run}_{\mathcal{A}}$. Then, there exist a SCC $P \subseteq U$ and $k \in \mathbb{N}$ such that $q_i \in P$ for all $i \geq k$. Moreover, $\iota \xrightarrow{*}_{\mathcal{A}} q_i \xrightarrow{*}_{\mathcal{A}} P$ for all $0 \leq i < k$. Then, $\rho \in (Q')^\omega$ and hence $\rho \in \mathrm{Run}_{\mathcal{B}}$.

Next, we prove that $[\![\mathcal{A}]\!]^\sharp = [\![\mathcal{B}]\!]^\sharp$. Let $\varepsilon > 0$. We show the following:

$$\forall \rho \in \mathrm{Run}_{\mathcal{B}}\, \exists \varrho \in \mathrm{Run}_{\mathcal{A}} : \mathrm{wt}_{\mathcal{A}}(\varrho) < \mathrm{wt}_{\mathcal{B}}(\rho) + \varepsilon. \tag{5.1}$$

Let $\rho \in \mathrm{Run}_{\mathcal{B}}$. If $\rho \in \mathrm{Run}_{\mathcal{A}}$ or $\mathrm{wt}_{\mathcal{B}}(\rho) = \infty$, then (5.1) is obvious. Now assume that $\rho \in \mathrm{Run}_{\mathcal{B}} \setminus \mathrm{Run}_{\mathcal{A}}$ and $\mathrm{wt}_{\mathcal{B}}(\rho) < \infty$. Then, there exists a simple cycle $\theta = u_0 ... u_n \in (Q')^+$ such that ρ visits each of its transitions infinitely often. Note that $\mathcal{Q}_\theta^{\mathcal{B}} \cap F = \emptyset$ (otherwise, $\rho \in \mathrm{Run}_{\mathcal{A}}$). Since $\mathcal{B}$ is non-singular, either $C(\theta) > 0$ or $D(\theta) < 1$.

We distinguish the following cases.

- $D(\theta) < 1$. Then, there exists $j \in \{0, ..., n-1\}$ with $\lambda := \mathrm{wt}_d(u_j, u_{j+1}) < 1$. Let $(i_j)_{j \in \mathbb{N}} \in \mathbb{N}^\omega$ be an infinite sequence such that $i_0 < i_1 < ...$ and $(q_{i_k}, q_{i_{k+1}}) = (u_j, u_{j+1})$ for all $k \in \mathbb{N}$. Then, $D_{i_0}(\rho) \leq 1$ and, for all $k \in \mathbb{N}$, $D_{i_{k+1}}(\rho) \leq \lambda \cdot D_{i_k}(\rho)$ and hence $0 \leq D_{i_k}(\rho) \leq \lambda^k$. This means that $\lim_{k \to \infty} D_{i_k}(\rho) = 0$. Since the sequence $(D_i(\rho))_{i \in \mathbb{N}}$ is monotonically non-increasing, we obtain $\lim_{i \to \infty} D_i(\rho) = 0$.

- $C(\theta) > 0$. Then, there exists $j \in \{0, ..., n-1\}$ with $c := \mathrm{wt}_c(u_j, u_{j+1}) > 0$. Suppose that $\delta := \lim_{i\to\infty} D_i(\rho) > 0$. Since $(D_i(\rho))_{i\in\mathbb{N}}$ is a monotonically non-increasing sequence, $D_i(\rho) \geq \delta$ for all $i \in \mathbb{N}$. Let $(i_j)_{j\in\mathbb{N}} \in \mathbb{N}^\omega$ be an infinite sequence such that $i_0 < i_1 < ...$ and $(q_{i_k}, q_{i_{k+1}}) = (u_j, u_{j+1})$ for all $k \in \mathbb{N}$. Then,

$$\mathrm{wt}_{\mathcal{A}'}(\rho) \geq \sum_{k=0}^{\infty} \mathrm{wt}_c(q_{i_k}, q_{i_{k+1}}) \cdot D_{i_k}(\rho) \geq \sum_{k=0}^{\infty} c \cdot \delta = \infty.$$

  A contradiction.

Let $k \in \mathbb{N}$. Then, $q_k \xrightarrow{*} U_k$ for some reachable and accepting SCC U_k of $\mathcal{A}'$. We fix a state $f_k \in U \cap F$. Then, $q_k \xrightarrow{*}_{\mathcal{B}} f_k \xrightarrow{+}_{\mathcal{B}} f_k$. Let $\eta = p_0...p_l$ be a run fragment of $\mathcal{A}$ such that $l \leq |Q|$, $p_0 = q_k$ and $p_l = f_k$. Moreover, since U_k is non-trivial, there exists a cycle $\theta = v_0...v_s$ of $\mathcal{A}$ such that $s \leq 2\cdot|Q|$, $v_0 = v_s = f_k$ and $\mathrm{wt}_d(v_j, v_{j+1}) \leq \lambda$ for some $j \in \{0, ..., s-1\}$. We construct an accepting run $\varrho^{(k)} = (u_i)_{i\in\mathbb{N}} \in \mathrm{Run}_{\mathcal{A}}(w)$ as follows: $\varrho^{(k)} = q_0...q_k p_1...p_{l-1}(f_k v_1...v_{s-1})^\omega$. For every $i \in \mathbb{N}$, let $c_i = \mathrm{wt}_c(q_i, q_{i+1})$ and $c_i' = \mathrm{wt}_c(u_i, u_{i+1})$. Then

$$\begin{aligned}\left|\mathrm{wt}_{\mathcal{B}}(\rho) - \mathrm{wt}_{\mathcal{A}}(\varrho^{(k)})\right| &= \left|\sum_{i=k}^{\infty}(c_i \cdot D_i(\rho) - c_i' \cdot D_i(\varrho^{(k)}))\right| \\ &< \sum_{i=k}^{\infty} c_i \cdot D_i(\pi) + M \cdot \sum_{i=k}^{\infty} D_i(\varrho^{(k)}).\end{aligned}$$

Since $\mathrm{wt}_{\mathcal{B}}(\rho) < \infty$, there exists $k_0 \in \mathbb{N}$ such that $\sum_{i=k'}^{\infty} c_i \cdot D_i(\pi) < \frac{\varepsilon}{2}$ for all $k' \geq k_0$. Moreover,

$$\begin{aligned}\sum_{i=k}^{\infty} D_i(\varrho^{(k)}) &< D_k(\varrho^{(k)}) \cdot \left(|Q| + 2 \cdot |Q| \cdot \sum_{i=0}^{\infty} \lambda^i\right) \\ &= D_k(\varrho^{(k)}) \cdot |Q| \cdot \frac{3-\lambda}{1-\lambda}.\end{aligned}$$

Since ρ and $\varrho^{(k)}$ have the same prefix $q_0...q_k$, we obtain $D_k(\rho) = D_k(\varrho^{(k)})$. Let $R = |Q| \cdot \frac{3-\lambda}{1-\lambda}$. Then, $\sum_{i=k}^{\infty} D_i(\varrho^{(k)}) < R \cdot D_k(\rho)$. Since $\lim_{k\to\infty} D_k(\rho) = 0$, we have $\lim_{k\to\infty} \sum_{i=k}^{\infty} D_i(\varrho^{(k)}) = 0$. Then, there exists $k_1 \in \mathbb{N}$ such that $\sum_{i=k'}^{\infty} D_i(\varrho^{(k')}) < \frac{\varepsilon}{2M}$. Let $k_2 = \max\{k_0, k_1\}$. Then,

$$|\,\mathrm{wt}_{\mathcal{B}}(\rho) - \mathrm{wt}_{\mathcal{A}}(\varrho^{(k_2)})| < \frac{\varepsilon}{2} + M \cdot \frac{\varepsilon}{2M} = \varepsilon$$

and we let $\varrho = \varrho^{k_2}$.

Let $\varepsilon > 0$ be an arbitrary real number. Then, there exists $\rho \in \mathrm{Run}_{\mathcal{B}}$ such that $\mathrm{wt}_{\mathcal{B}}(\rho) < [\![\mathcal{B}]\!]^\sharp + \varepsilon$. By (5.1), there exists $\varrho \in \mathrm{Run}_{\mathcal{A}}$ such that $\mathrm{wt}_{\mathcal{A}}(\varrho) < \mathrm{wt}_{\mathcal{B}}(\rho) + \varepsilon$. Then:

$$0 \leq [\![\mathcal{A}]\!]^\sharp - [\![\mathcal{B}]\!]^\sharp < \mathrm{wt}_{\mathcal{A}}(\varrho) - \mathrm{wt}_{\mathcal{B}}(\rho) + \varepsilon < 2\varepsilon.$$

Since $\varepsilon > 0$ is arbitrary, we obtain $[\![\mathcal{A}]\!]^\sharp = [\![\mathcal{B}]\!]^\sharp$. $\square$

Proof of Theorem 5.16. By subsequent application of Lemmas 5.18 and 5.19, Corollary 5.21, Lemmas 5.22 and 5.23, and Theorem 5.15. $\square$

As a corollary from Theorem 5.16, we obtain:

Corollary 5.24. *It is decidable, given an alphabet Σ, a MWBA $\mathcal{A}$ over Σ and $\mathbb{V}^{\omega\textsc{Disc}}$, and a threshold $\theta \in \mathbb{Q}_{\geq 0}$, whether $[\![\mathcal{A}]\!](w) < \theta$ for some $w \in \Sigma^\omega$.*

Chapter 6

Weight assignment logic

Contents

The goal of this chapter is to give a logical characterization for multi-weighted Büchi automata. We develop a *weight assignment logic* on infinite words; this logic allows us to assign multi-weights to positions of an ω-word. First, we show that *unambiguous* sentences of our weight assignment logic, i.e., sentences without any existential quantifiers, are expressively equivalent to unambiguous multi-weighted Büchi automata. Thereafter we show that our weight assignment logic is equally expressive as nondeterministic multi-weighted Büchi automata. The proofs of these results are based on our Nivat-like decomposition results for unambiguous and nondeterministic multi-weighted Büchi automata.

6.1 Partial mappings

In this section, we present some notions about partial mappings which will be used for the semantics of our weight assignment logic.

Let X, Y be non-empty sets. A *partial mapping* $f : X \dashrightarrow Y$ is a mapping $f : X' \to Y$ where $X' \subseteq X$. We say that the set X' is the *domain* of f and denote it by $\mathrm{dom}(f)$. We say that f is *total* if $\mathrm{dom}(f) = X$. Let $[X \dashrightarrow Y]$ denote the collection of all partial mappings $f : X \dashrightarrow Y$. Let $Y^{\uparrow} = [\mathbb{N} \dashrightarrow Y]$. Note that $Y^{\omega} \subsetneq Y^{\uparrow}$. Let $\perp$ denote the *empty mapping* with $\mathrm{dom}(\perp) = \emptyset$.

We say that mappings $f_1, f_2 : X \dashrightarrow Y$ are *compatible*, written $f_1 \uparrow f_2$, if $f_1(x) = f_2(x)$ for all $x \in \mathrm{dom}(f_1) \cap \mathrm{dom}(f_2)$. Clearly, the relation $\uparrow$ is reflexive and symmetric. For $f_1, f_2 : X \dashrightarrow Y$ with $f_1 \uparrow f_2$, the *union* (or *merging*) $f_1 \cup f_2 : X \dashrightarrow Y$ is defined as:

- $\mathrm{dom}(f_1 \cup f_2) = \mathrm{dom}(f_1) \cup \mathrm{dom}(f_2)$;
- and $\mathrm{dom}(f_1 \cup f_2)(x) = f_i(x)$ whenever $x \in \mathrm{dom}(f_i)$ for $i \in \{1, 2\}$.

This definition is correct, since f_1 and f_2 are compatible. Note that the union itself can be considered as a partial operation $\cup : [X \dashrightarrow Y]^2 \dashrightarrow [X \dashrightarrow Y]$ with $\mathrm{dom}(\cup) = \{(f_1, f_2) \mid f_1 \uparrow f_2\}$.

Now let I be an index set (possibly infinite) and $(f_i)_{i\in I}$ an *I-family* of partial mappings $f_i : X \dashrightarrow Y$. We say that $(f_i)_{i\in I}$ is *compatible* if $f_i \uparrow f_j$ for all $i, j \in I$. In this case we define the *union* $f := \left(\bigcup_{i\in I} f_i\right) : X \dashrightarrow Y$ as:

- $\mathrm{dom}(f) = \bigcup_{i\in I} \mathrm{dom}(f_i)$;
- $f(x) = f_i(x)$ whenever $i \in \mathrm{dom}(f_i)$ for some $i \in I$.

For I-families of partial mappings, we can consider the union as a partial mapping $\bigcup_I : [X \dashrightarrow Y]^I \dashrightarrow [X \dashrightarrow Y]$ with $\mathrm{dom}(\bigcup_I) = \{(f_i)_{i\in I} \mid (f_i)_{i\in I} \text{ is compatible}\}$.

Let $f : X \dashrightarrow Y$, $x_0 \in X$ and $y_0 \in Y$. Then, $f[x_0/y_0] : X \dashrightarrow Y$ is defined as $\mathrm{dom}(f[x_0/y_0]) = \mathrm{dom}(f) \cup \{x_0\}$, $f(x_0) = y_0$ and $f[x_0/y_0](x) = f(x)$ for all $x \in \mathrm{dom}(f) \setminus \{x_0\}$.

Example 6.1. *Let $\Sigma = \{a, b\}$ be an alphabet and $f_1, f_2, f_3 \in \Sigma^{\uparrow}$ partial mappings defined as follows:*

- $\mathrm{dom}(f_1) = \mathbb{N}$ *and* $f_1(i) = a$ *for all* $i \in \mathrm{dom}(f_1)$*, i.e.,* $f_1 = a^\omega$*;*
- $\mathrm{dom}(f_2) \subseteq \mathbb{N}$ *is the set of all odd numbers and* $f_2(i) = b$ *for all* $i \in \mathrm{dom}(f_2)$*;*
- $\mathrm{dom}(f_3) \subseteq \mathbb{N}$ *is the set of all even numbers and* $f_3(i) = a$ *for all* $i \in \mathrm{dom}(f_3)$*.*

Then, $f_1 \uparrow f_3$, $f_3 \uparrow f_2$ but $\neg(f_1 \uparrow f_2)$. This shows in particular that the relation $\uparrow$ is, in general, not transitive. Moreover, $(f_1 \cup f_3) = a^\omega$, $(f_2 \cup f_3) = (ab)^\omega$, and $(f_1 \cup f_2)$ is undefined.

6.2 Unambiguous weight assignment logic

In this section, we introduce our *unambiguous weight assignment logic* on infinite words and we show that it is expressively equivalent to *unambiguous* MWBA.

We start with *MSO logic* on infinite words. Its syntax is defined exactly as for MSO logic on finite words. Let Σ be an alphabet and $\mathbf{MSO}(\Sigma)$ the set of all MSO formulas over Σ. Here, we will interpret a formula $\varphi \in \mathbf{MSO}(\Sigma)$ over an ω-word $w \in \Sigma^\omega$. Let $\mathrm{dom}(w) = \mathbb{N}$, the *domain* of w. As opposed to finite words, all ω-words have the same domain. We define w-assignments σ and updates $\sigma[x/i]$, $\sigma[X/I]$ as it was done for finite words. Let Σ^ω_V denote the set of all pairs (w, σ) where $w \in \Sigma^\omega$ and σ is a w-assignment.

Let $(w, \sigma) \in \Sigma^\omega_V$ and $\varphi \in \mathbf{MSO}(\Sigma)$. The definition that (w, σ) satisfies φ, denoted by $(w, \sigma) \models \varphi$, is given inductively on the structure of φ in the same manner as for finite words. If φ is a sentence, then we simply write $w \models \varphi$ and let $\mathcal{L}_\omega(\varphi) = \{w \in \Sigma^\omega \mid w \models \varphi\}$, the ω-language *defined* by φ. We say that an ω-language $\mathcal{L} \subseteq \Sigma^\omega$ is *MSO-definable* if there exists a sentence $\varphi \in \mathbf{MSO}(\Sigma)$ such that $\mathcal{L}_\omega(\varphi) = \mathcal{L}$.

Theorem 6.2 (Büchi [25], Elgot [50])**.** *Let Σ be an alphabet and $\mathcal{L} \subseteq \Sigma^\omega$ an ω-language. Then, $\mathcal{L}$ is recognizable iff $\mathcal{L}$ is MSO-definable.*

Now we lift this result to the setting of unambiguous multi-weighted Büchi automata.

Let Σ be an alphabet and $\mathbb{V} = (M, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ an ω-valuation structure. We also consider a designated element $\mathbb{1} \in M$ which we call the *default weight*. We denote the pair $(\mathbb{V}, \mathbb{1})$ by $\mathbb{V}_{\mathbb{1}}$.

The set $\mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ of formulas of *unambiguous weight assignment logic* over Σ and $\mathbb{V}_{\mathbb{1}}$ is given by the grammar:

$$\varphi \ ::= \ \beta \mid x \mapsto m \mid \beta\,?\,(\varphi : \varphi) \mid \varphi \sqcap \varphi \mid \textstyle\bigsqcap x.\varphi \mid \bigsqcap X.\varphi$$

where $\beta \in \mathbf{MSO}(\Sigma)$, $m \in M$, $x \in V_1$ and $X \in V_2$. Such a formula φ will be called an *unambiguous weight assignment formula*. Note that here we use the weighted If-Then-Else operator of [18]. Let $\textsc{Const}(\varphi) \subseteq M$ be the set of all elements $m \in M$ occurring in φ. The set $\mathrm{Free}(\varphi) \subseteq V$ of *free variables* of φ is defined as usual. We say that φ is a *sentence* if $\mathrm{Free}(\varphi) = \emptyset$.

Next, we will define the semantics of unambiguous weighted assignment formulas. Recall that $M^\uparrow = [\mathbb{N} \dashrightarrow M]$. Let $\textsc{Undef} \notin M^\uparrow$ which will mean the undefined value (the value $\textsc{Undef}$ should not be mixed up with the empty mapping $\perp$ which is defined). Using this undefined value, we can naturally extend the partial mapping $\bigcup_I : (M^\uparrow)^I \dashrightarrow M^\uparrow$ as the total mapping $\overline{\bigcup}_I : (M^\uparrow \cup \{\textsc{Undef}\})^I \to M^\uparrow \cup \{\textsc{Undef}\}$ as follows. Consider a family $(f_i)_{i\in I} \in (M^\uparrow \cup \{\textsc{Undef}\})^I$. Then, we put

$$\overline{\bigcup}_{i\in I} f_i = \begin{cases} \bigcup_{i\in I} f_i, & \text{if } (f_i)_{i\in I} \in (M^\uparrow)^I \cap \mathrm{dom}(\bigcup_I), \\ \textsc{Undef}, & \text{otherwise.} \end{cases}$$

In particular, if some f_i is $\textsc{Undef}$, then the union is also $\textsc{Undef}$. For simplicity, we will omit the overline and write $\bigcup$ instead of $\overline{\bigcup}$.

Now let $\varphi \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$. The *auxiliary semantics* of φ is the mapping $\langle\!\langle \varphi \rangle\!\rangle : \Sigma_V^\omega \to M^\uparrow \cup \{\textsc{Undef}\}$ defined for all $(w, \sigma) \in \Sigma_V^\omega$ inductively on the structure of φ as shown in Table 6.1. Here, $\varphi, \varphi_1, \varphi_2 \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$, $m \in M$, $x \in V_1$ and $X \in V_2$.

The *proper semantics* $[\![\varphi]\!] : \Sigma_V^\omega \to K$ operates on the auxiliary semantics $\langle\!\langle \varphi \rangle\!\rangle$ as follows. Let $(w, \sigma) \in \Sigma_V^\omega$.

- If $\langle\!\langle \varphi \rangle\!\rangle(w, \sigma) \in M^\uparrow$, then we assign the default weight $\mathbb{1}$ to all undefined positions and evaluate the obtained sequence using val^ω.
- If $\langle\!\langle \varphi \rangle\!\rangle(w, \sigma) = \textsc{Undef}$, then we put $[\![\varphi]\!](w, \sigma) = \mathbb{0}$.

Note that if $\varphi \in \mathbf{uWAL}(\Sigma, \mathbb{S})$ is a sentence, then the values $\langle\!\langle \varphi \rangle\!\rangle(w, \sigma)$ and $[\![\varphi]\!](w, \sigma)$ do not depend on σ and we consider the auxiliary semantics of φ as

$$
\begin{aligned}
\langle\!\langle \beta \rangle\!\rangle(w,\sigma) &= \begin{cases} \bot, & \text{if } (w,\sigma) \models \beta, \\ \text{undef}, & \text{otherwise} \end{cases} \\
\langle\!\langle x \mapsto m \rangle\!\rangle(w,\sigma) &= \bot[\sigma(x)/m] \\
\langle\!\langle \beta \,?\, (\varphi_1 : \varphi_2) \rangle\!\rangle(w,\sigma) &= \begin{cases} \langle\!\langle \varphi_1 \rangle\!\rangle(w,\sigma), & \text{if } (w,\sigma) \models \beta, \\ \langle\!\langle \varphi_2 \rangle\!\rangle(w,\sigma), & \text{otherwise} \end{cases} \\
\langle\!\langle \varphi_1 \sqcap \varphi_2 \rangle\!\rangle(w,\sigma) &= \langle\!\langle \varphi_1 \rangle\!\rangle(w,\sigma) \;\cup\; \langle\!\langle \varphi_2 \rangle\!\rangle(w,\sigma) \\
\langle\!\langle \textstyle\bigsqcap x.\varphi \rangle\!\rangle(w,\sigma) &= \bigcup_{i \in \mathrm{dom}(w)} \langle\!\langle \varphi \rangle\!\rangle(w,\sigma[x/i]) \\
\langle\!\langle \textstyle\bigsqcap X.\varphi \rangle\!\rangle(w,\sigma) &= \bigcup_{I \subseteq \mathrm{dom}(w)} \langle\!\langle \varphi \rangle\!\rangle(w,\sigma[X/I])
\end{aligned}
$$

Table 6.1: The auxiliary semantics of the $\mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$-formulas

the mapping $\langle\!\langle \varphi \rangle\!\rangle : \Sigma^\omega \to M^\uparrow \cup \{\text{Undef}\}$ and the proper semantics of φ as the quantitative ω-language $[\![\varphi]\!] : \Sigma^\omega \to K$. We say that a quantitative ω-language $\mathbb{L} : \Sigma^\omega \to K$ is *unambiguously definable* over $\mathbb{V}$ if there exist a default weight $\mathbb{1} \in M$ and a sentence $\varphi \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ such that $[\![\varphi]\!] = \mathbb{L}$.

Remark 6.3. *Note that the merging $\sqcap$ extends the classical conjunction: as it is easy to see, for $\beta_1, \beta_2, \beta \in \mathbf{MSO}(\Sigma)$ and $\mathcal{X} \in V_1 \cup V_2$, we have $\langle\!\langle \beta_1 \sqcap \beta_2 \rangle\!\rangle = \langle\!\langle \beta_1 \wedge \beta_2 \rangle\!\rangle$ and $\langle\!\langle \bigsqcap \mathcal{X}.\beta \rangle\!\rangle = \langle\!\langle \forall \mathcal{X}.\beta \rangle\!\rangle$ and hence $[\![\beta_1 \sqcap \beta_2]\!] = [\![\beta_1 \wedge \beta_2]\!]$ and $[\![\bigsqcap \mathcal{X}.\beta]\!] = [\![\forall \mathcal{X}.\beta]\!]$. Moreover, the undefined value* Undef *corresponds to the Boolean value* False *and $\bot$ corresponds to the Boolean value* True.

Example 6.4. *Let $\Sigma = \{a\}$, $\mathbb{V}^{\omega\text{Disc}} = (M, (K, +, \mathbb{0}), \mathrm{val}^{\omega\text{Disc}})$ the ω-valuation structure of Example 5.4 (b) and the default weight $\mathbb{1} = (0, 1)$. Let $\lambda \in \mathbb{Q} \cap (0, 1)$. Consider the $\mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}}^{\omega\text{Disc}})$-sentence $\varphi = \bigsqcap x.(x \mapsto (1, \lambda))$. Then, $\langle\!\langle \varphi \rangle\!\rangle(a^\omega) = (1, \lambda)^\omega$ and $[\![\varphi]\!](a^\omega) = 1 + \lambda + \lambda^2 + ... = \frac{1}{1-\lambda}$.*

For $\beta \in \mathbf{MSO}(\Sigma)$ and $\varphi \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$, let $\beta \Rightarrow \varphi$ denote the formula $\beta \,?\, (\varphi : \text{True})$.

Example 6.5. *Let $\Sigma = \{a, b\}$. Consider the quantitative ω-language $\mathbb{L} : \Sigma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$ defined for all $w \in \Sigma^\omega$ as follows. If $w = a^\omega$, then we put $\mathbb{L}(w) = \infty$. Otherwise, let $\mathbb{L}(w)$ be the number of the earliest position in w labelled by b. For this, we consider the ω-valuation structure $\mathbb{V}^{\omega\text{Ratio}} = (M, (K, +, \mathbb{0}), \mathrm{val}^{\omega\text{Ratio}})$ of Example 5.4 (a) and the default weight $\mathbb{1} = (0, 0)$. Then, $\mathbb{L}$ is be defined by the $\mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$-sentence*

$$\varphi = (\forall x.P_a(x)) \sqcap \textstyle\bigsqcap x.\bigsqcap y.[(P_b^{\min}(x) \wedge (y < x)) \Rightarrow ([x \mapsto (1,1)] \sqcap [y \mapsto (1,0)])])$$

where $P_b^{\min}(x) = P_b(x) \wedge \forall z.((z < x) \to P_a(x))$. Indeed, let $w = a^\omega$. Then, $\langle\!\langle \varphi \rangle\!\rangle(w) = \text{Undef}$ and hence $[\![\varphi]\!](w) = \infty$. Now let $w = a^{n-1}bw'$ where $n \geq 1$

and $w' \in \Sigma^\omega$. Then, $f := \langle\!\langle\varphi\rangle\!\rangle(w) : \mathbb{N} \dashrightarrow M$ is the partial mapping with $\mathrm{dom}(f) = \{1, ..., n\}$*, $f(i) = (1, 0)$ for all $i < n$ and $f(n) = (1, 1)$. Then,* $[\![\varphi]\!](w) = n$.

Now we state our main result about unambiguous weight assignment logic.

Theorem 6.6. *Let Σ be an alphabet, $\mathbb{V} = (M, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ an ω-valuation structure, and $\mathbb{L} : \Sigma^\omega \to K$ a quantitative ω-language. Then, the following are equivalent.*

- *(a) $\mathbb{L}$ is unambiguously definable over $\mathbb{V}$.*
- *(b) $\mathbb{L}$ is unambiguously recognizable over $\mathbb{V}$.*

Using our Nivat-like result for unambiguously recognizable quantitative ω-languages, namely Corollary 5.10, we will deduce Theorem 6.6 from Theorem 6.2. For this, it suffices to show the following.

Lemma 6.7. *Let Σ be an alphabet, $\mathbb{V} = (M, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ an ω-valuation structure, $\mathbb{1}$ a default weight, and $\mathbb{L} : \Sigma^\omega \to K$ a quantitative ω-language. Then, the following are equivalent.*

- *(a) $\mathbb{L} = [\![\varphi]\!]$ for some sentence $\varphi \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$.*
- *(b) There exist an alphabet Γ, renamings $h : \Gamma \to \Sigma$ and $g : \Gamma \to M$, and an MSO-definable and h-unambiguous ω-language $\mathcal{L} \subseteq \Gamma^\omega$ such that $\mathbb{L} = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$.*

The proof of this lemma will be given in the rest of this section. First, we show that (a) implies (b). Let $\varphi \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ be a sentence.

We fix a new symbol $\# \notin M$ which we will use to mark the positions where a partial mapping $f \in M^\uparrow$ is undefined. Let $\Delta_\varphi = \textsc{Const}(\varphi) \cup \{\#\}$. If $\eta : \mathbb{N} \dashrightarrow \textsc{Const}(\varphi)$ is a partial mapping (clearly, $\eta \in M^\uparrow$), we encode η as the ω-word $\mathrm{code}(\eta) = (b_i)_{i \in \mathbb{N}} \in \Delta_\varphi^\omega$ where, for all $i \in \mathbb{N}$, $b_i = \eta(i)$ if $i \in \mathrm{dom}(\eta)$ and $b_i = \#$ otherwise. For any ω-word $w = (a_i)_{i \in \mathbb{N}} \in \Sigma^\omega$, we encode the pair (w, η) as the ω-word $\mathrm{code}(w, \eta) = ((a_i, b_i))_{i \in \mathbb{N}} \in (\Sigma \times \Delta_\varphi)^\omega$.

The following lemma will help us to construct a required MSO-definable and h-unambiguous ω-language $\mathcal{L} \subseteq \Gamma^\omega$.

Lemma 6.8. *Let $\varphi \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$. Then, there exists a formula $\zeta \in \mathbf{MSO}(\Sigma \times \Delta_\varphi)$ such that $\mathrm{Free}(\zeta) = \mathrm{Free}(\varphi)$ and, for all $(w, \sigma) \in \Sigma_V^\omega$ and all partial functions $\eta : \mathbb{N} \dashrightarrow \textsc{Const}(\varphi)$, the following holds:*

$$\langle\!\langle\varphi\rangle\!\rangle(w, \sigma) = \eta \quad \textit{iff} \quad (\mathrm{code}(w, \eta), \sigma) \models \zeta.$$

Note that $\langle\!\langle\varphi\rangle\!\rangle(w, \sigma) = \eta$ means in particular that $\langle\!\langle\varphi\rangle\!\rangle(w, \sigma) \neq \textsc{Undef}$.

Proof. For each $\beta \in \mathbf{MSO}(\Sigma)$, let $\beta^* \in \mathbf{MSO}(\Sigma \times \Delta_\varphi)$ be the formula obtained from β by replacing each subformula $P_a(x)$ of β by the formula $\bigvee\limits_{m \in \Delta_\varphi} P_{(a,m)}(x)$.

Let $Y \in V_2$ be a fresh variable which does not occur in φ. For each subformula ψ of φ, we construct the formula $r_Y(\psi) \in \mathbf{MSO}(\Sigma \times \Delta_\varphi)$ inductively on the structure of ψ as follows.

- Let $\psi = \beta \in \mathbf{MSO}(\Sigma)$. Then, we put $r_Y(\psi) = \beta^* \wedge (Y = \emptyset)$.
- Let $\psi = (x \mapsto m)$ with $m \in M$ and $x \in V_1$. Then, we put

$$r_Y(\psi) = \left(\bigvee_{a \in \Sigma} P_{(a,m)}(x) \right) \wedge (Y = \{x\}).$$

- Let $\psi = \beta\,?\,(\psi_1 : \psi_2)$ where $\beta \in \mathbf{MSO}(\Sigma)$. Then, we put

$$r_Y(\psi) = (\beta^* \wedge r_Y(\psi_1)) \vee ((\neg\beta^*) \wedge r_Y(\psi_2)).$$

- Let $\psi = \psi_1 \sqcap \psi_2$. Then, we put

$$r_Y(\psi) = \exists Y_1.\exists Y_2.(r_{Y_1}(\psi_1) \wedge r_{Y_2}(\psi_2) \wedge (Y = Y_1 \cup Y_2))$$

  where $Y_1, Y_2 \in V_2$ are two variables which do not occur in ψ and $Y_1 \neq Y_2$, $Y_1 \neq Y$, $Y_2 \neq Y$.
- Let $\psi = \bigsqcap \mathcal{X}.\psi'$ where $\mathcal{X} \in V_1 \cup V_2$. Let

$$\chi(Y) = \forall x.\exists Y'.(r_{Y'}(\psi') \wedge (Y' \subseteq Y))$$

  where $Y' \in V_2$ is a variable with $Y' \neq Y$ which does not occur in ψ. Then, we put

$$r_Y(\psi) = \chi(Y) \wedge \forall Z.(\chi(Z) \to (Y \subseteq Z)).$$

  where $Z \in V_2$ does not occur in ψ and $Z \notin \{Y, Y'\}$.

Let $w \in \Sigma^\omega$ and $\eta : \mathbb{N} \dashrightarrow \textsc{Const}(\varphi)$ a partial mapping. For $R \subseteq \mathbb{N}$, let $\eta|_R : \mathbb{N} \dashrightarrow \textsc{Const}(\varphi)$ be defined as $\mathrm{dom}(\eta|_R) = R \cap \mathrm{dom}(\eta)$ and $\eta|_R(i) = \eta(i)$ for all $i \in \mathrm{dom}(\eta|_R)$. We show by induction on the structure of ψ that, for all w-assignments σ, we have:

$$(\mathrm{code}(w,\eta),\sigma) \models r_Y(\psi) \ \text{ iff } \ \sigma(Y) \subseteq \mathrm{dom}(\eta) \text{ and } \langle\!\langle \psi \rangle\!\rangle(w,\sigma) = \eta|_{\sigma(Y)}. \quad (6.1)$$

- Let $\psi = \beta \in \mathbf{MSO}(\Sigma)$.
  - Assume that $(\mathrm{code}(w,\eta),\sigma) \models r_Y(\psi)$. Then, $(\mathrm{code}(w,\eta),\sigma) \models \beta^*$ and $\sigma(Y) = \emptyset$ which implies $(w,\sigma) \models \beta$ and $\sigma(Y) = \emptyset$. Hence, $\langle\!\langle \psi \rangle\!\rangle = \bot = \eta|_\emptyset$ and $\sigma(Y) = \emptyset \subseteq \mathrm{dom}(\eta)$.
  - Conversely, assume that $\sigma(Y) \subseteq \mathrm{dom}(\eta)$ and $\langle\!\langle \beta \rangle\!\rangle(w,\sigma) = \eta|_{\sigma(Y)}$. Then, $(w,\sigma) \models \beta$ and $\eta|_{\sigma(Y)} = \bot$. Since $\sigma(Y) \subseteq \mathrm{dom}(\eta)$, we obtain $\sigma(Y) = \emptyset$. Furthermore, $(\mathrm{code}(w,\eta),\sigma) \models \beta^*$. Then, $(\mathrm{code}(w,\eta),\sigma) \models r_Y(\beta)$.
- Let $\psi = (x \mapsto m)$ where $m \in \textsc{Const}(\varphi)$ and $x \in V_1$.
  - Assume that $(\mathrm{code}(w,\eta),\sigma) \models r_Y(\psi)$. Then, $\sigma(x) \in \mathrm{dom}(\eta)$, $\eta(\sigma(x)) = m$ and $\sigma(Y) = \{\sigma(x)\}$. Then, $\langle\!\langle \psi \rangle\!\rangle(w,\sigma) = \bot[\sigma(x)/m] = \eta|_{\sigma(Y)}$ and $\sigma(Y) = \{\sigma(x)\} \subseteq \mathrm{dom}(\eta)$.

  - Conversely, assume that the right hand side of (6.1) holds true. Then, $\eta|_{\sigma(Y)} = \bot[\sigma(x)/m]$. Since $\sigma(Y) \subseteq \mathrm{dom}(\eta)$, we have $\sigma(Y) = \{\sigma(x)\}$. Moreover, $\eta(\sigma(x)) = m$. Then, the left hand side of (6.1) holds true.

- Let $\psi = \beta\,?\,(\psi_1 : \psi_2)$ where $\beta \in \mathbf{MSO}(\Sigma)$.
  - Assume that the left hand side of (6.1) holds true. Then, one of the following situations is possible:
    - $(w, \sigma) \models \beta$ and $(\mathrm{code}(w, \eta), \sigma) \models r_Y(\psi_1)$. Then, by induction hypothesis, $\sigma(Y) \subseteq \mathrm{dom}(\eta)$ and $\langle\!\langle \psi_1 \rangle\!\rangle(w, \sigma) = \eta|_{\sigma(Y)}$. Since $\langle\!\langle \psi \rangle\!\rangle(w, \sigma) = \langle\!\langle \psi_1 \rangle\!\rangle(w, \sigma)$, the right hand side of (6.1) holds true.
    - $(w, \sigma) \not\models \beta$ and $(\mathrm{code}(w, \eta), \sigma) \models r_Y(\psi_2)$. This case is similar to the previous one.
  - Conversely, assume that the right hand side of (6.1) holds. Again, we distinguish between two cases.
    - $(w, \sigma) \models \beta$. Then, $\langle\!\langle \psi_1 \rangle\!\rangle(w, \sigma) = \langle\!\langle \psi \rangle\!\rangle(w, \sigma) = \eta|_{\sigma(Y)}$. By induction hypothesis for ψ_1, $(\mathrm{code}(w, \eta), \sigma) \models r_Y(\psi_1)$. Since $(\mathrm{code}(w, \eta), \sigma) \models \beta^*$, the left hand side of (6.1) holds.
    - $(w, \sigma) \not\models \beta$. This case is similar to the previous one.

- Let $\psi = \psi_1 \sqcap \psi_2$.
  - Assume that the left hand side of (6.1) holds. Then, there exist $R_1, R_2 \subseteq \mathrm{dom}(w)$ such that:
    - $\sigma(Y) = R_1 \cup R_2$,
    - $(\mathrm{code}(w, \eta), \sigma[Y_1/R_1]) \models r_{Y_1}(\psi_1)$,
    - $(\mathrm{code}(w, \eta), \sigma[Y_2/R_2]) \models r_{Y_2}(\psi_2)$.

    Then, by induction hypothesis for ψ_1 and ψ_2: $R_1 \subseteq \mathrm{dom}(\eta)$, $\langle\!\langle \psi_1 \rangle\!\rangle(w, \sigma) = \eta|_{R_1}$, $R_2 \subseteq \mathrm{dom}(\eta)$ and $\langle\!\langle \psi_2 \rangle\!\rangle(w, \sigma) = \eta|_{R_2}$. Hence, $\sigma(Y) \subseteq \mathrm{dom}(\eta)$ and, since $\eta|_{R_1}$ and $\eta|_{R_2}$ are compatible partial mappings, we have: $\langle\!\langle \psi_1 \sqcap \psi_2 \rangle\!\rangle(w, \sigma) = \eta|_{R_1} \cup \eta|_{R_2} = \eta|_{\sigma(Y)}$. This shows that the right hand side of (6.1) also holds true.
  - Conversely, assume that the right hand side of (6.1) holds. Let $\eta_1 = \langle\!\langle \psi_1 \rangle\!\rangle(w, \sigma)$ and $\eta_2 = \langle\!\langle \psi_2 \rangle\!\rangle(w, \sigma)$. Then, $\eta|_{\sigma(Y)} = \eta_1 \cup \eta_2$. Moreover, there exist $R_1, R_2 \subseteq \mathrm{dom}(w)$ such that:
    - $R_1 \cup R_2 = \sigma(Y)$,
    - $\eta_1 = \eta|_{R_1}$ and $\eta_2 = \eta|_{R_2}$.

    Since $R_1, R_2 \subseteq \sigma(Y) \subseteq \mathrm{dom}(\eta)$, by induction hypothesis, $(\mathrm{code}(w, \eta), \sigma[Y_i/R_i]) \models r_Y(\psi_i)$ for $i \in \{1, 2\}$. Since Y_2 does not occur in $r_Y(\psi_1)$ and Y_1 does not occur in $r_Y(\psi_2)$, we have: $(\mathrm{code}(w, \eta), \sigma[Y_1/R_1, Y_2/R_2]) \models r_Y(\psi_1)$ and $(\mathrm{code}(w, \eta), \sigma[Y_1/R_1, Y_2/R_2]) \models r_Y(\psi_2)$. Then, the left hand side of (6.1) holds.

- Let $\psi = \prod x.\psi'$ where $x \in V_1$. Let $\chi = \forall x.\exists Y'.(r_{Y'}(\psi') \wedge (Y' \subseteq Z))$.

– Assume that $(\text{code}(w,\eta),\sigma) \models r_Y(\psi)$. Then, $(\text{code}(w,\eta),\sigma) \models \chi(Y)$. This means that, for all $i \in \text{dom}(w)$, there exists $R_i \subseteq \sigma(Y)$ such that $(\text{code}(w,\eta),\sigma[x/i, Y'/R_i]) \models r'_Y(\psi')$. Then, by induction hypothesis, for all $i \in \text{dom}(w)$, $R_i \subseteq \text{dom}(\eta)$ and $\langle\!\langle\psi'\rangle\!\rangle(w,\sigma[x/i]) = \eta|_{R_i}$. Let $R = \bigcup\limits_{i\in\text{dom}(w)} R_i$. Then, $(\text{code}(w,\eta),\sigma[Z/R]) \models \chi(Z)$. Since $(\text{code}(w,\eta),\sigma) \models \forall Z.(\chi(Z) \to (Y \subseteq Z))$, we obtain: $\sigma(Y) \subseteq R$. Hence, $R = \sigma(Y)$ and

$$\langle\!\langle\psi\rangle\!\rangle(w,\sigma) = \bigcup_{i\in\text{dom}(w)} \eta|_{R_i} = \eta|_R = \eta|_{\sigma(Y)}.$$

Finally, $\sigma(Y) = \bigcup\limits_{i\in\text{dom}(w)} R_i \subseteq \text{dom}(\eta)$. This shows that the right hand side of (6.1) holds true.

– Conversely, assume that the right hand side of (6.1) holds. Then, there exists a family $(R_i)_{i\in\text{dom}(w)}$ with $R_i \subseteq \sigma(Y) \subseteq \text{dom}(\eta)$ for each $i \in \text{dom}(w)$, such that $\bigcup_{i\in\text{dom}(w)} R_i = \sigma(Y)$ and, for all $i \in \text{dom}(w)$, $\langle\!\langle\psi'\rangle\!\rangle(w,\sigma[x/i]) = \eta|_{R_i}$. Then, using the induction hypothesis, it is easy to see that, for all $i \in \text{dom}(w)$, $(\text{code}(w,\eta),\sigma[x/i, Y'/R_i]) \models r_{Y'}(\psi')$. Hence, $(\text{code}(w,\eta),\sigma) \models \chi(Y)$. It remains to show that

$$(\text{code}(w,\eta),\sigma) \models \forall Z.(\chi(Z) \to (Y \subseteq Z)).$$

Indeed, let $Q \subseteq \text{dom}(w)$ with $(\text{code}(w,\eta),\sigma[Z/Q]) \models \chi(Z)$. Then, for all $i \in \text{dom}(w)$, there exists a subset $Q_i \subseteq Q$ with $(\text{code}(w,\eta),\sigma[x/i, Y'/Q_i]) \models r_{Y'}(\psi')$. Then, by induction hypothesis, for all $i \in \text{dom}(w)$, $Q_i \subseteq \text{dom}(\eta)$ and

$$\eta|_{Q_i} = \langle\!\langle\psi'\rangle\!\rangle(w,\sigma[x/i]) = \eta|_{R_i}.$$

Hence, $Q_i = R_i$ for all $i \in \text{dom}(w)$, and

$$\sigma(Y) = \bigcup_{i\in\text{dom}(w)} R_i = \bigcup_{i\in\text{dom}(w)} Q_i \subseteq Q.$$

– The proof for $\psi = \prod X.\psi'$ with $X \in V_2$ is completely analogous to the proof of the previous case. The difference is that we consider "for all $I \subseteq \text{dom}(w)$" instead of "for all $i \in \text{dom}(w)$".

Finally, we construct the desired formula ζ as

$$\zeta = \exists Y. \left(r_Y(\varphi) \wedge \forall x. \left((\neg Y(x)) \to \bigvee\nolimits_{a\in\Sigma} P_{(a,\#)}(x) \right) \right).$$

Assume that $\langle\!\langle\varphi\rangle\!\rangle(w,\sigma) = \eta$. Let $R = \text{dom}(\eta)$ and consider $\sigma' = \sigma[Y/R]$. Then, $\sigma'(Y) \subseteq \text{dom}(\eta)$ and $\langle\!\langle\varphi\rangle\!\rangle(w,\sigma) = \eta|_{\sigma(Y)}$. Then, by (6.1), we have: $(\text{code}(w,\eta),\sigma') \models r_Y(\varphi)$. Moreover, for all $i \in \text{dom}(w) \setminus \sigma'(Y)$, $\eta(i)$ is undefined and hence $(\text{code}(w,\eta),\sigma') \models \forall x.(\neg Y(x) \to \bigvee_{a\in\Sigma} P_{(a,\#)}(x))$ which implies $(\text{code}(w,\eta),\sigma) \models \zeta$.

Conversely, let $(\mathrm{code}(w,\eta),\sigma) \models \zeta$. Then, there exists $R \subseteq \mathrm{dom}(w)$ such that $(\mathrm{code}(w,\eta),\sigma[Y/R]) \models r_Y(\varphi)$ and, for all $i \notin R$, $\eta(i)$ is undefined. By (6.1), we have: $R \subseteq \mathrm{dom}(\eta)$ and $\langle\!\langle\varphi\rangle\!\rangle(w,\sigma) = \eta|_R$. Since $\eta(i)$ is undefined for all $i \in \mathrm{dom}(w) \setminus R$, we have $\eta|_R = \eta$ and hence the claim follows. This finishes the proof of this lemma. □

The next lemma finishes the proof of the part (a) ⇒ (b) of Lemma 6.7. Recall that $\varphi \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ is a sentence.

Lemma 6.9. *There exist an alphabet* Γ*, renamings* $h : \Gamma \to \Sigma$ *and* $g : \Gamma \to M$*, and an MSO-definable and* h*-unambiguous* ω*-language* $\mathcal{L} \subseteq \Gamma^\omega$ *such that* $[\![\varphi]\!] = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$.

Proof. Let $\Gamma = \Sigma \times \Delta_\varphi$. Let $h : \Gamma \to \Sigma$ be defined for all $a \in \Sigma$ and $b \in \Delta_\varphi$ by $h(a,b) = a$. Let $g : \Gamma \to M$ be defined for all $a \in \Sigma$ and $b \in \Delta_\varphi$ by

$$g(a,b) = \begin{cases} b, & \text{if } b \in \textsc{Const}(\varphi), \\ \mathbb{1}, & \text{if } b = \#. \end{cases}$$

By Lemma 6.8, there exists a sentence $\zeta \in \mathbf{MSO}(\Gamma)$ such that $\mathcal{L}_\omega(\zeta) = \{\mathrm{code}(w,\eta) \mid w \in \mathrm{dom}(\langle\!\langle\varphi\rangle\!\rangle) \text{ and } \eta = \langle\!\langle\varphi\rangle\!\rangle(w)\}$. Note that, for each $w \in \Sigma^\omega$, there exists at most one $u \in \mathcal{L}_\omega(\zeta)$ with $h(u) = w$. Then, $\mathcal{L}_\omega(\zeta)$ is h-unambiguous. Let $\mathcal{L} = \mathcal{L}_\omega(\zeta)$. Then, $[\![\varphi]\!] = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$. Indeed, let $w \in \Sigma^\omega$. Then, we distinguish between the following two cases:

- $\langle\!\langle\varphi\rangle\!\rangle(w) = \textsc{Undef}$. Then, $[\![\varphi]\!](w) = \mathbb{0}$. On the other side, by definition of $\mathcal{L}$, there exists no $u \in \mathcal{L}$ with $h(u) = w$. Then, $h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})(w) = \mathbb{0}$ and hence $[\![\varphi]\!](w) = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})(w)$.
- $\langle\!\langle\varphi\rangle\!\rangle(w) \in M^\uparrow$. Then, since the mapping g assigns the default weight $\mathbb{1}$ to the undefined positions of $\langle\!\langle\varphi\rangle\!\rangle(w) \in M^\uparrow$ and $\mathcal{L}$ is h-unambiguous, we have $[\![\varphi]\!](w) = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})(w)$.

□

Now we turn to the implication (b) ⇒ (a) of Lemma 6.7; this result is used for the proof of the implication (b) ⇒ (a) of Theorem 6.6. Clearly, we can alternatively give a direct translation of an unambiguous MWBA into an **uWAL**-sentence. But our approach based on Nivat Theorem could be interesting since here we do not have to describe explicitly the behavior of the underlying Büchi automaton; it suffices to show that we can describe renamings and intersections.

Lemma 6.10. *Let* Γ *be an alphabet,* $h : \Gamma \to \Sigma$ *and* $g : \Gamma \to M$ *renamings, and* $\mathcal{L} \subseteq \Gamma^\omega$ *an* h*-unambiguous and* $\mathbf{MSO}(\Gamma)$*-definable* ω*-language. Then, there exist* $\mathbb{1} \in M$ *and a sentence* $\varphi \in \mathbf{WAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ *such that* $[\![\varphi]\!] = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$.

Proof. Let $\beta \in \mathbf{MSO}(\Gamma)$ be a sentence with $\mathcal{L}_\omega(\beta) = \mathcal{L}$ and $\mathcal{V} = \{X_\gamma \mid \gamma \in \Gamma\} \subseteq V_2$ a set of pairwise distinct variables not occurring in β. We will use them to describe the renaming h: each variable X_γ corresponds to the set of positions which were labelled by γ before the renaming). In order to transform β into an $\mathbf{MSO}(\Sigma)$-formula and not to lose the information about the labels of positions before the renaming, we introduce for each formula $\zeta \in \mathbf{MSO}(\Gamma)$, the *renaming* $h(\zeta) \in \mathbf{MSO}(\Sigma)$ obtained from ζ by replacing every predicate $P_\gamma(x)$ occurring in γ (with $\gamma \in \Gamma$ and $x \in V_1$) by the formula $P_{h(\gamma)}(x) \wedge X_\gamma(x)$.

For an ω-word $u = (\gamma_i)_{i\in\mathbb{N}}$, let $\sigma_u : \mathcal{V} \to 2^{\mathrm{dom}(u)}$ be defined for all $\gamma \in \Gamma$ as $\sigma_u(X_\gamma) = \{i \in \mathrm{dom}(u) \mid \gamma_i = \gamma\}$. It can be easily shown by induction on the structure of β that, for all ω-words $u \in \Gamma^\omega$ and all u-assignments σ with $\sigma|_\mathcal{V} = \sigma_u$, we have:

$$(u, \sigma) \models \beta \text{ iff } (h(u), \sigma) \models h(\beta). \tag{6.2}$$

Then, in order to describe the renamings, we additionaly define the following $\mathbf{MSO}(\Sigma)$-formulas.

- Let
$$\textsc{Partition} = \forall x.\Bigg(\bigvee_{\gamma\in\Gamma} X_\gamma(x) \wedge \bigwedge_{\substack{\gamma'\in\Gamma,\\ \gamma'\neq\gamma}} \neg X_{\gamma'}(x)\Bigg). \tag{6.3}$$
Note that this formula demands that the values of $\mathcal{V}$-variables form a partition of the domain of an ω-word (because every position of a word must be labelled exactly by one letter).
- Let
$$\textsc{Renaming} = \forall x.\Bigg(\bigvee_{\gamma\in\Gamma} (X_\gamma(x) \wedge P_{h(\gamma)}(x))\Bigg). \tag{6.4}$$
Note that the formula Renaming demands that whenever a position was labelled by γ before the renaming, then it must be labelled by $h(\gamma)$ after the renaming.

Let Boolean $=$ Partition $\wedge$ Renaming $\wedge\ h(\beta)$. Note that Boolean $\in$ $\mathbf{MSO}(\Sigma)$ and Free(Boolean) $= \mathcal{V}$.

Let $x \in V_1$ be a fresh variable and $\mathbb{1} \in M$ be defined arbitrarily. Let $(\gamma_i)_{1\leq i\leq|\Gamma|}$ be an enumeration of Γ. Now, we will assign weights to the positions of an ω-word using $g : \Gamma \to M$. For this, we apply the formula $\chi = \mathbf{uWAL}(\Sigma, \mathbb{V}_\mathbb{1})$ defined as follows. We let $\chi = \prod x.\chi_1$ where, for $i \in \{1, ..., \Gamma\}$, $\chi_i \in \mathbf{uWAL}(\Sigma, \mathbb{V}_\mathbb{1})$ is defined inductively as follows.

- For $i = |\Gamma|$, we let $\chi_{|\Gamma|} = (x \mapsto g(\gamma_{|\Gamma|}))$.
- Let $i \in \{1, ..., |\Gamma|\}$ and assume that χ_{i+1} is defined. Then, we let $\chi_i = X_{\gamma_i}(x)\,?\,((x \mapsto g(\gamma_i)) : \chi_{i+1})$.

Then, we define the desired sentence $\varphi \in \mathbf{uWAL}(\Sigma, \mathbb{V}_\mathbb{1})$ as

$$(\exists X_{\gamma_1}. \ldots \exists X_{\gamma_{|\Gamma|}}.\textsc{Boolean}) \sqcap \prod X_{\gamma_1}. \ldots \prod X_{\gamma_{|\Gamma|}}.(\textsc{Boolean}\,?\,(\chi : \textsc{True})).$$

We show that $[\![\varphi]\!] = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$. Let $w \in \Sigma^\omega$. We distinguish between the following two cases.

- There exists no $u \in \mathcal{L}$ such that $h(u) = w$. Then, by (6.2), we have $w \not\models \exists X_{\gamma_1}. \dots \exists X_{\gamma_{|\Gamma|}}.\textsc{Boolean}$ and hence $\langle\!\langle \varphi \rangle\!\rangle(w) = \textsc{Undef}$. Hence, $[\![\varphi]\!](w) = \mathbb{0} = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})(w)$.
- There exists $u \in \mathcal{L}$ such that $h(u) = w$. Since $\mathcal{L}$ is h-unambiguous, there exists exactly one such u. Note that, by (6.2), for each w-assignment σ, $(w, \sigma) \models \textsc{Boolean}$ iff $\sigma|_{\mathcal{V}} = \sigma_u$. Then, $w \models \exists X_{\gamma_1}. \dots \exists X_{\gamma_{|\Gamma|}}.\textsc{Boolean}$ and
$$[\![\varphi]\!](w) = \mathrm{val}^\omega(g(u)) = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L}).$$

□

Proof of Lemma 6.7. The implication (a) ⇒ (b) is shown in Lemma 6.9. The implication (b) ⇒ (a) is shown in Lemma 6.10. □

Proof of Theorem 6.6. Immediate by Lemma 6.7, Corollary 5.10 and Theorem 6.2. □

6.3 Weight assignment logic

In this section, we establish a Büchi-Elgot theorem for *nondeterministic* multi-weighted Büchi automata, i.e., we introduce a *weight assignment logic* which is expressively equivalent to them.

Let Σ be an alphabet, $\mathbb{V} = (M, (K, +, \mathbb{0}), \mathrm{val}^\omega)$ an ω-valuation structure, and $\mathbb{1} \in M$ a default weight. The set $\mathbf{WAL}(\Sigma, \mathbb{V}_\mathbb{1})$ of formulas of *weight assignment logic* (WAL) over Σ and $\mathbb{V}_\mathbb{1}$ is defined by the grammar

$$\varphi \ ::= \ \psi \mid \sqcup x.\varphi \mid \sqcup X.\varphi$$

where $\psi \in \mathbf{uWAL}(\Sigma, \mathbb{V}_\mathbb{1})$, $x \in V_1$ and $X \in V_2$. Here, $\sqcup x$ and $\sqcup X$ are quantitative versions of the existential quantifier. For $\varphi \in \mathbf{WAL}(\Sigma, \mathbb{V}_\mathbb{1})$, the *semantics* $[\![\varphi]\!] : \Sigma_V^\omega \to K$ is defined as follows.

- Let $\varphi = \psi \in \mathbf{uWAL}(\Sigma, \mathbb{V}_\mathbb{1})$. Then, $[\![\psi]\!]$ is defined as for unambiguous weight assignment formulas in the previous chapter.
- Let $\varphi = \sqcup x.\varphi$ with $x \in V_1$. Then, for all $(w, \sigma) \in \Sigma_V^\omega$, we let
$$[\![\sqcup x.\varphi]\!](w, \sigma) = \sum_{i \in \mathrm{dom}(w)} [\![\varphi]\!](w, \sigma[x/i]).$$
- Let $\varphi = \sqcup X.\varphi$ with $X \in V_2$. Then, for all $(w, \sigma) \in \Sigma_V^\omega$, we let
$$[\![\sqcup X.\varphi]\!](w, \sigma) = \sum_{I \subseteq \mathrm{dom}(w)} [\![\varphi]\!](w, \sigma[X/I]).$$

Given a formula $\varphi \in \mathbf{WAL}(\Sigma, \mathbb{V}_\mathbb{1})$, the set $\mathrm{Free}(\varphi) \subseteq V$ of free variables of φ is defined as usual. We say that φ is a *sentence* if $\mathrm{Free}(\varphi) = \emptyset$. If φ is a sentence, then we can eliminate the w-assignments from the domain of $[\![\varphi]\!]$ and consider

$[\![\varphi]\!]$ as a quantitative ω-language $[\![\varphi]\!] : \Sigma^\omega \to K$. We say that a *quantitative ω-language* $\mathbb{L} : \Sigma^\omega \to K$ is **WAL**-*definable* over $\mathbb{V}$ if there exist $\mathbb{1} \in M$ and a sentence $\varphi \in \mathbf{WAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ such that $[\![\varphi]\!] = \mathbb{L}$.

Let $\mathcal{V} = \{x_1, ..., x_k, X_1, ..., X_l\} \subseteq V$ be a set of pairwise distinct variables and $\psi \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$. We denote by $\sqcup\mathcal{V}.\psi$ the $\mathbf{WAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$-formula $\sqcup x_1. ... \sqcup x_k.\sqcup X_1. ... \sqcup X_l.\psi$.

Remark 6.11. *Let $\mathcal{V} \subseteq V$ be a finite set of variables, $\psi \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ and $\varphi = \sqcup\mathcal{V}.\psi$. Let $x \in V_1 \setminus \mathcal{V}$ be a variable not appearing in ψ. Then:*

$$[\![\varphi]\!] = [\![\sqcup(\mathcal{V} \cup \{x\}).(\psi \sqcap \min(x))]\!]$$

where $\min(x)$ *is an abbreviation for* $\forall y.(x \leq y)$.

Now let $X \in V_2 \setminus \mathcal{V}$ be a variable not appearing in ψ. Then:

$$[\![\varphi]\!] = [\![\sqcup(\mathcal{V} \cup \{X\}).(\psi \sqcap X = \emptyset)]\!]$$

where $X = \emptyset$ is an abbreviation for $\forall y.\neg X(y)$.

This shows we can transform arbitrary formulas $\varphi_1 = \sqcup\mathcal{V}_1.\psi_1$ and $\varphi_2 = \sqcup\mathcal{V}_2.\psi_2$ with $\psi_1, \psi_2 \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ to the form with $\mathcal{V}_1 = \mathcal{V}_2$.

Remark 6.12. *We consider several additional natural logical operators can be defined using* **WAL**-*formulas.*

(a) Consider a weighted If-Then-Else formula $\beta\ ?\ (\varphi_1 : \varphi_2)$ (cf. [18]) where $\beta \in \mathbf{MSO}(\Sigma)$ and $\varphi_1, \varphi_2 \in \mathbf{WAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$, whose semantics is defined for all $(w, \sigma) \in \Sigma_V^\omega$ as

$$[\![\beta\,?\,(\varphi_1 : \varphi_2)]\!](w, \sigma) = \begin{cases} [\![\varphi_1]\!](w, \sigma), & \text{if } (w, \sigma) \models \beta, \\ [\![\varphi_2]\!](w, \sigma), & \text{otherwise.} \end{cases}$$

Note that, in general, the formula $\beta\ ?\ (\varphi_1 : \varphi_2)$ is not in $\mathbf{WAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$. By Remark 6.11, we may assume that $\varphi_1 = \sqcup\mathcal{V}.\psi_1$ and $\varphi_2 = \sqcup\mathcal{V}.\psi_2$ where $\mathcal{V} \subseteq V$ is a finite set of variables and $\psi_1, \psi_2 \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$. We may also assume that the $\mathcal{V}$-variables do not appear in β. Then, the formula $\beta\ ?\ (\varphi_1 : \varphi_2)$ can be simulated by the $\mathbf{WAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$-formula $\sqcup\mathcal{V}.(\beta\ ?\ (\psi_1 : \psi_2))$. Note that $(\beta\ ?\ \psi_1 : \psi_2) \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$. For the sake of readability, we will sometimes denote the formula $\beta\,?\,(\varphi_1 : \varphi_2)$ by
$\beta\ ?\ \begin{cases} \varphi_1 \\ \varphi_2. \end{cases}$

(b) For $\varphi_1, \varphi_2 \in \mathbf{WAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$, consider the new formula $\varphi_1 \sqcup \varphi_2$ whose semantics is defined for all $(w, \sigma) \in \Sigma_V^\omega$ as

$$[\![\varphi_1 \sqcup \varphi_2]\!](w, \sigma) = [\![\varphi_1]\!](w, \sigma) + [\![\varphi_2]\!](w, \sigma).$$

Note that the formula $\varphi_1 \sqcup \varphi_2$ can be simulated by the $\mathbf{WAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$-formula

$$\sqcup X.(X = \emptyset)? \begin{cases} \varphi_1 \\ (X = \mathbb{N})? \begin{cases} \varphi_2 \\ \textsc{False} \end{cases} \end{cases}$$

where:

- $X \in V_2$ *is a fresh variable;*
- $X = \emptyset$ *is an abbreviation for* $\neg \exists x.X(x)$*;*
- $X = \mathbb{N}$ *is an abbreviation for* $\forall x.X(x)$*;*
- $\textsc{False}$ *is an abbreviaton for* $\exists x.(\neg(x \leq x))$.

Example 6.13. *Assume that a bus can operate two routes* A *and* B *which start and end at the same place. For* $R \in \{A, B\}$*, the route* R *lasts* $t_R \in \mathbb{Q}_{>0}$ *time units and profits* $p_R \in \mathbb{Q}_{>0}$ *money units on the average per trip. We may be interested in making an infinite schedule for this bus which is represented as an infinite sequence from* $\{A, B\}^\omega$. *This schedule must be fair in the sense that both routes* A *and* B *must occur infinitely often in this timetable (even if the route* A *or* B *is unprofitable). The optimality of the schedule is also preferred (we wish to profit per time unit as much as possible). We consider the* ω*-valuation structure* $\mathbb{V}^{\omega\text{Ratio}}$ *of Example 5.4 (a) together with a default weight* $\mathbb{1}$ *and a singleton alphabet* $\Sigma = \{\tau\}$*;* $\mathbb{1}$ *and* Σ *are irrelevant here. Now we construct a sentence* $\varphi \in \mathbf{mwMSO}^\omega(\Sigma, (\mathbb{V}^{\omega\text{Ratio}})_{\mathbb{1}})$ *to define the optimal income of the bus per time unit (supremum ratio between rewards and time):*

$$\varphi = \sqcup X.\left(\left(\overset{\infty}{\exists} x.X(x) \wedge \overset{\infty}{\exists} x.\neg(X(x))\right) \sqcap \prod x.(X(x)\,?\,(x \mapsto r_A : x \mapsto r_B))\right)$$

where $\overset{\infty}{\exists} x.\psi$ *is an abbreviation of the* $\mathbf{MSO}(\Sigma)$*-formula* $\forall y.\exists x.((y \leq x) \wedge \psi)$, $r_A = (p_A, t_A)$ *and* $r_B = (p_B, t_B)$. *Here, the second order variable* X *corresponds to the set of positions in an infinite schedule which can be assigned to the route* A. *Then,*

$$[\![\varphi]\!](\tau^\omega) = \sup\left\{ \limsup_{n\to\infty} \frac{p_A \cdot |I \cap \overline{n}| + p_B \cdot |I^c \cap \overline{n}|}{t_A \cdot |I \cap \overline{n}| + t_B \cdot |I^c \cap \overline{n}|} \;\middle|\; I \subseteq \mathbb{N} \text{ with } I, I^c \text{ infinite} \right\}$$

where $\overline{n} = \{1, ..., n\}$ *and* $I^c = \mathbb{N} \setminus I$.

Our Büchi-Elgot result for MWBA is the following theorem.

Theorem 6.14. *Let* Σ *be an alphabet,* $\mathbb{V} = (M, (K, +, \mathbb{0}), \text{val}^\omega)$ *an* ω*-valuation structure,* $\mathbb{1} \in M$ *a default weight, and* $\mathbb{L} : \Sigma^\omega \to K$ *a quantitative* ω*-language. Then, the following are equivalent.*

(a) $\mathbb{L}$ *is recognizable over* $\mathbb{V}$.
(b) $\mathbb{L}$ *is* **WAL***-definable over* $\mathbb{V}$.

We will use Theorem 5.6 to deduce Theorem 6.14 from Theorem 6.2. For this, it suffices to show the following.

Lemma 6.15. *Let* $\mathbb{L} : \Sigma^\omega \to K$ *be a quantitative language. Then, the following are equivalent.*

(a) $\mathbb{L} = [\![\varphi]\!]$ *for some sentence* $\varphi \in \mathbf{WAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$.

(b) There exist an alphabet Γ, renamings $h : \Gamma \to \Sigma$, $g : \Gamma \to M$, and an MSO-definable ω-language $\mathcal{L} \subseteq \Gamma^\omega$ such that $\mathbb{L} = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$.

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

Lemma 6.16. *Let $\varphi \in$ **WAL**$(\Sigma, \mathbb{V}_{\mathbb{1}})$ be a sentence. Then, there exists an alphabet Γ, renamings $h : \Gamma \to \Sigma$ and $g : \Gamma \to M$, and an* **MSO**(Γ)*-definable ω-language $\mathcal{L} \subseteq \Gamma^\omega$ such that $[\![\varphi]\!] = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$.*

Before we turn to the proof of Lemma 6.16, we give an example with illustrates the idea of the proof of this lemma.

Example 6.17. *Let $\Sigma = \{a\}$ be a singleton alphabet and $m, m' \in M$ with $m \neq m'$. Consider the* **WAL**$(\Sigma, \mathbb{V}_{\mathbb{1}})$*-sentence*

$$\varphi = \sqcup x. \sqcap y.(y \leq x) \ ? \begin{cases} y \mapsto m \\ y \mapsto m'. \end{cases}$$

The semantics of φ will be decomposed in the following way. First, we define the extended alphabet Γ which reflects labels, weights and the nondeterminism of the "existential" prefix. Let $\Gamma = \Sigma \times \{m, m', \#\} \times \{0, 1\}$. We need the additional component $\{0, 1\}$ since it can happen that we obtain the same assignment of weights for distinct values of the variable x. Let $h : \Gamma \to \Sigma$ be the projection to the Σ-component and let $g : \Gamma \to M$ be defined for all $\gamma = (a, b, c) \in \Gamma$ with $a \in \Sigma$, $b \in \{m, m', \#\}$ and $c \in \{0, 1\}$ by

$$g(\gamma) = \begin{cases} b, & \text{if } b \in \{m, m'\}, \\ \mathbb{1}, & \text{otherwise.} \end{cases}$$

In other words, the renaming h takes care of the labels and the renaming g takes care of the weights; note that g assigns the default weight $\mathbb{1}$ to all unassigned positions labelled by the special symbol $\#$. Finally, we construct the following **MSO**(Γ)*-formula which reflects the qualitative properties of the weight assignments:*

$$\exists x.(\forall y.([y \leq x \wedge P_{(*,m,*)}(y)] \vee [y > x \wedge P_{(*,m',*)}(y)]) \wedge \phi(x))$$

where

$$\phi(x) = \forall y.([y = x \wedge P_{(*,*,1)}(y)] \vee [y \neq x \wedge P_{(*,*,0)}(y)]).$$

and $*$ means that a given component can take all value from its domain. Note that $\phi(x)$ follows the idea of the standard Büchi encoding of variables: the value of the additional $\{0, 1\}$-component at the position x is 1, for all other positions this value is 0.

Proof. Our proof will be based on Lemma 6.8. We may assume that $\varphi = \sqcup x_1. \ldots \sqcup x_k. \sqcup X_1. \ldots \sqcup X_l. \psi$ where $k, l \geq 0$, $x_1, ..., x_k \in V_1$ and $X_1, ..., X_l \in V_2$ are pairwise distinct variables, and $\psi \in$ **uWAL**$(\Sigma, \mathbb{V}_{\mathbb{1}})$. Let $\mathcal{V} = \{x_1, ..., x_k, X_1, ..., X_l\}$. As in Lemma 6.8, let $\Delta_\varphi = \textsc{Const}(\varphi) \cup \{\#\}$ where $\# \notin M$. We define the alphabet Γ, the renamings h, g as follows.

- Let $\Gamma = \Sigma \times \Delta_\varphi \times \{0,1\}^{|\mathcal{V}|}$; we use the $\{0,1\}^{|\mathcal{V}|}$-component for the Büchi encoding of the values of $\mathcal{V}$-variables. Moreover, the Δ_φ-component of Γ will be used to encode weights of assigned positions (using the elements in $\text{Const}(\varphi)$) and unassigned positions (using $\#$).
- Let $\tilde{g} : \Delta_\varphi \to M$ be defined by $\tilde{g}(b) = b$ for all $b \in \text{Const}(\varphi)$ and $\tilde{g}(\#) = \mathbb{1}$. For all $\gamma = (a,b,c) \in \Gamma$ with $a \in \Sigma$, $b \in \Delta_\varphi$ and $c \in \{0,1\}^{|\mathcal{V}|}$, we let $h(\gamma) = a$ and $g(\gamma) = \tilde{g}(b)$.

It remains to define $\mathcal{L} \subseteq \Gamma^\omega$ such that $[\![\varphi]\!] = h((\text{val}^\omega \circ g) \cap \mathcal{L})$.

By Lemma 6.8, there exists a formula $\zeta \in \mathbf{MSO}(\Sigma \times \Delta_\varphi)$ such that $\text{Free}(\zeta) = \mathcal{V}$ and, for all $(w,\sigma) \in \Sigma_V^\omega$ and all partial mappings $\eta : \mathbb{N} \dashrightarrow \text{Const}(\varphi)$:

$$\langle\!\langle \psi \rangle\!\rangle(w,\sigma) = \eta \text{ iff } (\text{code}(w,\eta),\sigma) \models \zeta. \tag{6.5}$$

Note that formula ζ over the alphabet $\Sigma \times \Delta_\varphi$ encodes the auxiliary semantics $\langle\!\langle \psi \rangle\!\rangle$. We have to adopt this formula to the extended alphabet $\Sigma \times \Delta_\varphi \times \{0,1\}^{|\mathcal{V}|}$ and to connect the $\{0,1\}^{|\mathcal{V}|}$-component with $\mathcal{V}$-variables via the Büchi-encoding.

Let $\zeta^* \in \mathbf{MSO}(\Gamma)$ be the formula obtained from ζ by replacing each predicate $P_{(a,b)}(x)$ occurring in ζ (here, $a \in \Sigma$, $b \in \Delta_\varphi$ and $x \in V_1$) by the formula $\bigvee(P_{(a,b,c)}(x) \mid c \in \{0,1\}^{|\mathcal{V}|})$. Here we demand that the $\{0,1\}^{|\mathcal{V}|}$-component is arbitrary.

For all $1 \le i \le |\mathcal{V}|$, $d \in \{0,1\}$ and $x \in V_1$, let

$$R^*_{i,d}(x) = \bigvee \big(P_{(a,b,c)}(x) \mid a \in \Sigma, b \in \Delta_\varphi \text{ and } c = (c_1, ..., c_{|\mathcal{V}|}) \text{ with } c_i = d\big).$$

This formula demands that the i-th bit of a vector in $\{0,1\}^{|\mathcal{V}|}$ is d.

Let $y \in V_1$ be a fresh variable. Let $\phi \in \mathbf{MSO}(\Gamma)$ be defined as $\phi = \forall y.(\phi_1 \wedge \phi_2)$ where

$$\phi_1 = \bigwedge_{i=1}^{k} [(R^*_{i,1}(y) \wedge (y = x_i)) \vee (R^*_{i,0}(y) \wedge (y \neq x_i))],$$
$$\phi_2 = \bigwedge_{j=1}^{l} [(R^*_{k+j,1}(y) \wedge X_j(y)) \vee (R^*_{k+j,0}(y) \wedge (\neg X_j(y)))].$$

The formula ϕ encodes the values of $\mathcal{V}$-variables in the $\{0,1\}^{|\mathcal{V}|}$-component of an input word: the first-order variables $x_1, ..., x_k$ correspond to the first k bits of the $\{0,1\}^{|\mathcal{V}|}$-component and the second-order variables $X_1, ..., X_l$ correspond to the last l bits.

Let the sentence $\beta \in \mathbf{MSO}(\Gamma)$ be defined as $\beta = \exists \mathcal{V}.(\phi \wedge \zeta^*)$. Then, the desired language $\mathcal{L}$ is defined as $\mathcal{L} = \mathcal{L}(\beta)$. For $w = (a_i)_{i \in \mathbb{N}} \in \Sigma^\omega$ and $u = (b_i)_{i \in \mathbb{N}} \in \Delta_\varphi^\omega$, we will abuse the notation and write (w,u) for $((a_i,b_i))_{i \in \mathbb{N}}$.

If $w \in \Sigma^\omega$, let $\mathcal{V}_w$ be the set of all mappings $\mathcal{J} : \mathcal{V} \to \text{dom}(w) \cup 2^{\text{dom}(w)}$ such that $\mathcal{J}(\mathcal{V} \cap V_1) \subseteq \text{dom}(w)$ and $\mathcal{J}(\mathcal{V} \cap V_2) \subseteq 2^{\text{dom}(w)}$. For a w-assignment σ and $\mathcal{J} \in \mathcal{V}_w$, let $\sigma' := \sigma[\mathcal{V}/\mathcal{J}]$ denote the w-assignment such that $\sigma'|_\mathcal{V} = \mathcal{J}$ and $\sigma'|_{V \setminus \mathcal{V}} = \sigma|_{V \setminus \mathcal{V}}$.

Finally, we show that $[\![\varphi]\!] = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$. Let $w \in \Sigma^\omega$ and σ a fixed w-assignment. Then,

$$\begin{aligned} h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})(w) &= \sum \big(\mathrm{val}^\omega(\tilde{g}(u)) \mid \mathcal{J} \in \mathcal{V}_w \text{ and } ((w,u), \sigma[\mathcal{V}/\mathcal{J}]) \models \zeta \big) \\ &\overset{(!)}{=} \sum \big(\mathrm{val}^\omega(\eta) \mid \mathcal{J} \in \mathcal{V}_w \text{ and } \eta = \langle\!\langle \psi \rangle\!\rangle(w, \sigma[\mathcal{V}/\mathcal{J}]) \text{ is defined} \big) \\ &= \sum_{\mathcal{J} \in \mathcal{V}_w} [\![\psi]\!](w, \sigma[\mathcal{V}/\mathcal{J}]) \\ &= [\![\varphi]\!](w). \end{aligned}$$

Here, at the place (!), we apply (6.5). □

Now we show the implication (b) ⇒ (a) of Lemma 6.15.

Lemma 6.18. *Let Γ be an alphabet, $h : \Gamma \to \Sigma$ and $g : \Gamma \to M$ renamings, and $\mathcal{L} \subseteq \Gamma^\omega$ an $\mathbf{MSO}(\Gamma)$-definable ω-language. Then, there exist $\mathbb{1} \in \mathbb{M}$ and a sentence $\varphi \in \mathbf{WAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ such that $[\![\varphi]\!] = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})$.*

Proof. Here, we use a similar construction as in the proof of Lemma 6.10. Let $\beta \in \mathbf{MSO}(\Gamma)$ be a sentence with $\mathcal{L}_\omega(\beta) = \mathcal{L}$. Let $\mathcal{V} \subseteq V_2$, $(\gamma_i)_{i \in \{1,\dots,|\Gamma|}$ an enumeration of Γ, $\textsc{Boolean} \in \mathbf{MSO}(\Sigma)$ and $\chi \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ be defined as in the proof of Lemma 6.10. Let $\psi = (\textsc{Boolean}\,?\,(\chi : \textsc{True})) \in \mathbf{uWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$. Then, we define the sentence $\varphi \in \mathbf{WAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ as

$$\varphi = \sqcup X_{\gamma_1}. \dots \sqcup X_{\gamma_{|\Gamma|}}.\psi.$$

Note that $\mathrm{Free}(\varphi) = \mathcal{V}$ and, for all $(w, \sigma) \in \Sigma^\omega_{\mathcal{V}}$, $\langle\!\langle \psi \rangle\!\rangle(w, \sigma) \in M^\omega$.

Let $w \in \Sigma^\omega$ and σ a fixed w-assignment. We define the set $\mathcal{V}_w$ and the w-assignment $\sigma[\mathcal{V}/\mathcal{J}]$ for $\mathcal{J} \in \mathcal{V}_w$ as in the previous lemma. Then,

$$\begin{aligned} [\![\varphi]\!](w) = & \sum_{\substack{\mathcal{J} \in \mathcal{V}_w, \\ (w, \sigma[\mathcal{V}/\mathcal{J}]) \in \mathrm{dom}(\langle\!\langle \psi \rangle\!\rangle)}} \mathrm{val}^\omega(\langle\!\langle \psi \rangle\!\rangle(w, \sigma[\mathcal{V}/\mathcal{J}])) = \sum_{\substack{u \in \Gamma^\omega, h(u) = w, \\ (w, \sigma') \models h(\beta)}} \mathrm{val}^\omega(g(u)) \\ \overset{(!)}{=} & \sum_{\substack{u \in \mathcal{L}, \\ h(u) = w}} \mathrm{val}^\omega(g(u)) = h((\mathrm{val}^\omega \circ g) \cap \mathcal{L})(w). \end{aligned}$$

Here, σ' is a w-assignment with $\sigma'|_{\mathcal{V}} = \sigma_u$ and, at the place (!), we apply (6.2). □

Proof of Lemma 6.15. Immediate by Lemmas 6.16 and 6.18. □

Proof of Theorem 6.14. Immediate by Theorems 5.6, 6.2 and Lemma 6.15. □

Note that the proof of Theorem 6.14 is constructive. Then, we obtain the following corollary.

Corollary 6.19. *Let* $\mathbb{V}^{\omega\text{Disc}}$ *be the* ω*-valuation structure of Example 5.4 (c). Then, it is decidable, given an alphabet* Σ*, a default weight* $\mathbb{1} \in \mathbb{Q}_{\geq 0} \times ((0,1] \cap \mathbb{Q})$*, a sentence* $\varphi \in \mathbf{WAL}(\Sigma, (V^{\omega\text{Disc}})_{\mathbb{1}})$*, and a threshold* $\theta \in \mathbb{Q}_{\geq 0}$*, whether* $[\![\varphi]\!](w) < \theta$ *for some* $w \in \Sigma^\omega$*.*

Proof. Immediate from Theorem 6.14 and Corollary 5.24. □

Chapter 7

Multi-weighted timed automata

Contents

In this chapter, we develop a general framework for *multi-weighted timed automata* on finite timed words. Following the ideas of [38] and our ideas presented in Chapter 5, we give a Nivat-like characterization of quantitative languages recognizable by multi-weighted timed automata. Finally, using our Nivat theorem for multi-weighted timed automata, we study the connection via renamings between determinism and nondeterminism in the multi-weighted timed setting.

7.1 Timed automata

Let Σ be an alphabet. A (finite) *timed word* over Σ is a finite word over $\Sigma \times \mathbb{R}_{\geq 0}$, i.e., a finite sequence $(a_1, t_1)...(a_n, t_n)$ with $n \geq 0$, $a_1, ..., a_n \in \Sigma$ and $t_1, ..., t_n \in \mathbb{R}_{\geq 0}$. Let $|w| = n$, the *length* of w, and $\langle w \rangle = t_1 + ... + t_n$, the *time length* of w. If $|w| = 0$, then we say that the timed word w is *empty* and denote it by ε. For all i, j with $0 \leq i < j \leq n$, let $\langle w \rangle_{i,j} = t_{i+1} + ... + t_j$.

Let $\mathbb{T}\Sigma^* = (\Sigma \times \mathbb{R}_{\geq 0})^*$, the set of all finite timed words, and $\mathbb{T}\Sigma^+ = \mathbb{T}\Sigma^* \setminus \{\varepsilon\}$. Any set $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ of timed words is called a *timed language*. Like in the untimed case, we eliminate the empty word ε when considering timed languages. For timed words $w = (a_1, t_1)...(a_n, t_n) \in \mathbb{T}\Sigma^*$ and

$w' = (a'_1, t'_1)...(a'_n, t'_n) \in \mathbb{T}\Sigma^*$, the *concatenation* $(w \cdot w') \in \mathbb{T}\Sigma^*$ is defined as $(a_1, t_1)...(a_n, t_n)(a'_1, t'_1)...(a'_n, t'_n)$.

Remark 7.1. *In the literature, a timed word is represented sometimes as a sequence $(a_1, \tau_1)...(a_n, \tau_n)$ where $0 \leq \tau_1 \leq \tau_2 \leq ... \leq \tau_n$ where τ_i measures the time between the beginning and the i-th position. In contrast, here we represent a timed word as a sequence $(a_1, t_1)...(a_n, t_n)$ where $(t_1, ..., t_n)$ is an arbitrary finite sequence in $\mathbb{R}_{\geq 0}$. Here, t_i measures the time between the positions $i - 1$ and i.*

Let $\mathcal{I}$ denote the collection of all intervals of the form $[a, b]$, $[a, b)$, $(a, b]$, (a, b), $[a, \infty)$ or (a, ∞) where $a, b \in \mathbb{N}$. Note that $\emptyset \in \mathcal{I}$. Let C be a finite set of *clock variables* ranging over $\mathbb{R}_{\geq 0}$. A *clock constraint* over C is a mapping $\phi : C \to \mathcal{I}$. If $C = \emptyset$ or $\phi(x) = [0, \infty)$ for all $x \in C$, then we will denote such a mapping ϕ by TRUE. We denote by $\Phi(C)$ the set of all clock constaints over C. A *clock valuation* is a mapping $\nu : C \to \mathbb{R}_{\geq 0}$ which assigns a value to each clock variable. Let $\mathbb{R}^C_{\geq 0}$ be the set of all clock valuations over C. We say that a clock valuation $\nu \in \mathbb{R}^C_{\geq 0}$ *satisfies* a clock constraint $\phi \in \Phi(C)$, written $\nu \models \phi$, if $\nu(x) \in \phi(x)$ for all $x \in C$.

Now let $\nu \in \mathbb{R}^C_{\geq 0}$, $t \in \mathbb{R}_{\geq 0}$ and $\Lambda \subseteq C$. Let $\nu + t$ denote the clock valuation $\nu' \in \mathbb{R}^C_{\geq 0}$ such that $\nu'(x) = \nu(x) + t$ for all $x \in C$. Let $\nu[\Lambda := 0]$ denote the clock valuation $\nu' \in \mathbb{R}^C_{\geq 0}$ such that $\nu'(x) = 0$ for all $x \in \Lambda$ and $\nu'(x) = \nu(x)$ for all $x \notin \Lambda$.

Definition 7.2. *Let Σ be an alphabet. A* timed automaton *over Σ is a tuple $\mathcal{A} = (L, C, I, E, F)$ such that L is a finite set of* locations *, C is a finite set of* clocks*, $I, F \subseteq L$ are sets of* initial *resp.* final *locations and $E \subseteq L \times \Sigma \times \Phi(C) \times 2^C \times L$ is a finite set of* edges*.*

For an edge $e = (\ell, a, \phi, \Lambda, \ell')$, let label$(e) = a$, the *label* of e. A *run* of $\mathcal{A}$ is a finite sequence

$$\rho = (\ell_0, \nu_0) \xrightarrow{t_1} \xrightarrow{e_1} (\ell_1, \nu_1) \xrightarrow{t_2} \xrightarrow{e_2} ... \xrightarrow{t_n} \xrightarrow{e_n} (\ell_n, \nu_n) \tag{7.1}$$

where $n \geq 1$, $\ell_0, \ell_1, ..., \ell_n \in L$, $\nu_0, \nu_1, ..., \nu_n \in \mathbb{R}^C_{\geq 0}$, $t_1, ..., t_n \in \mathbb{R}_{\geq 0}$ and $e_1, ..., e_n \in E$ satisfy the following conditions:

- $\ell_0 \in I$, $\nu_0(x) = 0$ for all $x \in C$, $\ell_n \in F$;
- for all $1 \leq i \leq n$: $e_i = (\ell_{i-1}, a_i, \phi_i, \Lambda_i, \ell_i)$ for some $a_i \in \Sigma$, $\phi_i \in \Phi(C)$ and $\Lambda_i \subseteq C$ such that $\nu_{i-1} + t_i \models \phi_i$ and $\nu_i = (\nu_{i-1} + t_i)[\Lambda_i := 0]$.

The *label* of ρ is the timed word label$(\rho) = (\text{label}(e_1), t_1)...(\text{label}(e_n), t_n) \in \mathbb{T}\Sigma^+$. Let $\text{Run}_\mathcal{A}$ denote the set of all runs of $\mathcal{A}$. For any timed word $w \in \mathbb{T}\Sigma^+$, let $\text{Run}_\mathcal{A}(w)$ denote the set of all runs ρ of $\mathcal{A}$ such that label$(\rho) = w$. Let $\mathcal{L}(\mathcal{A}) = \{w \in \mathbb{T}\Sigma^+ \mid \text{Run}_\mathcal{A}(w) \neq \emptyset\}$. We say that an arbitrary timed language $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ is *recognizable* if there exists a timed automaton over Σ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. Let tREC$(\Sigma)$ denote the collection of all recognizable timed languages.

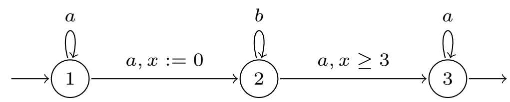

Figure 7.1: Timed automaton $\mathcal{A}$ of Example 7.3

Example 7.3. *Let $\Sigma = \{a, b\}$ and $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ be the timed language consisting of all timed words $w = xyz$ where $x, z \in \mathbb{T}\{a\}^+$ and $y \in \mathbb{T}\{b\}^*$ such that either $y = \varepsilon$ or $\langle y \rangle \geq 3$. We show that this timed language is recognizable. For this, we construct the timed automaton $\mathcal{A} = (L, C, I, E, F)$ over Σ where $L = \{1, 2, 3\}$, $C = \{c\}$, $I = \{1\}$, $F = \{3\}$ and*

$$E = \{(1, a, \textsc{True}, \emptyset, 1), (1, a, \textsc{True}, \{x\}, 2), (2, b, \textsc{True}, \emptyset, 2),\\ (2, a, \phi_{\geq 3}, \emptyset, 3), (3, a, \textsc{True}, \emptyset, 3)\}$$

where $\phi_{\geq 3} \in \Phi(C)$ with $\phi_{\geq 3}(c) = [3, \infty)$. Then, $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. The timed automaton $\mathcal{A}$ is depicted in Figure 7.1. Here, we omit all irrelevant information (i.e., when a set of clocks to be reset is empty or a clock constraint is True*).*

Let $\mathcal{A} = (L, C, I, E, F)$ be a timed automaton over Σ. We call $\mathcal{A}$:

- *unambiguous* if $|\operatorname{Run}_{\mathcal{A}}(w)| \leq 1$ for all $w \in \mathbb{T}\Sigma^+$;
- *deterministic* if $|I| = 1$ and, for all $e_1 = (\ell, a, \phi_1, \Lambda_1, \ell_1) \in E$ and $e_2 = (\ell, a, \phi_2, \Lambda_2, \ell_2) \in E$ with $e_1 \neq e_2$, there exists no clock valuation $\nu \in \mathbb{R}_{\geq 0}^C$ with $\nu \models \phi_1$ and $\nu \models \phi_2$;
- *sequential* if $|I| = 1$ and, for all $e_1 = (\ell, a, \phi_1, \Lambda_1, \ell_1) \in E$ and $e_2 = (\ell, a, \phi_2, \Lambda_2, \ell_2) \in E$, we have $e_1 = e_2$ (this property can be viewed as a strong form of determinism).

Let $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ be a timed language. We say that $\mathcal{L}$ is *unambiguously (deterministically, sequentially, respectively) recognizable* if there exists an unambiguous (deterministic, sequential, respectively) timed automaton $\mathcal{A}$ over Σ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.

Let $\textsc{tRec}^{\textsc{Unamb}}(\Sigma)$ denote the collection of all unambiguously recognizable timed languages over Σ, $\textsc{tRec}^{\textsc{Det}}(\Sigma)$ the collection of deterministically recognizable timed languages, and $\textsc{tRec}^{\textsc{Seq}}(\Sigma)$ the collection of sequentially recognizable timed languages. Clearly,

$$\textsc{tRec}^{\textsc{Seq}}(\Sigma) \subseteq \textsc{tRec}^{\textsc{Det}}(\Sigma) \subseteq \textsc{tRec}^{\textsc{Unamb}}(\Sigma) \subseteq \textsc{tRec}(\Sigma).$$

Lemma 7.4 ([84])**.** *There exists an alphabet Σ such that*

$$\textsc{tRec}^{\textsc{Det}}(\Sigma) \subsetneq \textsc{tRec}^{\textsc{Unamb}}(\Sigma) \subsetneq \textsc{tRec}(\Sigma).$$

We complete the chain of strict inclusions of Lemma 7.4 by the following strict inclusion.

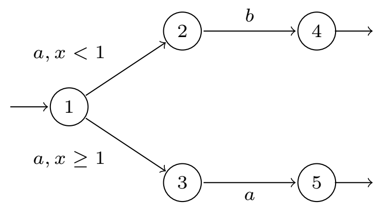

Figure 7.2: Timed automaton $\mathcal{A}_{\mathcal{L}}$ from the proof of Lemma 7.5

Lemma 7.5. *There exists an alphabet* Σ *such that*

$$\text{tRec}^{\text{Seq}}(\Sigma) \subsetneq \text{tRec}^{\text{Det}}(\Sigma).$$

Proof. Let $\Sigma = \{a, b\}$. Consider the timed language $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ defined as

$$\mathcal{L} = \{(a,t)(b,t') \mid t < 1 \text{ and } t' \in \mathbb{R}_{\geq 0}\} \cup \{(a,t)(a,t') \mid t \geq 1 \text{ and } t' \in \mathbb{R}_{\geq 0}\}.$$

We show that $\mathcal{L} \in \text{tRec}^{\text{Det}}(\Sigma) \setminus \text{tRec}^{\text{Seq}}(\Sigma)$.

First, we show that $\mathcal{L} \in \text{tRec}^{\text{Det}}(\Sigma)$. Consider the timed automaton $\mathcal{A}_{\mathcal{L}}$ over Σ with the only clock x depicted in Figure 7.2. Clearly, $\mathcal{A}_{\mathcal{L}}$ is deterministic and $\mathcal{L}(\mathcal{A}_{\mathcal{L}}) = \mathcal{L}$. Then, $\mathcal{L} \in \text{tRec}^{\text{Det}}(\Sigma)$.

Next, we show that $\mathcal{L} \notin \text{tRec}^{\text{Seq}}(\Sigma)$. Assume that there exists a sequential timed automaton $\mathcal{A} = (L, C, \{\ell_0\}, E, F)$ over Σ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. Consider $w = (a, 0.9)(b, 0.2) \in \mathcal{L}(\mathcal{A})$. Then, there exist $e = (\ell_0, a, \phi, \Lambda, \ell_1) \in E$ and $e' = (\ell_1, b, \phi', \Lambda', \ell_2) \in E$ such that

$$\rho = (\ell_0, \nu_0) \xrightarrow{0.9} \xrightarrow{e} (\ell_1, \nu_1) \xrightarrow{0.2} \xrightarrow{e'} (\ell_2, \nu_2)$$

is a run in $\text{Run}_{\mathcal{A}}(w)$. Here, $\nu_0(x) = 0$ for all $x \in C$, $\nu_1 = (\nu_0 + 0.9)[\Lambda := 0]$ and $\nu_2 = (\nu_1 + 0.2)[\Lambda' := 0]$. Moreover, $(\nu_0 + 0.9) \models \phi$ and $(\nu_1 + 0.2) \models \phi'$ and $\ell_2 \in F$.

Consider now the timed word $u = (a, 1)(a, 0.1) \in \mathcal{L}(\mathcal{A})$. Since $\mathcal{A}$ is sequential, there exists $e'' = (\ell_1, a, \phi'', \Lambda'', \ell_3)$ such that

$$\varrho = (\ell_0, \nu_0) \xrightarrow{1} \xrightarrow{e} (\ell_1, \nu_1') \xrightarrow{0.1} \xrightarrow{e''} (\ell_3, \nu_2')$$

is a run in $\text{Run}_{\mathcal{A}}(u)$. Here, $\nu_1' = (\nu_0 + 1)[\Lambda := 0]$ and $\nu_2' = (\nu_1' + 0.1)[\Lambda'' := 0]$. Moreover, $(\nu_0 + 1) \models \phi$ and $(\nu_1' + 0.1) \models \phi''$ and $\ell_3 \in F$. Then, for all $x \in \Lambda$, we have: $(\nu_1 + 0.2)(x) = 0.2 \in \phi'(x)$ and $(\nu_1' + 0.1)(x) = 0.1 \in \phi''(x)$ and, for all $x \in C \setminus \Lambda$, we have: $(\nu_1 + 0.2)(x) = 1.1 \in \phi'(x)$ and $(\nu_1' + 0.1)(x) = 1.1 \in \phi''(x)$. Note that $0.1 \in \phi''(x)$ implies $0.2 \in \phi''(x)$. Then, $(\nu_1 + 0.2) \models \phi''(x)$. Let

$$\pi = (\ell_0, \nu_0) \xrightarrow{0.9} \xrightarrow{e} (\ell_1, \nu_1) \xrightarrow{0.2} \xrightarrow{e''} (\ell_3, \nu_2')$$

where $\nu_2' = (\nu_1 + 0.2)[\Lambda'' := 0]$. Clearly, π is a run of $\mathcal{A}$. Since $\text{label}(\pi) = (a, 0.9)(a, 0.2)$, we obtain $(a, 0.9)(a, 0.2) \in \mathcal{L}(\mathcal{A}) = \mathcal{L}$. A contradiction. Thus, $\mathcal{L} \notin \text{tRec}^{\text{Seq}}(\Sigma)$.

$\square$

7.2 A general framework and examples of multi-weighted timed automata

In this section, we introduce a general model of *multi-weighted timed automata* over *timed valuation structures*. For the clarity of presentation, as in the framework of multi-weighted Büchi automata, we will not generalize the notion of evaluators to the timed setting but, as in [42], we define our algebraic structure for multi-weighted timed automata in the spirit of *valuation monoids* [40]. We introduce a timed valuation function to compute the weights of runs. The non-determinism on the weights of runs will be resolved by means of a commutative operation (e.g., minimum or maximum). As in [40], the weight of a run depends on the history of weights that occurred in this run. We also will model the property that staying in a location invokes costs depending on the length of the stay and that the subsequent transition also invokes costs but happens instantaneously. In addition, we must take into account the multi-weighted setting, i.e., the fact that the weight constants (e.g., tuples of weights) in an automaton and the weights of runs (e.g., single values) are not necessarily taken from the same set.

Definition 7.6. *A* timed valuation structure *is a tuple* $\mathbb{V} = (M_L, M_E, (K, +, \mathbb{0}), \mathrm{val}^{\mathbb{T}})$ *where:*

- $(K, +, \mathbb{0})$ *is a commutative monoid;*
- M_L *and* M_E *are non-empty sets of* location weights *and* edge weights *respectively;*
- $\mathrm{val}^{\mathbb{T}} : \mathbb{T}(M_L \times M_E)^+ \to K$ *is a* timed valuation function.

We say that $\mathbb{V}$ is *idempotent* if $+$ is idempotent, i.e., $k + k = k$ for all $k \in K$.

Remark 7.7. *In [42], a* timed valuation monoid *is defined as a timed valuation structure with* $M_L = M_E = K$. *In this thesis, we extend this model in order to be able:*

- *to consider the multi-weighted setting (e.g., the weight constants can be the tuples of weights and the behavior takes on single values);*
- *to incorporate situations where the weights of staying in locations can be defined by arbitrary functions, as in [77, 78].*

Definition 7.8. *Let* Σ *be an alphabet and* $\mathbb{V} = (M_L, M_E, (K, +, \mathbb{0}), \mathrm{val}^{\mathbb{T}})$ *a timed valuation structure. A* multi-weighted timed automaton (MWTA) *over* Σ *and* $\mathbb{V}$ *is a tuple* $\mathcal{A} = (L, C, I, E, F, \mathrm{wt})$ *where:*

- (L, C, I, E, F) *is a timed automaton over* Σ*;*
- $\mathrm{wt} : (L \cup E) \to (M_L \cup M_E)$ *is a* weight function *with* $\mathrm{wt}(L) \subseteq M_L$ *and* $\mathrm{wt}(E) \subseteq M_E$.

We call $\mathcal{A}$ *unambiguous* (*deterministic*, *sequential*, respectively) if the underlying timed automaton (L, C, I, E, F) is unambiguous (deterministic, sequential, respectively). Let ρ be a run of $\mathcal{A}$ of the form

$$\rho = (\ell_0, \nu_0) \xrightarrow{t_1} \xrightarrow{e_1} (\ell_1, \nu_1) \xrightarrow{t_2} \xrightarrow{e_2} \ldots \xrightarrow{t_n} \xrightarrow{e_n} (\ell_n, \nu_n).$$

We associate with ρ the timed word

$$\mathrm{wt}^{\#}_{\mathcal{A}}(\rho) = ((\mathrm{wt}(\ell_0), \mathrm{wt}(e_1)), t_1)...((\mathrm{wt}(\ell_{n-1}), \mathrm{wt}(e_n)), t_n) \in \mathbb{T}(M_L \times M_E)^+$$

which contains all the history of weights and time delays of the run ρ. Then, the *weight* of ρ is defined as $\mathrm{wt}_{\mathcal{A}}(\rho) = \mathrm{val}^{\mathbb{T}}(\mathrm{wt}^{\#}_{\mathcal{A}}(\rho)) \in K$. The *behavior* of $\mathcal{A}$ is the mapping $[\![\mathcal{A}]\!] : \mathbb{T}\Sigma^+ \to K$ defined for all $w \in \mathbb{T}\Sigma^+$ by

$$[\![\mathcal{A}]\!](w) = \sum(\mathrm{wt}_{\mathcal{A}}(\rho) \mid \rho \in \mathrm{Run}_{\mathcal{A}}(w)).$$

A mapping $\mathbb{L} : \mathbb{T}\Sigma^+ \to K$ is called a *quantitative timed language* (QTL). We say that $\mathbb{L}$ is:

- *recognizable* over $\mathbb{V}$ if there exists a MWTA $\mathcal{A}$ over Σ and $\mathbb{V}$ such that $[\![\mathcal{A}]\!] = \mathbb{L}$;
- *unambiguously (deterministically, sequentially) recognizable* if there exists an unambiguous (deterministic, sequential) MWTA $\mathcal{A}$ over Σ and $\mathbb{V}$ such that $[\![\mathcal{A}]\!] = \mathbb{L}$.

Let $\mathrm{tRec}(\Sigma, \mathbb{V})$ denote the collection of all QTL $\mathbb{L} : \mathbb{T}\Sigma^+ \to K$ recognizable over Σ and $\mathbb{V}$. In the same manner, we define the collections $\mathrm{tRec}^{\mathrm{Unamb}}(\Sigma, \mathbb{V})$, $\mathrm{tRec}^{\mathrm{Det}}(\Sigma, \mathbb{V})$ and $\mathrm{tRec}^{\mathrm{Seq}}(\Sigma, \mathbb{V})$ for unambiguous, deterministic and sequential MWTA. Clearly,

$$\mathrm{tRec}^{\mathrm{Seq}}(\Sigma, \mathbb{V}) \subseteq \mathrm{tRec}^{\mathrm{Det}}(\Sigma, \mathbb{V}) \subseteq \mathrm{tRec}^{\mathrm{Unamb}}(\Sigma, \mathbb{V}) \subseteq \mathrm{tRec}(\Sigma, \mathbb{V}).$$

The next examples shows that various models of single-weighted timed automata and multi-weighted timed automata fit into the framework of timed valuation structures.

Example 7.9. *(a)* Linearly priced timed automata *were studied in [7, 11, 68]. The weights of edges in these automata have the same meaning as the transition weights of weighted untimed automata, i.e., happen instantly and time independently. In contrast, the weight of a delay in a location is time-dependent and grows linearly in time. The behavior of linearly priced timed automata can be described by means of the timed valuation structure*

$$\mathbb{V}^{\mathrm{tSum}} = \left(\mathbb{Q}, \mathbb{Q}, (\mathbb{R} \cup \{\infty\}, \min, \infty), \mathrm{val}^{\mathrm{tSum}}\right)$$

where $\mathrm{val}^{\mathrm{tSum}} : \mathbb{T}(\mathbb{Q} \times \mathbb{Q})^+ \to \mathbb{R} \cup \{\infty\}$ *is defined for all timed words* $v = ((m_1, m'_1), t_1)...((m_n, m'_n), t_n) \in \mathbb{T}(\mathbb{Q} \times \mathbb{Q})^+$ *by*

$$\mathrm{val}^{\mathrm{tSum}}(v) = \sum_{i=1}^{n}(m_i \cdot t_i + m'_i).$$

(b) Now we consider the reward-cost ratio measure for timed automata investigated in [21, 22], i.e., the objective is the ratio between accumulated

rewards and costs. The behavior of ratio timed automata on finite words can be described using the timed valuation structure

$$\mathbb{V}^{\text{tRatio}} = \Big(M, M, (\mathbb{R} \cup \{\infty\}, \min, \infty), \mathrm{val}^{\text{tRatio}}\Big)$$

where $M = \mathbb{Q} \times \mathbb{Q}_{\geq 0}$ *and* $\mathrm{val}^{\text{tRatio}} : \mathbb{T}(M \times M)^+ \to \mathbb{R} \cup \{\infty\}$ *is defined for all* $v = ((m_1, m_1'), t_1)...((m_n, m_n'), t_n) \in \mathbb{T}(M \times M)^+$ *with* $m_i = (r_i, c_i)$ *and* $m_i' = (r_i', c_i')$ *by*

$$\mathrm{val}^{\text{tRatio}}(v) = \frac{\sum_{i=1}^{n}(r_i \cdot t_i + r_i')}{\sum_{i=1}^{n}(c_i \cdot t_i + c_i')}.$$

Here, we put $\frac{x}{0} = \infty$ *for all* $x \in \mathbb{R}$.

(c) *Single-weighted timed automata with discounting in time were investigated in [53, 54] (cf. also [6]). Their behavior on finite words can be described as follows. For a discounting factor* $\lambda \in \mathbb{Q} \cap (0, 1)$*, we consider the timed valuation structure*

$$\mathbb{V}^{\text{tDisc}(\lambda)} = \Big(\mathbb{Q}, \mathbb{Q}, (\mathbb{R} \cup \{\infty\}, \min, \infty), \mathrm{val}^{\text{tDisc}(\lambda)}\Big).$$

Here, the timed valuation function $\mathrm{val}^{\text{tDisc}(\lambda)} : \mathbb{T}(\mathbb{Q} \times \mathbb{Q})^+ \to \mathbb{R} \cup \{\infty\}$ *is defined for all* $v = ((m_1, m_1'), t_1)...((m_n, m_n'), t_n) \in \mathbb{T}(\mathbb{Q} \times \mathbb{Q})^+$ *by*

$$\mathrm{val}^{\text{tDisc}(\lambda)}(v) = \sum_{i=1}^{n} \lambda^{t_1 + ... + t_{i-1}} \cdot \Big(\int_0^{t_i} m_i \cdot \lambda^{\tau} d\tau + \lambda^{t_i} \cdot m_i'\Big).$$

(d) *Double-priced timed automata with a knapsack-like objective, i.e., with an upper bound* $\eta \in \mathbb{Q}_{\geq 0}$ *on the accumulated weight for the secondary weight parameter were investigated in [69]. Their behavior can be described by means of the timed valuation structure*

$$\mathbb{V}^{\text{tKnap}(\eta)} = \Big(M, M, (\mathbb{R}_{\geq 0} \cup \{\infty\}, \min, \infty), \mathrm{val}^{\text{tKnap}(\eta)}\Big)$$

where $M = \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$ *and* $\mathrm{val}^{\text{Knap}(\eta)}$ *is defined for all* $v = ((m_1, m_1'), t_1)...((m_n, m_n'), t_n) \in \mathbb{T}(M \times M)^+$ *with* $m_i = (x_i, y_i)$ *and* $m_i' = (x_i', y_i')$ *by*

$$\mathrm{val}^{\text{tKnap}(\eta)}(v) = \begin{cases} \sum_{i=1}^{n}(x_i \cdot t_i + x_i'), & \text{if } \sum_{i=1}^{n}(y_i \cdot t_i + y_i') \leq \eta, \\ \infty, & \text{otherwise.} \end{cases}$$

(e) *A semiring model for weighted timed automata (which covers, for instance, the case of linearly priced timed automata) was considered in [77, 78]. In this model, the weights of edges are taken from a semiring* $\mathbb{S} = (S, +, \cdot, \mathbb{0}, \mathbb{1})$ *and the weights of locations are taken from a set of functions* $\mathcal{F} \subseteq S^{\mathbb{R}_{\geq 0}}$. *Our model of MWTA over timed valuation structures also covers this case. Here, we consider the timed valuation structure*

$$\mathbb{V}^{\mathbb{S},\mathcal{F}} = \Big(\mathcal{F}, S, (S, +, \mathbb{0}), \mathrm{val}^{\mathbb{S},\mathcal{F}}\Big)$$

where $\mathrm{val}^{\mathbb{S},\mathcal{F}} : \mathbb{T}(S \times \mathcal{F})^+ \to S$ *is defined for all timed words* $v = ((f_1, s_1), t_1)...((f_n, s_n), t_n)$ *by*

$$\mathrm{val}^{\mathbb{S},\mathcal{F}}(v) = \prod_{i=1}^{n} s_i \cdot f_i(t_i).$$

Now we show that the threshold problem for multi-weighted ratio timed automata of Example (b) 7.9 is decidable (note that, in [21, 22], these multi-weighted automata were considered for nonterminating computations with some additional properties which we do not need for finite timed words).

Lemma 7.10. *It is decidable, given an alphabet* Σ*, a MWTA* $\mathcal{A}$ *over* Σ *and* $\mathbb{V}^{\text{tRatio}}$*, and a threshold* $\theta \in \mathbb{Q}$*, whether* $[\![\mathcal{A}]\!](w) < \theta$ *for some* $w \in \mathbb{T}\Sigma^+$*.*

Proof. Let $\mathcal{A} = (L, C, I, E, F, \mathrm{wt})$ be a MWTA over an alphabet Σ and $\mathbb{V}^{\text{tRatio}}$. For every $u \in L \cup E$, let $\mathrm{wt}(u) = (r_u, c_u)$. For a run

$$\rho = (\ell_0, \nu_0) \xrightarrow{t_1}\xrightarrow{e_1} (\ell_1, \nu_1) \xrightarrow{t_2}\xrightarrow{e_2} ... \xrightarrow{t_n}\xrightarrow{e_n} (\ell_n, \nu_n)$$

of $\mathcal{A}$, let $\textsc{Reward}_{\mathcal{A}}(\rho) = \sum_{i=1}^{n}(r_{\ell_{i-1}} \cdot t_i + r_{e_i})$, the *reward* of ρ, and $\textsc{Cost}_{\mathcal{A}}(\rho) = \sum_{i=1}^{n}(c_{\ell_{i-1}} \cdot t_i + c_{e_i})$, the *cost* of ρ. Note that $\mathrm{wt}_{\mathcal{A}}(\rho) = \frac{\textsc{Reward}_{\mathcal{A}}(\rho)}{\textsc{Cost}_{\mathcal{A}}(\rho)}$.

Following the idea of part (i) of the proof of Lemma 2.10, we construct a MWTA $\mathcal{A}' = (L', C', I', E', F', \mathrm{wt}')$ over Σ and $\mathbb{V}^{\text{tRatio}}$ such that $[\![\mathcal{A}']\!] = [\![\mathcal{A}]\!]$ and $\textsc{Cost}_{\mathcal{A}}(\rho) \neq 0$ for all $\rho \in \mathrm{Run}_{\mathcal{A}'}$ as follows:

- $L' = L \times \{0, 1\}$, $C' = C \cup \{x\}$ with $x \notin C$, $I' = I \times \{0\}$, $F' = F \times \{1\}$;
- $\mathrm{wt}'(\ell, j) = \mathrm{wt}(\ell)$ for all $\ell \in L$ and $j \in \{0, 1\}$;
- E' is constructed from E as follows. Let $e = (\ell, a, \phi, \Lambda, \ell') \in E$. Then, we add the following three transitions to E':
  - $e_1 := ((\ell, 1), a, \phi_1, \Lambda, (\ell', 1))$ with $\phi_1|_C = \phi|_C$ and $\phi_1(x) = [0, \infty)$. We put $\mathrm{wt}'(e_1) = \mathrm{wt}(e)$;
  - $e_2 := ((\ell, 0), a, \phi_2, \Lambda \cup \{x\}, (\ell', j))$ with $\phi_2|_C = \phi|_C$, $\phi_2(x) = [0, 0]$ and
  $$j = \begin{cases} 0, & \text{if } c_e = 0, \\ 1, & \text{otherwise.} \end{cases}$$
  We also put $\mathrm{wt}'(e_2) = \mathrm{wt}(e)$;
  - $e_3 := ((\ell, 0), a, \phi_3, \Lambda \cup \{x\}, (\ell', j))$ with $\phi_3|_C = \phi|_C$, $\phi_3(x) = (0, \infty)$ and
  $$j = \begin{cases} 0, & \text{if } c_\ell = c_e = 0, \\ 1, & \text{otherwise.} \end{cases}$$
  Again, we put $\mathrm{wt}'(e_3) = \mathrm{wt}(e)$.

Note that $\mathcal{A}'$ simulates all runs ρ of $\mathcal{A}$ with $\textsc{Cost}_{\mathcal{A}}(\rho) > 0$ and rejects all runs ρ with $\textsc{Cost}_{\mathcal{A}}(\rho) = 0$. Then, clearly, $[\![\mathcal{A}']\!] = [\![\mathcal{A}]\!]$.

The rest of the proof follows a similar idea as the proof of part (ii) of Lemma 2.10. Let $\theta \in \mathbb{Q}$ be a threshold. Note that

$$\exists w \in \mathbb{T}\Sigma^+ : [\![\mathcal{A}']\!](w) < \theta \text{ iff } \exists \rho \in \mathrm{Run}_{\mathcal{A}'} : \mathrm{wt}_{\mathcal{A}'}(\rho) < \theta.$$

We construct the linearly priced timed automaton $\mathcal{A}'' = (L', C', I', E', F', \mathrm{wt}'')$ over Σ and $\mathbb{V}^{\mathrm{T}\textsc{Sum}}$ (as defined in Example 7.9 (a)) as follows. For $u \in L' \cup E'$, let $\mathrm{wt}'(u) = (r_u, c_u)$. Then, we put $\mathrm{wt}''(u) = r_u - \theta \cdot c_u$. Since $\textsc{Cost}(\rho) > 0$ for all $\rho \in \mathrm{Run}_{\mathcal{A}'}$, we have:

$$(\exists \rho \in \mathrm{Run}_{\mathcal{A}'} : \mathrm{wt}_{\mathcal{A}'}(\rho) < \theta) \Leftrightarrow (\exists \rho \in \mathrm{Run}_{\mathcal{A}''} : \mathrm{wt}_{\mathcal{A}'}(\rho) < 0).$$

Moreover, it follows from the results of, e.g., [11], that the $<$-threshold problem for linearly priced timed automata is decidable. Then, the claim follows.

□

7.3 Closure properties

In this section, we consider some closure properties for recognizable quantitative timed languages. We fix an alphabet Σ and a timed valuation structure $\mathbb{V} = (M_L, M_E, (K, +, \mathbb{0}), \mathrm{val}^{\mathbb{T}})$.

Let Γ be an alphabet and $h : \Gamma \to \Sigma$ a mapping called a *renaming*. For a timed word $v = (\gamma_1, t_1)...(\gamma_n, t_n) \in \mathbb{T}\Gamma^+$, we let $h(v)$ be the timed word $(h(\gamma_1), t_1)...(h(\gamma_n), t_n) \in \mathbb{T}\Sigma^+$. Now assume that Γ is finite. Then, for a QTL $r : \mathbb{T}\Gamma^+ \to K$, we define the *renaming* $h(r) : \mathbb{T}\Sigma^+ \to K$ for all $w \in \mathbb{T}\Sigma^+$ by

$$h(r)(w) = \sum (r(v) \mid v \in \mathbb{T}\Gamma^+ \text{ and } h(v) = w).$$

Observe that, for any $w \in \mathbb{T}\Sigma^+$, there are only finitely many $v \in \mathbb{T}\Gamma^+$ with $h(v) = w$, hence the sum exists in $(K, +, \mathbb{0})$.

Lemma 7.11. *Let Γ be an alphabet and $h : \Gamma \to \Sigma$ a renaming. If a QTL $r : \mathbb{T}\Gamma^+ \to K$ is recognizable over $\mathbb{V}$, then the renaming $h(r) : \mathbb{T}\Sigma^+ \to K$ is also recognizable over $\mathbb{V}$.*

Proof. Let $\mathcal{A} = (L, C, I, E, F, \mathrm{wt})$ be a MWTA over Γ and $\mathbb{V}$ such that $[\![\mathcal{A}]\!] = r$. To construct a MWTA for the renaming $h(r)$, we use a similar idea as in [47], Lemma 1. Consider the MWTA $\mathcal{A}' = (L \times \Gamma, C, I', E', F \times \Gamma, \mathrm{wt}')$ over Σ and $\mathbb{V}$ where:

- $I' = I \times \{\gamma_0\}$ for some fixed $\gamma_0 \in \Gamma$;
- E' consists of all edges $e' = ((\ell_1, \gamma_1), h(\gamma_2), \phi, \Lambda, (\ell_2, \gamma_2))$ such that $e = (\ell_1, \gamma_2, \phi, \Lambda, \ell_2) \in E$ and $\gamma_1 \in \Gamma$. For such an edge e', we put $\mathrm{wt}'(e') = \mathrm{wt}(e)$;
- for any $\ell \in L$ and $\gamma \in \Gamma$: $\mathrm{wt}'(\ell, \gamma) = \mathrm{wt}(\ell)$.

Let $w = (a_1, t_1)...(a_n, t_n) \in \mathbb{T}\Sigma^+$ be a timed word. We define the mapping $\pi : \mathrm{Run}_{\mathcal{A}'}(w) \to \bigcup_{v \in h^{-1}(w)} \mathrm{Run}_{\mathcal{A}}(v)$ as follows. Let

$$\rho : ((\ell_0, \gamma_0), \nu_0) \xrightarrow{t_1} \xrightarrow{e_1} ((\ell_1, \gamma_1), \nu_1) \xrightarrow{t_2} \xrightarrow{e_2} ... \xrightarrow{t_n} \xrightarrow{e_n} ((\ell_n, \gamma_n), \nu_n)$$

be a run in $\mathrm{Run}_{\mathcal{A}'}(w)$. Assume that $e_i = ((\ell_{i-1}, \gamma_{i-1}), a_i, \phi_i, \Lambda_i, (\ell_i, \gamma_i))$ for all $1 \leq i \leq n$. Let $\overline{e}_i = (\ell_{i-1}, \gamma_i, \phi_i, \Lambda_i, \ell_i)$ for each $1 \leq i \leq n$. Then, by definition of $\mathcal{A}'$, $h(\gamma_i) = a_i$ for all $1 \leq i \leq n$ and

$$\overline{\rho} : (\ell_0, \nu_0) \xrightarrow{t_1} \xrightarrow{\overline{e}_1} (\ell_1, \nu_1) \xrightarrow{t_2} \xrightarrow{\overline{e}_2} ... \xrightarrow{t_n} \xrightarrow{\overline{e}_n} (\ell_n, \nu_n)$$

is a run of $\mathcal{A}$ with $h(\mathrm{label}(\overline{\rho})) = (h(\gamma_1), t_1)...(h(\gamma_n), t_n) = w$, i.e., $\mathrm{label}(\overline{\rho}) \in h^{-1}(w)$. Then, we put $\pi(\rho) = \overline{\rho}$. Clearly, π is bijective. Moreover, $\mathrm{wt}_{\mathcal{A}}(\pi(\rho)) = \mathrm{wt}_{\mathcal{A}'}(\rho)$ for all $\rho \in \mathrm{Run}_{\mathcal{A}'}(w)$. Then:

$$\begin{aligned} [\![\mathcal{A}']\!](w) &= \sum_{\rho \in \mathrm{Run}_{\mathcal{A}'}(w)} \mathrm{wt}_{\mathcal{A}'}(\rho) = \sum \Big(\mathrm{wt}_{\mathcal{A}'}(\pi^{-1}(\overline{\rho})) \mid \overline{\rho} \in \bigcup_{v \in h^{-1}(w)} \mathrm{Run}_{\mathcal{A}}(v) \Big) \\ &= \sum_{v \in h^{-1}(w)} \sum_{\overline{\rho} \in \mathrm{Run}_{\mathcal{A}}(v)} \mathrm{wt}_{\mathcal{A}}(\overline{\rho}) = \sum_{v \in h^{-1}(w)} [\![\mathcal{A}]\!](v) = h(r)(w) \end{aligned}$$

and hence $h(r) = [\![\mathcal{A}']\!]$. □

Let $g : \Gamma \to M_L \times M_E$ be a renaming. We define the *composition* $\mathrm{val}^{\mathbb{T}} \circ g : \mathbb{T}\Gamma^+ \to K$ for all $w = (\gamma_1, t_1)...(\gamma_n, t_n) \in \mathbb{T}\Gamma^+$ by

$$(\mathrm{val}^{\mathbb{T}} \circ g)(w) = \mathrm{val}^{\mathbb{T}}((g(\gamma_1), t_1)...(g(\gamma_n), t_n)).$$

We say that the timed valuation structure $\mathbb{V}$ is *location-independent* if, for any $v = ((m_1, m_1'), t_1)...((m_n, m_n'), t_n) \in \mathbb{T}(M_L \times M_E)^+$ and $v' = ((k_1, k_1'), t_1)...((k_n, k_n'), t_n) \in \mathbb{T}(M_L \times M_E)^+$ with $m_i' = k_i'$ for all $i \in \{1, ..., n\}$, we have $\mathrm{val}^{\mathbb{T}}(v) = \mathrm{val}^{\mathbb{T}}(v')$. If $\mathbb{V}$ is not location-independent, then we say that $\mathbb{V}$ is *location-dependent*.

Lemma 7.12. *Let Γ be a finite alphabet and $g : \Gamma \to M_L \times M_E$ a renaming. Then:*

(a) $\mathrm{val}^{\mathbb{T}} \circ g$ *is unambiguously recognizable over* $\mathbb{V}$.
(b) If $\mathbb{V}$ *is location-independent, then* $\mathrm{val}^{\mathbb{T}} \circ g$ *is sequentially recognizable over* $\mathbb{V}$.

Proof. Note that for multi-weighted Büchi automata and a renaming $g : \Gamma \to M$ which takes care of weights, we constructed a deterministic MWBA with one state. In the timed setting, the situation is different since we first stay in a location and then take an edge; moreover, the locations of MWTA are also equipped with weights. Then, we have to guess which letter we will read after staying in a location.

Let $g_L : \Gamma \to M_L$ and $g_E : \Gamma \to M_E$ be mappings such that $g(\gamma) = (g_L(\gamma), g_E(\gamma))$ for all $\gamma \in \Gamma$.

(a) Consider the MWTA $\mathcal{A} = (L, \emptyset, I, E, F, \mathrm{wt})$ over Γ and $\mathbb{V}$ such that:

- $L = I = \Gamma$ and $F = \{\gamma_f\}$ where $\gamma_f \in \Gamma$ is fixed.
- $E = \{(\gamma, \gamma, \textsc{True}, \emptyset, \gamma') \mid \gamma, \gamma' \in \Gamma\}$.
- $\mathrm{wt}(\ell) = g_L(\ell)$ for all $\ell \in L = \Gamma$ and $\mathrm{wt}(e) = g_E(\gamma)$ for all $e = (\gamma, \gamma, \textsc{True}, \emptyset, \gamma') \in E$

Let $w = (\gamma_1, t_1)...(\gamma_n, t_n) \in \mathbb{T}\Gamma^+$. Let $e_i = (\gamma_i, \gamma_i, \textsc{True}, \emptyset, \gamma_{i+1})$ for all $1 \leq i \leq n$ where $\gamma_{n+1} = \gamma_f$. Then, $\mathrm{Run}_{\mathcal{A}}(w) = \{\rho\}$ where

$$\rho = (\gamma_1, \nu_1) \xrightarrow{t_1} \xrightarrow{e_1} (\gamma_2, \nu_2) \xrightarrow{t_2} \xrightarrow{e_2} ... \xrightarrow{t_{n-1}} \xrightarrow{e_{n-1}} (\gamma_n, \nu_n) \xrightarrow{t_n} \xrightarrow{e_n} (\gamma_f, \nu_{n+1})$$

with $\nu_i = \emptyset$ for all $1 \leq i \leq n+1$. Then, $\mathcal{A}$ is unambiguous and

$$\begin{aligned} [\![\mathcal{A}]\!](w) &= \mathrm{wt}_{\mathcal{A}}(\rho) = \mathrm{val}^{\mathbb{T}}[((g_L(\gamma_1), g_E(\gamma_1)), t_1)...((g_L(\gamma_n), g_E(\gamma_n)), t_n)] \\ &= (\mathrm{val}^{\mathbb{T}} \circ g)(w) \end{aligned}$$

and hence $\mathrm{val}^{\mathbb{T}} \circ g = [\![\mathcal{A}]\!]$. This shows that the QTL $\mathrm{val}^{\mathbb{T}} \circ g : \mathbb{T}\Gamma^+ \to K$ is unambiguously recognizable over $\mathbb{V}$.

(b) Let $\mathcal{A} = (\{1\}, \emptyset, \{1\}, E, \{1\}, \mathrm{wt})$ be a MWTA over Γ and $\mathbb{V}$ with $E = \{(1, \gamma, \textsc{True}, \emptyset, 1) \mid \gamma \in \Gamma\}$, $\mathrm{wt}(1, \gamma, \textsc{True}, \emptyset, 1) = g_E(\gamma)$ for all $\gamma \in \Gamma$ and $\mathrm{wt}(1) = m_0$ where $m_0 \in M_L$ is fixed. Clearly, $\mathcal{A}$ is sequential. Let $w = (\gamma_1, t_1)...(\gamma_n, t_n) \in \mathbb{T}\Gamma^+$. Then, since $\mathbb{V}$ is location-independent, we have

$$\begin{aligned} [\![\mathcal{A}]\!](w) &= \mathrm{val}^{\mathbb{T}}[((m_0, g_E(\gamma_1)), t_1)...((m_0, g_E(\gamma_n)), t_n)] \\ &= \mathrm{val}^{\mathbb{T}}((g(\gamma_1), t_1)...(g(\gamma_n), t_n)) = (\mathrm{val}^{\mathbb{T}} \circ g)(w). \end{aligned}$$

This shows that $\mathrm{val}^{\mathbb{T}} \circ g$ is sequentially recognizable.

□

However, in general, the QTL $\mathrm{val}^{\mathbb{T}} \circ g$ is not sequentially recognizable over $\mathbb{V}$. Moreover, we can show that, in general, $\mathrm{val}^{\mathbb{T}} \circ g$ is not deterministically recognizable.

Lemma 7.13. *There exist an alphabet Γ, a location-dependent timed valuation structure $\mathbb{V} = (M_L, M_E, (K, +, \mathbb{0}), \mathrm{val}^{\mathbb{T}})$, and a renaming $g : \Gamma \to M_L \times M_E$ such that the QTL $\mathrm{val}^{\mathbb{T}} \circ g$ is not deterministically recognizable over $\mathbb{V}$.*

Proof. Let $\Gamma = \{a, b\}$ and $\mathbb{V} = \mathbb{V}^{\mathrm{T{\small SUM}}}$ as in Example 7.9 (a). Let $g(a) = (1, 0)$ and $g(b) = (2, 0)$. Suppose that there exists a deterministic MWTA $\mathcal{A} = (L, C, I, E, F, \mathrm{wt})$ over Γ and $\mathbb{V}^{\mathrm{T{\small SUM}}}$ such that $[\![\mathcal{A}]\!] = \mathrm{val}^{\mathrm{T{\small SUM}}} \circ g$. Let $I = \{\ell_0\}$, $A = \mathrm{wt}(\ell_0)$ and $B = \max\{|\mathrm{wt}(e)| \mid e \in E\}$. Let $\gamma \in \Gamma$ and $t \in \mathbb{R}_{\geq 0}$. Then,

$$A \cdot t - B \leq [\![\mathcal{A}]\!](\gamma, t) \leq A \cdot t + B.$$

By assumption, $[\![\mathcal{A}]\!](a,t) = \mathrm{val}^{\mathrm{tSum}}(g(a),t) = t$ and $[\![\mathcal{A}]\!](b,t) = 2 \cdot t$. Then, every $t \in \mathbb{R}_{\geq 0}$ is a solution of the following system of inequations:

$$\begin{cases} -B \leq (1-A) \cdot t \leq B, \\ -B \leq (2-A) \cdot t \leq B. \end{cases}$$

Then, every $t \in \mathbb{R}_{\geq 0}$ is a solution of the system

$$\begin{cases} |A-1| \cdot t \leq |B|, \\ |A-2| \cdot t \leq |B|. \end{cases} \tag{7.2}$$

There are two possibilities:

- $A \notin \{1,2\}$. Then, every $t > \min\left\{\frac{|B|}{|A-1|}, \frac{|B|}{|A-2|}\right\}$ is not a solution of system (7.2).
- $A \in \{1,2\}$. Then, every $t > |B|$ is not a solution of (7.2).

Thus, the QTL $\mathrm{val}^{\mathrm{tSum}} \circ g$ is not deterministically recognizable over $\mathbb{V}^{\mathrm{tSum}}$. $\square$

Let $\mathcal{L} \subseteq \mathbb{T}\Gamma^+$ be a timed language and $r : \mathbb{T}\Gamma^+ \to K$ a QTL. The *intersection* $(r \cap \mathcal{L}) : \mathbb{T}\Gamma^+ \to K$ is the QTL defined by $(r \cap \mathcal{L})(w) = r(w)$ if $w \in \mathcal{L}$ and $(r \cap \mathcal{L})(w) = \mathbb{0}$ if $w \in \mathbb{T}\Gamma^+ \setminus \mathcal{L}$.

As opposed to weighted untimed automata, the class of recognizable quantitative timed languages is not closed under the intersection with recognizable timed languages.

Lemma 7.14. *There exists an alphabet Γ, a non-idempotent timed valuation structure $\mathbb{V} = (M_L, M_E, (K, +, \mathbb{0}), \mathrm{val}^{\mathbb{T}})$, a recognizable timed language $\mathcal{L} \subseteq \mathbb{T}\Gamma^+$ and a QTL $r : \mathbb{T}\Gamma^+ \to K$ sequentially recognizable over $\mathbb{V}$ such that $r \cap \mathcal{L}$ is not recognizable over $\mathbb{V}$.*

Proof. Our proof will be based on the fact that $\mathrm{tRec}^{\mathrm{Unamb}}(\Gamma) \subsetneq \mathrm{tRec}(\Gamma)$. Let $\Gamma = \{a\}$ be a singleton alphabet. Recall that $\mathbb{N} = (\mathbb{N}, +, \cdot, 0, 1)$ is the semiring of natural numbers. Let $\mathcal{F} = \{f\} \subseteq \mathbb{N}^{\mathbb{R}_{\geq 0}}$ with $f(t) = 1$ for all $t \in \mathbb{R}_{\geq 0}$. Let $\mathbb{V} = \mathbb{V}^{\mathbb{N},\mathcal{F}}$ (cf. Example 7.9 (e)). For a timed language $\mathcal{L} \subseteq \mathbb{T}\Gamma^+$, let $\mathrm{char}(\mathcal{L}) : \mathbb{T}\Gamma^+ \to \mathbb{N}$ be the characteristic function of $\mathcal{L}$, i.e., $\mathrm{char}(\mathcal{L})(w) = 1$ for all $w \in \mathcal{L}$ and $\mathrm{char}(\mathcal{L})(w) = 0$ for all $w \notin \mathcal{L}$. We claim that, for any timed language $\mathcal{L} \subseteq \mathbb{T}\Gamma^+$:

$$\mathcal{L} \in \mathrm{tRec}^{\mathrm{Unamb}}(\Gamma) \text{ iff } \mathrm{char}(\mathcal{L}) \in \mathrm{tRec}(\Gamma, \mathbb{V}). \tag{7.3}$$

First, let $\mathcal{L} = \mathcal{L}(\mathcal{A})$ for an unambiguous timed automaton $\mathcal{A} = (L, C, I, E, F)$ over Γ. We construct from $\mathcal{A}$ the MWTA $\mathcal{A}' = (L, C, I, E, F, \mathrm{wt})$ over Γ and $\mathbb{V}$ where $\mathrm{wt}(\ell) = f$ for all $\ell \in L$ and $\mathrm{wt}(e) = 1$ for all $e \in E$. Clearly, for every run of $\mathcal{A}'$, $\mathrm{wt}_{\mathcal{A}'}(\rho) = 1$. Then, since $\mathcal{A}$ is unambiguous, for every $w \in \mathcal{L}$, we have $|\mathrm{Run}_{\mathcal{A}'}(w)| = 1$ and hence $[\![\mathcal{A}']\!](w) = 1 = \mathrm{char}(\mathcal{L})(w)$. Similarly, for every $w \notin \mathcal{L}$, $\mathrm{Run}_{\mathcal{A}'}(w) = \emptyset$ and hence $[\![\mathcal{A}']\!](w) = 0 = \mathrm{char}(\mathcal{L})(w)$. Conversely, assume that $\mathrm{char}(\mathcal{L})$ is recognizable over $\mathbb{V}$. Then, there exists a MWTA $\mathcal{A} = (L, C, I, E, F, \mathrm{wt})$ over Γ and $\mathbb{V}$ such that $[\![\mathcal{A}]\!] = \mathrm{char}(\mathcal{L})$. Let $E' = E \setminus \{e \in E \mid \mathrm{wt}(e) = 0\}$, $\mathrm{wt}' = \mathrm{wt}|_{E'}$ and $\mathcal{A}' = (L, C, I, E', F, \mathrm{wt}')$ be the MWTA over Γ and $\mathbb{V}$. Then, for all $w \in \mathbb{T}\Gamma^+$, $\mathrm{Run}_{\mathcal{A}'}(w) = \{\rho \in \mathrm{Run}_{\mathcal{A}}(w) \mid \mathrm{wt}_{\mathcal{A}}(\rho) \neq 0\}$ and hence $[\![\mathcal{A}']\!](w) = [\![\mathcal{A}]\!](w)$. Let $w \in \mathcal{L}$ and $\rho \in \mathrm{Run}_{\mathcal{A}'}(w)$. Assume that ρ has the form

$$\rho = (\ell_0, \nu_0) \xrightarrow{t_1}\xrightarrow{e_1} (\ell_1, \nu_1) \xrightarrow{t_2}\xrightarrow{e_2} \ldots \xrightarrow{t_n}\xrightarrow{e_n} (\ell_n, \nu_n).$$

Then, $\mathrm{wt}_{\mathcal{A}'}(\rho) = \mathrm{wt}(e_1) \cdot \ldots \cdot \mathrm{wt}(e_n) \geq 1$ and hence $1 = [\![\mathcal{A}']\!](w) \geq |\,\mathrm{Run}_{\mathcal{A}'}(w)|$ which implies $|\,\mathrm{Run}_{\mathcal{A}'}(w)| = 1$. Now let $w \notin \mathcal{L}$. Since $[\![\mathcal{A}']\!](w) = 0$ and $\mathrm{wt}_{\mathcal{A}'}(\rho) \geq 1$ for all $\rho \in \mathrm{Run}_{\mathcal{A}'}(w)$, we obtain $\mathrm{Run}_{\mathcal{A}'}(w) = \emptyset$. Then, the timed automaton $\mathcal{A}' = (L, C, I, E', F)$ is unambiguous and $\mathcal{L}(\mathcal{A}') = \mathcal{L}$. Hence, $\mathcal{L}$ is unambiguously recognizable.

Let $\mathcal{L} \in \mathrm{tRec}(\Gamma) \setminus \mathrm{tRec}^{\mathrm{Unamb}}(\Gamma) \neq \emptyset$ (cf. Lemma 7.4). Let the QTL $r : \mathbb{T}\Gamma^+ \to \mathbb{N}$ be defined by $r(w) = 1$ for all $w \in \mathbb{T}\Gamma^+$. Consider the sequential MWTA $\mathcal{A} = (\{1\}, \emptyset, \{1\}, \{(1, a, \mathrm{True}, \emptyset, 1)\}, \{1\}, \mathrm{wt})$ over Γ and $\mathbb{V}$ with $\mathrm{wt}(1) = \mathrm{wt}(1, a, \mathrm{True}, \emptyset, 1) = 1$. Clearly, $[\![\mathcal{A}]\!] = r$. Note that $r \cap \mathcal{L} = \mathrm{char}(\mathcal{L})$. Suppose that $r \cap \mathcal{L}$ is recognizable over $\mathbb{V}$. Then, by (7.3), $\mathcal{L} \in \mathrm{tRec}^{\mathrm{Unamb}}(\Gamma)$. A contradiction. $\square$

Lemma 7.15. *Let Γ be an alphabet, $\mathcal{L} \subseteq \mathbb{T}\Gamma^+$ a recognizable timed language, and $r : \mathbb{T}\Gamma^+ \to K$ a QTL recognizable over $\mathbb{V}$.*

(a) If $\mathbb{V}$ is idempotent, then $(r \cap \mathcal{L}) \in \mathrm{tRec}(\Gamma, \mathbb{V})$.
(b) Suppose that $\mathcal{L} \in \mathrm{tRec}^{\mathrm{Unamb}}(\Gamma)$. Then $(r \cap \mathcal{L}) \in \mathrm{tRec}(\Gamma, \mathbb{V})$.
(c) Suppose that $\mathcal{L} \in \mathrm{tRec}^{\mathrm{Unamb}}(\Gamma)$ and $r \in \mathrm{tRec}^{\mathrm{Unamb}}(\Gamma, \mathbb{V})$. Then $(r \cap \mathcal{L}) \in \mathrm{tRec}^{\mathrm{Unamb}}(\Gamma, \mathbb{V})$.
(d) If $\mathcal{L} \in \mathrm{tRec}^{\mathrm{Seq}}(\Gamma)$ and $r \in \mathrm{tRec}^{\mathrm{Seq}}(\Gamma, \mathbb{V})$, then $(r \cap \mathcal{L}) \in \mathrm{tRec}^{\mathrm{Seq}}(\Gamma, \mathbb{V})$.
(e) If $\mathcal{L} \in \mathrm{tRec}^{\mathrm{Det}}(\Gamma)$ and $r \in \mathrm{tRec}^{\mathrm{Det}}(\Gamma, \mathbb{V})$, then $(r \cap \mathcal{L}) \in \mathrm{tRec}^{\mathrm{Det}}(\Gamma, \mathbb{V})$.

Proof. Let $\mathcal{A} = (L, C, I, E, F)$ be a timed automaton over Γ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. Let $\mathcal{A}' = (L', C', I', E', F', \mathrm{wt}')$ be a MWTA over Γ and $\mathbb{V}$ such that $[\![\mathcal{A}']\!] = r$. We may assume that C and C' are disjoint sets of clocks. Let $\mathcal{A} \times \mathcal{A}' = (L \times L', C \cup C', I \times I', E'', F \times F', \mathrm{wt}'')$ be the MWTA over Γ and $\mathbb{V}$ where E'' consists of all edges $e'' = ((\ell_1, \ell_1'), a, \phi \wedge \phi', \Lambda \cup \Lambda', (\ell_2, \ell_2'))$ such that $e = (\ell_1, a, \phi, \Lambda, \ell_2) \in E$ and $e' = (\ell_1', a, \phi', \Lambda', \ell_2') \in E'$. For such an edge e'', we put $\mathrm{wt}''(e'') = \mathrm{wt}'(e')$. For any $\ell'' = (\ell, \ell') \in L \times L'$, we put $\mathrm{wt}''(\ell'') = \mathrm{wt}'(\ell')$.

(a) Let $\mathbb{V}$ be idempotent. Consider a timed word $w \in \mathbb{T}\Gamma^+ \setminus \mathcal{L}$. Then, $\mathrm{Run}_{\mathcal{A}}(w) = \emptyset$ and hence $\mathrm{Run}_{\mathcal{A} \times \mathcal{A}'}(w) = \emptyset$. Then,

$$[\![\mathcal{A} \times \mathcal{A}']\!](w) = \mathbb{0} = (r \cap \mathcal{L})(w).$$

If $w \in \mathcal{L}$, then $\mathrm{Run}_{\mathcal{A}}(w) \neq \emptyset$. Let $N = |\,\mathrm{Run}_{\mathcal{A}}(w)| > 0$. Since $\mathbb{V}$ is idempotent, $\sum_{i=1}^{N} \mathrm{wt}_{\mathcal{A}'}(\rho) = \mathrm{wt}_{\mathcal{A}'}(\rho)$ for all $\rho \in \mathrm{Run}_{\mathcal{A}'}(w)$. Then,

$$\begin{aligned} [\![\mathcal{A} \times \mathcal{A}']\!](w) &= \sum_{\rho \in \mathrm{Run}_{\mathcal{A}\times\mathcal{A}'}(w)} \mathrm{wt}_{\mathcal{A}\times\mathcal{A}'}(\rho) = \sum_{\rho \in \mathrm{Run}_{\mathcal{A}'}(w)} \sum_{i=1}^{N} \mathrm{wt}_{\mathcal{A}'}(\rho) \\ &= \sum_{\rho \in \mathrm{Run}_{\mathcal{A}'}(w)} \mathrm{wt}_{\mathcal{A}'}(\rho) = [\![\mathcal{A}']\!](w) = r(w) = (r \cap \mathcal{L})(w). \end{aligned}$$

This shows that $[\![\mathcal{A} \times \mathcal{A}']\!] = r \cap \mathcal{L}$. Thus, $r \cap \mathcal{L}$ is recognizable over $\mathbb{V}$.

(b) Let $\mathcal{A}$ be unambiguous. Then, as in the previous case, for all $w \in \mathbb{T}\Gamma^+ \setminus \mathcal{L}$, $[\![\mathcal{A} \times \mathcal{A}']\!](w) = \mathbb{0} = (r \cap \mathcal{L})(w)$. For $w \in \mathcal{L}$, $|\mathrm{Run}_{\mathcal{A}}(w)| = 1$. Then,

$$\begin{aligned} [\![\mathcal{A} \times \mathcal{A}']\!](w) &= \sum_{\rho \in \mathrm{Run}_{\mathcal{A}\times\mathcal{A}'}(w)} \mathrm{wt}_{\mathcal{A}\times\mathcal{A}'}(\rho) = \sum_{\rho \in \mathrm{Run}_{\mathcal{A}'}(w)} \mathrm{wt}_{\mathcal{A}'}(\rho) \\ &= [\![\mathcal{A}']\!](w) = r(w) = (r \cap \mathcal{L})(w). \end{aligned}$$

Then, $r \cap \mathcal{L}$ is recognizable over $\mathbb{V}$.

(c) Let $\mathcal{A}, \mathcal{A}'$ be unambiguous. Then, clearly, $\mathcal{A} \times \mathcal{A}'$ is unambiguous. As it was shown in (b), $[\![\mathcal{A} \times \mathcal{A}']\!] = r \cap \mathcal{L}$. Thus, $r \cap \mathcal{L}$ is unambiguously recognizable over $\mathbb{V}$.

(d) Let $\mathcal{A}, \mathcal{A}'$ be sequential. Then, clearly, $\mathcal{A} \times \mathcal{A}'$ is sequential. As it was shown in (b), $[\![\mathcal{A} \times \mathcal{A}']\!] = r \cap \mathcal{L}$. Thus, $r \cap \mathcal{L}$ is sequentially recognizable over $\mathbb{V}$.

(e) Let $\mathcal{A}, \mathcal{A}'$ be deterministic. It can be easily shown that $\mathcal{A} \times \mathcal{A}'$ is also deterministic. Then, like in (b), $[\![\mathcal{A} \times \mathcal{A}']\!] = r \cap \mathcal{L}$. Thus, $r \cap \mathcal{L}$ is deterministically recognizable over $\mathbb{V}$.

$\square$

7.4 A Nivat theorem for multi-weighted timed automata

In this section, we present a Nivat-like characterization of multi-weighted timed automata. This result illustrates the connection between recognizable quantitative and qualitative languages (cf. the Nivat theorem for semiring-weighted automata [38] and Theorem 5.6 for multi-weighted Büchi automata).

Let Σ be an alphabet and $\mathbb{V} = (M_L, M_E, (K, +, \mathbb{0}), \mathrm{val}^{\mathbb{T}})$ a timed valuation structure. Let $\mathcal{N}(\Sigma, \mathbb{V})$ (with $\mathcal{N}$ standing for Nivat) denote the class of all QTL $\mathbb{L} : \mathbb{T}\Sigma^+ \to K$ such that there exist an alphabet Γ, renamings $h : \Gamma \to \Sigma$ and $g : \Gamma \to M_L \times M_E$, and a recognizable timed language $\mathcal{L} \subseteq \mathbb{T}\Gamma^+$ such that $\mathbb{L} = h((\mathrm{val}^{\mathbb{T}} \circ g) \cap \mathcal{L})$. Notice that these decompositions are defined in a similar way as for ω-words (cf. Theorem 5.6). Here, val^{ω} is replaced by $\mathrm{val}^{\mathbb{T}}$ and $g : \Gamma \to M$ by $g : \Gamma \to (M_L \times M_E)$. However, in contrast to Theorem 5.6, the equality $\mathrm{tRec}(\Sigma, \mathbb{V}) = \mathcal{N}(\Sigma, \mathbb{V})$ does not always hold.

Lemma 7.16. *There exist an alphabet Σ and a non-idempotent and location-independent timed valuation structure $\mathbb{V}$ such that* $\text{tRec}(\Sigma, \mathbb{V}) \neq \mathcal{N}(\Sigma, \mathbb{V})$.

Proof. Let $\Sigma = \{a\}$ be a singleton alphabet and $\mathbb{V} = (M_L, M_E, (K, +, \mathbb{0}), \mathrm{val}^{\mathbb{T}})$ be as in the proof of Lemma 7.14. Let $\mathcal{L} \in \text{tRec}(\Sigma) \setminus \text{tRec}^{\text{Unamb}}(\Sigma) \neq \emptyset$ (cf. Lemma 7.4). Consider the alphabet $\Gamma = \Sigma$ and the identity mapping $\mathrm{id} : \Gamma \to \Sigma$, i.e. $\mathrm{id}(\gamma) = \gamma$ for all $\gamma \in \Gamma$. Consider also the mapping $g : \Gamma \to M_L \times M_E$ defined by $g(\gamma) = (1, 1)$ for all $\gamma \in \Gamma$ and the QTL $r : \mathbb{T}\Gamma^+ \to K$ with $r(u) = 1$ for all $u \in \mathbb{T}\Gamma^+$. Then, $\mathrm{val}^{\mathbb{T}} \circ g = r$ and $r \cap \mathcal{L} = \mathrm{id}((\mathrm{val}^{\mathbb{T}} \circ g) \cap \mathcal{L}) \in \mathcal{N}(\Sigma, \mathbb{V})$. However, it was shown in Lemma 7.14 that $r \cap \mathcal{L} \notin \text{tRec}(\Sigma, \mathbb{V})$. This proves that $\text{tRec}(\Sigma, \mathbb{V}) \neq \mathcal{N}(\Sigma, \mathbb{V})$. $\square$

Let the collection $\mathcal{N}^{\text{Seq}}(\Sigma, \mathbb{V})$ be defined like $\mathcal{N}(\Sigma, \mathbb{V})$ with the only difference that $\mathcal{L}$ is sequential. Similarly, let the collections $\mathcal{N}^{\text{Unamb}}(\Sigma, \mathbb{V})$ and $\mathcal{N}^{\text{Det}}(\Sigma, \mathbb{V})$ be defined like $\mathcal{N}(\Sigma, \mathbb{V})$ with the only difference that $\mathcal{L}$ is unambiguously recognizable resp. deterministically recognizable. Clearly,

$$\mathcal{N}^{\text{Seq}}(\Sigma, \mathbb{V}) \subseteq \mathcal{N}^{\text{Det}}(\Sigma, \mathbb{V}) \subseteq \mathcal{N}^{\text{Unamb}}(\Sigma, \mathbb{V}) \subseteq \mathcal{N}(\Sigma, \mathbb{V}). \tag{7.4}$$

Our Nivat theorem for MWTA is the following.

Theorem 7.17. *Let Σ be an alphabet and $\mathbb{V}$ a timed valuation structure. Then,*

(a) $\text{tRec}(\Sigma, \mathbb{V}) = \mathcal{N}^{\text{Seq}}(\Sigma, \mathbb{V}) = \mathcal{N}^{\text{Det}}(\Sigma, \mathbb{V}) = \mathcal{N}^{\text{Unamb}}(\Sigma, \mathbb{V})$.
(b) $\text{tRec}(\Sigma, \mathbb{V}) \subseteq \mathcal{N}(\Sigma, \mathbb{V})$. *If $\mathbb{V}$ is idempotent, then* $\text{Rec}(\Sigma, \mathbb{V}) = \mathcal{N}(\Sigma, \mathbb{V})$.

The proof of this theorem follows from the next two lemmas.

Lemma 7.18. $\text{tRec}(\Sigma, \mathbb{V}) \subseteq \mathcal{N}^{\text{Seq}}(\Sigma, \mathbb{V})$.

Proof. Let $\mathcal{A} = (L, C, I, E, F, \mathrm{wt})$ be a MWTA over Σ and $\mathbb{V}$ such that $\mathbb{L} = [\![\mathcal{A}]\!]$. Let $\Gamma = E$. We define the renamings $h : \Gamma \to \Sigma$ and $g : \Gamma \to M_L \times M_E$ for all $\gamma = (\ell, a, \phi, \Lambda, \ell') \in \Gamma$ by $h(\gamma) = a$ and $g(\gamma) = (\mathrm{wt}(\ell), \mathrm{wt}(\gamma))$. We construct the timed automaton $\mathcal{A}' = (L, C, I, E', F)$ over Γ with

$$E' = \{(\ell, \gamma, \phi, \Lambda, \ell') \mid \exists a \in \Sigma : \gamma = (\ell, a, \phi, \Lambda, \ell') \in E\}.$$

We put $\mathcal{L} = \mathcal{L}(\mathcal{A}')$. We show that $\mathcal{L}$ is sequentially recognizable. Indeed, let $\perp \notin L$ and consider the timed automaton $\mathcal{A}'' = (L \cup \{\perp\}, C, \{\perp\}, E'', F)$ over Γ with

$$E'' = E' \cup \{(\perp, \gamma, \phi, \Lambda, \ell') \mid \exists \ell \in I : (\ell, \gamma, \phi, \Lambda, \ell') \in E'\}.$$

It is easy to see that $\mathcal{A}''$ is sequential and $\mathcal{L}(\mathcal{A}'') = \mathcal{L}$.

It remains to show that $\mathbb{L} = h((\mathrm{val}^{\mathbb{T}} \circ g) \cap \mathcal{L})$. Let $w = (a_1, t_1)...(a_n, t_n) \in \mathbb{T}\Sigma^+$. We define the mapping $\pi : \mathrm{Run}_{\mathcal{A}}(w) \to \{v \in \mathcal{L} \mid h(v) = w\}$ as follows. Let $\rho \in \mathrm{Run}_{\mathcal{A}}(w)$ be a run of the form

$$\rho = (\ell_0, \nu_0) \xrightarrow{t_1} \xrightarrow{\gamma_1} (\ell_1, \nu_1) \xrightarrow{t_2} \xrightarrow{\gamma_2} ... \xrightarrow{t_n} \xrightarrow{\gamma_n} (\ell_n, \nu_n). \tag{7.5}$$

Then, we put $\pi(\rho) = (\gamma_1, t_1)...(\gamma_n, t_n)$. Clearly, for all $i \in \{1, ..., n\}$, $\gamma_i = (\ell_{i-1}, a_i, \phi_i, \Lambda_i, \ell_i)$ for some $\phi_i \in \Phi(C)$ and $\Lambda_i \subseteq C$, and so $h(\pi(\rho)) = w$. For $1 \leq i \leq n$, let $e_i = (\ell_{i-1}, \gamma_i, \phi_i, \Lambda_i, \ell_i) \in E'$. Then,

$$(\ell_0, \nu_0) \xrightarrow{t_1} \xrightarrow{e_1} (\ell_1, \nu_1) \xrightarrow{t_2} \xrightarrow{e_2} ... \xrightarrow{t_n} \xrightarrow{e_n} (\ell_n, \nu_n) \tag{7.6}$$

is a run in $\mathrm{Run}_{\mathcal{A}'}(\pi(\rho))$ and hence $\pi(\rho) \in \mathcal{L}(\mathcal{A}') = \mathcal{L}$. This shows that π is correctly defined. Now we show that π is bijective. Indeed, let $\rho, \rho' \in \mathrm{Run}_{\mathcal{A}}(w)$ where ρ is defined as in (7.5) and

$$\rho' = (\ell_0', \nu_0') \xrightarrow{t_1} \xrightarrow{\gamma_1'} (\ell_1', \nu_1') \xrightarrow{t_2} \xrightarrow{\gamma_2'} ... \xrightarrow{t_n} \xrightarrow{\gamma_n'} (\ell_n', \nu_n').$$

Assume that $\rho \neq \rho'$. Then, there exists $i \in \{1, ..., n\}$ such that $\gamma_i \neq \gamma_i'$. Then, $\pi(\rho) \neq \pi(\rho')$ and hence π is injective. It remains to show that π is onto. Let $v' = (\delta_1, t_1)...(\delta_n, t_n) \in \mathcal{L}$ with $h(v') = w$. Then, there exists a run $\varrho \in \mathrm{Run}_{\mathcal{A}'}(v')$ of the form (7.6) where, for all $1 \leq i \leq n$, $e_i = (\ell_{i-1}, \delta_i, \phi_i, \Lambda_i, \ell_i)$ for some $\phi_i \in \Phi(C)$ and $\Lambda_i \subseteq C$. Then, there exists $b_i \in \Sigma$ with $\delta_i = (\ell_{i-1}, b_i, \phi_i, \Lambda_i, \ell_i) \in E$. Since $h(v') = w$, we have $b_i = a_i$. Then,

$$\varrho' = (\ell_0, \nu_0) \xrightarrow{t_1} \xrightarrow{\delta_1} (\ell_1, \nu_1) \xrightarrow{t_2} \xrightarrow{\delta_2} ... \xrightarrow{t_n} \xrightarrow{\delta_n} (\ell_n, \nu_n)$$

is a run in $\mathrm{Run}_{\mathcal{A}}(w)$ with $\pi(\varrho') = v'$.

Now let $\rho \in \mathrm{Run}_{\mathcal{A}}(w)$ be a run of the form (7.5). Then,

$$(\mathrm{val}^{\mathbb{T}} \circ g)(\pi(\rho)) = \mathrm{val}^{\mathbb{T}}(g(\pi(\rho))) = \mathrm{val}^{\mathbb{T}}((g(\gamma_1), t_1)...(g(\gamma_n), t_n)) = \mathrm{wt}_{\mathcal{A}}(\rho).$$

Thus,

$$\begin{aligned} h((\mathrm{val}^{\mathbb{T}} \circ g) \cap \mathcal{L})(w) &= \sum((\mathrm{val}^{\mathbb{T}} \circ g)(v) \mid v \in \mathcal{L} \text{ and } h(v) = w) \\ &= \sum(\mathrm{wt}_{\mathcal{A}}(\rho) \mid \rho \in \mathrm{Run}_{\mathcal{A}}(w)) = [\![\mathcal{A}]\!](w) = \mathbb{L}(w). \end{aligned}$$

□

Lemma 7.19. *(a)* $\mathcal{N}^{\mathrm{UNAMB}}(\Sigma, \mathbb{V}) \subseteq \mathrm{tREC}(\Sigma, \mathbb{V})$.
(b) Let $\mathbb{V}$ be idempotent. Then, $\mathcal{N}(\Sigma, \mathbb{V}) \subseteq \mathrm{tREC}(\Sigma, \mathbb{V})$.

Proof. (a) The result follows by successive application of Lemmas 7.11, 7.12 and 7.15(b).

(b) The result follows by successive application of 7.11, 7.12 and 7.15(a).

□

Proof of Theorem 7.17. Immediate by Lemmas 7.18 and 7.19 and the chain of inclusions (7.4).

7.5 Renamings of recognizable quantitative timed languages

In Corollary 5.11, we stated the coincidence of the class of recognizable quantitative ω-languages with the class of the renamings of deterministically recognizable quantitative ω-languages. Here, we will study this connection in the context of multi-weighted timed automata.

Let Σ be an alphabet and $\mathbb{V} = (M_L, M_E, (K, +, \mathbb{0}), \mathrm{val}^{\mathbb{T}})$ a timed valuation structure. We introduce the following abbreviations.

- Let $\mathcal{H}^{\textsc{Unamb}}(\Sigma, \mathbb{V})$ denote the collection of all QTL $\mathbb{L} : \mathbb{T}\Sigma^+ \to K$ such that there exist an alphabet Γ, a renaming $h : \Gamma \to \Sigma$ and an unambiguously recognizable QTL $r : \mathbb{T}\Gamma^+ \to K$ over $\mathbb{V}$ such that $\mathbb{L} = h(r)$.
- The collection $\mathcal{H}^{\textsc{Det}}(\Sigma, \mathbb{V})$ is also defined like $\mathcal{H}^{\textsc{Unamb}}(\Sigma, \mathbb{V})$ with the only difference that r is deterministically recognizable over $\mathbb{V}$.
- The collection $\mathcal{H}^{\textsc{Seq}}(\Sigma, \mathbb{V})$ is defined like $\mathcal{H}^{\textsc{Unamb}}(\Sigma, \mathbb{V})$ with the only difference that r is sequentially recognizable over $\mathbb{V}$.

The following theorem compares the classes $\mathcal{H}^{\textsc{Seq}}(\Sigma, \mathbb{V})$, $\mathcal{H}^{\textsc{Det}}(\Sigma, \mathbb{V})$, $\mathcal{H}^{\textsc{Unamb}}(\Sigma, \mathbb{V})$, and $\textsc{tRec}(\Sigma, \mathbb{V})$.

Theorem 7.20. *(a) For every alphabet Σ and a timed valuation structure $\mathbb{V}$:*

$$\mathcal{H}^{\textsc{Seq}}(\Sigma, \mathbb{V}) = \mathcal{H}^{\textsc{Det}}(\Sigma, \mathbb{V}) \subseteq \mathcal{H}^{\textsc{Unamb}}(\Sigma, \mathbb{V}) = \textsc{tRec}(\Sigma, \mathbb{V}).$$

Moreover, if $\mathbb{V}$ is location-independent, then $\mathcal{H}^{\textsc{Det}}(\Sigma, \mathbb{V}) = \mathcal{H}^{\textsc{Unamb}}(\Sigma, \mathbb{V})$.

(b) There exist an alphabet Σ and a location-dependent timed valuation structure $\mathbb{V}$ such that $\mathcal{H}^{\textsc{Det}}(\Sigma, \mathbb{V}) \neq \mathcal{H}^{\textsc{Unamb}}(\Sigma, \mathbb{V})$.

The proof of this theorem will follow from the lemmas of the rest of this section.

Lemma 7.21. $\mathcal{H}^{\textsc{Unamb}}(\Sigma, \mathbb{V}) = \textsc{tRec}(\Sigma, \mathbb{V})$.

Proof. The inclusion from left to right follows from Lemma 7.11. It remains to show that $\textsc{tRec}(\Sigma, \mathbb{V}) \subseteq \mathcal{H}^{\textsc{Unamb}}(\Sigma, \mathbb{V})$. Let $\mathbb{L} \in \textsc{tRec}(\Sigma, \mathbb{V})$. Then, by Theorem 7.17, $\mathbb{L} \in \mathcal{N}^{\textsc{Unamb}}(\Sigma, \mathbb{V})$, i.e., there exist an alphabet Γ, renamings $h : \Gamma \to \Sigma$ and $g : \Gamma \to M_L \times M_E$, and a timed language $\mathcal{L} \in \textsc{tRec}^{\textsc{Unamb}}(\Sigma)$ such that $\mathbb{L} = h((\mathrm{val}^{\mathbb{T}} \circ g) \cap \mathcal{L})$. By Lemma 7.12 (a), the QTL $(\mathrm{val}^{\mathbb{T}} \circ g)$ is unambiguously recognizable over $\mathbb{V}$. Consider the QTL $r = ((\mathrm{val}^{\mathbb{T}} \circ g) \cap \mathcal{L}) : \mathbb{T}\Gamma^+ \to K$. By Lemma 7.15 (c), r is unambiguously recognizable over $\mathbb{V}$. Since $\mathbb{L} = h(r)$, we have: $\mathbb{L} \in \mathcal{H}^{\textsc{Unamb}}(\Sigma, \mathbb{V})$. □

Lemma 7.22. *Let $\mathbb{V}$ be location-independent. Then, $\mathcal{H}^{\textsc{Seq}}(\Sigma, \mathbb{V}) = \textsc{tRec}(\Sigma, \mathbb{V})$.*

Proof. The inclusion $\mathcal{H}^{\textsc{Seq}}(\Sigma, \mathbb{V}) \subseteq \textsc{tRec}(\Sigma, \mathbb{V})$ follows from Lemma 7.11. We show that $\textsc{tRec}(\Sigma, \mathbb{V}) \subseteq \mathcal{H}^{\textsc{Seq}}(\Sigma, \mathbb{V})$. Let $\mathbb{L} \in \textsc{tRec}(\Sigma, \mathbb{V})$. Then, by Theorem 7.17, there exist an alphabet Γ, renamings $h : \Gamma \to \Sigma$, $g : \Gamma \to M_L \times M_E$ and a sequentially recognizable timed language $\mathcal{L} \subseteq \mathbb{T}\Gamma^+$

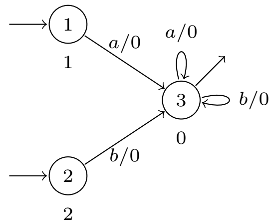

Figure 7.3: MWTA $\mathcal{A}_{\mathbb{L}}$ from the proof of Lemma 7.23

such that $\mathbb{L} = h((\mathrm{val}^{\mathbb{T}} \circ g) \cap \mathcal{L})$. Since $\mathbb{V}$ is location-independent, by Lemma 7.12 (b), the QTL $(\mathrm{val}^{\mathbb{T}} \circ g) : \mathbb{T}\Gamma^+ \to K$ is sequentially recognizable over $\mathbb{V}$. Consider the QTL $r = ((\mathrm{val}^{\mathbb{T}} \circ g) \cap \mathcal{L}) : \mathbb{T}\Gamma^+ \to K$. By Lemma 7.15 (d), $r \in \mathrm{tRec}^{\mathrm{Seq}}(\Sigma, \mathbb{V})$. Since $\mathbb{L} = h(r)$, we have: $\mathbb{L} \in \mathcal{H}^{\mathrm{Seq}}(\Sigma, \mathbb{V})$. □

Lemma 7.23. *There exists an alphabet Σ and a location-dependent timed valuation structure $\mathbb{V}$ such that $\mathcal{H}^{\mathrm{Seq}}(\Sigma, \mathbb{V}) \neq \mathrm{Rec}(\Sigma, \mathbb{V})$.*

Proof. Let $\Sigma = \{a, b\}$ and let $\mathbb{V} = (M_L, M_E, (K, +, \mathbb{0}), \mathrm{val}^{\mathbb{T}}) = \mathbb{V}^{\mathrm{tSum}}$ as in Example 7.9 (a). Consider the QTL $\mathbb{L} : \mathbb{T}\Sigma^+ \to K$ over $\mathbb{V}$ defined by

$$\mathbb{L}((a_1, t_1)...(a_n, t_n)) = \begin{cases} t_1, & \text{if } a_1 = a, \\ 2 \cdot t_1, & \text{otherwise.} \end{cases}$$

Consider the MWTA $\mathcal{A}_{\mathbb{L}}$ over Σ and $\mathbb{V}$ depicted in Figure 7.3 whose set of clocks is empty (here, the numbers, depicted under the locations of $\mathcal{A}_{\mathbb{L}}$, mean the weights of these locations). Then, $[\![\mathcal{A}]\!]_{\mathbb{L}} = \mathbb{L}$ and hence $\mathbb{L} \in \mathrm{tRec}(\Sigma, \mathbb{V})$.

Suppose that there exist a finite alphabet Γ, a mapping $h : \Gamma \to \Sigma$ and a sequentially recognizable QTL $r : \mathbb{T}\Gamma^+ \to K$ over $\mathbb{V}$ such that $\mathbb{L} = h(r)$. Let $\mathcal{A}' = (L', C', I', E', F', \mathrm{wt}')$ be a sequential MWTA over Γ and $\mathbb{V}$ such that $[\![\mathcal{A}']\!] = r$. Let $I' = \{\perp\}$. Let $\nu_0 : C \to \mathbb{R}_{\geq 0}$ be the clock valuation with $\nu_0(x) = 0$ for all $x \in C$. Let $A = \mathrm{wt}'(\perp)$ and, for $\sigma \in \Sigma$ and $t \in \mathbb{R}_{\geq 0}$, $B_{\sigma,t} = \min\{\mathrm{wt}'(e) \mid e = (\perp, \gamma, \phi, \Lambda, \ell) \in E', h(\gamma) = \sigma, \nu_0 + t \models \phi \text{ and } \ell \in F'\}$. Note that, for any $\sigma \in \Sigma$, the set $\{B_{\sigma,t} \mid t > 0\}$ is finite. For any $t > 0$,

$$t = [\![\mathcal{A}]\!](a, t) = \min\{[\![\mathcal{A}']\!](\gamma, t) \mid \gamma \in \Gamma, h(\gamma) = a\} = A \cdot t + B_{a,t}$$

and $2 \cdot t = [\![\mathcal{A}]\!](b, t) = A \cdot t + B_{b,t}$. Then, $A \neq \infty$ and for any $t > 0$, $B_{a,t} \neq \infty$ and $B_{b,t} \neq \infty$. We consider the following possibilities:

- $A = 1$. Then, the set $\{B_{b,t} \mid t > 0\} = \{t \mid t > 0\}$ is infinite which is impossible;
- $A \neq 1$. Then, the set $\{B_{a,t} \mid t > 0\} = \{(1 - A) \cdot t \mid t > 0\}$ is infinite which is also impossible.

Hence $\mathbb{L} \notin \mathcal{H}^{\mathrm{Seq}}(\Sigma, \mathbb{V})$. □

Lemma 7.24. $\mathcal{H}^{\text{SEQ}}(\Sigma, \mathbb{V}) = \mathcal{H}^{\text{DET}}(\Sigma, \mathbb{V})$.

Proof. The inclusion $\mathcal{H}^{\text{SEQ}}(\Sigma, \mathbb{V}) \subseteq \mathcal{H}^{\text{DET}}(\Sigma, \mathbb{V})$ is trivial. We show the converse inclusion. Let $\mathbb{L} \in \mathcal{H}^{\text{DET}}(\Sigma, \mathbb{V})$. Then, there exist an alphabet Γ, a renaming $h : \Gamma \to \Sigma$ and a deterministic MWTA $\mathcal{A} = (L, C, I, E, F, \text{wt})$ over Γ and $\mathbb{V}$ such that $\mathbb{L} = h([\![\mathcal{A}]\!])$. Assume that $C = \{x_1, ..., x_m\} \neq \emptyset$ and $|C| = m$. For all $1 \leq i \leq m$, let $\mathcal{K}_i'$ be the set of all $k \in \mathbb{N}$ such that there exists an edge $e \in E$ with the clock constraint $\phi \in \Phi(C)$ such that either $k = \inf \phi(x_i)$ or $k = \sup \phi(x_i)$; note that $\phi(x_i) \in \mathcal{I}$ is an interval. Let $\mathcal{K}_i = \mathcal{K}_i' \cup \{0, \infty\}$. Assume that, for all $1 \leq i \leq m$, $\mathcal{K}_i = \{k_{i,1}, ..., k_{i,l_i}\}$ with $0 = k_{i,1} < k_{i,2} < ... < k_{i,l_i} = \infty$. For any $1 \leq i \leq m$, let $\mathcal{R}_i = \{[k_{i,j}, k_{i,j}] \mid 1 \leq j < l_i\} \cup \{(k_{i,j}, k_{i,j+1}) \mid 1 \leq j < l_i\}$ be a finite set of intervals. Let $\mathcal{R} = \mathcal{R}_1 \times ... \times \mathcal{R}_m$. For $(r_1, ..., r_m) \in \mathcal{R}$ and a clock valuation $\nu \in \mathbb{R}_{\geq 0}^C$, we will write $\nu \models (r_1, ..., r_m)$ iff $\nu(x_i) \in r_i$ for all $1 \leq i \leq m$. Note that, for all $\nu \in \mathbb{R}_{\geq 0}^C$ there exists exactly one tuple $(r_1, ..., r_m) \in \mathcal{R}$ such that $\nu \models (r_1, ..., r_m)$. For $\varrho = (r_1, ..., r_m) \in \mathcal{R}$ and $\phi \in \Phi(C)$, we will write $\varrho \models \phi$ if $r_i \subseteq \phi(x_i)$ for all $1 \leq i \leq m$.

Let $\phi \in \Phi(C)$ be a clock constraint which appears in E. Then, for all clock valuations $\nu \in \mathbb{R}_{\geq 0}^C$, we have: $\nu \models \phi$ if and only if there exists exactly one $\varrho \in \mathcal{R}$ such that $\nu \models \varrho$ and $\varrho \models \phi$. Consider the alphabet $\Gamma' = \Gamma \times \mathcal{R}$ and the mapping $h' : \Gamma' \to \Sigma$ defined for all $\gamma \in \Gamma$, $\varrho \in \mathcal{R}$ by $h'(\gamma, \varrho) = h(\gamma)$. We define the MWTA $\mathcal{A}' = (L, C, I, E', F, \text{wt}')$ over Γ' and $\mathbb{V}$ where E' consists of all edges $e' = (\ell, (\gamma, \varrho), \phi, \Lambda, \ell')$ such that $e := (\ell, \gamma, \phi, \Lambda, \ell') \in E$ and $\varrho \models \phi$. For such an edge e', we put $\text{wt}'(e') = \text{wt}(e)$.

First, we show that $\mathcal{A}'$ is sequential. Indeed, let $e_1' = (\ell, (\gamma, \varrho), \phi_1, \Lambda_1, \ell_1')$ and $e_2' = (\ell, (\gamma, \varrho), \phi_2, \Lambda_2, \ell_2')$ be edges in E'. Then, $(\ell, \gamma, \phi_1, \Lambda_1, \ell_1') \in E$ and $(\ell, \gamma, \phi_2, \Lambda_2, \ell_2') \in E$. Moreover, there exists $\nu \in \mathbb{R}_{\geq 0}^C$ with $\nu \models \varrho$. Since $\varrho \models \phi_1$ and $\varrho \models \phi_2$, we have $\nu \models \phi_1$ and $\nu \models \phi_2$. Since $\mathcal{A}$ is deterministic, we have $\phi_1 = \phi_2$, $\Lambda_1 = \Lambda_2$ and $\ell_1' = \ell_2'$. This shows that $e_1' = e_2'$.

Let $g : \Gamma' \to \Gamma$ be defined for all $\gamma \in \Gamma$ and $\varrho \in \mathcal{R}$ by $g(\gamma, \varrho) = \gamma$, so $h' = h \circ g$. It is easy to see that $[\![\mathcal{A}]\!] = g([\![\mathcal{A}']\!])$. Then:

$$\mathbb{L} = h([\![\mathcal{A}]\!]) = h(g([\![\mathcal{A}']\!])) = (h \circ g)([\![\mathcal{A}']\!]) = h'([\![\mathcal{A}']\!]).$$

Thus, $\mathbb{L} \in \mathcal{H}^{\text{SEQ}}(\Sigma, \mathbb{V})$. $\square$

Consider the Boolean semiring $\mathbb{B}$ and the mapping $f : \{0, 1\} \to \{1\}$. Let $\mathcal{F} = \{f\}$. Then, $\mathbb{V}^{\mathbb{B}, \mathcal{F}}$ (cf. Example 7.9(e)) is a location independent timed valuation structure. Note that every recognizable language $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ can be interpreted as a recognizable QTL over $\mathbb{V}^{\mathbb{B}, \mathcal{F}}$. Let $\mathcal{H}^{\text{UNAMB}}(\Sigma)$ denote the class of all timed languages $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ such that there exist an alphabet Γ, a renaming $h : \Gamma \to \Sigma$ and an unambiguously recognizable language $\mathcal{L}' \subseteq \mathbb{T}\Gamma^+$ with $\mathcal{L} = h(\mathcal{L}')$. The classes $\mathcal{H}^{\text{DET}}(\Sigma)$ and $\mathcal{H}^{\text{SEQ}}(\Sigma)$ are defined similarly, using languages $\mathcal{L}'$ which are deterministically recognizable resp. sequentially recognizable.

As a corollary from Theorem 7.20, we obtain:

Corollary 7.25. *Let Σ be an alphabet. Then,*

$$\mathcal{H}^{\text{SEQ}}(\Sigma) = \mathcal{H}^{\text{DET}}(\Sigma) = \mathcal{H}^{\text{UNAMB}}(\Sigma) = \text{tREC}(\Sigma).$$

Chapter 8

Timed weight assignment logic

Contents

In this chapter, we establish a Büchi-Elgot characterization for multi-weighted timed automata of Chapter 7. We introduce a *timed weight assignment logic* which extends the weight assignment logic of Chapter 6 to the timed setting; the qualitative basis of this logic is Wilke's *relative distance logic*. We prove that our timed weight assignment logic is equally expressive as multi-weighted timed automata.

8.1 Relative distance logic

In [84, 85], Wilke introduced *relative distance logic* on timed words and showed that this logic is equally expressive as timed automata. In this section, we present basic notions about this logic.

Recall that V_1 is a countable set of first-order variables and V_2 is a countable set of second-order variables. Here, we also add a countable set $\mathcal{D}$ of (second-order) *relative distance variables* such that $\mathcal{D} \cap (V_1 \cup V_2) = \emptyset$. Let $\mathcal{W} = V_1 \cup V_2 \cup \mathcal{D}$.

Let Σ be an alphabet. Recall from the previous chapter that $\mathcal{I}$ means the collection of intervals. The set $\mathbf{RDL}(\Sigma)$ of *relative distance formulas* over Σ is defined by the grammar

$$\varphi \ ::= \ P_a(x) \mid x \leq y \mid \mathcal{X}(x) \mid \ \mathrm{d}^I(X, x) \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x.\varphi \mid \exists X.\varphi$$

$(w,\sigma) \models P_a(x)$	iff	$a_{\sigma(x)} = a$
$(w,\sigma) \models x \leq y$	iff	$\sigma(x) \leq \sigma(y)$
$(w,\sigma) \models \mathcal{X}(x)$	iff	$\sigma(x) \in \sigma(\mathcal{X})$
$(w,\sigma) \models \mathrm{d}^I(D,x)$	iff	$(\sigma(D),\sigma(x)) \in \mathrm{d}^I(w)$
$(w,\sigma) \models \varphi_1 \vee \varphi_2$	iff	$(w,\sigma) \models \varphi_1$ or $(w,\sigma) \models \varphi_2$
$(w,\sigma) \models \neg\varphi$	iff	$(w,\sigma) \models \varphi$ does not hold
$(w,\sigma) \models \exists x.\varphi$	iff	$\exists j \in \mathrm{dom}(w) : (w,\sigma[x/j]) \models \varphi$
$(w,\sigma) \models \exists X.\varphi$	iff	$\exists J \subseteq \mathrm{dom}(w) : (w,\sigma[X/J]) \models \varphi$

Table 8.1: The semantics of relative distance formulas

where $a \in \Sigma$, $x,y \in V_1$, $X \in V_2$, $\mathcal{X} \in V_2 \cup \mathcal{D}$, and $I \in \mathcal{I}$. The formulas of the form $\mathrm{d}^I(X,x)$ are called *relative distance predicates*.

Let $w = (a_1,t_1)...(a_n,t_n) \in \mathbb{T}\Sigma^+$ be a timed word. The *domain* of w is the set $\mathrm{dom}(w) = \{1,...,n\}$ of *positions* of w. Let $j \in \mathrm{dom}(w)$, $J \subseteq \mathrm{dom}(w)$, and $I \in \mathcal{I}$. Then, we write $(J,j) \in \mathrm{d}^I(w)$ if $\langle w \rangle_{i,j} \in I$ for the greatest value $i \in J \cup \{0\}$ with $i < j$.

To define a w-*assignment* σ, we also take into account relative distance variables, i.e., $\sigma : \mathcal{W} \to \mathrm{dom}(w) \cup 2^{\mathrm{dom}(w)}$ where $\sigma(V_1) \subseteq \mathrm{dom}(w)$ and $\sigma(V_2 \cup \mathcal{D}) \subseteq 2^{\mathrm{dom}(w)}$. The updates $\sigma[x/j]$ and $\sigma[\mathcal{X}/J]$ with $x \in V_1$, $\mathcal{X} \in V_2 \cup \mathcal{D}$, $j \in \mathrm{dom}(w)$ and $J \subseteq \mathrm{dom}(w)$ are defined as for usual finite words.

Let $\mathbb{T}\Sigma^+_{\mathcal{W}}$ denote the set of all pairs (w,σ) where $w \in \mathbb{T}\Sigma^+$ and σ is a w-assignment. The definition that a pair $(w,\sigma) \in \mathbb{T}\Sigma^+_{\mathcal{W}}$ satisfies the formula φ, written $(w,\sigma) \models \varphi$, is given inductively on the structure of φ as shown in Table 8.1. Here, $a \in \Sigma$, $x,y \in V_1$, $X \in V_2$, $D \in \mathcal{D}$, $\mathcal{X} \in V_2 \cup \mathcal{D}$, and $I \in \mathcal{I}$.

Let $\exists\mathbf{RDL}(\Sigma)$ denote the set of all formulas $\varphi = \exists D_1. \ ... \ \exists D_k.\psi$ where $k \geq 0$, $D_1,...,D_k \in \mathcal{D}$ and $\varphi \in \mathbf{RDL}(\Sigma)$. Note that $\mathbf{RDL}(\Sigma) \subseteq \exists\mathbf{RDL}(\Sigma)$. For $(w,\sigma) \in \mathbb{T}\Sigma^+_{\mathcal{W}}$, the satisfaction relation $(w,\sigma) \models \varphi$ is defined as usual.

For a formula $\varphi \in \exists\mathbf{RDL}(\Sigma)$, the set $\mathrm{Free}(\varphi) \subseteq \mathcal{W}$ of *free variables* of φ is defined as the set of all variables of φ not bound by a quantifier. We say that φ is a *sentence* if $\mathrm{Free}(\varphi) = \emptyset$. In this case, we can simply write $w \models \varphi$. For a sentence $\varphi \in \exists\mathbf{RDL}(\Sigma)$, let $\mathcal{L}(\varphi) = \{w \in \mathbb{T}\Sigma^+ \mid w \models \varphi\}$, the timed language *defined* by φ. We say that a timed language $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ is *definable* if there exists a sentence $\varphi \in \exists\mathbf{RDL}(\Sigma)$ such that $\mathcal{L}(\varphi) = \mathcal{L}$.

Theorem 8.1 (Wilke [84]). *Let Σ be an alphabet and $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ a timed language. Then, $\mathcal{L}$ is recognizable iff $\mathcal{L}$ is definable.*

8.2 Timed weight assignment logic

In this section, we introduce a quantitative logic which is equivalent to the multi-weighted timed automata we considered in the previous chapter. As the qualitative basis, we take Wilke's *relative distance logic*. As the quantitative basis, we take the *weight assignment logic* (cf. Chapter 6). As a weight structure, we use a *timed valuation structure* investigated in the previous chapter. As in the case of weight assignment logic, we will assign weights to the positions of a timed word. In order to reflect both discrete and continuous weights, we have to augment our logic with the possibility to assign these both sorts of weights.

Let Σ be an alphabet and $\mathbb{V} = (M, S, (K, +, \mathbb{0}), \mathrm{val}^{\mathbb{T}})$ a timed valuation monoid. Note that the set M corresponds to the set of location weights of a MWTA and S corresponds to the set of edge weights. Let $\mathbb{1} = (\mathbb{1}_M, \mathbb{1}_S) \in M \times S$, a pair of *default weights*. Let $\mathbb{V}_{\mathbb{1}}$ denote the pair $(\mathbb{V}, \mathbb{1})$.

The set $\mathbf{tWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ of *timed weight assignment logic* formulas over Σ and $\mathbb{V}_{\mathbb{1}}$ is given by the grammar

$$\begin{aligned} \varphi &::= \beta \mid x \mapsto m \mid x \Mapsto s \mid \beta\,?\,(\varphi : \varphi) \mid \varphi \sqcap \varphi \mid \textstyle\bigsqcap x.\varphi \mid \bigsqcap X.\varphi \\ \psi &::= \varphi \mid \sqcup x.\psi \mid \sqcup \mathcal{X}.\psi \end{aligned}$$

where $\beta \in \mathbf{RDL}(\Sigma)$, $m \in M$, $s \in S$, $x \in V_1$, $X \in V_2$ and $\mathcal{X} \in V_2 \cup \mathcal{D}$. We denote by $\mathbf{tWAL}^{\sqcap}(\Sigma, \mathbb{V}_{\mathbb{1}}) \subseteq \mathbf{tWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ the set of all formulas generated from the nonterminal φ.

For a formula $\varphi \in \mathbf{tWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$, let $\mathrm{CONST}_M(\varphi) \subseteq M$ be the set of all $m \in M$ such that $x \mapsto m$ appears in φ for some $x \in V_1$. Similarly, let $\mathrm{CONST}_S(\varphi) \subseteq S$ be the set of all constants $s \in S$ such that $x \Mapsto s$ appears in φ for some $x \in V_1$.

Now we are going to define the semantics of formulas in $\mathbf{tWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$. The approach here is similar to our approach to the semantics of weight assignment logic on ω-words. What is new here is that:

- the domain of a finite timed word is not fixed;
- we must assign to the positions of a timed word both continuous weights from M and discrete weighs from S, therefore we need two partially defined mappings.

For the definitions about partial mappings, we refer the reader to Sect. 6.1. For a set Y and $n > 0$, let $Y_n^{\uparrow}$ denote the set of all partial mappings $f : \{1, ..., n\} \dashrightarrow Y$. Now let $\mu, \mu' \in M_n^{\uparrow}$ and $\delta, \delta' \in S_n^{\uparrow}$. We say that pairs $r = (\mu, \delta)$ and $r' = (\mu', \delta')$ are *compatible*, written $r \uparrow r'$ if $\mu \uparrow \mu'$ and $\delta \uparrow \delta'$. If $r \uparrow r'$, then we define the *union* (or *merging*) $(r \cup r') \in M_n^{\uparrow} \times S_n^{\uparrow}$ by $r \cup r' = (\mu \cup \mu', \delta \cup \delta')$. Let $\Delta : \mathbb{T}\Sigma_V^+ \dashrightarrow \bigcup_{n \geq 1} M_n^{\uparrow} \times S_n^{\uparrow}$ be a partial mapping with $\Delta(w, \sigma) \in M_{|w|}^{\uparrow} \times S_{|w|}^{\uparrow}$ for all $(w, \sigma) \in \mathbb{T}\Sigma_{\mathcal{W}}^+$. We define $\mathrm{val}_{\mathbb{0},\mathbb{1}}^{\mathbb{T}}(\Delta) : \mathbb{T}\Sigma_{\mathcal{W}}^+ \to K$ for all $w = (a_1, t_1)...(a_n, t_n) \in \mathbb{T}\Sigma^+$ and all w-assignments σ as follows. If $\Delta(w, \sigma)$ is undefined, then we let $\mathrm{val}_{\mathbb{0},\mathbb{1}}^{\mathbb{T}}(\Delta) = \mathbb{0}$. Otherwise, $\Delta(w, \sigma) = (\mu, \delta)$ for some $\mu \in M_{|w|}^{\uparrow}$ and $\delta \in S_{|w|}^{\uparrow}$, and we let

$$\mathrm{val}_{\mathbb{0},\mathbb{1}}^{\mathbb{T}}(\Delta) = \mathrm{val}^{\mathbb{T}}(((m_1, s_1), t_1)...((m_n, s_n), t_n))$$

where $m_i = \begin{cases} \mu(i), & \text{if } i \in \mathrm{dom}(\mu), \\ \mathbb{1}_M, & \text{otherwise} \end{cases}$ and $s_i = \begin{cases} \delta(i), & \text{if } i \in \mathrm{dom}(\delta), \\ \mathbb{1}_S, & \text{otherwise} \end{cases}$ for all $i \in \{1, ..., n\}$.

$$\langle\!\langle \beta \rangle\!\rangle(w,\sigma) = \begin{cases} (\perp,\perp), & \text{if } (w,\sigma) \models \beta, \\ \text{undef}, & \text{otherwise} \end{cases}$$
$$\langle\!\langle x \mapsto m \rangle\!\rangle(w,\sigma) = (\perp[\sigma(x)/m],\perp)$$
$$\langle\!\langle x \Mapsto s \rangle\!\rangle(w,\sigma) = (\perp,\perp[\sigma(x)/s])$$
$$\langle\!\langle \beta\,?\,(\varphi_1 : \varphi_2) \rangle\!\rangle(w,\sigma) = \begin{cases} \langle\!\langle \varphi_1 \rangle\!\rangle(w,\sigma), & \text{if } (w,\sigma) \models \beta \\ \langle\!\langle \varphi_2 \rangle\!\rangle(w,\sigma), & \text{otherwise} \end{cases}$$
$$\langle\!\langle \varphi_1 \sqcap \varphi_2 \rangle\!\rangle(w,\sigma) = \langle\!\langle \varphi_1 \rangle\!\rangle(w,\sigma) \cup \langle\!\langle \varphi_2 \rangle\!\rangle(w,\sigma)$$
$$\langle\!\langle \textstyle\bigsqcap x.\varphi \rangle\!\rangle(w,\sigma) = \bigcup_{i \in \mathrm{dom}(w)} \langle\!\langle \varphi \rangle\!\rangle(w,\sigma[x/i])$$
$$\langle\!\langle \textstyle\bigsqcap X.\varphi \rangle\!\rangle(w,\sigma) = \bigcup_{I \subseteq \mathrm{dom}(w)} \langle\!\langle \varphi \rangle\!\rangle(w,\sigma[X/I])$$

Table 8.2: The semantics of $\mathbf{tWAL}^{\sqcap}(\Sigma, \mathbb{V}_{\mathbb{1}})$-formulas.

As in Sect. 6.2, we introduce the undefined value UNDEF and extend the union operations of partial mappings also to UNDEF: we assume that the union with UNDEF is UNDEF as well as $r \cup r' = \text{UNDEF}$ for all $r, r' \in M_n^{\uparrow} \times S_n^{\uparrow}$ $(n > 0)$ which are not compatible.

First, we start with the semantics for $\varphi \in \mathbf{tWAL}^{\sqcap}(\Sigma, \mathbb{V}_{\mathbb{1}})$. Let $w \in \mathbb{T}\Sigma^+$ and σ be a w-assignment. The *semantics* of φ is the mapping $\langle\!\langle \varphi \rangle\!\rangle : \mathbb{T}\Sigma^+_{\mathcal{W}} \to \bigcup_{n \geq 1}(M_n^{\uparrow} \times S_n^{\uparrow}) \cup \{\text{UNDEF}\}$ defined for all $(w,\sigma) \in \mathbb{T}\Sigma^+_{\mathcal{W}}$ inductively on the structure of φ as shown in Table 8.2. Here $\beta \in \mathbf{RDL}(\Sigma)$, $m \in M$, $s \in S$, $x \in V_1$ and $X \in V_2$.

Now we turn to the semantics of $\psi \in \mathbf{tWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ which is the mapping $[\![\psi]\!] : \mathbb{T}\Sigma^+_{\mathcal{W}} \to K$ defined for all $(w,\sigma) \in \mathbb{T}\Sigma^+_{\mathcal{W}}$ as shown in Table 8.3. Here, $\varphi \in \mathbf{tWAL}^{\sqcap}(\Sigma, \mathbb{V}_{\mathbb{1}})$, $x \in V_1$ and $\mathcal{X} \in V_2 \cup \mathcal{D}$ and we let $\mathrm{val}^{\mathbb{T}}_{\mathbb{0},\mathbb{1}}(\text{UNDEF}) = \mathbb{0}$.

We say that $\psi \in \mathbf{tWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ is a *sentence* if every variable of ψ is bound by a quantifier. If ψ is a sentence, then its semantics does not depend on variable assignments; so we can consider $[\![\psi]\!]$ as the mapping $[\![\psi]\!] : \mathbb{T}\Sigma^+ \to K$. We say that a QTL $\mathbb{L} : \mathbb{T}\Sigma^+ \to K$ is $\mathbf{tWAL}$-*definable* over $\mathbb{V}$ if there exist a pair of the default weights $\mathbb{1} \in M \times S$ and a sentence $\psi \in \mathbf{tWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ such that $[\![\psi]\!] = \mathbb{L}$.

Example 8.2. *A plane makes flights between two airports X and Y. Let τ_f be the flight time from X to Y and also from Y to X. During a flight the air company bears the following expenses:*

- *the fuel costs $f \in \mathbb{Q}_{\geq 0}$ per time unit;*
- *the crew salary $c \in \mathbb{Q}_{\geq 0}$ per time unit;*

In the airports, the plane can stay between $\tau_s^{\min}$ and $\tau_s^{\max}$ time units. Assume that $\tau_s^{\min}, \tau_s^{\max} \in \mathbb{N}$ While staying there are the following costs:

$$\llbracket\varphi\rrbracket(w,\sigma) = \mathrm{val}^{\mathbb{T}}_{\mathbb{0},\mathbb{1}}(\langle\!\langle\varphi\rangle\!\rangle(w,\sigma))$$
$$\llbracket\sqcup x.\psi\rrbracket(w,\sigma) = \sum_{i\subseteq \mathrm{dom}(w)} \llbracket\psi\rrbracket(w,\sigma[x/i])$$
$$\llbracket\sqcup \mathcal{X}.\psi\rrbracket(w,\sigma) = \sum_{I\subseteq \mathrm{dom}(w)} \llbracket\psi\rrbracket(w,\sigma[\mathcal{X}/I])$$

Table 8.3: The semantics of $\mathbf{tWAL}(\Sigma,\mathbb{V}_{\mathbb{1}})$-formulas

- *the crew salary $c \in \mathbb{Q}_{\geq 0}$ per time unit;*
- *the airport fee $a \in \mathbb{Q}_{\geq 0}$ per time unit;*
- *one-time costs $b \in \mathbb{Q}_{\geq 0}$ for beverages, snacks, cleaning service, etc.*

A flight schedule *is a timed word $w = (A,t_1)(F,t_2)...(A,t_{2n-1})(F,t_{2n})$ over the alphabet $\{A,F\}$ where, for all $1 \leq i \leq n$, $\tau_s^{\min} \leq t_{2i-1} \leq \tau_s^{\max}$ and $t_{2i} = \tau_f$. Here, the pair (A,t) means that the plane stays in an airport t time units and the pair (F,t) means that the plane makes a flight which takes t time units. The average cost of w is defined by*

$$C(w) = c + \frac{a\cdot t_1 + b + f\cdot t_2 + ... + a\cdot t_{2n-1} + b + f\cdot t_{2n}}{t_1 + t_2 + ... + t_{2n-1} + t_{2n}}$$

Given a timed word $w \in \mathbb{T}\{A,F\}^+$, our goal is to check whether w is a flight schedule and to compute the average cost of w. For this, we will use our weighted relative distance logic over the alphabet $\Sigma = \{A,F\}$, the timed valuation structure $\mathbb{V} = \mathbb{V}^{\mathrm{RATIO}}$ of Example 7.9 (b) together with the pair of the default weights $\mathbb{1} = ((0,0),(0,0))$. Let β_1 be the sentence which checks whether we have a sequence of the form $AF...AF$. Let D_A be a relative distance variable which will mean the set of all positions labeled by A. Analogously, let $D_F \in \mathcal{D}$ be a variable which corresponds to the set of all positions labeled by F. This can be expressed by the formula

$$\beta_2 = \forall x.[(P_A(x) \leftrightarrow D_A(x)) \wedge (P_F(x) \leftrightarrow D_F(x))].$$

Let $I = [\tau_s^{\min},\tau_s^{\max}] \in \mathcal{I}$. Next we check the correctness of a time sequence using the formula

$$\beta_3 = \forall x.[\{P_A(x) \rightarrow \mathrm{d}^I(D_F,x)\} \wedge \{P_F(x) \rightarrow \mathrm{d}^{[\tau_f,\tau_f]}(D_A,x)\}].$$

Finally, we construct the timed weight assignment sentence $\varphi \in \mathbf{tWAL}(\Sigma,\mathbb{V}_{\mathbb{1}})$ as

$$\varphi = \sqcup D_A.\sqcup D_F.(\beta \sqcap \textstyle\bigsqcap x.[X_A(x)\,?\,([x \mapsto (a+c,1) \sqcap x \Mapsto (b,0)] : x \mapsto (f+c,1))]).$$

Then, for all $w \in \mathbb{T}\{A,F\}^+$:

$$\llbracket\varphi\rrbracket(w) = \begin{cases} C(w), & \text{if } w \text{ is a flight schedule,}\\ \infty, & \text{otherwise.}\end{cases}$$

Our Büchi-Elgot theorem for multi-weighted timed automata is the following.

Theorem 8.3. *Let Σ be an alphabet, $\mathbb{V} = (M, S, (K, +, \mathbb{0}), \mathrm{val}^{\mathbb{T}})$ a timed valuation structure, $\mathbb{1} \in M \times S$ a pair of default weights, and $\mathbb{L} : \mathbb{T}\Sigma^{+} \to K$ a QTL. Then, $\mathbb{L}$ is recognizable over $\mathbb{V}$ iff $\mathbb{L}$ is* **tWAL**-*definable over $\mathbb{V}$.*

The proof of this theorem will we given in Sect. 8.4.

The proof of Theorem 8.3 that we present in Sect. 8.4 gives us effective translation procedures from logic to automata (and vice versa). Then, we obtain the following corollary.

Corollary 8.4. *Let $\mathbb{V}^{\text{tRatio}}$ be the timed valuation structure of Example 7.9 (b) and $\mathbb{1}$ an arbitary pair of default weights in $\mathbb{V}^{\text{tRatio}}$. Then, it is decidable, given an alphabet Σ, a sentence $\psi \in$* **tWAL**$(\Sigma, \mathbb{V}_{\mathbb{1}})$*, and a threshold $\theta \in \mathbb{Q}$, whether $[\![\varphi]\!](w) < \theta$ for some $w \in \mathbb{T}\Sigma^{+}$.*

8.3 Unambiguously definable timed languages

As in the case of weight assignment logic on ω-words, we will prove Theorem 8.3 using our Nivat Theorem 7.17 (a). Note that the qualitative basis of this theorem is given by a proper subclass of recognizable timed languages (sequentially, deterministically or unambiguously recognizable) whereas Wilke's Theorem 8.1 characterizes the full class of recognizable timed languages. A similar problem with the unambiguity occured in [77] for semiring-weighted relative distance logic over non-idempotent semirings; a solution was given by restricting the Boolean part of the logic to formulas of bounded variability. Here, we show that this restriction can be avoided. For this, we introduce a fragment of relative distance logic which is equivalent to unambiguous timed automata and applicable for the proof of Theorem 8.3.

Definition 8.5. *Let Σ be an alphabet. A timed language $\mathcal{L} \subseteq \mathbb{T}\Sigma^{+}$ is called* unambiguously definable *if there exists a formula $\varphi \in$* **RDL**(Σ) *such that $|\operatorname{Free}(\varphi)| = k \geq 0$, $\operatorname{Free}(\varphi) = \{D_1, ..., D_k\} \subseteq \mathcal{D}$ and, for every $w \in \mathbb{T}\Sigma^{+}$, there exists at most one tuple $(I_1, ..., I_k) \in (2^{\operatorname{dom}(w)})^k$ with $(w, \sigma[D_1/I_1, ..., D_k/I_k]) \models \varphi$, and $\mathcal{L} = \mathcal{L}(\exists D_1. \, ... \, \exists D_k.\varphi)$.*

The goal of the present section is to show the following theorem.

Theorem 8.6. *Let Σ be a finite alphabet and $\mathcal{L} \subseteq \mathbb{T}\Sigma^{+}$ a timed language. Then, $\mathcal{L}$ is unambiguously recognizable iff $\mathcal{L}$ is unambiguously definable.*

The proof of this theorem will be given in the rest of this section.

Lemma 8.7. *Let $\mathcal{A}$ be an unambiguous timed automaton over Σ. Then, the timed language $\mathcal{L}(\mathcal{A})$ is unambiguously definable.*

Proof. Let $\mathcal{A} = (L, C, I, E, F)$ and $(c_i)_{i\in\{1,...,|C|\}}$ be a enumeration of C. Let $\{D_c \mid c \in C\} \subseteq \mathcal{D}$ be a set of pairwise distinct relative distance variables. As it was shown in [84], $\mathcal{L}(\mathcal{A})$ can be defined by a $\exists\mathbf{RDL}(\Sigma)$-sentence of the form $\exists D_{c_1}. \ldots \exists D_{c_{|C|}}.\varphi$ where $\varphi \in \mathbf{RDL}(\Sigma)$ describes an accepting run in $\mathcal{A}$. For a timed word $w \in \mathbb{T}\Sigma^+$, a run $\rho \in \mathrm{Run}_{\mathcal{A}}(w)$ and a clock $c \in C$, D_c corresponds to the set of all positions of w where, along the run ρ, the clock c was reset. Since $\mathcal{A}$ is unambiguous, for every word there exists at most one accepting run. Then, if such a run exists, then the sets D_c $(c \in C)$ are uniquely determined. Thus, $\mathcal{L}(\mathcal{A})$ is unambiguously definable. $\square$

Now we turn to the converse direction. For this, we will use the idea of determinizable *event-recording* automata introduced in [4] and their logical characterization [30]. Let $\mathcal{V}$ be a finite set. A $\mathcal{V}$*-event-recording* automaton over Σ is a timed automaton $\mathcal{A} = (L, \mathcal{V}, I, E, F)$ over the alphabet $\Sigma \times 2^{\mathcal{V}}$ such that, for each $(\ell, (a, U), \phi, \Lambda, \ell') \in E$ with $a \in \Sigma$ and $U \subseteq \mathcal{V}$, we have: $U = \Lambda$. Our definition of $\mathcal{V}$-event-recording automata differs from event-recording automata of [4] in that we do not associate a clock with each letter in $\Sigma \times 2^{\mathcal{V}}$. However, our definition follows the same idea that the history of clock resets is uniquely determined by an input timed word.

Lemma 8.8. *Let $\mathcal{V}$ be a finite set.*

(a) Let $\mathcal{L}_1, \mathcal{L}_2 \in \mathbb{T}(\Sigma \times 2^{\mathcal{V}})^+$ be timed languages recognizable by $\mathcal{V}$-event-recording automata. Then, the intersection $\mathcal{L}_1 \cap \mathcal{L}_2$ is also recognizable by a $\mathcal{V}$-event-recording automaton.
(b) For each $\mathcal{V}$-event-recording automaton $\mathcal{A}$ over Σ there exists a deterministic $\mathcal{V}$-event-recording automaton $\mathcal{A}'$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.
(c) Let $\mathcal{L}$ be a timed language recognizable by a $\mathcal{V}$-event-recording automaton. Then, the complement $\mathbb{T}\Sigma^+ \setminus \mathcal{L}$ is also recognizable by a $\mathcal{V}$-event-recording automaton.

Proof. (a) We use the standard product construction for timed automata.
(b) As in [4], we adopt the standard powerset construction to the timed setting. Assume that $\mathcal{A} = (L, \mathcal{V}, I, E, F)$. We construct $\mathcal{A}' = (L', \mathcal{V}', I', E', F')$ as follows.

- $L' = 2^L$, $I' = \{I\}$, $F' = \{\tilde{L} \subseteq L \mid \tilde{L} \cap F \neq \emptyset\}$;
- E' is defined in the following way. We denote by $K \subseteq \mathbb{N} \setminus \{0\}$ the set of all positive natural numbers k such that k is the lower or upper bound of the interval $\phi(x)$ where $x \in \mathcal{V}$ and $\phi \in \Phi(\mathcal{V})$ is some clock constraint appearing in $\mathcal{A}$. We will use these points for a partition of the interval $[0, \infty)$. Assume that $K = \{k_1, ..., k_l\}$ where $l \geq 0$ and $k_1 < k_2 < ... < k_l$. Let $\mathbb{P}(K) \subseteq \mathcal{I}$ be defined as

$$\mathbb{P}(K) = \{[0, 0], (0, k_1), [k_1, k_1], (k_1, k_2), ..., [k_l, k_l], (k_l, \infty)\}$$

(if $l = 0$, then $\mathbb{P}(K) = \{[0, 0], (0, \infty)\}$). Let $Q, Q' \subseteq L$, $(a, U) \in \Sigma \times 2^{\mathcal{V}}$ and $\phi \in \Phi(\mathcal{V})$. Then, we let $(Q, (a, U), \phi, U, Q') \in E'$

iff $\phi : \mathcal{V} \to \mathbb{P}(K)$ and Q' is the set of all $\ell' \in L$ such that there exists an edge $(\ell, (a, U), \overline{\phi}, U, \ell') \in E$ where $\ell \in Q$ and $\phi(x) \subseteq \overline{\phi}(x)$ for all $x \in \mathcal{V}$.

Then, $\mathcal{A}'$ is deterministic and $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.

(c) This fact follows from (b). □

As it was shown in [4], event-recording automata are not closed under renaming. In contrast, our $\mathcal{V}$-event-recording automata enjoy this property if a renaming is applied only to the Σ-component. Let Γ, Δ be alphabets and $h : \Gamma \to \Delta$ a renaming. For a timed word $w = ((\gamma_1, U_1), t_1)...((\gamma_n, U_n), t_n) \in \mathbb{T}(\Gamma \times 2^{\mathcal{V}})^+$, let $h(w) \in \mathbb{T}(\Delta \times 2^{\mathcal{V}})^+$ denote the timed word $((h(\gamma_1), U_1), t_1)...((h(\gamma_n), U_n), t_n)$. Then, let $\mathcal{L} \subseteq \mathbb{T}(\Gamma \times 2^{\mathcal{V}})^+$, let $h(\mathcal{L}) = \{h(w) \mid w \in \mathcal{L}\}$. Similarly, for $\mathcal{L}' \subseteq \mathbb{T}(\Gamma \times 2^{\mathcal{V}})^+$, let $h^{-1}(\mathcal{L}') = \{w \in \mathbb{T}(\Sigma \times 2^{\mathcal{V}})^+ \mid h(w) \in \mathcal{L}'\}$.

Lemma 8.9. *Let Γ, Δ be alphabets, $h : \Gamma \to \Delta$ a renaming and $\mathcal{V}$ a finite set.*

(a) Let $\mathcal{A}$ be a $\mathcal{V}$-event-recording automaton over Γ. Then, there exists a $\mathcal{V}$-event-recording automaton $\mathcal{A}'$ over Δ with $\mathcal{L}(\mathcal{A}') = h(\mathcal{L}(\mathcal{A}))$.
(b) Let $\mathcal{B}$ be a $\mathcal{V}$-event-recording automaton over Δ. Then, there exists a $\mathcal{V}$-event-recording automaton $\mathcal{B}'$ over Γ with $\mathcal{L}(\mathcal{B}') = h^{-1}(\mathcal{L}(\mathcal{B}))$.

Proof. (a) Here, we apply the standard renaming construction where we replace each edge $(\ell, (\gamma, U), \phi, U, \ell')$ of $\mathcal{A}$ with $\gamma \in \Gamma$ by the edge $(\ell, (h(\gamma), U), \phi, U, \ell')$.
(b) Here, we replace each edge $(\ell, (\delta, U), \phi, U, \ell')$ of $\mathcal{B}$ with $\delta \in \Delta$ by the set of edges $(\ell, (\gamma, U), \phi, U, \ell')$ where $\gamma \in \Gamma$ and $h(\gamma) = \delta$.

□

Lemma 8.10. *Let $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ be an unambiguously definable timed language. Then, $\mathcal{L}$ is unambiguously recognizable.*

Proof. Let $\mathcal{L} = \mathcal{L}(\exists D_1. ... \exists D_k.\varphi)$ where $\{D_1, ..., D_k\} \subseteq \mathcal{D}$ are pairwise distinct variables and $\varphi \in \mathbf{RDL}(\Sigma)$ such that $\mathrm{Free}(\varphi) = \mathcal{D}$ and, for every $w \in \mathbb{T}\Sigma^+$, there exists at most one tuple $(I_1, ..., I_k) \in (2^{\mathrm{dom}(w)})^k$ with $(w, \sigma[D_1/I_1, ..., D_k/I_k]) \models \varphi$.

Let $\mathcal{V} = \{D_1, ..., D_k\}$ and, for every $w \in \mathbb{T}\Sigma^+$ and $\mathcal{J} : \mathcal{V} \to 2^{\mathrm{dom}(w)}$, let $\mathrm{code}(w, \mathcal{J}) \in \mathbb{T}(\Sigma \times 2^{\mathcal{V}})^+$ denote the Büchi encoding of the pair $(w, \mathcal{J})$ as the timed word.

Using the standard Büchi encoding technique, Lemmas 8.8, 8.9 and structural induction, we can show that there exists a deterministic $\mathcal{V}$-event-recording automaton $\mathcal{A}$ over Σ such that

$$\mathcal{L}(\mathcal{A}) = \{\mathrm{code}(w, \mathcal{J}) \mid w \in \mathbb{T}\Sigma^+, \mathcal{J} : \mathcal{V} \to 2^{\mathrm{dom}(w)} \text{ and } (w, \mathcal{J}) \models \varphi\}.$$

Assume that $\mathcal{A} = (L, \mathcal{V}, I, E, F)$. Then, we define the timed automaton $\mathcal{A}' = (L, \mathcal{V}, I, E', F)$ over Σ by letting

$$E' = \{(\ell, a, \phi, U, \ell') \mid (\ell, (a, U), \phi, U, \ell') \in E\}.$$

Note that $\mathcal{A}$ is deterministic and, for each $w \in \mathbb{T}\Sigma^+$, there exists at most one assignment $\mathcal{J} : \mathcal{V} \to 2^{\mathrm{dom}(w)}$ such that $\mathrm{code}(w, \mathcal{J}) \in \mathcal{L}(\mathcal{A})$. Then, for each $w \in \mathbb{T}\Sigma^+$, there exists at most one run $\rho \in \mathrm{Run}_{\mathcal{A}'}(w)$ and hence $\mathcal{A}'$ is unambiguous. Moreover, as it easy to see, $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\exists D_1. ... \exists D_k.\varphi)$. Thus, $\mathcal{L}$ is unambiguously recognizable. $\square$

Then, Theorem 8.6 follows immediately from Lemmas 8.7 and 8.10.

8.4 Definability equals recognizability

In this section, we give a proof of Theorem 8.3. The proof idea is similar to the idea used for weighted assignment logic on ω-words, i.e., we proceed via our Nivat-like result for MWTA. What is new here is that we have two sorts of weights as well as that we deal with unambiguously recognizable timed languages.

Recall that Σ is an alphabet, $\mathbb{V} = (M, S, (K, +, \mathbb{0}), \mathrm{val}^{\mathbb{T}})$ is a timed valuation structure, and $\mathbb{1} \in M \times S$ is a pair of default weights. First, we show that definability implies recognizability.

Theorem 8.11. *Let $\psi \in \mathbf{tWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ be a sentence. Then, the QTL $[\![\psi]\!]$ is recognizable over Σ and $\mathbb{V}$.*

The proof of this theorem will be given below.

Let $\psi = \sqcup x_1. ... \sqcup x_k.\sqcup X_1. ... \sqcup X_l.\varphi$ be a sentence with $k, l \geq 0$, $x_1, ..., x_k \in V_1$, $X_1, ..., X_l \in V_2 \cup \mathcal{D}$ and $\varphi \in \mathbf{tWAL}^{\sqcap}(\Sigma, \mathbb{V}_{\mathbb{1}})$. First of all, we fix a letter $\# \notin (M \cup S)$. We will use it to mark positions where partial mappings $\mu \in M_n^{\uparrow}$ and $\delta \in S_n^{\uparrow}$ are undefined. Let $\psi \in \mathbf{tWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$, $\Delta_\varphi^M = \textsc{Const}_M(\varphi) \cup \{\#\}$ and $\Delta_\varphi^S = \textsc{Const}_S(\varphi) \cup \{\#\}$.

Let $w = (a_1, t_1)...(a_n, t_n) \in \mathbb{T}\Sigma^+$ be a timed word. Consider partial mappings $\mu : \{1, ..., n\} \dashrightarrow \textsc{Const}_M(\varphi)$ and $\delta : \{1, ..., n\} \dashrightarrow \textsc{Const}_D(\varphi)$ (clearly, $\mu \in M_n^{\uparrow}$ and $\delta \in S_n^{\uparrow}$). We encode μ as the word $\mathrm{code}(\mu) = m_1...m_n \in (\Delta_\varphi^M)^+$ such that, for all $i \in \{1, ..., n\}$, $m_i = \mu(i)$ if $i \in \mathrm{dom}(\mu)$ and $m_i = \#$ otherwise. Similarly, we encode δ as the word $\mathrm{code}(\delta) = s_1...s_n \in (\Delta_\varphi^S)^+$ such that, for all $i \in \{1, ..., n\}$, $s_i = \delta(i)$ if $i \in \mathrm{dom}(\delta)$ and $s_i = \#$ otherwise. Then, we encode the triple (w, μ, δ) as the timed word $\mathrm{code}(w, \mu, \delta) = ((a_1, m_1, s_1), t_1)...((a_n, m_n, s_n), t_n) \in \mathbb{T}(\Sigma \times \Delta_\varphi^M \times \Delta_\varphi^S)^+$.

Lemma 8.12. *Let $\varphi \in \mathbf{tWAL}^{\sqcap}(\Sigma, \mathbb{V}_{\mathbb{1}})$. Then, there exists a formula $\zeta \in \mathbf{RDL}(\Sigma \times \Delta_\varphi^M \times \Delta_\varphi^S)$ such that $\mathrm{Free}(\zeta) = \mathrm{Free}(\varphi)$ and, for all $(w, \sigma) \in \mathbb{T}\Sigma_{\mathcal{W}}^+$ and all partial functions $\mu \in (\textsc{Const}_M(\varphi))_{|w|}^{\uparrow}$ and $\delta \in (\textsc{Const}_S(\varphi))_{|w|}^{\uparrow}$, the following holds:*

$$\langle\!\langle \varphi \rangle\!\rangle(w, \sigma) = (\mu, \delta) \;\; \textit{iff} \; (\mathrm{code}(w, \mu, \delta), \sigma) \models \zeta. \tag{8.1}$$

Proof. The proof of this lemma follows the same idea as the proof of Lemma 6.8. First, let $\beta \in \mathbf{RDL}(\Sigma)$. We denote by β^* the $\mathbf{RDL}(\Sigma \times \Delta_\varphi^M \times \Delta_\varphi^S)$-formula obtained from β by replacing each subformula $P_a(x)$ of β by the formula $\bigvee \big(P_{(a,m,d)}(x) \mid m \in \Delta_\varphi^M \text{ and } d \in \Delta_\varphi^S\big)$. Clearly, $\mathrm{Free}(\beta^*) = \mathrm{Free}(\beta)$.

Since μ and δ are partially defined mappings, we need to keep track of the positions where these mappings are undefined. For this, we introduce two fresh second-order variables $Y, Z \in V_2$: Y for μ and Z for δ. For each subformula γ of φ, we define a formula $r_{Y,Z}(\gamma) \in \mathbf{RDL}(\Sigma \times \Delta_\varphi^M \times \Delta_\varphi^S)$ with $\mathrm{Free}(r_{Y,Z}(\gamma)) = \mathrm{Free}(\gamma) \cup \{Y, Z\}$. We proceed by induction as follows.

- Let $\gamma = \beta \in \mathbf{RDL}(\Sigma)$. Then, we put $r_{Y,Z}(\gamma) = \beta^* \wedge (Y = \emptyset) \wedge (Z = \emptyset)$.
- Let $\gamma = x \mapsto m$ with $m \in \Delta_\varphi^M$ and $x \in V_1$. Then, we put

$$r_{Y,Z}(\gamma) = \bigvee \left(P_{(a,m,s)}(x) \mid a \in \Sigma \text{ and } s \in \Delta_\varphi^M\right) \wedge (Y = \{x\}) \wedge (Z = \emptyset).$$

- Let $\gamma = x \Mapsto s$ with $s \in \Delta_\varphi^S$ and $x \in V_1$. Then, we put

$$r_{Y,Z}(\gamma) = \bigvee \left(P_{(a,m,s)}(x) \mid a \in \Sigma \text{ and } m \in \Delta_\varphi^M\right) \wedge (Y = \emptyset) \wedge (Z = \{x\}).$$

- Let $\gamma = \beta\,?\,(\gamma_1 : \gamma_2)$ with $\beta \in \mathbf{RDL}(\Sigma)$. Then, we put

$$r_{Y,Z}(\gamma) = (\beta^* \wedge r_{Y,Z}(\gamma_1)) \vee ((\neg\beta^*) \wedge r_{Y,Z}(\gamma_2)).$$

- Let $\gamma = \gamma_1 \sqcap \gamma_2$. Then, we put

$$\begin{aligned} r_{Y,Z}(\gamma) = \exists Y_1.\exists Y_2.\exists Z_1.\exists Z_2.\big[&r_{Y_1,Z_1}(\gamma_1) \wedge r_{Y_2,Z_2}(\gamma_2) \wedge \\ &(Y = Y_1 \cup Y_2) \wedge (Z = Z_1 \cup Z_2)\big] \end{aligned}$$

  where $Y_1, Y_2, Z_1, Z_2 \in V_2$ are fresh and pairwise distinct variables.
- Let $\gamma = \bigsqcap \mathcal{X}.\gamma'$ where $\mathcal{X} \in V_1 \cup V_2$. Let

$$\xi(Y, Z) = \forall \mathcal{X}.\exists Y'.\exists Z'.(r_{Y',Z'}(\gamma') \wedge (Y' \subseteq Y) \wedge (Z' \subseteq Z))$$

  where $Y', Z' \in V_2$ are fresh and distinct variables. Then, we put

$$r_{Y,Z}(\gamma) = \xi(Y, Z) \wedge \forall U.\forall W.[\xi(U, W) \to ((Y \subseteq U) \wedge (Z \subseteq W))].$$

Let $w \in \mathbb{T}\Sigma^+$, $\mu \in (\textsc{Const}_M(\varphi))^{\uparrow}_{|w|}$ and $\eta \in (\textsc{Const}_S(\varphi))^{\uparrow}_{|w|}$. For $R \subseteq \{1, ..., |w|\}$, let $\mu|_R \in (\textsc{Const}_M(\varphi))^{\uparrow}_{|w|}$ be defined such that $\mathrm{dom}(\mu|_R) = R \cap \mathrm{dom}(\mu)$ and $\mathrm{dom}(\mu|_R)(i) = \mu(i)$ for all $i \in \mathrm{dom}(\mu|_R)$. The partial mapping $\delta|_R \in (\textsc{Const}_S(\varphi))^{\uparrow}_{|w|}$ is defined similarly. As in the proof of Lemma 6.8, it can be shown by induction on the structure of γ that, for any w-assignment σ, $(\mathrm{code}(w, \mu, \delta), \sigma) \models r_{Y,Z}(\gamma)$ iff the following hold:

- $\sigma(Y) \subseteq \mathrm{dom}(\mu)$ and $\sigma(Z) \subseteq \mathrm{dom}(\delta)$,
- $\langle\!\langle \gamma \rangle\!\rangle(w, \sigma) = (\mu|_{\sigma(Y)}, \delta|_{\sigma(Z)})$.

Then, the desired formula $\zeta \in \mathbf{RDL}(\Sigma \times \Delta_\varphi^M \times \Delta_\varphi^S)$ is defined as

$$\begin{aligned} \zeta = \exists Y.\exists Z.\Bigg[&r_{Y,Z}(\varphi) \wedge \forall x.\Bigg((\neg Y(x)) \to \bigvee_{a \in \Sigma,\ s \in \Delta_\varphi^S} P_{(a,\#,s)}(x)\Bigg) \\ &\wedge \forall x.\Bigg((\neg Z(x)) \to \bigvee_{a \in \Sigma,\ m \in \Delta_\varphi^M} P_{(a,m,\#)}(x)\Bigg)\Bigg]. \end{aligned}$$

Note that $\mathrm{Free}(\zeta) = \mathrm{Free}(\varphi)$ and (8.1) holds. $\square$

Lemma 8.13. *There exist an alphabet* Γ*, renamings* $h : \Gamma \to \Sigma$ *and* $g : \Gamma \to M \times S$*, and an unambiguously definable timed language* $\mathcal{L} \subseteq \mathbb{T}\Gamma^+$ *such that* $[\![\psi]\!] = h((\mathrm{val}^{\mathbb{T}} \circ g) \cap \mathcal{L})$.

Proof. The proof is similar to the proof of Lemma 6.16. The main differences are that we have two types of weights and that $\mathcal{L}$ must be unambiguously definable. Let $\mathbb{1} = (\mathbb{1}_M, \mathbb{1}_S)$.

Recall that $\psi = \sqcup x_1. \ldots \sqcup x_k. \sqcup X_1. \ldots \sqcup X_l.\varphi$ with $\varphi \in \mathbf{tWAL}^{\sqcap}(\Sigma, \mathbb{V}_{\mathbb{1}})$. We may assume that $x_1, \ldots, x_k, X_1, \ldots, X_l$ are pairwise distinct variables. Let $\mathcal{V} = \{x_1, \ldots, x_k, X_1, \ldots, X_l\}$. We define Γ, h and g as follows.

- Let $\Gamma = \Sigma \times \Delta_\varphi^M \times \Delta_\varphi^S \times \{0,1\}^{|\mathcal{V}|}$.
- Let $h : \Gamma \to \Sigma$ be defined for all $\gamma = (a, m, s, u)$ with $a \in \Sigma$, $m \in \Delta_\varphi^M$, $s \in \Delta_\varphi^S$ and $u \in \{0,1\}^{|\mathcal{V}|}$ as $h(\gamma) = a$.
- Let $g^M : \Delta_\varphi^M \to M$ be defined by $g^M(m) = m$ for all $m \in \mathrm{C{\scriptsize ONST}}_M(\varphi)$ and $g^M(\#) = \mathbb{1}_M$. Similarly, let $g^S : \Delta_\varphi^S \to S$ be defined by $g^S(s) = s$ for all $s \in \mathrm{C{\scriptsize ONST}}_S(\varphi)$ and $g^S(\#) = \mathbb{1}_S$. Then, $g : \Gamma \to M \times S$ is defined for all $\gamma = (a, m, s, u)$ with $a \in \Sigma$, $m \in \Delta_\varphi^M$, $s \in \Delta_\varphi^S$ and $u \in \{0,1\}^{|\mathcal{V}|}$ by letting $g(\gamma) = (g^M(m), g^S(s))$.

It remains to define $\mathcal{L} \subseteq \mathbb{T}\Gamma^+$. By Lemma 8.12, there exists a formula $\zeta \in \mathbf{RDL}(\Sigma \times \Delta_\varphi^M \times \Delta_\varphi^S)$ such that $\mathrm{Free}(\zeta) = \mathcal{V}$ and, for all $(w, \sigma) \in \mathbb{T}\Sigma_{\mathcal{W}}^+$, all partial mappings $\mu \in (\mathrm{C{\scriptsize ONST}}_M(\varphi))_{|w|}^{\uparrow}$ and $\delta \in (\mathrm{C{\scriptsize ONST}}_S(\varphi))_{|w|}^{\uparrow}$ and all w-assignments σ, we have: $\langle\!\langle \varphi \rangle\!\rangle(w, \sigma) = (\mu, \delta)$ iff $(\mathrm{code}(w, \mu, \delta), \sigma) \models \zeta$.

Let $\zeta^* \in \mathbf{MSO}(\zeta)$ be the formula obtained from ζ by replacing each predicate $P_{(a,m,s)}(x)$ occurring in ζ (here, $a \in \Sigma$, $m \in \mathrm{C{\scriptsize ONST}}_M(\varphi)$, $s \in \mathrm{C{\scriptsize ONST}}_S(\varphi)$ and $x \in V_1$) by the formula $\bigvee(P_{(a,m,s,u)}(x) \mid u \in \{0,1\}^{|\mathcal{V}|})$. For all $1 \leq i \leq |\mathcal{V}|$, $d \in \{0,1\}$ and $x \in V_1$, let

$$R_{i,d}^*(x) = \bigvee \big(P_{(a,m,s,u)}(x) \mid a \in \Sigma, m \in \Delta_\varphi^M, s \in \Delta_\varphi^S \text{ and } u = (u_1, \ldots, u_{|\mathcal{V}|}) \text{ with } u_i = d\big)$$

Using our new formulas $R_{i,d}^*(x)$, we define the formulas ϕ_1, ϕ_2 and ϕ for the Büchi encoding of $\mathcal{V}$-variables as in the proof of Lemma 6.16. The formula φ encodes the values of $\mathcal{V}$-variables. Let the sentence $\beta \in \exists\mathbf{RDL}(\Gamma)$ be defined as

$$\beta = \exists D_1. \ldots \exists D_r. \exists Y_1. \ldots \exists Y_{l-r}. \exists x_1. \ldots \exists x_k. (\phi \wedge \zeta^*)$$

with $r = |\mathcal{V} \cap \mathcal{D}|$, $\{D_1, \ldots, D_r\} = \mathcal{V} \cap \mathcal{D}$ and $\{Y_1, \ldots, Y_{l-r}\} = \mathcal{V} \cap V_2$. Then, we put $\mathcal{L} = \mathcal{L}(\beta)$. Since ϕ uniquely associates values of $\mathcal{V}$-variables with an input timed word, $\mathcal{L}$ is unambiguously definable. Moreover, it follows from Lemma 8.12 that $h((\mathrm{val}^{\mathbb{T}} \circ g) \cap \mathcal{L}) = [\![\varphi]\!]$. $\square$

Proof of Theorem 8.11: follows from Lemma 8.13, Theorem 8.6 and the inclusion $\mathcal{N}^{\mathrm{U{\scriptsize NAMB}}}(\Sigma, \mathbb{V}) \subseteq \mathrm{{\scriptsize T}R{\scriptsize EC}}(\Sigma, \mathbb{V})$ of our Nivat Theorem 7.17 (a). $\square$

Now we turn to the converse direction of Theorem 8.3

Theorem 8.14. *Let $\mathbb{L} : \mathbb{T}\Sigma^+ \to K$ be a QTL recognizable over Σ and $\mathbb{V}$. Then, there exists a sentence $\psi \in \mathbf{tWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ such that $[\![\psi]\!] = \mathbb{L}$.*

Using the inclusion $\textsc{tRec}(\Sigma, \mathbb{V}) \subseteq \mathcal{N}^{\textsc{Unamb}}(\Sigma, \mathbb{V})$ of Theorem 7.17 (a), it suffices to show the following.

Lemma 8.15. *Let Γ be an alphabet, $h : \Gamma$ and $g : \Gamma \to M \times S$ renamings, and $\mathcal{L} \subseteq \mathbb{T}\Gamma^+$ an unambiguously definable timed language. Then, there exists a sentence $\psi \in \mathbf{tWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ such that $[\![\psi]\!] = ((\mathrm{val}^{\mathbb{T}} \circ g) \cap \mathcal{L})$.*

Proof. Here we use a similar proof technique as in Lemma 6.18. For each $\gamma \in \Gamma$ with $g(\gamma) = (m, s)$, let $g_M(\gamma) = m$ and $g_S(\gamma) = s$. Let $\beta \in \mathbf{RDL}(\Gamma)$ be a formula such that:

(i) $\mathrm{Free}(\beta) = \{D_1, ..., D_k\} \subseteq \mathcal{D}$ where $k \geq 0$ and $D_1, ..., D_k$ are pairwise distinct variables;
(ii) for each $w \in \mathbb{T}\Sigma^+$ there exists at most one tuple $(I_1, ..., I_k) \in (2^{\mathrm{dom}(w)})^k$ such that $(w, \sigma[D_1/I_1, ..., D_k/I_k]) \models \beta$; here σ is a fixed w-assignment;
(iii) $\mathcal{L} = \mathcal{L}(\exists D_1. \ldots \exists D_k.\beta)$.

Let $\mathcal{V} = \{X_\gamma \mid \gamma \in \Gamma\} \subseteq V_2$ be a set of pairwise distinct variables which do not appear in β. For any formula $\zeta \in \mathbf{RDL}(\Gamma)$, we denote by $h(\zeta) \in \mathbf{RDL}(\Gamma)$ the formula obtained from ζ by replacing each predicate $P_\gamma(x)$ occurring in γ by the formula $P_{h(\gamma)}(x) \wedge X_\gamma(x)$.

For a timed word $u = (\gamma_1, t_1)...(\gamma_n, t_n) \in \mathbb{T}\Gamma^+$, let $\sigma_u : \mathcal{V} \to 2^{\mathrm{dom}(u)}$ be defined for all $\gamma \in \Gamma$ as $\sigma_u(X_\gamma) = \{i \in \mathrm{dom}(u) \mid \gamma_i = \gamma\}$. It can be shown by induction on the structure of β that, for all timed words $u \in \mathbb{T}\Gamma^+$ and all u-assignments σ with $\sigma|_{\mathcal{V}} = \sigma_u$, we have:

$$(u, \sigma) \models \beta \text{ iff } (h(u), \sigma) \models h(\beta). \tag{8.2}$$

Let the formula $\textsc{Partition}$ be defined as in Equation (6.3) and the formula $\textsc{Renaming}$ as in Equation (6.4). Let

$$\textsc{Boolean} = \textsc{Partition} \wedge \textsc{Renaming} \wedge h(\beta).$$

Note that $\textsc{Boolean} \in \mathbf{RDL}(\Sigma)$. Let $(\gamma_i)_{1 \leq i \leq |\Gamma|}$ be an enumeration of Γ and $x \in V_1$ a fresh variable. For $1 \leq i \leq |\Gamma|$, we define the formula $\chi_i \in \mathbf{tWAL}^{\sqcap}(\Sigma, \mathbb{V}_{\mathbb{1}})$ inductively as follows.

- For $i = |\Gamma|$, we let $\chi_i = (x \mapsto g_M(\gamma_i)) \sqcap (x \Mapsto g_S(\gamma_i))$.
- Let $1 \leq i < |\Gamma|$ and assume that χ_{i+1} is defined. Then, we let $\chi_i = X_{\gamma_i}(x) \,?\, (((x \mapsto g_M(\gamma_i)) \sqcap (x \Mapsto g_S(\gamma_i))) : \chi_{i+1})$.

Let $\chi = \bigsqcap x.\chi_i$. Note that $\chi \in \mathbf{tWAL}^{\sqcap}(\Sigma, \mathbb{V}_{\mathbb{1}})$. Then, we define the desired sentence $\psi \in \mathbf{tWAL}(\Sigma, \mathbb{V}_{\mathbb{1}})$ as

$$\psi = \sqcup D_1. \ldots \sqcup D_k.\sqcup X_1. \ldots \sqcup X_{|\Gamma|}.(\textsc{Boolean} \sqcap \chi).$$

Taking into account (ii) and Equation 8.2, it can be shown as in the proof of Lemma 6.18 that $[\![\psi]\!] = ((\mathrm{val}^{\mathbb{T}} \circ g) \cap \mathcal{L})$. $\square$

Now Theorem 7.17 (a) and Lemma 8.15 prove Theorem 8.14.

Proof of Theorem 8.3. Immediate by Theorems 8.11 and 8.14.

Remark 8.16. *Alternatively, we could prove Theorem 8.14 by a direct translation of a MWTA into a timed weight assignment sentence. However, our proof has the advantage that we do not have to provide a bulky description of an accepting run of a timed automaton.*

Chapter 9

Timed pushdown automata and timed matching logic

Contents

In this chapter, we provide a logical characterization for *timed pushdown automata* investigated in [2] (cf. also [19, 31, 51] for related models). For our purpose, we introduce a *timed matching logic*. As in the logic of Lautemann, Schwentick and Thérien [70], we handle the stack functionality by means of a binary *matching* predicate. As in the logic of Wilke [84], we use *relative distance predicates* 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 are able compare the time distance between matched positions with a constant. The main result of this chapter will be the expressive equivalence of timed pushdown automata and timed matching logic.

9.1 Timed pushdown automata

In this section, we consider *timed pushdown automata* which have been introduced and investigated in [1]. These machines are nondeterministic automata equipped with finitely many *global* clocks (like timed automata) and a stack (like pushdown automata). In contrast to untimed pushdown automata, in the model of TPDA we push together with a letter a *local* clock whose initial age can be an arbitrary real number from some interval. Like in timed automata, the values of global clocks and the ages of local clocks grow in time. Then, we can pop this letter only if its age belongs to a given interval. Note that, when considering all possible runs of a TPDA, the number of used local clocks is in general not bounded by any constant. We slightly extend the definition of TPDA presented in [1] by allowing labels of edges. This, however, does not harm the decidability of the reachability problem which was shown in [1]. Note that the model of TPDA of [1] extends the model of timed automata with untimed stack proposed in [19].

For basic notions about timed languages, clock constraints and clock valuations, we refer the reader to Section 7.1. Let Γ be a *stack alphabet*. We denote by $\mathcal{S}(\Gamma) = (\{\downarrow\} \times \Gamma \times \mathcal{I}) \cup \{\#\} \cup (\{\uparrow\} \times \Gamma \times \mathcal{I})$ the set of *stack commands* over Γ. For $u = (\gamma_1, t_1)...(\gamma_k, t_k) \in \mathbb{T}\Gamma^*$ and $t \in \mathbb{R}_{\geq 0}$, let $u+t = (\gamma_1, t_1+t)...(\gamma_k, t_k+t) \in \mathbb{T}\Gamma^*$.

Definition 9.1. *Let Σ be an alphabet. A* timed pushdown automaton (TPDA) *over Σ is a tuple $\mathcal{A} = (L, \Gamma, C, L_0, E, L_f)$ where L is a finite set of* locations*, Γ is a finite* stack alphabet*, C is a finite set of* clocks*, $L_0, L_f \subseteq L$ are sets of* initial *resp.* final *locations, and $E \subseteq L \times \Sigma \times \mathcal{S}(\Gamma) \times \mathcal{I}^C \times 2^C \times L$ is a finite set of* edges*.*

Let $e = (\ell, a, s, \phi, \Lambda, \ell') \in E$ be an edge of $\mathcal{A}$ with $\ell, \ell' \in L$, $a \in \Sigma$, $s \in \mathcal{S}(\Gamma)$, $\phi \in \mathcal{I}^C$ and $\Lambda \subseteq C$. We will denote e by $\ell \xrightarrow[s]{a,\phi,\Lambda} \ell'$. We say that a is the *label* of e and denote it by label(e). We also let stack$(e) = s$, the stack command of e. Let $E^{\downarrow} \subseteq E$ denote the set of all *push* edges e with stack$(e) = (\downarrow, \gamma, I)$ for some $\gamma \in \Gamma$ and $I \in \mathcal{I}$. Similarly, let $E^{\#} = \{e \in E \mid \text{stack}(e) = \#\}$ be the set of *local* edges and $E^{\uparrow} = \{e \in E \mid \text{stack}(e) = (\uparrow, \gamma, I) \text{ for some } \gamma \in \Gamma \text{ and } I \in \mathcal{I}\}$ the set of *pop* edges. Then, we have $E = E^{\downarrow} \cup E^{\#} \cup E^{\uparrow}$.

A *configuration* c of $\mathcal{A}$ is described by the present location, the values of the clocks, and the stack, which is a timed word over Γ. That is, c is a triple $\langle \ell, \nu, u \rangle$ where $\ell \in L$, $\nu \in \mathbb{R}_{\geq 0}^C$ and $u \in \mathbb{T}\Gamma^*$. We say that c is *initial* if $\ell \in L_0$, $\nu(x) = 0$ for all $x \in C$ 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 $\mathcal{A}$, $\mathcal{C}_{\mathcal{A}}^0$ the set of all initial configurations of $\mathcal{A}$ and $\mathcal{C}_{\mathcal{A}}^f \subseteq \mathcal{C}_{\mathcal{A}}$ the set of all final configurations.

Let $c = \langle \ell, \nu, u \rangle$ and $c' = \langle \ell', \nu', u' \rangle$ be two configurations with $u = (\gamma_1, t_1)(\gamma_2, t_2)...(\gamma_k, t_k)$ and let $e = (q, a, s, \phi, \Lambda, q') \in E$ be an edge. We say that $c \vdash_e c'$ is a *switch transition* if $\ell = q$, $\ell' = q'$, $\nu \models \phi$, $\nu' = \nu[\Lambda := 0]$, and:

- if $s = (\downarrow, \gamma, I)$ for some $\gamma \in \Gamma$ and $I \in \mathcal{I}$, then $u' = (\gamma, \tau)u$ for some $\tau \in I$;
- if $s = \#$, then $u' = u$;
- if $s = (\uparrow, \gamma, I)$ with $\gamma \in \Gamma$ and $I \in \mathcal{I}$, then $k \geq 1$, $\gamma = \gamma_1$, $t_1 \in I$ and $u' = (\gamma_2, t_2)...(\gamma_k, t_k)$.

For $t \in \mathbb{R}_{\geq 0}$, we say that $c \vdash_t c'$ is a *delay transition* if $\ell = \ell'$, $\nu' = \nu + t$ 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 $c'' \vdash_e c'$.

A *run* ρ 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 $n \geq 1$, $c_0 \in \mathcal{C}_{\mathcal{A}}^0$,

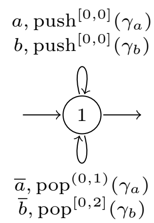

Figure 9.1: TPDA $\mathcal{A}$ of Example 9.2

$c_1, ..., c_{n-1} \in \mathcal{C}_{\mathcal{A}}$, $c_n \in \mathcal{C}^f_{\mathcal{A}}$, $t_1, ..., t_n \in \mathbb{R}_{\geq 0}$ and $e_1, ..., e_n \in E$. The *label* of ρ is the timed word $\text{label}(\rho) = (\text{label}(e_1), t_1)...(\text{label}(e_n), t_n) \in \mathbb{T}\Sigma^+$. Let $\mathcal{L}(\mathcal{A}) = \{w \in \mathbb{T}\Sigma^+ \mid \text{there exists a run } \rho \text{ of } \mathcal{A} \text{ with } \text{label}(\rho) = w\}$, the timed language *recognized* by $\mathcal{A}$. We say that a timed language $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ is a *timed pushdown language* if there exists a TPDA $\mathcal{A}$ over Σ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.

Note that every timed automaton $\mathcal{A} = (L, C, I, E, F)$ can be considered as a TPDA $\mathcal{A} = (L, \Gamma, C, I, E, F)$ where Γ is an arbitrary alphabet and $E = E^{\#}$.

Example 9.2. *Here, we consider a timed extension of the well-known* Dyck languages*. Let $\Sigma = \{a_1, ..., a_m\}$ be a set of opening brackets and $\overline{\Sigma} = \{\overline{a}_1, ..., \overline{a}_m\}$ a set of corresponding closing brackets. Let $I_{a_1}, ..., I_{a_m} \in \mathcal{I}$ be intervals. We will consider the* timed Dyck language $\mathcal{D}_\Sigma(I_{a_1}, ..., I_{a_m}) \subseteq \mathbb{T}(\Sigma \cup \overline{\Sigma})^+$ *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 every $i \in \{1, ..., m\}$, the time distance between any two matching brackets a_i and $\overline{a}_i$ is in I_{a_i}. It is not difficult to see that the timed language $\mathcal{D}_\Sigma(I_{a_1}, ..., I_{a_m})$ is a timed pushdown language. We illustrate this on the following example. Let $\Sigma = \{a, b\}$, $\overline{\Sigma} = \{\overline{a}, \overline{b}\}$, $I_a = (0, 1)$ and $I_b = [0, 2]$. Consider the TPDA $\mathcal{A} = (L, \Gamma, \emptyset, L_0, E, L_f)$ with:*

- $L = L_0 = L_f = \{1\}$, $\Gamma = \{\gamma_a, \gamma_b\}$*;*
- $E = \{e_\alpha \mid \alpha \in \Sigma \cup \overline{\Sigma}\}$ *such that, for* $\alpha \in \Sigma$, $e_\alpha = \big(1 \xrightarrow[(\downarrow, \gamma_\alpha, [0,0])]{\alpha, \emptyset, \emptyset} 1\big)$ *and* $e_{\overline{\alpha}} = \big(1 \xrightarrow[(\uparrow, \gamma_\alpha, I_\alpha)]{\overline{\alpha}, \emptyset, \emptyset} 1\big)$.

The TPDA $\mathcal{A}$ is depicted in Fig. 9.1. Note that $\mathcal{A}$ does not contain any global clocks. Then, $\mathcal{L}(\mathcal{A}) = \mathcal{D}_\Sigma(I_a, I_b)$. Consider, for instance, the timed word $w = (b, 0)(a, 0.2)(\overline{a}, 0.9)(\overline{b}, 0.9) \in \mathbb{T}(\Sigma \cup \overline{\Sigma})^+$. Then,

$$\begin{aligned}\langle 1, \varepsilon \rangle &\vdash_0 \langle 1, \varepsilon \rangle \vdash_{e_b} \langle 1, (\gamma_b, 0) \rangle \vdash_{0.2} \langle 1, (\gamma_b, 0.2) \rangle \vdash_{e_a} \langle 1, (\gamma_a, 0)(\gamma_b, 0.2) \rangle \\ &\vdash_{0.9} \langle 1, (\gamma_a, 0.9)(\gamma_b, 1.1) \rangle \vdash_{e_{\overline{a}}} \langle 1, (\gamma_b, 1.1) \rangle \vdash_{0.9} \langle 1, (\gamma_b, 2) \rangle \vdash_{e_{\overline{b}}} \langle 1, \varepsilon \rangle\end{aligned}$$

is an accepting run of $\mathcal{A}$ with the label w. Note that here we omit the empty clock valuation of configurations.

9.2 Timed matching logic

The goal of this section is to develop a logical formalism which is expressively equivalent to TPDA defined in Sect. 9.1. Our new logic will incorporate Wilke's *relative distance logic* [84] for timed automata as well as the logic with *matchings* [70] introduced by Lautemann, Schwentick and Thérien for context-free languages. Moreover, we augment our logic with the possibility to measure the time distance between matched positions.

Recall that $V_1, V_2, \mathcal{D}$ denote the countable and pairwise disjoint sets of first-order, second-order and relative distance variables, respectively. We also fix a *matching variable* $\mu \notin V_1 \cup V_2 \cup \mathcal{D}$. Let $\mathcal{U} = V_1 \cup V_2 \cup \mathcal{D} \cup \{\mu\}$.

Let Σ be an alphabet. The set $\mathbf{tMSO}(\Sigma)$ of *timed matching MSO formulas* is defined by the grammar

$$\varphi \ ::= \ P_a(x) \mid x \le y \mid \mathcal{X}(x) \mid \mathrm{d}^I(D,x) \mid \mu^I(x,y) \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x.\varphi \mid \exists X.\varphi$$

where $a \in \Sigma$, $x, y \in V_1$, $X \in V_2$, $D \in \mathcal{D}$, $\mathcal{X} \in V_2 \cup \mathcal{D}$ and $I \in \mathcal{I}$. The formulas of the form $\mathrm{d}^I(D,x)$ are called *relative distance predicates* and the formulas of the form $\mu^I(x,y)$ are called *distance matchings*. For $\mu^{[0,\infty)}(x,y)$, we will write simply $\mu(x,y)$.

The $\mathbf{tMSO}(\Sigma)$-formulas are interpreted over timed words over Σ and assignments of variables. Let $w \in \mathbb{T}\Sigma^+$ be a timed word. Recall that $\mathrm{dom}(w) = \{1,...,|w|\}$ is the domain of w. A $(w,\mathcal{U})$-*assignment* is a mapping $\sigma : \mathcal{U} \to \mathrm{dom}(w) \cup 2^{\mathrm{dom}(w)} \cup 2^{(\mathrm{dom}(w))^2}$ such that $\sigma(V_1) \subseteq \mathrm{dom}(w)$, $\sigma(V_2 \cup \mathcal{D}) \subseteq 2^{\mathrm{dom}(w)}$ and $\sigma(\mu) \subseteq 2^{(\mathrm{dom}(w))^2}$. Let σ be a $(w,\mathcal{U})$-assignment. For $x \in V_1$ and $j \in \mathrm{dom}(w)$, the *update* $\sigma[x/j]$ is the $(w,\mathcal{U})$-assignment defined by $\sigma[x/j](x) = j$ and $\sigma[x/j](y) = \sigma(y)$ for all $y \in \mathcal{U} \setminus \{x\}$. Similarly, for $\mathcal{X} \in V_2 \cup \mathcal{D}$ and $J \subseteq \mathrm{dom}(w)$, we define the update $\sigma[\mathcal{X}/J]$ and, for $M \subseteq (\mathrm{dom}(w))^2$, the update $\sigma[\mu/M]$.

Let $w \in \mathbb{T}\Sigma^+$ be a timed word and $I \in \mathcal{I}$ an interval. Recall that, for $j \in \mathrm{dom}(w)$ and $J \subseteq \mathrm{dom}(w)$, we write $(J,j) \in \mathrm{d}^I(w)$ if $\langle w \rangle_{i,j} \in I$ for the greatest value $i \in J \cup \{0\}$ with $i < j$. For $i, j \in \mathrm{dom}(w)$, $M \subseteq (\mathrm{dom}(w))^2$ and $I \in \mathcal{I}$, we will write $(i,j,M) \in \mu^I(w)$ if $i < j$, $(i,j) \in M$ and $\langle w \rangle_{i,j} \in I$.

Given a formula $\varphi \in \mathbf{tMSO}(\Sigma)$, a timed word $w = (a_1,t_1)...(a_n,t_n) \in \mathbb{T}\Sigma^+$ and a $(w,\mathcal{U})$-assignment σ; the satisfaction relation $(w,\sigma) \models \varphi$ is defined inductively on the structure of φ as shown in Table 9.1. Here, $a \in \Sigma$, $x, y \in V_1$, $X \in V_2$, $D \in \mathcal{D}$, $\mathcal{X} \in V_2 \cup \mathcal{D}$ and $I \in \mathcal{I}$.

For $\varphi \in \mathbf{tMSO}(\Sigma)$ and $y \in V_1$, let $\exists^{\le 1} y.\varphi$ denote the formula $\neg\exists y.\varphi \vee \exists y.(\varphi \wedge \forall z.(z \ne y \to \neg\varphi[y/z]))$ where $z \in V_1$ does not occur in φ and $\varphi[y/z]$ is the formula obtained from φ by replacing y by z. Let $\textsc{Matching}(\mu) \in \mathbf{tMSO}(\Sigma)$ denote the formula

$$\begin{aligned}\textsc{Matching}(\mu) =& \forall x.\forall y.(\mu(x,y) \to x < y) \wedge \forall x.\exists^{\le 1} y.(\mu(x,y) \vee \mu(y,x)) \wedge \\ & \forall x.\forall y.\forall u.\forall v.((\mu(x,y) \wedge \mu(u,v) \wedge x < u < y) \to x < v < y).\end{aligned}$$

This formula demands that a binary relation μ on a timed word domain is a *matching* (cf. [70]), i.e., it is compatible with $<$, each element of the domain belongs to at most one pair in μ and μ is noncrossing.

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

Table 9.1: The semantics of $\mathbf{tMSO}(\Sigma)$-formulas

The set $\mathbf{TML}(\Sigma)$ of the formulas of *timed matching logic* over Σ is defined to be the set of all formulas of the form

$$\psi = \exists\mu.\exists D_1.\ \dots\ \exists D_m.(\varphi \wedge \textsc{Matching}(\mu))$$

where $m \geq 0$, $D_1,\dots,D_m \in \mathcal{D}$ and $\varphi \in \mathbf{tMSO}(\Sigma)$. Let $w \in \mathbb{T}\Sigma^+$ and σ be a $(w,\mathcal{U})$-assignment. Then, $(w,\sigma) \models \psi$ iff there exist $J_1,\dots,J_m \subseteq \mathrm{dom}(w)$ and a matching $M \subseteq (\mathrm{dom}(w))^2$ such that $(w,\sigma[D_1/J_1,\dots,D_m/J_m,\mu/M]) \models \varphi$. For simplicity, we will denote ψ by $\exists^{\mathrm{match}}\mu.\exists D_1.\ \dots\ \exists D_m.\varphi$. Note that $\exists\mathbf{RDL}(\Sigma)$-formulas are exactly the $\mathbf{TML}(\Sigma)$-formulas not containing μ.

For a formula $\psi \in \mathbf{TML}(\Sigma)$, the set $\mathrm{Free}(\psi) \subseteq \mathcal{U}$ of *free variables* of ψ is defined as usual. We say that $\psi \in \mathbf{TML}(\Sigma)$ is a *sentence* if $\mathrm{Free}(\psi) = \emptyset$. Note that, for a sentence ψ, the satisfaction relation $(w,\sigma) \models \psi$ does not depend on a $(w,\mathcal{U})$-assignment σ. Then, we will simply write $w \models \psi$. Let $\mathcal{L}(\psi) = \{w \in \mathbb{T}\Sigma^+ \mid w \models \psi\}$, the language *defined* by ψ. We say that a timed language $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ is $\mathbf{TML}$-*definable* if there exists a sentence $\psi \in \mathbf{TML}(\Sigma)$ such that $\mathcal{L}(\psi) = \mathcal{L}$.

Example 9.3. *Consider the* timed Dyck language $\mathcal{D}_\Sigma(I_{a_1},\dots,I_{a_m}) \subseteq \mathbb{T}(\Sigma\cup\overline{\Sigma})^+$ *defined in Example 9.2. The timed language* $\mathcal{D}_\Sigma(I_{a_1},\dots,I_{a_m})$ *can be defined by the* $\mathbf{TML}(\Sigma)$*-sentence*

$$\exists^{\mathrm{match}}\mu.\Big(\forall x.\exists y.(\mu(x,y) \vee \mu(y,x)) \wedge$$
$$\forall x.\forall y.\Big(\mu(x,y) \rightarrow \bigvee\nolimits_{j=1}^{m}(P_{a_j}(x) \wedge P_{\overline{a}_j}(y) \wedge \mu^{I_{a_j}}(x,y))\Big)\Big).$$

Our main result is the following theorem.

Theorem 9.4. *Let Σ be an alphabet and $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ a timed language. Then $\mathcal{L}$ is a timed pushdown language iff $\mathcal{L}$ is* $\mathbf{TML}$*-definable.*

Note that Theorem 9.4 extends the result of [70] for context-free languages as well as the result of [84] for regular timed languages. As already mentioned in the introduction, we will use the logical characterization result for visibly pushdown languages [5]. In Sect. 9.3, for the convenience of the reader, we recall this result. In Sect. 9.4, we show a Nivat-like decomposition theorem for timed pushdown languages. Finally, in Sect. 9.5, we give a proof of Theorem 9.4.

It was shown in [1] that the emptiness problem for TPDA is decidable. Moreover, as we will see later, our proof of Theorem 9.4 is constructive. Then, we obtain the decidability of the satisfiability problem for our timed matching logic.

Corollary 9.5. *It is decidable, given an alphabet* Σ *and a sentence* $\psi \in \mathbf{TML}(\Sigma)$*, whether there exists a timed word* $w \in \mathbb{T}\Sigma^+$ *such that* $w \models \psi$*.*

9.3 Visibly pushdown languages

For the rest of the chapter, we fix a special stack symbol $\perp$.

A *pushdown alphabet* is a triple $\tilde{\Sigma} = \langle \Sigma^{\downarrow}, \Sigma^{\#}, \Sigma^{\uparrow} \rangle$ with pairwise disjoint sets $\Sigma^{\downarrow}$, $\Sigma^{\#}$ and $\Sigma^{\uparrow}$ of *push*, *local* and *pop* letters, respectively. Let $\Sigma = \Sigma^{\downarrow} \cup \Sigma^{\#} \cup \Sigma^{\uparrow}$. 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, Γ is a stack alphabet with $\perp \notin \Gamma$, and $T = T^{\downarrow} \cup T^{\#} \cup T^{\downarrow}$ is a set of transitions where $T^{\downarrow} \subseteq Q \times \Sigma^{\downarrow} \times \Gamma \times Q$ is a set of push transitions, $T^{\#} \subseteq Q \times \Sigma^{\#} \times Q$ is a set of local transitions and $T^{\uparrow} \subseteq Q \times \Sigma^{\uparrow} \times (\Gamma \cup \{\perp\}) \times Q$ is a set of pop transitions.

We define the label of a transition $\tau \in T$ depending on its sort as follows. If $\tau = (p, c, \gamma, p') \in T^{\downarrow} \cup T^{\uparrow}$ or $\tau = (p, c, p') \in T^{\#}$, we let $\text{label}(\tau) = c$, so $c \in \Sigma^{\downarrow} \cup \Sigma^{\uparrow}$ 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 $\tau = (p, a, \gamma, p') \in T^{\downarrow}$, 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 $\tau = (p, a, \gamma, p') \in T^{\uparrow}$ with $\gamma \in \Gamma \cup \{\perp\}$, then we put $c \vdash_\tau c'$ iff $p = q$, $p' = q'$ and either $\gamma \neq \perp$ and $u = \gamma u'$, or $\gamma = \perp$ 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 $\rho = c_0 \vdash_{\tau_1} c_1 \vdash_{\tau_2} \dots \vdash_{\tau_n} c_n$ where $c_0, c_1, \dots, c_n$ are configurations of $\mathcal{A}$ such that c_0 is initial, c_n is final and $\tau_1, \dots, \tau_n \in T$. Let $\text{label}(\rho) = \text{label}(\tau_1) \dots \text{label}(\tau_n) \in \Sigma^+$, the *label* of ρ. Let $\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^+ \mid \text{there exists a run } \rho \text{ of } \mathcal{A} \text{ with } \text{label}(\rho) = w\}$. 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 Σ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.

Remark 9.6. *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* $\perp$*). This permits to consider the situations where some pop letters are not balanced by push letters and vice versa.*

We note that the visibly pushdown languages over $\tilde{\Sigma}$ form a proper subclass of the context-free languages over Σ, cf. [5] 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 $a_i \in \Sigma^{\downarrow}$, $b_i = 0$ if $a_i \in \Sigma^{\#}$, and $b_i = -1$ otherwise. Let $\mathbb{L} \subseteq \{-1, 0, 1\}^*$ be the language which contains ε 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, $\mathbb{L}$ 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 [5] to be expressively 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 $\sigma : V_1 \cup V_2 \to \mathrm{dom}(w) \cup 2^{\mathrm{dom}(w)}$. We will write $(w, \sigma) \models \mathbb{L}(x, y)$ iff $\sigma(x) < \sigma(y)$, $a_{\sigma(x)} \in \Sigma^{\downarrow}$, $a_{\sigma(y)} \in \Sigma^{\uparrow}$ and $\textsc{Mask}(a_{\sigma(x)+1}...a_{\sigma(y)-1}) \in \mathbb{L}$. For other formulas, the satisfaction relation is defined as usual. If φ 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 9.7 (Alur, Madhusudan [5])**.** *Let* $\tilde{\Sigma} = (\Sigma^{\downarrow}, \Sigma^{\#}, \Sigma^{\uparrow})$ *be a pushdown alphabet,* $\Sigma = \Sigma^{\downarrow} \cup \Sigma^{\#} \cup \Sigma^{\uparrow}$*, 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.*

9.4 Decomposition of timed pushdown automata

In this section we prove a Nivat-like (cf. [74, 12]) decomposition theorem for timed pushdown automata. This result establishes a connection between timed pushdown languages and visibly pushdown languages. We will use this theorem for the proof of our Theorem 9.4.

The key idea is to consider a timed pushdown language as a renaming of a timed pushdown language over an extended alphabet which encodes the information about clocks and stack; on the level of this extended alphabet we can separate the setting of visibly pushdown languages from the 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.

In the rest of this chapter, we fix an alphabet Σ (which we will understand as the alphabet of Theorem 9.4).

Consider a TPDA $\mathcal{A} = (L, \Gamma, C, L_0, E, L_f)$ over Σ. We may assume that $C = \{1, ..., m\}$. Let $X \subseteq \mathbb{N}$ be the set of all natural numbers which are lower or upper bounds of some interval $I \in \mathcal{I}$ appering in E (either in a clock constraint or in a stack command). Clearly, X is a finite set. Let $k = \max(X)$ (if $X = \emptyset$, then we let $k = 0$). 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 $\mathbb{R}_{\geq 0}$. Note that $\mathbb{P}(k)$ is a finite non-empty set since $[0,0] \in \mathbb{P}(k)$ for any $k \in \mathbb{N}$. The extended alphabet for such a TPDA $\mathcal{A}$ will be a pushdown alphabet augmented with the following additional components reflecting the performance of the clocks and the stack:

- the partition of the pushdown alphabet will be induced by the component $\{\downarrow, \#, \uparrow\}$;
- for every global clock $c \in \{1, ..., m\}$, we add two components:
  - a component $\mathbb{P}(k)$ which indicates the interval containing a value of the clock c before taking an edge of $\mathcal{A}$;
  - a component $\{0, 1\}$ which indicates whether the clock c was reset after taking an edge of $\mathcal{A}$ or not;
- to handle the local clocks of the stack, we add the component $\mathbb{P}(k)$ which indicates:
  - for all push letters (i.e. with the $\downarrow$-component) the interval containing an initial value of the local clock which will be pushed into the stack;
  - for all pop letters (i.e. with the $\uparrow$-component) the interval containing a value of the clock on the top of the stack.
  - for all letters with $\#$, the stack is not touched and the $\mathbb{P}(k)$-component of this letter is useless. So in this case we can restrict ourselves to the interval $[0,0]$.

Formally, we consider the pushdown alphabet $\tilde{\mathcal{R}}_{m,k} = \langle \mathcal{R}^{\downarrow}_{m,k}, \mathcal{R}^{\#}_{m,k}, \mathcal{R}^{\uparrow}_{m,k} \rangle$ where, for $\delta \in \{\downarrow, \#, \uparrow\}$: $\mathcal{R}^{\delta}_{m,k} = \Sigma \times (\mathbb{P}(k))^m \times \{0,1\}^m \times \mathbb{P}(k) \times \{\delta\}$. Let $\mathcal{R}_{m,k} = \bigcup_{\delta \in \{\downarrow, \#, \uparrow\}} \mathcal{R}^{\delta}_{m,k}$.

Now consider a "simple" TPDA over $\mathcal{R}_{m,k}$ with a single state, a single stack symbol and m clocks $\{1, ..., m\}$; for every letter in $\mathcal{R}_{m,k}$, this TPDA processes the clocks and the stack according to the information encoded in the additional components of $\mathcal{R}_{m,k}$. Let $\mathcal{T}_{m,k} \subseteq \mathbb{T}(\mathcal{R}_{m,k})^+$ denote the timed language accepted by this TPDA.

For intervals $I, I' \in \mathcal{I}$, let $I - I' = \{x - x' \mid x \in I \text{ and } x' \in I'\}$.

The timed language $\mathcal{T}_{m,k}$ can be described formally as follows. Let $w = (b_1, t_1)...(b_n, t_n) \in \mathbb{T}(\mathcal{R}_{m,k})^+$ where, for all $i \in \{1, ..., n\}$, $b_i = (a_i, \overline{G}_i, \overline{R}_i, s_i, \delta_i)$ with $a_i \in \Sigma$, $\overline{G}_i = (g^1_i, ..., g^m_i) \in (\mathbb{P}(k))^m$ (corresponds to the intervals for the global clocks), $\overline{R}_i = (r^1_i, ..., r^m_i) \in \{0,1\}^m$ (corresponds to the resets of global clocks), $s_i \in \mathbb{P}(k)$ (corresponds to the intervals for the local clocks in the stack), $\delta_i \in \{\downarrow, \#, \uparrow\}$ and $t_i \in \mathbb{R}_{\geq 0}$. Then, $w \in \mathcal{T}_{m,k}$ iff the following hold:

- $\textsc{Mask}(b_1...b_n) \in \mathbb{L}$ (with respect to the pushdown alphabet $\tilde{\mathcal{R}}_{m,k}$);
- for all $i \in \{1, ..., n\}$ and $j \in \{1, ..., m\}$, letting $r_0^j = 1$, we have $\langle w \rangle_{i',i} \in g_i^j$ for the greatest $i' \in \{0, 1, ..., i-1\}$ with $r_{i'}^j = 1$;
- for all $i, i' \in \{1, ..., n\}$ with $i < i'$, $\delta_i = \downarrow$, $\delta_{i'} = \uparrow$ and $\textsc{Mask}(b_{i+1}...b_{i'-1}) \in \mathbb{L}$, we have $\langle w \rangle_{i,i'} \in s_{i'} - s_i$.

Clearly, the timed language $\mathcal{T}_{m,k}$ is a non-empty timed pushdown language. Let Δ be an alphabet, $\mathcal{L} \subseteq \Delta^+$ a language and $\mathcal{L}' \subseteq \mathbb{T}\Delta^+$ a timed language. Let $(\mathcal{L} \cap \mathcal{L}') \subseteq \mathbb{T}\Delta^+$ be the "restriction" of $\mathcal{L}'$ to $\mathcal{L}$, i.e., the timed language consisting of all timed words $w = (b_1, t_1)...(b_n, t_n) \in \mathcal{L}'$ such that $b_1...b_n \in \mathcal{L}$. Let Δ, Δ' be alphabets and $h : \Delta \to \Delta'$ a renaming. For a timed word $w = (b_1, t_1)...(b_n, t_n) \in \mathbb{T}\Delta^+$, let $h(w) = (h(b_1), t_1)...(h(b_n), t_n)$. Then, for a timed language $\mathcal{L} \subseteq \mathbb{T}\Delta^+$, let $h(\mathcal{L}) = \{h(w) \mid w \in \mathcal{L}\}$, so $h(\mathcal{L}) \subseteq \mathbb{T}(\Delta')^+$.

Now we formulate our decomposition theorem. This result permits to separate the discrete part of TPDA from their timed part. We show that the discrete part can be described by visibly pushdown languages whereas the timed part can be described by means of timed languages $\mathcal{T}_{m,k}$ which have the following interesting property. We can decide whether a timed word w belongs to $\mathcal{T}_{m,k}$ by analyzing the components of w. In contrast, if we have a TPDA $\mathcal{A}$ and can use it only as a "black box", then we cannot say whether a timed word w is accepted by this TPDA $\mathcal{A}$ without passing w through $\mathcal{A}$.

Theorem 9.8. *Let Σ be an alphabet and $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ a timed language. Then the following are equivalent.*

(a) $\mathcal{L}$ is a timed pushdown language.
(b) There exist $m, k \in \mathbb{N}$, a renaming $h : \mathcal{R}_{m,k} \to \Sigma$, and a visibly pushdown language $\mathcal{L}' \subseteq (\mathcal{R}_{m,k})^+$ over the pushdown alphabet $\tilde{\mathcal{R}}_{m,k}$ such that $\mathcal{L} = h(\mathcal{L}' \cap \mathcal{T}_{m,k})$.

Before we turn to the proof of Theorem 9.8, we give an example of the decomposition for a TPDA.

Example 9.9. *Consider the TPDA $\mathcal{A} = (L, \Gamma, C, L_0, E, L_f)$ over the alphabet $\Sigma = \{a, b\}$ depicted in Fig. 9.2. Formally, $\mathcal{A}$ is defined as follows:*

- $L = \{1, 2\}$, $L_0 = \{1\}$, $L_f = \{2\}$, $\Gamma = \{\gamma\}$, $C = \{x\}$;
- *E consists of the following edges:* $1 \xrightarrow[(\downarrow,\gamma,(0,1))]{a,\textsc{True},\emptyset} 1$, $1 \xrightarrow[\#]{b,\textsc{True},\{x\}} 1$, $1 \xrightarrow[\#]{a,x\geq 1,\emptyset} 2$, $2 \xrightarrow[(\uparrow,\gamma,[1,1])]{a,\textsc{True},\emptyset} 2$.

The timed language $\mathcal{L}(\mathcal{A})$ can be decomposed in the following way. As already mentioned before, m is the number of global clocks of $\mathcal{A}$, i.e., $m = 1$ and k is the maximal constant appearing in the intervals of $\mathcal{A}$, i.e., $k = 1$. Then, $\mathcal{R}_{1,1} = \Sigma \times \mathbb{P}(1) \times \{0, 1\} \times \mathbb{P}(1) \times \{\downarrow, \#, \uparrow\}$. Then, $\mathcal{L} = h(\mathcal{L}' \cap \mathcal{T}_{m,k})$ where:

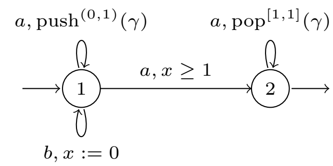

Figure 9.2: TPDA $\mathcal{A}$ of Example 9.9

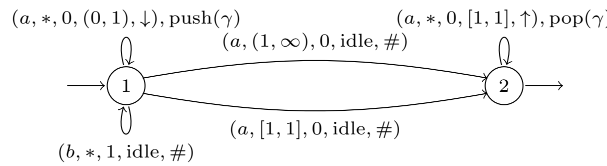

Figure 9.3: TPDA $\mathcal{A}_{\mathcal{L}'}$ of Example 9.9

- *$h : \mathcal{R}_{1,1} \to \Sigma$ is the projection to the first component;*
- *the language $\mathcal{L}' \subseteq (\mathcal{R}_{1,1})^+$ is recognized by the visibly pushdown automaton $\mathcal{A}_{\mathcal{L}'} = (L, \Gamma, L_0, T', L_f)$ over the pushdown alphabet $\tilde{\mathcal{R}}_{1,1}$ depicted in Fig. 9.3. Here, the component $*$ in the transition labels means an arbitrary element of $\mathbb{P}(1)$ and* idle $= [0,0]$ *denotes the idle stack interval for the letters with the #-component. We also would like to point out that every edge of the TPDA $\mathcal{A}$ is simulated by several transitions of the VPA $\mathcal{A}_{\mathcal{L}'}$. For instance, we simulate the edge from the location 1 to the location 2 of the TPDA $\mathcal{A}$ by two edges, since, for the condition $x \geq 1$ we have two intervals $[1,1], (1,\infty)$ in the partition $\mathbb{P}(1)$ which satisfy this condition.*
- *The timed language $\mathcal{T}_{1,1} \subseteq \mathbb{T}(\mathcal{R}_{1,1})^+$ (as defined before) can be recognized by the TPDA $\mathcal{A}_{\mathcal{T}_{1,1}} = (\{1\}, \{\alpha\}, C, \{1\}, E', \{1\})$ depicted in Fig. 9.4. Here, I, J are arbitrary intervals in $\mathbb{P}(1)$. By using a new stack letter α, we want to point out that the stack alphabet of the TPDAs for $\mathcal{T}_{m,k}$ is a singleton alphabet and does not depend on Γ.*

First, we show the implication (a) $\Rightarrow$ (b) of Theorem 9.8.

Lemma 9.10. *Let $\mathcal{A}$ be a timed pushdown automaton over Σ. Then, there exist $m, k \in \mathbb{N}$, a renaming $h : \mathcal{R}_{m,k} \to \Sigma$, and a visibly pushdown automaton $\mathcal{A}'$ over the pushdown alphabet $\tilde{\mathcal{R}}_{m,k}$ such that $\mathcal{L}(\mathcal{A}) = h(\mathcal{L}(\mathcal{A}') \cap \mathcal{T}_{m,k})$.*

Proof. The idea of our decomposition is illustrated in Example 9.9. Now we give a formal proof. Let $\mathcal{A} = (L, \Gamma, C, L_0, E, L_f)$. We may assume without loss of generality that $C = \{1, ..., m\}$. As defined in the beginning of this section, let $X \subseteq \mathbb{N}$ be the set of all natural numbers which are lower or upper bounds of some interval $I \in \mathcal{I}$ appearing in E and let $k = \max(X)$ (for $X = \emptyset$, we let $k = 0$).

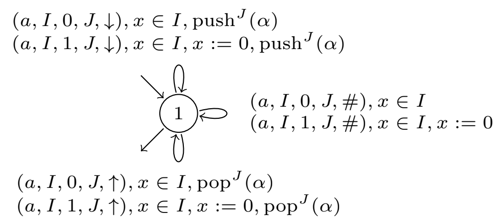

Figure 9.4: TPDA $\mathcal{A}_{\mathcal{T}_{1,1}}$ of Example 9.9

Recall that $\mathcal{R}_{m,k} = \Sigma \times (\mathbb{P}(k))^m \times \{0,1\}^m \times \mathbb{P}(k) \times \{\downarrow, \#, \uparrow\}$. Let $\mathcal{R} = \mathcal{R}_{m,k}$ and $h : \mathcal{R} \to \Sigma$ be the projection to the Σ-component. We define the visibly pushdown automaton $\mathcal{A}' = (L, \Gamma, L_0, T, L_f)$ over the pushdown alphabet $\tilde{\mathcal{R}}_{m,k}$ where the set $T = T^\downarrow \cup T^\# \cup T^\uparrow$ is defined as follows. We simulate every edge $e = \left(\ell \xrightarrow[s]{a,\phi,\Lambda} \ell'\right) \in E$ with $\ell, \ell' \in L$, $a \in \Sigma$, $\phi : \{1, ..., m\} \to \mathcal{I}$, $\Lambda \subseteq \{1, ..., m\}$ and $s \in \mathcal{S}(\Gamma)$ by (possibly multiple) transitions in T depending on the sort of e as follows.

- If $e \in E^\downarrow$ and $s = (\downarrow, \gamma, I)$ for some $\gamma \in \Gamma$ and $I \in \mathcal{I}$, then we let $(\ell, a^\downarrow, \gamma, \ell') \in T^\downarrow$ for all $a^\downarrow = (a, (g^1, ..., g^m), (r^1, ..., r^m), \sigma, \downarrow) \in \mathcal{R}$ such that:
  - for all $j \in \{1, ..., m\}$, g^j is any interval in $\mathbb{P}(k)$ such that $g^j \subseteq \phi(j)$;
  - for all $j \in \{1, ..., m\}$: $r^j \in \{0, 1\}$, and $r^j = 1$ iff $j \in \Lambda$;
  - σ is an interval in $\mathbb{P}(k)$ with $\sigma \subseteq I$.
- If $e \in E^\#$, then we let $(\ell, a^\#, \ell') \in T^\#$ for all $a^\# = (a, (g^1, ..., g^m), (r^1, ..., r^m), \sigma, \#) \in \mathcal{R}$ where $g^1, ..., g^m, r^1, ..., r^m$ are defined as in the previous case and $\sigma = [0, 0]$ is the idle interval.
- If $e \in E^\uparrow$ and $s = (\uparrow, \gamma, I)$ for some $\gamma \in \Gamma$ and $I \in \mathcal{I}$, then we let $(l, a^\uparrow, \gamma, \ell') \in T^\uparrow$ for all $a^\uparrow = (a, (g^1, ..., g^m), (r^1, ..., r^m), \sigma, \uparrow) \in \mathcal{R}$ where $g^1, ..., g^m$, r^1, ..., r^m and σ are defined as in the first case. Note that we do not have transitions in $T^\uparrow$ whose stack letter is $\perp$.

Note that although the emptiness of the stack at the end of a run is not required by visibly pushdown automata, it is checked by intersection with the timed language $\mathcal{T}_{m,k}$.

In the rest of the proof, we denote the timed language $\mathcal{T}_{m,k}$ simply by $\mathcal{T}$. It remains to prove that $\mathcal{L}(\mathcal{A}) = h(\mathcal{L}(\mathcal{A}') \cap \mathcal{T})$.

First, we show $\subseteq$. Let $w = (a_1, t_1)...(a_n, t_n) \in \mathbb{T}\Sigma^+$ be a timed word such that $w \in \mathcal{L}(\mathcal{A})$. Then, there exists a run

$$\rho = \langle \ell_0, \nu_0, u_0 \rangle \vdash_{t_1, e_1} \langle \ell_1, \nu_1, u_1 \rangle \vdash_{t_2, e_2} ... \vdash_{t_n, e_n} \langle \ell_n, \nu_n, u_n \rangle$$

of $\mathcal{A}$ with label$(\rho) = w$ such that

- for all $i \in \{0,...,n\}$, $\ell_i \in L$, $\nu_i \in \mathbb{R}_{\geq 0}^C$ and $u_i \in \mathbb{T}\Gamma^*$;
- for all $i \in \{1,...,n\}$, $e_i = \big(\ell_{i-1} \xrightarrow[s_i]{a_i,\phi_i,\Lambda_i} \ell_i\big)$ for some $\phi_i \in \Phi(C)$, $\Lambda_i \subseteq C$ and $s_i \in \mathcal{S}(\Gamma)$. Assume that

$$s_i = \begin{cases} (\downarrow, \gamma_i, I_i), & \text{if } e_i \in E^{\downarrow}, \\ \#, & \text{if } e_i \in E^{\#}, \\ (\uparrow, \gamma_i, I_i), & \text{if } e_i \in E^{\uparrow}. \end{cases}$$

where $\gamma_i \in \Gamma$ and $I_i \subseteq \mathcal{I}$.

For all $i \in \{0,...,n\}$, assume that $u_i = (p_i^1, \tau_i^1)...(p_i^{x_i}, \tau_i^{x_i})$ where $x_i \geq 0$, $p_i^1,...,p_i^{x_i} \in \Gamma$ and $\tau_i^1,...,\tau_i^{x_i} \in \mathbb{R}_{\geq 0}$. Let $\overline{p}_i = p_i^1...p_i^{x_i} \in \Gamma^*$. Clearly, $\overline{p}_0 = \varepsilon$. For all $i \in \{1,...,n\}$ with $e_i \in E^{\downarrow}$, let $\xi_i = \tau_i^1$, the initial age of a clock which was pushed into the stack at the position i. Note that $\xi_i \in I_i$.

We show that there exist $\pi_1,...,\pi_n \in \mathcal{R}$ such that:

(i) $h(\pi_i) = a_i$ for all $i \in \{1,...,n\}$;
(ii) $\pi_1...\pi_n \in \mathcal{L}(\mathcal{A}')$;
(iii) $(\pi_1, t_1)...(\pi_n, t_n) \in \mathcal{T}$.

For every $i \in \{1,...,n\}$, we let $\pi_i = (a_i, (g_i^1,...,g_i^m), (r_i^1,...,r_i^m), \sigma_i, \delta_i)$ defined as follows.

- For all $j \in \{1,...,m\}$, let $g_i^j \in \mathbb{P}(k)$ be the interval such that $\nu_{i-1}(j) + t_i \in g_i^j$. Since ρ is a run of $\mathcal{A}$, we have $g_i^j \subseteq \phi(j)$.
- If $e_i \in E^{\#}$, then we let $\sigma_i = [0,0]$.
- If $e_i \in E^{\downarrow}$, then we let σ_i be the interval containing ξ_i. Note that $\sigma_i \subseteq I_i$.
- If $e_i \in E^{\uparrow}$, then we let $\sigma_i \in \mathbb{P}(k)$ be the interval containing $\tau_{i-1}^1 + t_i$ (note that $x_{i-1} \geq 1$ since ρ is a run of $\mathcal{A}$). Since ρ is a run of $\mathcal{A}$, we have $\sigma_i \subseteq I_i$.
- For all $j \in \{1,...,m\}$, $r_i^j = \begin{cases} 1, & \text{if } j \in \Lambda_i, \\ 0, & \text{otherwise.} \end{cases}$
- Let $\delta_i = \begin{cases} \downarrow, & \text{if } e_i \in E^{\downarrow}, \\ \#, & \text{if } e_i \in E^{\#}, \\ \uparrow, & \text{if } e_i \in E^{\uparrow} \end{cases}$

Clearly, (i) holds. Now we show (ii). Let

$$\varrho = \langle \ell_0, \overline{p}_0 \rangle \vdash_{e'_1} \langle \ell_1, \overline{p}_1 \rangle \vdash_{e'_2} ... \vdash_{e'_n} \langle \ell_n, \overline{p}_n \rangle$$

where, for all $i \in \{1,...,n\}$ with $e_i \in E^{\downarrow} \cup E^{\uparrow}$, we have $e'_i = (\ell_{i-1}, \pi_i, \gamma_i, \ell_i) \in T$ and, for all $i \in \{1,...,n\}$ with $e_i \in E^{\#}$, we have $e'_i = (\ell_{i-1}, \pi_i, \ell_i) \in T$. Then, ϱ is a run of $\mathcal{A}'$ and $\text{label}(\varrho) = \pi_1...\pi_n$. Hence $\pi_1...\pi_n \in \mathcal{L}(\mathcal{A}')$.

Next, we show (iii).

- Since ρ is a run of $\mathcal{A}$, it is easy to see that $\textsc{Mask}(\pi_1...\pi_n) \in \mathbb{L}$.

- Let $i \in \{1, ..., n\}$ and $j \in \{1, ..., m\}$ and assume that $r_0^j = 1$. Then, for the greatest $i' \in \{0, 1, ..., i-1\}$ with $r_{i'}^j = 1$, we have: $\langle w \rangle_{i',i} = \nu_{i-1}(j) + t_i \in g_i^j$.
- Let $i, i' \in \{1, ..., n\}$ with $i < i'$, $\delta_i = \downarrow$, $\delta_{i'} = \uparrow$ and $\textsc{Mask}(\pi_{i+1}...\pi_{i'-1}) \in \mathbb{L}$. This means that, at the position i, some letter $\gamma \in \Gamma$ was pushed into the stack and at the position i', this letter was popped. Then, $\langle w \rangle_{i,i'} = (\tau_{i'-1}^1 + t_{i'}) - \xi_i \in I_{i'} - I_i$.

Then, $(\pi_1, t_1)...(\pi_n, t_n) \in \mathcal{T}$. Hence, $\mathcal{L}(\mathcal{A}) \subseteq h(\mathcal{L}(\mathcal{A}') \cap \mathcal{T})$.

Now we show the converse inclusion $\mathcal{L}(\mathcal{A}) \supseteq h(\mathcal{L}(\mathcal{A}') \cap \mathcal{T})$. Let $w = (a_1, t_1)...(a_n, t_n) \in h(\mathcal{L}(\mathcal{A}') \cap \mathcal{T})$. Then, there exist $\pi_1, ..., \pi_n \in \mathcal{R}$ such that:

(i) $h(\pi_i) = a_i$ for all $i \in \{1, ..., n\}$;
(ii) $\pi_1...\pi_n \in \mathcal{L}(\mathcal{A}')$;
(iii) $(\pi_1, t_1)...(\pi_n, t_n) \in \mathcal{T}$.

By (i), assume that, for all $i \in \{1, ..., n\}$, $\pi_i = (a_i, (g_i^1, ..., g_i^m), (r_i^1, ..., r_i^m), \sigma_i, \delta_i)$ where $g_i^1, ..., g_i^m, \sigma_i \in \mathbb{P}(k)$, $r_i^1, ..., r_i^m \in \{0, 1\}$ and $\delta_i \in \{\downarrow, \#, \uparrow\}$. By (ii), there exists a run ϱ of $\mathcal{A}'$ with $\text{label}(\varrho) = \pi_1...\pi_n$. Assume that

$$\varrho = \langle \ell_0, \overline{p}_0 \rangle \vdash_{e_1'} \langle \ell_1, \overline{p}_1 \rangle \vdash_{e_2'} ... \vdash_{e_n'} \langle \ell_n, \overline{p}_n \rangle$$

where $\ell_0, ..., \ell_n \in L$, $\overline{p}_0, ..., \overline{p}_n \in \Gamma^*$ and $e_1', ..., e_n' \in T$. We assume that, for all $i \in \{1, ..., n\}$ with $\delta_i \in \{\downarrow, \uparrow\}$, $e_i' = (\ell_{i-1}, \pi_i, \gamma_i, \ell_i)$ for some $\gamma_i \in \Gamma$ and, for all $i \in \{1, ..., n\}$ with $\delta_i = \#$, we have $e_i' = (\ell_{i-1}, \pi_i, \ell_i)$.

By definition of $\mathcal{A}'$, for every $i \in \{1, ..., n\}$, there exists an edge $e_i = \left(\ell_{i-1} \xrightarrow[s_i]{a_i, \phi_i, \Lambda_i} \ell_i\right) \in E$ such that:

- $s_i = \begin{cases} (\downarrow, \gamma_i, I_i), & \text{if } \delta_i = \downarrow, \\ \#, & \text{if } \delta_i = \#, \text{ where } I_i \in \mathcal{I} \text{ with } \sigma_i \subseteq I_i; \\ (\uparrow, \gamma_i, I_i), & \text{if } \delta_i = \uparrow \end{cases}$
- $g_i^j \subseteq \phi_i(j)$ for all $j \in \{1, ..., m\}$;
- $\Lambda_i = \{j \mid j \in \{1, ..., m\} \text{ and } r_i^j = 1\}$;

Now we show that there exists a run ρ of $\mathcal{A}$ such that $\text{label}(\rho) = w$. We let

$$\rho = \langle \ell_0, \nu_0, u_0 \rangle \vdash_{t_1, e_1} \langle \ell_1, \nu_1, u_1 \rangle \vdash_{t_2, e_2} ... \vdash_{t_n, e_n} \langle \ell_n, \nu_n, u_n \rangle$$

where $\nu_i \in \mathbb{R}_{\geq 0}^C$ and $u_i \in \mathbb{T}\Gamma^*$ are defined as follows.

- We let $\nu_0(j) = 0$ for all $j \in \{1, ..., m\}$ and $u_0 = \varepsilon$.
- Assume that $i \geq 1$ and ν_{i-1} is defined. Then, by (iii), we have for all $j \in \{1, ..., m\}$: $\nu_{i-1}(j) + t_i \in g_i^j \subseteq \phi_i(j)$. Hence, $\nu_{i-1} + t_i \models \phi_i$. Then, we let $\nu_i = (\nu_{i-1} + t_i)[\Lambda_i := 0]$.
- Assume that $i \geq 1$ and u_{i-1} is defined. We distinguish between the following cases:

- If $\delta_i = \downarrow$, then, we let $u_i = (\gamma_i, \xi_i)(u_{i-1} + t_i)$ for some unknown nonnegative real number $\xi_i \in \sigma_i$ which will be determined when the letter γ_i will be popped.
- If $\delta_i = \#$, then we let $u_i = u_{i-1} + t_i$.
- If $\delta_i = \uparrow$, then, using (iii) and the fact that ϱ is a run of $\mathcal{A}'$, we have: $u_{i-1} = (\gamma_i, \tau')u'$ such that there exists a position $i' < i$ with $\delta_{i'} = \uparrow$, $\textsc{Mask}(\pi_{i'+1}...\pi_{i-1}) \in \mathbb{L}$ and $\langle w \rangle_{i',i} = \tau' - \xi_{i'} \in \sigma_i - \sigma_{i'}$. Then, there exist $y \in \sigma_i$ and $y' \in \sigma_{i'}$ such that $\tau' - \xi_{i'} = y - y'$. By letting $\xi_{i'} = y' \in \sigma_{i'}$ (which determines the unknown value of ξ_i), we have $\tau' = y \in \sigma_i \subseteq I_i$. So we let $u_i = u'$.

Finally, since $\textsc{Mask}(\pi_1...\pi_n) \in \mathbb{L}$, we have $u_n = \varepsilon$. This shows that ρ is a run of $\mathcal{A}$. Since label$(\rho) = w$, we obtain $w \in \mathcal{L}(\mathcal{A})$.

□

Now we turn to the converse direction of Theorem 9.8.

Lemma 9.11. *Let $m, k \in \mathbb{N}$ and $\mathcal{A}$ a visibly pushdown automaton over the pushdown alphabet $\tilde{\mathcal{R}}_{m,k}$. Then, there exists a TPDA $\mathcal{A}'$ over the alphabet $\mathcal{R}_{m,k}$ such that $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A}) \cap \mathcal{T}_{m,k}$.*

Proof. The proof idea is the following. Since we work here with the extended alphabet $\mathcal{R}_{m,k}$ (which corresponds to m global clocks), every letter of this alphabets contains the information about the guards and resets of global clocks as well as performance of the timed stack, we can rewrite the transitions of a VPA (over the extended pushdown alphabet $\tilde{\mathcal{R}}_{m,k}$) as edges of a TPDA (over the extended alphabet $\mathcal{R}_{m,k}$).

Let $\mathcal{A} = (L, \Gamma, L_0, T, L_f)$. We put $\mathcal{A}' = (L, \Gamma, C, L_0, E, L_f)$ where $C = \{1, ..., m\}$ and $E = E^{\downarrow} \cup E^{\#} \cup E^{\uparrow}$ is defined as follows.

- For every $t = (\ell, a^{\downarrow}, \gamma, \ell') \in T^{\downarrow}$ with $a^{\downarrow} = (a, (g^1, ..., g^m), (r^1, ..., r^m), \sigma, \downarrow)$ (where $g^1, ..., g^m, \sigma \in \mathbb{P}(k)$ and $r^1, ..., r^m \in \{0, 1\}$), we let $\left(\ell \xrightarrow[(\downarrow,\gamma,\sigma)]{a^{\downarrow},\phi,\Lambda} \ell'\right) \in E^{\downarrow}$ where $\phi(j) = g^j$ for all $j \in C$ and $\Lambda = \{j \in C \mid r^j = 1\}$.
- For every $t = (\ell, a^{\#}, \ell') \in T^{\#}$ with $a^{\#} = (a, (g^1, ..., g^m), (r^1, ..., r^m), \sigma, \#)$, we let $\left(\ell \xrightarrow[\#]{a^{\#}\phi,\Lambda} \ell'\right) \in E^{\#}$ where ϕ and Λ are defined as in the previous case.
- For every $t = (\ell, a^{\uparrow}, \gamma, \ell') \in T^{\uparrow}$ with $a^{\uparrow} = (a, (g^1, ..., g^m), (r^1, ..., r^m), \sigma, \uparrow)$, we let $\left(\ell \xrightarrow[(\uparrow,\gamma,\sigma)]{a^{\uparrow},\phi,\Lambda} \ell'\right) \in E^{\uparrow}$ where ϕ and Λ are defined as in the first case.

Let $\mathcal{R} = \mathcal{R}_{m,k}$ and $\mathcal{T} = \mathcal{T}_{m,k}$. We prove that $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A}) \cap \mathcal{T}$.

First, we show the inclusion $\subseteq$. Let $v = (\pi_1, t_1)...(\pi_n, t_n) \in \mathcal{L}(\mathcal{A}')$ where, for all $i \in \{1, ..., n\}$, $\pi_i = (a_i, (g_i^1, ..., g_i^m), (r_i^1, ..., r_i^m), \sigma_i, \delta_i)$. Then, there exists a run ρ of $\mathcal{A}'$ with label$(\rho) = v$. Assume that

$$\rho = \langle \ell_0, \nu_0, u_0 \rangle \vdash_{t_1,e_1} \langle \ell_1, \nu_1, u_1 \rangle \vdash_{t_2,e_2} ... \vdash_{t_n,e_n} \langle \ell_n, \nu_n, u_n \rangle.$$

where, for all $i \in \{1, ..., n\}$, $e_i = \big(\ell_{i-1} \xrightarrow[s_i]{\pi_i, \phi_i, \Lambda_i} \ell_i\big)$ with

$$s_i = \begin{cases} (\downarrow, \gamma_i, I_i), & \text{if } \delta_i = \downarrow, \\ \#, & \text{if } \delta_i = \#, \\ (\uparrow, \gamma_i, I_i), & \text{if } \delta_i = \uparrow \end{cases}$$

For all $i \in \{0, ..., n\}$, assume that $u_i = (p_i^1, \tau_i^1)...(p_i^{x_i}, \tau_i^{x_i})$ where $x_i \geq 0$, $p_i^1, ..., p_i^{x_i} \in \Gamma$ and $\tau_i^1, ..., \tau_i^{x_i} \in \mathbb{R}_{\geq 0}$. Let $\overline{p}_i = p_i^1...p_i^{x_i} \in \Gamma^*$.

Note that, for all $i \in \{1, ..., n\}$ with $\delta_i \in \{\downarrow, \uparrow\}$, we have $e'_i := (\ell_{i-1}, \pi_i, \gamma_i, \ell_i) \in T$ and, for all $i \in \{1, ..., n\}$ with $\delta_i = \#$, we have $e'_i := (\ell_{i-1}, \pi_i, \ell_i) \in T$. Then,

$$\langle \ell_0, \overline{p}_0 \rangle \vdash_{e'_1} \langle \ell_1, \overline{p}_1 \rangle \vdash_{e'_2} ... \vdash_{e'_n} \langle \ell_n, \overline{p}_n \rangle$$

is a run of $\mathcal{A}$ and hence $\pi_1...\pi_n \in \mathcal{L}(\mathcal{A})$. Since, for all $i \in \{1, ..., n\}$, $\phi_i(j) = g_i^j$ for all $j \in C$, $\Lambda_i = \{j \in C \mid r_i^j = 1\}$ and $\sigma_i = I_i$, it is easy to see that $v \in \mathcal{T}$. Then, $v \in \mathcal{L}(\mathcal{A}) \cap \mathcal{T}$ and hence $\mathcal{L}(\mathcal{A}') \subseteq \mathcal{L}(\mathcal{A}) \cap \mathcal{T}$.

Second, we show the inclusion $\mathcal{L}(\mathcal{A}) \cap \mathcal{T} \subseteq \mathcal{L}(\mathcal{A}')$. Let $v = (\pi_1, t_1)...(\pi_n, t_n) \in \mathcal{L}(\mathcal{A}) \cap \mathcal{T}$ such that $\pi_i = (a_i, (g_i^1, ..., g_i^m), (u_i^1, ..., u_i^m), \sigma_i, \delta_i)$ for all $i \in \{1, ..., n\}$. Then, $\pi_1...\pi_n \in \mathcal{L}(\mathcal{A})$ and hence there exists a run ϱ of $\mathcal{A}$ with $\text{label}(\varrho) = v$. Assume that

$$\varrho = \langle \ell_0, \overline{p}_0 \rangle \vdash_{e'_1} \langle \ell_1, \overline{p}_1 \rangle \vdash_{e'_2} ... \vdash_{e'_n} \langle \ell_n, \overline{p}_n \rangle$$

such that:

- for all $i \in \{0, ..., n\}$: $\ell_i \in L$ and $\overline{p}_i \in \Gamma^*$ (since $v \in \mathcal{T}$, we have $\overline{p}_n = \varepsilon$)
- for all $i \in \{1, ..., n\}$ with $\delta_i \in \{\downarrow, \uparrow\}$: $e'_i = (\ell_{i-1}, \pi_i, \gamma_i, \ell_i)$ for some $\gamma_i \in \Gamma$ (note that $\gamma_i \neq \perp$ since $v \in \mathcal{T}$; recall that $\perp$ is a special bottom-of-stack symbol of a VPA);
- for all $i \in \{1, ..., n\}$ with $\delta_i = \#$: $e'_i = (\ell_{i-1}, \pi_i, \ell_i)$.

We define edges $e_1, ..., e_n \in E$ as follows. For every $i \in \{1, ..., n\}$, let $e_i = \big(\ell_{i-1} \xrightarrow[s_i]{\pi_i, \phi_i, \Lambda_i} \ell_i\big)$ where:

- $s_i = \begin{cases} (\downarrow, \gamma_i, \sigma_i), & \text{if } \delta_i = \downarrow, \\ \#, & \text{if } \delta_i = \#, \\ (\uparrow, \gamma_i, \sigma_i), & \text{if } \delta_i = \uparrow; \end{cases}$
- $\phi_i(j) = g_i^j$ for all $j \in C$;
- $\Lambda_i = \{j \in C \mid r_i^j = 1\}$.

Then, the condition $v \in \mathcal{T}$ guarantees that there exists a run ρ of $\mathcal{A}'$ of the form

$$\rho = \langle \ell_0, \nu_0, u_0 \rangle \vdash_{t_1, e_1} \langle \ell_1, \nu_1, u_1 \rangle \vdash_{t_2, e_2} ... \vdash_{t_n, e_n} \langle \ell_n, \nu_n, u_n \rangle$$

where $\nu_0, ..., \nu_n \in \mathbb{R}_{\geq 0}^C$ and $u_0, ..., u_n \in \mathbb{T}\Gamma^*$. Since $\text{label}(\rho) = v$, we have $v \in \mathcal{L}(\mathcal{A}')$. Then, $\mathcal{L}(\mathcal{A}) \cap \mathcal{T} \subseteq \mathcal{L}(\mathcal{A}')$. $\square$

It is easy to see that the class of timed pushdown languages is closed under renamings:

Lemma 9.12. *Let Δ, Δ' be alphabets, $h : \Delta' \to \Delta$ a renaming, and $\mathcal{A}$ a TPDA over Δ'. Then, there exists a TPDA $\mathcal{A}'$ over Δ such that $\mathcal{L}(\mathcal{A}') = h(\mathcal{L}(\mathcal{A}))$.*

Proof. Let $\mathcal{A} = (L, \Gamma, C, L_0, E, L_f)$. Then, we put $\mathcal{A}' = (L, \Gamma, C, L_0, E', L_f)$ with $E' = \{(\ell, h(a), s, \phi, \Lambda, \ell') \mid (\ell, a, s, \phi, \Lambda, \ell') \in E\}$. Then, it is not difficult to see that $\mathcal{L}(\mathcal{A}') = h(\mathcal{L}(\mathcal{A}))$. □

Proof of Theorem 9.8. Immediate by Lemmas 9.10, 9.11 and 9.12. □

Remark 9.13. *As it can be observed from the proof of Theorem 9.8, instead of the k-interval partition $\mathbb{P}(k)$, for every global clock or the timed stack, one could take a partition induced by bounds of the intervals which correspond to this clock or timed stack. For instance, if the intervals $(0, 1)$ and $(8, 15)$ appear in the commands for the timed stack, then we could take the partition $\{[0, 0], (0, 1), [1, 1], (1, 8), [8, 8], (8, 15), [15, 15], (15, \infty)\}$. However, for the simplicity of our notations, we considered the same partition $\mathbb{P}(k)$ for all global clocks and the timed stack.*

As a corollary of Theorem 9.8 and its proof, we deduce a decomposition theorem for timed automata. These may be considered as TPDA whose sets of push and pop edges are empty (and hence a stack alphabet is irrelevant for their definition). We slightly modify the extended alphabet needed for the decomposition by excluding the components relevant for the stack. Moreover, instead of visibly pushdown languages we consider classical regular languages. For $m, k \in \mathbb{N}$, let $\mathcal{R}^0_{m,k} = \Sigma \times (\mathbb{P}(k))^m \times \{0, 1\}^m$. We define the timed language $\mathcal{T}^0_{m,k} \subseteq \mathbb{T}(\mathcal{R}^0_{m,k})^+$ as follows. Let $w = (b_1, t_1)...(b_n, t_n) \in \mathbb{T}(\mathcal{R}^0_{m,k})^+$ where, for all $i \in \{1, ..., n\}$, $b_i = (a_i, (g^1_i, ..., g^m_i), (r^1_i, ..., r^m_i))$ with $a_i \in \Sigma$, $g^1_i, ..., g^m_i \in \mathbb{P}(k)$ and $r^1_i, ..., r^m_i \in \{0, 1\}$. Then, $w \in \mathcal{T}^0_{m,k}$ iff, for all $i \in \{1, ..., n\}$ and $j \in \{1, ..., m\}$, letting $r^j_0 = 1$, we have $\langle w \rangle_{i',i} \in g^j_i$ for the greatest $i' \in \{0, 1, ..., i-1\}$ with $r^j_{i'} = 1$.

Corollary 9.14. *Let Σ be an alphabet and $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ a timed language. Then the following are equivalent.*

(a) $\mathcal{L}$ is recognizable by a timed automaton.
(b) There exist $m, k \in \mathbb{N}$, a renaming $h : \mathcal{R}^0_{m,k} \to \Sigma$ and a regular language $\mathcal{L}' \subseteq (\mathcal{R}^0_{m,k})^+$ such that $\mathcal{L} = h(\mathcal{L}' \cap \mathcal{T}^0_{m,k})$.

9.5 Definability equals recognizability

In this section, we prove Theorem 9.4. First, we show that **TML**-definable timed languages are pushdown recognizable.

Theorem 9.15. *Let Σ be an alphabet and $\psi \in$ **TML**(Σ) a sentence. Then, there exists a TPDA $\mathcal{A}$ over Σ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\psi)$.*

To prove this theorem, let $\psi = \exists^{\mathrm{match}}\mu.\exists D_1. \ldots \exists D_m.\varphi \in \mathbf{TML}(\Sigma)$ with $m \geq 0$. We may assume that $D_1, \ldots, D_m \in \mathcal{D}$ are pairwise distinct variables.

We wish to use Theorem 9.8. As preparation for this, we prove the following technical lemma which provides a decomposition of a **TML**-sentence. For the definitions of $\mathcal{R}_{m,k}$, $\tilde{R}_{m,k}$ and $\mathcal{T}_{m,k}$ we refer the reader to the previous section.

Lemma 9.16. *Let $\psi \in \mathbf{TML}(\Sigma)$ be a sentence as defined above. Then, there exist $k \in \mathbb{N}$, a renaming $h : \mathcal{R}_{m,k} \to \Sigma$ and a sentence $\varphi^* \in \mathbf{MSO}_{\mathbb{L}}(\tilde{\mathcal{R}}_{m,k})$ such that $\mathcal{L}(\psi) = h(\mathcal{L}(\varphi^*) \cap \mathcal{T}_{m,k})$.*

Before we turn to the proof of Lemma 9.16, we give an informal explanation and an example which illustrates such a decomposition of ψ. For decomposition, we will consider the extended alphabet $\mathcal{R}_{m,k}$ where m is the number of relative distance variables of ψ and k is the maximal natural number which is a lower or upper bound of some interval appearing in ψ (if ψ does not contain any intervals, then we let $k = 0$). So, our extended alphabet is $\Sigma \times (\mathbb{P}(k))^m \times \{0,1\}^m \times \mathbb{P}(k) \times \{\downarrow, \#, \uparrow\}$. The additional components will have the following meaning.

- Using a vector $(g^1, \ldots, g^m) \in (\mathbb{P}(k))^m$, we will encode the intervals which appear in the relative distance predicates of ψ. Here the component g^i ($i \in \{1, \ldots, m\}$) is responsible for the relative distance predicates with the variable D_i.
- Using a vector $(r^1, \ldots, r^m) \in \{0,1\}^m$, we will implement the standard Büchi-encoding of the variables $D_1, \ldots, D_m$.
- Using the component $\mathbb{P}(k) \times \{\downarrow, \#, \uparrow\}$, we will model quantitative matchings. Here $\mathbb{P}(k)$ is responsible for the intervals of quantitative matchings. The component $\{\downarrow, \#, \uparrow\}$ will have the following task. If a position does not belong to any pair in a matching relation, then it is marked by $\#$. If a position is on the left side in a matched pair, then it is marked by $\downarrow$. If a position is on the right side in a matched pair, then it is marked by $\uparrow$.

Example 9.17. *Let $\Sigma = \{a, b\}$ and*

$$\psi = \exists^{\mathrm{match}}\mu.\exists D.\forall x.(D(x) \wedge [(\exists y.\mu^{(1,\infty)}(x,y)) \vee \mathrm{d}^{(0,1]}(D,x) \vee P_b(x)]).$$

In this example, we have $m = 1$ and $k = 1$, i.e. the extended alphabet is $\mathcal{R}_{1,1} = \Sigma \times \mathbb{P}(k) \times \{0,1\} \times \mathbb{P}(k) \times \{\downarrow, \#, \uparrow\}$. As a renaming $h : \mathcal{R}_{m,k} \to \Sigma$, we take the projection to the Σ-component. The sentence $\varphi^* \in \mathbf{MSO}_{\mathbb{L}}(\tilde{\mathcal{R}}_{1,1})$ is defined:

$$\forall x.(\varphi_1 \wedge [(\exists y.\varphi_2) \vee \varphi_3 \vee \varphi_4])$$

where

$$\begin{aligned}
\varphi_1 &= P_{(*,*,1,*,*)}(x) \\
\varphi_2 &= \mathbb{L}(x,y) \wedge P_{(*,*,*,[0,0],\downarrow)}(y) \wedge P_{(*,*,*,(1,\infty),\uparrow)}(y) \\
\varphi_3 &= P_{(*,*,(0,1),*,*)}(x) \vee P_{(*,*,[1,1],*,*)}(x) \\
\varphi_4 &= P_{(b,*,*,*,*)}(x).
\end{aligned}$$

where we denote by $*$ the components which can take arbitrary values from their domains. Then, $\mathcal{L}(\psi) = h(\mathcal{L}(\varphi^*) \cap \mathcal{T}_{m,k})$.

Proof of Lemma 9.16. Let $\psi = \exists^{\text{match}}\mu.\exists D_1. \ldots \exists D_m.\varphi$ as defined above. Let k be the maximal natural number which is a lower or upper bound of some interval appearing in ψ (if ψ does not contain any intervals, then we let $k = 0$). For simplicity, let $\mathcal{R} = \mathcal{R}_{m,k}$, $\tilde{\mathcal{R}} = \tilde{\mathcal{R}}_{m,k}$ and $\mathcal{T} = \mathcal{T}_{m,k}$. Let $\Delta = \{\downarrow, \#, \uparrow\}$. We will denote a letter of $\mathcal{R}$ by $(a, \overline{G}, \overline{R}, s, \delta)$ where $a \in \Sigma$, $\overline{G} \in (\mathbb{P}(k))^m$, $\overline{R} \in \{0,1\}^m$, $s \in \mathbb{P}(k)$ and $\delta \in \Delta$. Let $h : \mathcal{R} \to \Sigma$ be the projection to the Σ-component. Finally, we define the formula $\varphi^* \in \mathbf{MSO}_{\mathbb{L}}(\tilde{\mathcal{R}})$ from the formula φ by the following substitutions.

- If $P_a(x)$ with $a \in \Sigma$ and $x \in V_1$ is a subformula of φ, then $P_a(x)$ is replaced by the formula
$$\bigvee(P_{(a,\overline{G},\overline{R},s,\delta)}(x) \mid \overline{G} \in (\mathbb{P}(k))^m, \overline{R} \in \{0,1\}^m, s \in \mathbb{P}(k), \delta \in \Delta).$$
- If $D_j(x)$ with $j \in \{1, \ldots, m\}$ and $x \in V_1$ appears in φ, then $D_j(x)$ is replaced by the formula
$$\bigvee(P_{(a,\overline{G},\overline{R},s,\delta)}(x) \mid a \in \Sigma, \overline{G} \in (\mathbb{P}(k))^m, \overline{R} = (r^1, \ldots, r^m) \in \{0,1\}^m \text{ with } r^j = 1, s \in \mathbb{P}(k), \delta \in \Delta).$$
- If $\mathrm{d}^I(D_j, x)$ is a subformula of φ where $j \in \{1, \ldots, m\}$, $x \in V_1$ and $I \in \mathcal{I}$, then we replace $\mathrm{d}^I(D_j, x)$ by the formula
$$\bigvee(P_{(a,\overline{G},\overline{R},s,\delta)}(x) \mid a \in \Sigma, \overline{G} = (g^1, \ldots, g^m) \in (\mathbb{P}(k))^m \text{ with } g^j \subseteq I, \overline{R} \in \{0,1\}^m, s \in \mathbb{P}(k), \delta \in \Delta).$$
Note that we remove the variable D_j from φ.
- If $\mu^I(x, y)$ is a subformula of φ where $x, y \in V_1$ and $I \in \mathcal{I}$, then $\mu^I(x, y)$ is replaced by the formula
$$\mathbb{L}(x, y) \wedge \bigvee(P_{(a,\overline{G},\overline{R},[0,0],\downarrow)}(x) \mid a \in \Sigma, \overline{G} \in (\mathbb{P}(k))^m, \overline{R} \in \{0,1\}^m) \wedge \bigvee(P_{(a,\overline{G},\overline{R},s,\uparrow)}(y) \mid a \in \Sigma, \overline{G} \in (\mathbb{P}(k))^m, \overline{R} \in \{0,1\}^m, s \in \mathbb{P}(k) \text{ with } s \subseteq I).$$
Note that here we replace the matching relation μ by the matching relation $\mathbb{L}$ with respect to the visibly pushdown alphabet $\tilde{\mathcal{R}}$ and measure the time distance using the $\mathbb{P}(S)$-component of the extended alphabet.

Note that φ^* is a sentence, since $\mathrm{Free}(\varphi) \subseteq \{D_1, \ldots, D_m, \mu\}$ and we removed $D_1, \ldots, D_m, \mu$ when constructing φ^*. It remains to show that $\mathcal{L}(\psi) = h(\mathcal{L}(\varphi^*) \cap \mathcal{T})$.

First, we show the inclusion $\subseteq$. Let $w = (a_1, t_1) \ldots (a_n, t_n) \in \mathcal{L}(\psi)$ and σ be a fixed $(w, \mathcal{U})$-assignment. Then, there exist a matching $M \subseteq \mathrm{dom}(w) \times \mathrm{dom}(w)$ and sets $J_1, \ldots, J_m \subseteq \mathrm{dom}(w)$ such that $(w, \sigma[D_1/J_1, \ldots, D_m/J_m, \mu/M]) \models \varphi$. We construct a word $w^* = \pi_1 \ldots \pi_n \in \mathcal{R}^+$ where, for all $i \in \{1, \ldots, n\}$, $\pi_i = (a_i, (g_i^1, \ldots, g_i^m), (r_i^1, \ldots, r_i^m), s_i, \delta_i)$ with $a_i \in \Sigma$, $g_i^1, \ldots, g_i^m, s_i \in \mathbb{P}(k)$, $r_i^1, \ldots, r_i^m \in \{0,1\}$ and $\delta_i \in \Delta$ as follows.

- For all $i, i' \in \{1, ..., n\}$ with $(i, i') \in M$, we put $\delta_i = \downarrow$, $\delta_{i'} = \uparrow$, $s_i = [0, 0]$ and let $s_{i'}$ be the interval in the partition $\mathbb{P}(k)$ with $\langle w \rangle_{i,i'} \in s_{i'}$ (uniquely determined).
- For all $i \in \{1, ..., n\}$ such that i does not belong to any pair in M, we put $\delta_i = \#$ and $s_i = [0, 0]$.
- For all $i \in \{1, ..., n\}$ and $j \in \{1, ..., m\}$, we let $r_i^j = 1$ iff $i \in J_j$.
- For all $i \in \{1, ..., n\}$ and $j \in \{1, ..., m\}$, we let g_i^j be the interval in the partition $\mathbb{P}(k)$ such that $(J_j, i) \in \mathrm{d}^{k_i^j}(w)$ (note that g_i^j is uniquely determined).

Let $w^{**} := (\pi_1, t_1)...(\pi_n, t_n) \in \mathbb{T}\mathcal{R}^+$. Clearly, since $I - [0, 0] = I$ for all $I \in \mathcal{I}$, we have $w^{**} \in \mathcal{T}$. Moreover, $h(w^{**}) = w$.

For a $(w,\mathcal{U})$-assignment $\sigma : \mathcal{U} \to \mathrm{dom}(w) \cup 2^{\mathrm{dom}(w)} \cup 2^{(\mathrm{dom}(w))^2}$, let $\overline{\sigma} = \sigma[D_1/J_1, ..., D_m/J_m, \mu/M]$ and $\overline{\overline{\sigma}}$ be the restriction of σ to $V_1 \cup V_2$, i.e., $\overline{\overline{\sigma}} = \sigma|_{V_1 \cup V_2}$.

Now we show that $w^* \in \mathcal{L}(\varphi^*)$. For this, we show that, for every w-assignment σ and every subformula η of φ, the following holds:

$$(w, \overline{\sigma}) \models \eta \ \text{ iff } \ (w^*, \overline{\overline{\sigma}}) \models \eta^*. \tag{9.1}$$

We proceed by induction on the structure of η.

(i) The cases $\eta = P_a(x)$, $\eta = (x \leq y)$ and $\eta = X(x)$ with $a \in \Sigma$, $x, y \in V_1$ and $X \in V_2$ are straightforward.

(ii) Let $\eta = D_j(x)$ with $j \in \{1, ..., m\}$ and $x \in V_1$. Then:

$$(w, \overline{\sigma}) \models \eta \;\Leftrightarrow\; \sigma(x) \in J_j \;\Leftrightarrow\; r^j_{\sigma(x)} = 1 \;\Leftrightarrow\; (w^*, \overline{\overline{\sigma}}) \models \eta^*$$

(iii) Let $\eta = \mathrm{d}^I(D_j, x)$ where $j \in \{1, ..., m\}$, $x \in V_1$ and $I \in \mathcal{I}$. Then:

$$(w, \overline{\sigma}) \models \eta \;\Leftrightarrow\; g^j_{\sigma(x)} \subseteq I \;\Leftrightarrow\; (w^*, \overline{\overline{\sigma}}) \models \eta^*.$$

(iv) Let $\eta = \mu^I(x, y)$ where $x, y \in V_1$ and $I \in \mathcal{I}$. Then:

$$\begin{aligned}(w, \overline{\sigma}) \models \eta \quad &\Leftrightarrow \quad (\sigma(x) < \sigma(y) \wedge \delta_{\sigma(x)} = \downarrow \wedge \delta_{\sigma(y)} = \uparrow \\ &\qquad\quad \wedge \textsc{Mask}(r_{\sigma(x)+1}...r_{\sigma(y)-1}) \in \mathbb{L} \wedge s_{\sigma(y)} \subseteq I) \\ &\Leftrightarrow \quad (w, \overline{\overline{\sigma}}) \models \eta^*\end{aligned}$$

(v) Let $\eta = \eta_1 \vee \eta_2$. By induction hypothesis, $(w, \overline{\sigma}) \models \eta_i \;\Leftrightarrow\; (w^*, \overline{\overline{\sigma}}) \models (\eta_i)^*$ for $i \in \{1, 2\}$. Then:

$$\begin{aligned}(w, \overline{\sigma}) \models \eta \;&\Leftrightarrow\; (w, \overline{\sigma}) \models \eta_1 \text{ or } (w, \overline{\sigma}) \models \eta_2 \\ &\overset{(!)}{\Leftrightarrow}\; (w^*, \overline{\overline{\sigma}}) \models (\eta_1)^* \text{ or } (w^*, \overline{\overline{\sigma}}) \models (\eta_2)^* \\ &\Leftrightarrow\; (w^*, \overline{\overline{\sigma}}) \models (\eta_1)^* \vee (\eta_2)^* \models (w^*, \overline{\overline{\sigma}}) \models \eta^*\end{aligned}$$

Here, at the place (!), we apply induction hypothesis.

(vi) Let $\eta = \neg\eta'$. By induction hypothesis, $(w, \overline{\sigma}) \models \eta' \;\Leftrightarrow\; (w^*, \overline{\overline{\sigma}}) \models (\eta')^*$. Then:

$$(w, \overline{\sigma}) \models \eta \;\Leftrightarrow\; (w, \overline{\sigma}) \not\models \eta' \;\Leftrightarrow\; (w^*, \overline{\overline{\sigma}}) \not\models (\eta')^* \;\Leftrightarrow\; (w^*, \overline{\overline{\sigma}}) \models \eta.$$

Here, at the place (!), we apply induction hypothesis.

(vii) Let $\eta = \exists x.\eta'$ with $x \in V_1$. For every $i \in \{1, ..., n\}$, by induction hypothesis, we have $(w, \overline{\sigma}[x/i]) \models \eta' \;\Leftrightarrow\; (w, \overline{\overline{\sigma}}[x/i]) \models (\eta')^*$. Then, since $\eta^* = \exists x.(\eta')^*$, (9.1) holds true.

(viii) The case $\eta = \exists X.\eta'$ with $X \in V_2$ is similar to the previous case.

Since φ^* is a sentence and $(w, \overline{\sigma}) \models \varphi$ for any $(w, \mathcal{U})$-assignment σ, it follows from (9.1) that $w^* \models \varphi^*$ and hence $w^* \in \mathcal{L}(\varphi^*)$. Thus, $w \in h(\mathcal{L}(\varphi^*) \cap \mathcal{T})$.

Second, we show the converse inclusion $\mathcal{L}(\psi) \supseteq h(\mathcal{L}(\varphi^*) \cap \mathcal{T})$. Let $w = (a_1, t_1)...(a_n, t_n) \in h(\mathcal{L}(\varphi^*) \cap \mathcal{T})$. Then, there exists a timed word $w^{**} = (\pi_1, t_1)...(\pi_n, t_n) \in \mathcal{L}(\varphi^*) \cap \mathcal{T}$ such that $h(w^{**}) = w$. Assume that, for all $i \in \{1, ..., n\}$, $\pi_i = (a_i, (g_i^1, ..., g_i^m), (r_i^1, ..., r_i^m), s_i, \delta_i)$. Let $M = \{(i, j) \in (\mathrm{dom}(w))^2 \mid i < j, \delta_i = \downarrow, \delta_j = \uparrow \text{ and } \textsc{Mask}(r_{i+1}...r_{j-1}) \in \mathbb{L}\}$. Since $w^{**} \in \mathcal{T}$, M is a matching. For every $j \in \{1, ..., m\}$, let $J_j = \{i \in \mathrm{dom}(w) \mid r_i^j = 1\}$. Let σ be a $(w, \mathcal{U})$-assignment. To prove that $w \in \mathcal{L}(\psi)$, we show that $(w, \sigma[D_1/J_1, ..., D_m/J_m, \mu/M]) \models \varphi$. As in the proof of the inclusion $\subseteq$, for any $(w, \mathcal{U})$-assignment $\sigma : \mathcal{U} \to \mathrm{dom}(w) \cup 2^{\mathrm{dom}(w)} \cup 2^{(\mathrm{dom}(w))^2}$, let $\overline{\sigma} = \sigma[D_1/J_1, ..., D_m/J_m, \mu/M]$ and $\overline{\overline{\sigma}}$ be the restriction of σ to $V_1 \cup V_2$. Let $w^* = \pi_1...\pi_n \in \mathcal{L}(\varphi^*)$. We show that, for every w-assignment σ and every subformula η of φ, (9.1) holds. We proceed by induction on the structure of η in the same fashion as it was done in (i) - (viii) with the only difference that we apply the fact that $w^{**} \in \mathcal{T}$. Then, $(w, \sigma[D_1/J_1, ..., D_m/J_m, \mu/M]) \models \varphi$ and hence $w \models \psi$. This finishes the proof of this lemma. $\square$

Proof of Theorem 9.15. Immediate by Lemma 9.16, Theorem 9.7 and Theorem 9.8, implication (b) $\Rightarrow$ (a). $\square$

Now, we show the converse direction of Theorem 9.4.

Theorem 9.18. *Let Σ be an alphabet and $\mathcal{A}$ a timed pushdown automaton over Σ. Then, there exists a sentence $\psi \in \mathbf{TML}(\Sigma)$ such that $\mathcal{L}(\psi) = \mathcal{L}(\mathcal{A})$.*

The proof of this theorem will be given in the rest of this section. Again, we will apply our decomposition Theorem 9.8 for TPDA.

Lemma 9.19. *Let Σ be an alphabet, $m, k \in \mathbb{N}$, $h : \mathcal{R}_{m,k} \to \Sigma$ a renaming, and $\varphi \in \mathbf{MSO}_{\mathbb{L}}(\tilde{\mathcal{R}}_{m,k})$ a sentence. Then, there exists a sentence $\psi \in \mathbf{TML}(\Sigma)$ such that $\mathcal{L}(\psi) = h(\mathcal{L}(\varphi) \cap \mathcal{T}_{m,k})$.*

Proof. For the proof, we follow a similar approach as in the proof of Lemma 6.10. Let $\Gamma = \mathcal{R}_{m,k}$, $\overline{X} = \{X_\gamma \in V_2 \mid \gamma \in \Gamma\}$ be a set of pairwise distinct variables not appearing in φ, and $\overline{D} = \{D_i \in \mathcal{D} \mid 1 \leq i < m\}$ be a set of pairwise distinct relative distance variables. Using the $\overline{X}$-variables, we will describe the renaming

h (i.e., we store in these second-order variables the positions of the letters of the extended alphabet Γ before the renaming). We will use the $\overline{D}$-variables as well as the matching variable μ to describe the timed language $\mathcal{T}_{m,k}$. Moreover, we transform φ to a $\mathbf{tMSO}(\Sigma)$-formula.

Let $\varphi^* \in \mathbf{tMSO}(\Sigma)$ be the formula obtained from φ by the following transformations.

- Every subformula $\mathbb{L}(x,y)$ of φ with $x, y \in V_1$ is replaced by $\mu(x,y)$.
- Every subformula $P_\gamma(x)$ of φ with $x \in V_1$ and $\gamma \in \Gamma$ is replaced by the formula $P_{h(\gamma)}(x) \wedge X_\gamma(x)$.

The formula

$$\textsc{Partition} = \Big(\forall x. \Big[\bigvee_{\gamma\in\Gamma}\Big(X_\gamma(\gamma) \wedge \bigwedge_{\gamma'\neq\gamma} \neg X_{\gamma'}(x)\Big)\Big]\Big)$$

demands that values of $\overline{X}$-variables form a partition of the domain. The formula

$$\textsc{Renaming} = \forall x. \Big(\bigvee_{\gamma\in\Gamma}\big(X_\gamma(\gamma) \to P_{h(\gamma)}(x)\big)\Big)$$

correlates values of $\overline{X}$-variables with an input word. It remains to handle $\mathcal{T}_{m,k}$. For a letter $\gamma = (a, (g^1, ..., g^m), (r^1, ..., r^m), s, \delta) \in \Gamma$ with $a \in \Sigma$, $g^1, ..., g^m, s \in \mathbb{P}(k)$, $r^1, ..., r^m \in \{0,1\}$ and $\delta \in \{\downarrow, \#, \uparrow\}$, let $g^j(\gamma) = g^j$ $(1 \le j \le m)$, $s(\gamma) = s$, $r^j(\gamma) = r^j$ $(1 \le j < m)$ and $\delta(\gamma) = \delta$.

For $I, I' \in \mathcal{I}$, let $I \ominus I' = (I - I') \cap [0, \infty)$. Clearly, $(I \ominus I') \in \mathcal{I}$. For $j \in \{1, ..., m\}$ and $x \in V_1$, let $D_j^1(x) = D_j(x)$ and $D_j^0(x) = \neg D_j(x)$. Consider the $\mathbf{tMSO}(\Sigma)$-formula

$$\begin{aligned}
\xi =& \forall x. \Big(\bigwedge_{\gamma\in\Gamma} X_\gamma(x) \to \bigwedge\nolimits_{j=1}^{m} (\mathrm{d}^{g^j(\gamma)}(D_j, x) \wedge D_j^{r^j(\gamma)}(x))\Big) \wedge \\
& \forall x. \Big(\bigwedge_{\substack{\gamma\in\Gamma,\\ \delta(\gamma)=\#}} X_\gamma(x) \to (\neg\exists y.(\mu(x,y) \vee \mu(y,x)))\Big) \wedge \\
& \forall x. \Big(\bigwedge_{\substack{\gamma\in\Gamma,\\ \delta(\gamma)=\uparrow}} X_\gamma(x) \to \exists y. \bigvee_{\substack{\gamma'\in\Gamma,\\ \delta(\gamma')=\downarrow}} (X_{\gamma'}(y) \wedge \mu^{s(\gamma)\ominus s(\gamma')}(y,x))\Big)
\end{aligned}$$

which takes care of matchings and time. Let $(\gamma_i)_{i\in\{1,...,|\Gamma|\}}$ be an enumeration of Γ. Then, we define the desired sentence ψ as

$$\psi = \exists^{\text{match}}\mu.\exists D_1. \ldots \exists D_m.\exists X_1. \ldots \exists X_{|\Gamma|}.(\varphi^* \wedge \textsc{Partition} \wedge \textsc{Renaming} \wedge \xi).$$

We show that $\mathcal{L}(\psi) = h(\mathcal{L}(\varphi) \cap \mathcal{T}_{m,k})$.

Again, for the sake of simplicity, let $\mathcal{T} = \mathcal{T}_{m,k}$. First, we show that $\mathcal{L}(\psi) \subseteq h(\mathcal{L}(\varphi) \cap \mathcal{T})$. Let $w = (a_1, t_1)...(a_n, t_n) \in \mathcal{L}(\psi)$ and σ be a fixed $(w, \mathcal{U})$-assignment. Then, there exist sets $J_1, ..., J_m, K_1, ..., K_{|\Gamma|} \subseteq \mathrm{dom}(w)$ and a matching relation $M \subseteq (\mathrm{dom}(w))^2$ such that:

(i) the sets $K_1, ..., K_{|\Gamma|}$ form a partition of $\mathrm{dom}(w)$;
(ii) for all $i \in \mathrm{dom}(w)$: whenever $i \in K_v$ for some $v \in \{1, ..., |\Gamma|\}$, we have $a_i = h(\gamma_v)$;
(iii) $(w, \sigma[X_1/K_1, ..., X_{|\Gamma|}/K_{|\Gamma|}, \mu/M]) \models \varphi^*$;
(iv) $(w, \sigma[D_1/J_1, ..., D_m/J_m, X_1/K_1, ..., K_{|\Gamma|}/J_{|\Gamma|}, \mu/M]) \models \xi$; moreover, note that the sets $J_1, ..., J_m$ are uniquely determined by the sets $K_1, ..., K_{|\Gamma|}$.

For each $i \in \{1, ..., n\}$, let $\pi_i = \gamma_j$ for $j \in \{1, ..., |\Gamma|\}$ such that $i \in K_j$ (here we take into account (i)). Then, by (iii), we have $\pi_1 ... \pi_n \in \mathcal{L}(\varphi)$. Let $u = (\pi_1, t_1)...(\pi_n, t_n)$. Then, by (iv), $u \in \mathcal{T}$ and, by (ii), $h(u) = w$. Hence, $w \in h(\mathcal{L}(\varphi) \cap \mathcal{T})$.

Second, we show that $h(\mathcal{L}(\varphi) \cap \mathcal{T}) \subseteq \mathcal{L}(\psi)$. Assume that $w = (a_1, t_1)...(a_n, t_n) \in h(\mathcal{L}(\varphi) \cap \mathcal{T})$. Then, there exist $\pi_1, ..., \pi_n \in \Gamma$ such that:

(a) $\pi_1 ... \pi_n \in \mathcal{L}(\varphi)$;
(b) $v := (\pi_1, t_1)...(\pi_n, t_n) \in \mathcal{T}$;
(c) $h(\pi_i) = a_i$ for all $i \in \{1, ..., n\}$.

Assume that, for all $i \in \{1, ..., n\}$, $\pi_i = (a_i, (g_i^1, ..., g_i^m), (r_i^1, ..., r_i^m), s_i, \delta_i)$. For each $j \in \{1, ..., m\}$, let $J_j = \{i \in \{1, ..., n\} \mid r_i^j = 1\}$. For each $v \in \{1, ..., |\Gamma|\}$, let $K_v = \{i \in \{1, ..., n\} \mid \pi_i = \gamma_v\}$. Then, clearly, the sets $K_1, ..., K_{|\Gamma|}$ form a partition of $\mathrm{dom}(w)$. We define a relation $M \subseteq (\mathrm{dom}(w))^2$ as follows:

$$M = \{(i, i') \mid i < i', \delta_i = \downarrow, \delta_{i'} = \uparrow \text{ and } \textsc{Mask}(\pi_{i+1} ... \pi_{i'-1}) \in \mathbb{L}\}.$$

By (b), M is a matching relation. Note also that M is uniquely determined by the sets $K_1, ..., K_{|\Gamma|}$.

Let σ be a fixed $(w, \mathcal{U})$-assignment. Then, the following holds.

- Since the sets $K_1, ..., K_{|\Gamma|}$ form a partition of $\mathrm{dom}(w)$, we have $(w, \sigma[X_1/K_1, ..., X_{|\Gamma|}/K_{|\Gamma|}]) \models \textsc{Partition}$.
- By (c), we have $(w, \sigma[X_1/K_1, ..., X_{|\Gamma|}/K_{|\Gamma|}]) \models \textsc{Renaming}$.
- Using (a), it is not difficult to check that $(w, \sigma[X_1/K_1, ..., X_{|\Gamma|}/K_{|\Gamma|}, \mu/M]) \models \varphi^*$.
- By (b): $(w, \sigma[D_1/J_1, ..., D_m/J_m, X_1/K_1, ..., X_{|\Gamma|}/K_{|\Gamma|}, \mu/M]) \models \xi$.

Let $\chi = \varphi^* \wedge \textsc{Partition} \wedge \textsc{Renaming} \wedge \xi$. Then,

$$(w, \sigma[D_1/J_1, ..., D_m/J_m, X_1/K_1, ..., X_{|\Gamma|}/K_{|\Gamma|}, \mu/M]) \models \chi$$

and hence $w \in \mathcal{L}(\psi)$. This finishes the proof of this theorem. □

Proof of Theorem 9.18. Immediate by Theorems 9.8 and 9.7 and Lemma 9.19. □

Remark 9.20. *Alternatively, Theorem 9.18 can be proved by a direct translation of $\mathcal{A}$ into ψ. However, by using Theorem 9.8, it suffices to describe a simpler timed language $\mathcal{T}_{m,k}$ and a projection h to adopt the logical description of a visibly pushdown language of [5]. In particular, here we do not have to describe some technical details like initial, final states as well as concatenations of transitions.*

Proof of Theorem 9.4. Immediate by Theorems 9.15 and 9.18. □

Chapter 10

Conclusion and future work

In this thesis, we studied various models of multi-weighted automata and the model of pushdown timed automata from the automata-theoretic point of view. Our main results concern logical characterization of these compound automata models. From the theoretical point of view, these results show the robustness of the automata-theoretic approach. From the practical point of view, the developed logical formalisms with an effective translation process from formulas into automata could be used as a tool for the specification and verification of compound properties of systems. In addition, we studied several algorithmic problems for multi-weighted automata. In particular, we showed that, despite the loss of distributivity, the behavior of ratio automata on finite words can be evaluated in polynomial time. We also investigated the optimal weight problem for multi-weighted Büchi automata with discounting. The methods, ideas and auxiliary results presented in this thesis could be of independent interest and could be helpful for further developments in the theory of complex automata models combining time, costs, stacks, data etc. In particular, our Nivat-like results provide a way to separate quantitative components of models from their qualitative parts and could be applied to the extension of results, known for the qualitative setting, to the quantitative setting.

Now we briefly summarize the contents of this thesis and mention some directions for the future work.

In *Chapter 2*, we introduced a general algebraic model of *multi-weighted automata* which extends single-weighted automata over semirings [36] and valuation monoids [40]. We also studied some algorithmic properties of multi-weighted automata with the reward-cost ratio measure. As we already mentioned, we presented a polynomial time algorithm for the evaluation problem for ratio automata. It could be interesting to investigate whether the behavior of multi-weighted automata with discounting (cf. Example 2.5) or averaging (cf. Example 2.6) on a finite word can be also computed in polynomial time. The exploration of further decision problems for multi-weighted automata (e.g., the equivalence problem or the quantitative language inclusion problem [27]) is also encouraged. In particular, since the ratio automata of Example 2.3 contain the semiring-weighted automata over the tropical semiring, the equivalence problem for these automata is undecidable [63]. However, one could investigate whether the result of Hashiguchi et al. [60] for finitely ambiguous automata also holds in the multi-weighted setting (cf. also [62]).

In *Chapter 3*, we extended the use of the semiring-weighted logic of Droste and Gastin [34] to the multi-weighted setting. We introduced a multi-weighted MSO logic on finite words and characterized the class of behaviors of multi-weighted automata by means of a syntactically restricted fragment of this logic. The logical fragment we propose here could be of independent interest for various weighted settings, since it permits constant-preserving translations of formulas into automata. For a subclass of evaluators which contains all semirings as well as various multi-weighted settings known from literature, we could enlarge the fragment of multi-weighted MSO logic which is expressively equivalent to multi-weighted automata; the same fragment was used in [34] for a logical characterization of semiring-weighted automata. This shows the robustness of our multi-weighted approach. Further research in this topic should compare the complexity of the translation processes from single- and multi-weighted MSO formulas into automata. Moreover, it could be also interesting to extend the results of Bollig, Gastin, Monmege and Zeitoun [17, 18] about the equivalence of weighted pebble automata and weighted transitive closure logic to the multi-weighted setting as well as the Kleene-Schützenberger theorem for weighted pebble automata [58].

In *Chapter 4*, we lifted the classical Kleene-Schützenberger theorem [81] to the multi-weighted setting. In addition, we showed that the basis of a multi-weighted automaton can be characterized by a single-weighted automaton over the semiring of natural numbers. We also showed that, for the classes of idempotent evaluators and non-idempotent infinite semirings, the setting of natural numbers in this characterization is irrelevant and can be replaced by classical finite-state automata, i.e., non-idempotent infinite semifields are closely related to idempotent evaluators. It could be also interesting to investigate other classes of non-idempotent evaluators which are also closely related to idempotent evaluators.

In *Chapter 5*, we investigated multi-weighted Büchi automata on infinite words. Following the ideas of a Nivat theorem of [38] for semirings, we proved a Nivat theorem for multi-weighted automata. Using operations like renamings and intersections, we could characterize recognizable and unambiguously recognizable quantitative ω-languages by means of recognizable qualitative ω-languages. As an application of this theorem, we showed that multi-weighted ω-automata with Büchi and Muller acceptance conditions are expressively equivalent. Moreover, we considered an algorithm for the optimal weight problem for multi-weighted Büchi automata with discounting. The further research may investigate the complexity of this algorithm. Moreover, multi-weighted Büchi automata with various objectives should be further explored with respect to decidability and complexity results.

In *Chapter 6*, we introduced a concept of weight assignment logic on ω-words. We showed that this logic is expressively equivalent to multi-weighted Büchi automata. Using our Nivat theorem for multi-weighted Büchi automata, we could deduce the proof of this result from the classical Büchi-Elgot theorem [25, 50]. Note that we considered formulas with a weighted existential prefix where, in the scope of this existential prefix, we can use the conjunction-like operators without any restrictions. If we disallow the use of weighted existential quantifiers, then we obtain a logic which is expressively equivalent to unambiguous multi-weighted automata; these automata could have better decidability and complexity properties than their nondeterministic extensions. Following a similar approach, we can also define a weight assignment logic on finite words. Further research may investigate multi-weighted extensions of temporal logics like LTL and CTL, for instance, in the context of the model-checking problem.

In *Chapter 7*, we developed a general framework for multi-weighted timed automata on finite words and gave their Nivat-like characterization. We had to overcome the difficulty that unambiguous timed automata are weaker than nondeterministic timed automata. Based on this result, we illustrated the following difference between the weighted and weighted timed settings: whereas recognizable quantitative languages are exactly the renamings of deterministically recognizable quantitative languages, this result does not hold true in the context of timed automata. In our Nivat theorem for multi-weighted timed automata, we showed that, in general, if in a Nivat decomposition of the form $h((\mathrm{val}^{\mathbb{T}} \circ g) \cap \mathcal{L})$ we take an intersection with a nondeterministically recognizable timed language $\mathcal{L}$, then we can obtain an unrecognizable quantitative timed language. Then, it could be interesting to study which extension of multi-weighted timed automata can be characterized by such decompositions. It seems likely that the results of this chapter can be easily extended to the setting of infinite timed words. Multi-weighted timed automata should be further studied in the context of decidability and complexity.

In *Chapter 8*, we characterized multi-weighted timed automata by means of a timed extension of weight assignment logic of Chapter 6 based on Wilke's relative distance logic [84]. An open question is whether it is possible to extend the results of [17, 18] to the timed setting. Moreover, one could try to establish a Kleene-Schützenberger theorem for multi-weighted timed automata. For the special case of semirings, based on the approach of [24], such a characterization was presented in [43].

Finally, in *Chapter 9*, we introduced a timed matching logic and showed that this logic is equally expressive as timed pushdown automata of [1]. Hence, the satisfiability problem for our timed matching logic is decidable. When proving our main result, we showed a Nivat-like decomposition theorem for timed pushdown automata. This theorem seems to be the first algebraic characterization of timed pushdown languages. Based on the ideas presented in [34, 42, 71, 78] and the ideas of this thesis, our ongoing research concerns a logical characterization for *weighted timed pushdown automata* [2]. It could be also interesting to prove a version of the Chomsky-Schützenberger theorem for timed pushdown automata (cf. [48]). One could also investigate such an extension of timed pushdown automata where each edge permits to push or pop several stack elements. Our conjecture is that this extended model is more expressive than timed pushdown automata as considered in this chapter.

Bibliography

[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., Fix, L., Henzinger, T.: Event-clock automata: a determinizable class of timed automata. Theoretical Computer Science 211(1-2), 253–273 (1999).

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

[6] Alur, R., Triverdi, A.: Relating average and discounted costs for quantitative analysis of timed systems. In: EMSOFT 2011, pp. 165–174. IEEE (2011).

[7] Alur, R., La Torre, S., Pappas, G.J.: Optimal paths in weighted timed automata. In: HSCC 2001. LNCS, vol. 2034, pp. 49–62. Springer (2001).

[8] Andersson, D.: Improved combinatorial algorithms for discounted payoff games. Master's thesis, Uppsala University, Department of Information Technology (2006).

[9] Baier, C., Größer, M.: Recognizing omega-regular languages with probabilistic automata. In: LICS 2005, pp. 137–146. IEEE (2005).

[10] Bauer, S., Juhl, L., Larsen, K., Legay, A., Srba, J.: A logic for accumulated-weight reasoning on multiweighted modal automata. In: TASE 2012, pp. 77–84. IEEE (2012).

[11] Behrmann, G., Fehnker, A., Hune, T., Larsen, K.G., Petterson, P., Romijn, J., Vaandrager, F.: Minimum-cost reachability for priced timed automata. In: HSCC 2001. LNCS, vol. 2034, pp. 147–161. Springer (2001).

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

[13] Berstel, J., Reutenauer, C.: Rational Series and Their Languages. EATCS Monographs on Theoretical Computer Science, vol. 12. Springer (1988).

[14] Bloem, R., Greimel, K., Henzinger, T.A., Jobstmann, B.: Synthesizing robust systems. In: FMCAD 2009, pp. 85–92. IEEE (2009).

[15] Boker, U., Henzinger, T.: Determinizing discounted-sum automata. In: CSL 2011. LIPIcs, vol. 12, pp. 82–96. Schloss Dagstuhl (2011).

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

[17] Bollig, B., Gastin, P., Monmege, B., Zeitoun, M.: Pebble weighted automata and transitive closure logics. In: ICALP 2010. LNCS, vol. 6199, pp. 587–598. Springer (2010).

[18] Bollig, B., Gastin, P., Monmege, B., Zeitoun, M.: Logical characterization of weighted pebble walking automata. In: CSL/LICS'14. ACM Press (2014).

[19] 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).

[20] Bouyer, P.: A logical characterization of data languages. Inf. Process. Lett. 84(2), 75–85 (2002).

[21] Bouyer, P., Brinksma, E., Larsen, K.G.: Staying alive as cheaply as possible. In: HSCC 2004. LNCS, vol. 2993, pp. 203–218. Springer (2004).

[22] Bouyer, P., Brinksma, E., Larsen, K.G.: Optimal infinite scheduling for multi-priced timed automata. Formal Methods in System Design 32, 3–23 (2008).

[23] Bouyer, P., Fahrenberg, U., Larsen, K.G., Markey, N., Srba, J.: Infinite runs in weighted timed automata with energy constraints. In: FORMATS 2008. LNCS, vol. 5215, pp. 33-47. Springer, Heidelberg (2008).

[24] Bouyer, P., Petit, A.: A Kleene/Büchi-like theorem for clock languages. Journal of Automata, Languages and Combinatorics 7(2), 167–186 (2001).

[25] Büchi, J.R.: Weak second-order arithmetic and finite automata. Z. Math. Logik und Grundl. Math. 6, 66–92 (1960).

[26] Carton, O., Michel, M.: Unambiguous Büchi automata. In: LATIN 2000. LNCS, vol. 1776, pp. 407–416. Springer (2000).

[27] Chatterjee, K., Doyen, L., Henzinger, T.A.: Quantitative languages. In: CSL 2008. LNCS, vol. 5213, pp. 385–400. Springer (2008).

[28] Chatterjee, K., Doyen, L., Henzinger, T.A.: Expressiveness and closure properties for quantitative languages. Logical Methods in Comp. Sci. 6(3) (2010).

[29] Culik, K., Kari, J.: Image compression using weighted finite automata. Comput. Graphics 17, 305–313 (1993).

[30] D'Souza, D.: A logical characterization of event clock automata. Int. Journal of Foundations of Computer Science 14(4), 625–639. World Scientific (2003).

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

[32] De Alfaro, L., Henzinger, T.A., Majumdar, R.: Discounting the future in systems theory. In: ICALP 2003. LNCS, vol. 2719, pp. 1022–1037. Springer (2003).

[33] Droste, M., Gastin, P.: Weighted automata and weighted logics. In: ICALP 2005. LNCS, vol. 3580, pp. 513–525. Springer (2005).

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

[35] Droste, M., Gastin, P.: Weighted automata and weighted logics. In: Droste, M., Kuich, W., Vogler, H. (eds.) [36], chapter 5.

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

[37] Droste, M., Kuske, D.: Skew and infinitary formal power series. In: ICALP 2003. LNCS, vol. 2719, pp. 426–438. Springer (2003).

[38] Droste, M., Kuske, D.: Weighted automata. In: Pin, J.-E. (ed.) Handbook: "Automata: from Mathematics to Applications", European Mathematical Society, to appear.

[39] Droste, M., Meinecke, M.: Weighted automata and regular expressions over valuation monoids. Int. J. Found. Comput. Sci. 22(8), 1829–1844 (2011).

[40] Droste, M., Meinecke, I.: Weighted automata and weighted MSO logics for average and long-time behaviors. Inf. Comput. 220-221, 44–59 (2012).

[41] Droste, M., Perevoshchikov, V.: Multi-weighted automata and MSO logic. In: CSR 2013. LNCS, vol. 7913, pp. 418–430. Springer (2013).

[42] Droste, M., Perevoshchikov, V.: A Nivat theorem for weighted timed automata and relative distance logic. In: ICALP 2014. LNCS, vol. 8573, pp. 171–182. Springer (2009).

[43] Droste, M., Quaas, K.: A Kleene-Schützenberger theorem for weighted timed automata. Theor. Comput. Sci. 412(12-14), 1140–1153 (2011).

[44] Droste, M., Rahonis, G.: Weighted automata and weighted logics on infinite words. In: DLT 2006. LNCS, vol. 4036, pp. 49–58. Springer (2006).

[45] Droste, M., Rahonis, G.: Weighted automata and weighted logics with discounting. Theoretical Computer Science 410(37), 3481–3494 (2009).

[46] Droste, M., Sakarovitch, J., Vogler, H.: Weighted automata with discounting. Inf. Process. Lett. 108(1), 23–28 (2008).

[47] Droste, M., Vogler, H.: Weighted automata and multi-valued logics over arbitrary bounded lattices. Theoret. Comp. Science 418, 14–36 (2012).

[48] Droste, M., Vogler, H.: The Chomsky-Schützenberger theorem for quantitative context-free languages. In: DLT 2012. LNCS, vol. 7907, pp. 203–214. Springer (2013).

[49] Eilenberg, S.: Automata, Languages and Machines, volume A. Academic Press, New York (1974).

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

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

[52] Fahrenberg, U., Juhl, L., Larsen, K.G., Srba, J.: Energy games in multiweighted automata. In: ICTAC 2011. LNCS, vol. 6916, pp. 95–115. Springer (2011).

[53] Fahrenberg, U., Larsen, K.G.: Discount-optimal infinite runs in priced timed automata. Electr. Notes Theor. Comput. Sci. 239, 179–191 (2009).

[54] Fahrenberg, U., Larsen, K.G.: Discounting in time. Electr. Notes Theor. Comput. Sci. 253, 25–31 (2009).

[55] Fahrenberg, U., Larsen, K.G., Thrane, C.R.: Model-based verification and analysis for real-time systems. In: Software and Systems Safety - Specification and Verification. NATO Science for Peace and Security Series - D: Information and Communication Securita, vol. 30, pp. 231–259. IOS Press (2011).

[56] Filiot, E., Gentilini, R., Raskin, J.-F.: Quantitative languages defined by functional automata. In: CONCUR 2012. LNCS, vol. 7454, pp. 132–146. Springer (2012).

[57] Fränzle, M., Swaminathan, M.: Revisiting decidability and optimum reachability for multi-priced timed automata. In: FORMATS 2009. LNCS, vol. 5813, pp. 149–163. Springer (2009).

[58] Gastin, P., Monmege, B.: Adding pebbles to weighted automata. In: CIAA 2012. LNCS, vol. 7382, pp. 28–51. Springer (2012).

[59] Giammarresi, D., Restivo, A., Seibert, S., Thomas, W.: Monadic second-order logic over pictures and recognizability by tiling systems. In: STACS 1994. LNCS, vol. 775, pp. 365–375. Springer (1994).

[60] Hashiguchi, K., Ishiguro, K., Jimbo, S.: Decidability of the equivalence problem for finitely ambiguous finance automata. Int. Journal of Algebra and Computation 12(3), 445–461 (2002).

[61] Kleene S.C.: Representation of events in nerve nets and finite automata. Automata Studies, 3–42. Princeton University Press (1956).

[62] Klimann, I., Lombardy, S., Mairesse, J., Prieur, C.: Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. Theor. Comput. Sci. 327(3), 349–373 (2004).

[63] Krob, D.: The equality problem for rational series with multiplicities in the tropical semiring is undecidable. International Journal of Algebra and Computation 4(3), 405–425 (1994).

[64] Kuich, W.: Semirings and formal power series: their relevance to formal languages and automata. In: Handbook of Formal Languages, vol. 1, pp. 609–677. Springer (1997).

[65] Kuich, W., Salomaa, A.: Semirings, Automata and Languages. EATCS Monographs on Theoretical Computer Science, vol. 5. Springer (1986).

[66] Kreutzer, S., Riveros, C.: Quantitative monadic second-order logic. In: LICS 2013, pp. 113–122. IEEE (2013).

[67] Kuske, D.: Schützenberger's theorem on formal power series follows from Kleene's theorem. Theor. Comput. Sci. 401(1-3), 243-248 (2008).

[68] Larsen, K.G., Behrmann, G., Brinksma, E., Fehnker, A., Hune, T., Pettersson, P., Romijn, J.: As cheap as possible: efficient cost-optimal reachability for priced timed automata. In: CAV 2001. LNCS, vol. 2102, pp. 493–505. Springer (2001).

[69] Larsen, K.G., Rasmussen, J.I.: Optimal conditional reachability for multi-priced timed automata. In: FOSSACS 2005. LNCS, vol. 3441, pp. 234–249. Springer (2005).

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

[71] Mathissen, Ch.: Weighted logics for nested words and algebraic formal power series. In: ICALP 2008. LNCS, vol. 5126, pp 221–232. Springer (2008).

[72] Megiddo, N.: Combinatorial optimization with rational objective functions. In: STOC 1978. ACM, pp. 1–12 (1978).

[73] Mohri, M.: Finite-state transducers in language and speech processing. Computational Linguistics 23, 269–311 (1997).

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

[75] Puri, A., Tripakis, S.: Algorithms for the multi-constrained routing problem. In: SWAT 2002. LNCS, vol. 2368, pp. 338–347. Springer (2002).

[76] Quaas, K.: On the supports of recognizable timed series. In: FORMATS 2009. LNCS, vol. 5813, pp. 243–257. Springer (2009).

[77] Quaas, K.: Kleene-Schützenberger and Büchi theorems for weighted timed automata. PhD thesis, Universität Leipzig (2010).

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

[79] Rasmussen, J., Larsen, K.G., Subramani, K.: Resource-optimal scheduling using priced timed automata. In: TACAS 2004. LNCS, vol. 2988, pp. 220–235. Springer (2004).

[80] Salomaa, A., Soittola, M.: Automata-Theoretic Aspects of Formal Power Series. Text and Monographs in Computer Science. Springer (1978).

[81] Schützenberger, M.P.: On the definition of a family of automata. Information and control 4, 245–270 (1961).

[82] Stüber, T., Vogler, H., Fülöp, Z.: Decomposition of weighted multioperator tree automata. Int. J. Foundations of Computer Sci. 20(2), 221–245 (2009).

[83] Thatcher, J.W., Wright, J.B.: Generalized finite automata theory with an application to a decision problem of second-order logic. Mathematical System Theory, 2:57–81 (1968).

[84] Wilke, T.: Automaten und Logiken zur Beschreibung zeitabhängiger Systeme. PhD thesis, Christian-Albrecht-Universität Kiel (1994).

[85] 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).

Index

List of Figures

List of Tables

Selbstständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

Leipzig, 11. Dezember 2014

Vitaly Perevoshchikov

Wissenschaftlicher Werdegang

Sep.1998 - Jun.2005	*Gymnasium 32 Kaliningrad, Russland*
Sep.2005 - Jun.2010	*Russische Staatliche Immanuel-Kant-Universität Kaliningrad* Studium der Angewandten Mathematik und Informatik Diplomarbeit "Über eine effiziente parallelisierte Implementierung der Pollard-Rho-Methode zur Berechnung des diskreten Logarithmus auf elliptischen Kurven" Diplom mit Auszeichnung
Sep.2010 - Sep.2012	*Universität Leipzig* Wissenschaftlicher Mitarbeiter und Doktorand am Institut für Informatik Abteilung Automaten und Sprachen Betreuer: Prof. Dr. M. Droste
Okt.2012 - Apr.2015	*Universität Leipzig* Doktorand am Institut für Informatik Abteilung Automaten und Sprachen Stipendiat des DFG Graduiertenkollegs "QuantLA" Betreuer: Prof. Dr. Manfred Droste

Publikationsliste

Begutachtete Veröffentlichungen

1. Weight assignment logic. Angenommen bei *International Conference on Developments in Language Theory (DLT) 2015*.
2. A logical characterization of timed pushdown languages (mit M. Droste). Angenommen bei *International Computer Science Symposium in Russia (CSR) 2015*.
3. A Nivat theorem for weighted timed automata and weighted relative distance logic (mit M. Droste). In: *International Colloquium on Automata, Languages and Programming (ICALP) 2014*. Lecture Notes in Computer Science, Volume 8573, S. 171–182. Springer (2014).
4. Multi-weighted automata and MSO logic (mit M. Droste). In: *International Computer Science Symposium in Russia (CSR) 2013*. Lecture Notes in Computer Science, Volume 7913, S. 418–430. Springer (2013).
5. On the problems of efficient realization of parallelized Pollard's rho-method (mit A. Gritsenko). In: *Proceedings of the Junior Scientist Conference 2010*, S. 95–96. Technische Universität Wien (2010).
6. On the efficient realization of an elliptic curve discrete logarithm problem solution (mit A. Gritsenko). In: *Scientific Conference at the Tomsk State University of Control Systems and Radioelectronics 2009*, S. 319–327 (2009) (auf Russisch).

Eingereichte Arbeiten

1. Logics for weighted timed pushdown automata (mit M. Droste). April 2015.
2. Multi-weighted automata and MSO logic (mit M. Droste). Eingereicht bei *Theory of Computing Systems (TOCS), Special Issue of CSR 2013*. September 2014 (Revision).
3. Adding operations to weighted MSO-logic. April 2014.

Liste der Vorträge

1. Decomposition of weighted timed automata. FFM 2015, Aachen, 26.02.15.
2. A logical characterization of timed pushdown languages. Post WATA 2014, Dresden, 26.11.14.
3. A Nivat theorem for weighted timed automata and weighted relative distance logic. ICALP 2014, Kopenhagen, 09.07.14.
4. Adding operations to weighted MSO-logic. YR-ICALP 2014, Kopenhagen, 09.07.14.
5. A Nivat theorem for weighted timed automata and weighted relative distance logic. WATA 2014, Leipzig, 07.05.14.
6. Multi-weighted automata and MSO logic. Dagstuhl Seminar 14041 "Quantitative Models: Expressiveness, Analysis, and New Applications", Schloss Dagstuhl, 28.01.14.
7. Multi-weighted automata and MSO logic. CSR 2013, Jekaterinburg, 29.06.13.
8. Büchi-type theorems for unambiguous, functional and multi-weighted automata. WATA 2012, Dresden, 02.06.12.
9. On the problems of efficient realization of parallelized Pollard's rho method. Junior Scientist Conference 2010, Wien, 08.04.10.