Quantitative vs. Weighted Automata

Udi Boker

Reichman University, Herzliya, Israel
udiboker@idc.ac.il
https://faculty.idc.ac.il/udiboker

Abstract. Weighted automata are widely researched, but with a variety of different semantics, which mostly fit into either the "quantitative view" or the "algebraic view". We argue that the two views result with incomparable automata families, each providing a different conceptual generalization of Boolean automata and having different natural extensions.

We propose to term the former "quantitative automata" and the latter "weighted automata".

In both views, transitions are labeled with weights and the value of a path of transitions is given by some value function on the traversed weights. However, the main conceptual difference is in the generalization of nondeterminism and its dual (universality, in alternating automata).

Quantitative automata keep the *preference* meaning of *choice* and *obligation* to nondeterminism and universality, interpreted as supremum and infimum, respectively, and accordingly restrict weights and value functions to the totally ordered domain of real numbers.

Weighted automata, on the other hand, generalize nondeterminism to an arbitrary *commutative operation* (of a semiring or valuation monoid), and generally have no interpretation of universality. The weights and value functions can be from arbitrary domains.

On several aspects the algebraic view generalizes the quantitative one, allowing for richer weight domains and interpretations of nondeterminism, whereas on different aspects the quantitative view is more general, having alternation, inherent compatibility with games and adequacy to approximations.

We argue that clarifying the conceptual difference between the two automata families can enlighten their possible future extensions.

Keywords: Quantitative automata · Weighted automata · Nondeterminism · Alternation · Games · Logic.

1 Introduction

A bit of history.

1959 *Nondeterministic automata* (on finite words) introduced by Michael Rabin and Dana Scott [65].

1961 *Weighted automata* (with integer weights on finite words) introduced by Marcel-Paul Schützenberger [67].

1962 *Automata on infinite words* introduced by Julius Richard Büchi [19].

1963 *Probabilistic automata* (on finite words) introduced by Michael Rabin [64].

1970s *Weighted automata over semirings* (on finite words) have evolved (e.g., [40, 66]).

1976 *Alternating automata* introduced by Ashok Chandra, Dexter Kozen and Larry Stockmeyer [22, 21].

2000s *Weighted automata over semirings on infinite words* have evolved (e.g., [30, 35, 39]).

2008 *Quantitative automata*, as we refer to them, introduced by Krishnendu Chatterjee, Laurent Doyen, and Thomas Henzinger [23–25].

2010 *Weighted automata over valuation monoids* introduced by Manfred Droste and Ingmar Meinecke [36, 37].

Along the years:

– The various automata types were generalized to operate not only on words, but also on trees, graphs, and other structures.
– Counter automata of various types have evolved, which are inherently different from quantitative and weighted automata in the sense that in counter automata the counter value along a run can allow or forbid certain transitions.
– Automata were shown to be closely related to other entities and especially to logic and games, in both the Boolean and weighted settings.
– Automata on infinite words proved very useful in verification and synthesis.
– Alternation was shown to be particularly related to logic and games, and also proved very useful in verification.

The relations between quantitative and weighted automata.

The main conceptual difference between quantitative and weighted automata is in the interpretation of nondeterminism and its dual (universality). Quantitative automata keep their *preference* meaning with *choice* and *obligation*, respectively, thus considering weights and value functions over the totally ordered domain of real numbers. Weighted automata, on the other hand, allow for an arbitrary commutative interpretation of nondeterminism and generally have no interpretation of universality.

Quantitative automata usually have no acceptance condition, which is generalized by the numerical value of the automaton on the input. Weighted automata generally do have a Boolean acceptance condition, and only the values of accepting runs are considered by the commutative operation.

The automata families are formally defined in Section 2 and their relations are illustrated in Fig. 1.

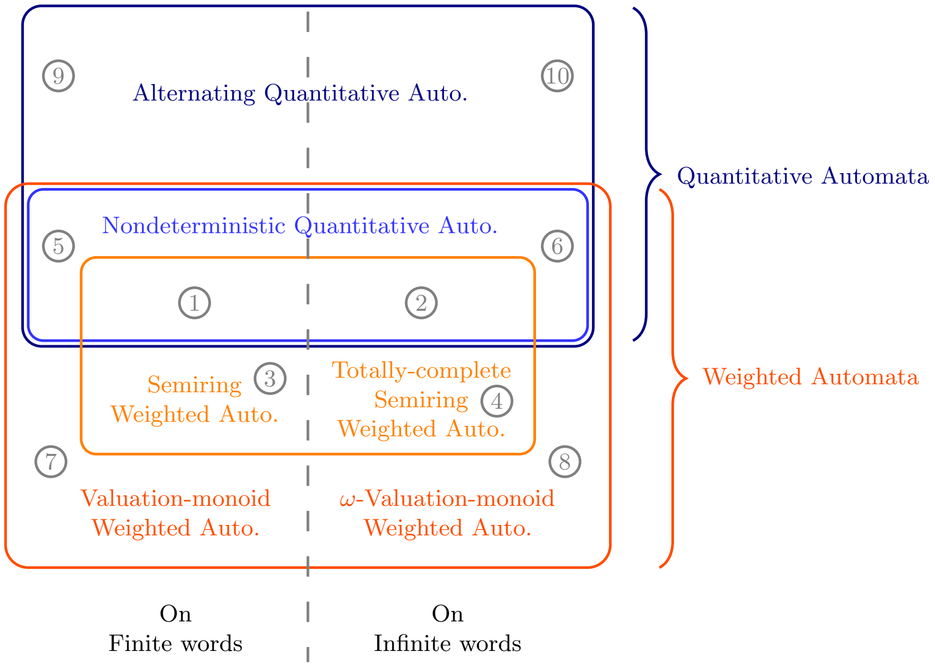
9
Alternating Quantitative Auto.
10
Quantitative Automata
Nondeterministic Quantitative Auto.
5
6
1
2
Semiring Weighted Auto.
3
Totally-complete Semiring Weighted Auto.
4
Weighted Automata
7
8
Valuation-monoid Weighted Auto.
ω-Valuation-monoid Weighted Auto.
On Finite words
On Infinite words

Fig. 1. Quantitative and weighted automata relations. Examples of some automata types that belong to the different (intersection of) automata families, as marked by the circled numbers, are given below.

For a quantitative automaton, one may first consider whether it is nondeterministic, universal, or alternating, and then consider its specific type, which is determined by its *value function*—a function Val from finite or infinite sequences of real numbers to a real number, defining how to value a path of transition weights. For example, an alternating discounted-sum automaton.

For a weighted automaton, one needs to first consider which subfamily it belongs to, depending on the algebraic restrictions on the function $\prod$ to value a path of transition weights and the function $\sum$ to aggregate the values of the accepting paths on an input word. The most common subfamily is of *semiring* weighted automata, operating on finite words and requiring $\prod$ and $\sum$ to correspond to the $\otimes$ and $\oplus$ operations of a semiring. For automata on infinite words, one needs to restrict the semiring to a *totally-complete semiring*, which properly extends $\prod$ and $\sum$ to infinite sequences and sets. In order to extend weighted automata to allow for the value functions that are used by quantitative automata, one needs to replace semirings with the more liberal *valuation monoid* for finite words and ω-valuation monoid for infinite words.

Example of automata types in the different (intersection of) automata families. The numbers below correspond to the circled numbers in Fig. 1. Some of the value functions, semirings, and valuation monoids mentioned in the examples are formally defined in Section 2. For ④ and ⑧, requiring a commutative operation on infinitely many elements that is different from supremum and infimum, we are not aware of many natural examples in the literature.

1,2. The intersection of [totally-complete] semiring weighted automata and nondeterministic quantitative automata restricts the former to interpret $\oplus$ as max (and $\sum$ as supremum) and the latter to use value functions (which take the role of $\prod$ in the semiring) that distribute over supremum. On finite words (① in Fig. 1), it includes various common automata types, among which are Boolean finite automata (NFAs) and Sum-automata, which are the same as tropical/arctic/max-plus automata. On infinite words (②), there are fewer examples, among which are Sup automata.

3,4. [Totally-complete] semiring weighted automata that are not quantitative automata interpret the $\oplus$ and $\sum$ operations differently from max/min and supremum/infimum. For finite words (③), there are many interesting examples, such as weighted automata over the log semiring.

5,6. Nondeterministic quantitative automata that are not [totally-complete] semiring weighted automata use value functions that do not distribute over max/supremum. There are many such examples, among which are Avg automata on finite words and LimInfAvg automata on infinite words.

7,8. [ω-]valuation-monoid weighted automata that are not [totally-complete] semiring weighted automata and not quantitative automata interpret the $\oplus$ and $\sum$ operations differently from max/min and supremum/infimum, and their value functions do not distribute over $\sum$. On finite words (⑦), one can take for example the domain of weights to be $\{0, 1\}$, use the value function Avg, and interpret $\oplus$ as multiplication.

9,10. All alternating quantitative automata that indeed use alternation (namely, automata that do not have only nondeterministic or universal transitions) are not [ω-]valuation-monoid weighted automata.

Terminology mismatch in the literature. The usage in the literature of "weighted automata" to different automata families can often be confusing. Some papers on "weighted automata" are specific to the tropical/arctic semiring (which is in ① of Fig. 1), for example [20, 6, 3]; some speak of quantitative automata, for example [24, 27]; and a significant segment of them refer to semiring weighted automata on finite words. There are also cases of using "quantitative automata" when referring to the algebraic view, for example [8].

Different conceptual views leading to different extensions.

As a result of the conceptual differences between the two automata families, they are naturally related to and extended with different notions and entities.

In particular, quantitative automata, which incorporate preference and the dual roles of nondeterminism and universality, naturally relate to two-player turn-based zero-sum games and to formal verification and synthesis, and allow for approximations with respect to standard distance functions over the real numbers.

Weighted automata, on the other hand, having an algebraic structure and a flexible interpretation of nondeterminism, are naturally related to various algebraic areas and have established equivalences with monadic second order logic.

In Section 3 we elaborate on several such notions and entities that are differently related to each of the automata families, and put forward possible extensions of the relations with the "less related" family.

We believe that understanding the conceptual difference between the two automata families (and making a terminological distinction between them) can help clarity, and furthermore enlighten the possible future extensions of each family, taking inspiration from natural extensions of the other family.

2 Definitions of Quantitative and Weighted Automata

We start with defining transition-labeled automata[1], and then extend them separately to quantitative automata and to weighted automata.

Remark 1. We describe automata on finite or infinite *words*, while both quantitative and weighted automata have orthogonal generalizations to more involved input structures, such as trees, graphs, and pictures. Likewise, we speak of automata with a *single weight* on each transition, while both automata families have natural extensions that allow for multiple weights on each transition.

Nondeterministic and alternating transition-labeled automata.

A *nondeterministic transition-labeled automaton* is a tuple $\mathcal{A} = (\Sigma, Q, I, \delta)$, where Σ is an alphabet set; Q is a finite nonempty set of states; $I \subseteq Q$ is a set of initial states; and $\delta\colon Q \times \Sigma \to 2^{W \times Q}$ is a transition function, where W is a set of labels.

A *transition* is a tuple $(q, \sigma, x, q') \in Q \times \Sigma \times W \times Q$, also written $q \xrightarrow{\sigma:x} q'$. (Note that there might be several transitions with different weights over the

[1] We define first an automaton without an acceptance-condition/value-function/semiring/valuation-monoid, but with initial state(s). Usually, the term 'automaton' refers to an entity with them, while 'semiautomaton' to an entity that also lacks initial state(s).

same letter between the same pair of states[2].) We write $\gamma(t) = x$ for the *weight* of a transition $t = (q, \sigma, x, q')$.

A *run* (or *path*) of the automaton on a word w is a sequence π of transitions that starts in an initial state and respects the transition function; that is $\pi = t_0, t_1, t_2, \ldots$, such that $t_0 = q \xrightarrow{w[0]:x} q'$ for a transition $t_0 \in \delta$ and $q \in I$, and for every $i > 0$, we have $t_{i-1} = q \xrightarrow{w[i-1]:x} q'$ and $t_i = q' \xrightarrow{w[i]:x} q''$, such that $t_{i-1}, t_i \in \delta$.

A nondeterministic automaton is *deterministic* if its set of initial states is a singleton and its transition function maps every state and letter to a singleton (a weight-state pair).

An *alternating transition-labeled automaton* is also a tuple $\mathcal{A} = (\Sigma, Q, \iota, \delta)$, where Σ and Q are as in the nondeterministic case, $\iota \in Q$ is an initial state[3], and $\delta \colon Q \times \Sigma \to \mathsf{B}^+(\mathbb{R} \times Q)$ is a transition function, where $\mathsf{B}^+(\mathbb{R} \times Q)$ is the set of positive Boolean formulas (*transition conditions*) over weight-state pairs.

A *transition* is as in the nondeterministic case a tuple $(q, \sigma, x, q') \in Q \times \Sigma \times \mathbb{R} \times Q$. (A transition condition generally yields many transitions.)

A run of the automaton on a word w is intuitively a play between Adam and Eve in a game denoted by $G_{\mathcal{A}}(w)$[4]. It starts in the initial state ι, and in each round, when the automaton is in state q and the next letter of w is σ, Eve resolves the nondeterminism (disjunctions) of the transition condition $\delta(q, \sigma)$ and Adam resolves its universality (conjunctions), yielding a transition $q \xrightarrow{\sigma:x} q'$. The output of a play is thus a path $\pi = t_0 t_1 t_2 \ldots$ of transitions.

A nondeterministic (resp. universal) automaton is a special case of an alternating automaton, in which all transition conditions are disjunctions (resp. conjunctions).

A nondeterministic/alternating automaton is *complete* if for every state $q \in Q$ and letter $\sigma \in \Sigma$, there is at least one transition $q \xrightarrow{\sigma:x} q'$ to some state q'.

2.1 Quantitative Automata

A quantitative automaton is defined with respect to a *value function* $\mathsf{Val} : \mathbb{R}^* \to \mathbb{R}$ or $\mathsf{Val} : \mathbb{R}^\omega \to \mathbb{R}$. It is then called a nondeterministic/alternating Val automaton (e.g., a nondeterministic discounted-sum automaton).

Examples of common value functions over sequences of real values. [5]

For finite sequences $v = v_0 v_1 \ldots v_{n-1}$:

[2] This extra flexibility of allowing for "parallel" transitions with different weights is often omitted, since it is redundant for some value functions while important for others.

[3] Nondeterministic automata are also often defined with a single initial state.

[4] An equivalent definition goes via trees instead of games.

[5] There are value functions that are more naturally defined over sequences of tuples of real values (see Remark 1), for example lexicographic-mean-payoff [9] and discounted-summation with multiple discount factors [13].

– $\mathsf{Sum}(v) = \sum_{i=0}^{n-1} v_i$ – $\mathsf{Avg}(v) = \frac{1}{n}\sum_{i=0}^{n-1} v_i$ – $\mathsf{Prod}(v) = \prod_{i=0}^{n-1} v_i$

For finite and infinite sequences $v = v_0 v_1 \ldots$:

– $\mathsf{Inf}(v) = \inf\{v_n \mid n \geq 0\}$ – $\mathsf{Sup}(v) = \sup\{v_n \mid n \geq 0\}$

– For a discount factor $\lambda \in \mathbb{Q} \cap (0,1)$, ${}_\lambda\mathsf{DSum}(v) = \sum_{i \geq 0} \lambda^i v_i$

For infinite sequences $v = v_0 v_1 \ldots$:

– $\mathsf{LimInf}(v) = \lim_{n\to\infty} \inf\{v_i \mid i \geq n\}$ – $\mathsf{LimSup}(v) = \lim_{n\to\infty} \sup\{v_i \mid i \geq n\}$

– $\mathsf{LimInfAvg}(v) = \mathsf{LimInf}\left(\frac{1}{n}\sum_{i=0}^{n-1} v_i\right)$ – $\mathsf{LimSupAvg}(v) = \mathsf{LimSup}\left(\frac{1}{n}\sum_{i=0}^{n-1} v_i\right)$

(LimInfAvg and LimSupAvg are also called $\underline{\mathsf{MeanPayoff}}$ and $\overline{\mathsf{MeanPayoff}}$.)

A *nondeterministic quantitative automaton* is a complete nondeterministic transition-labeled automaton with labels of real numbers[6] and some value function Val.

The value of a run π is $\mathsf{Val}(\gamma(\pi))$. The value of $\mathcal{A}$ on a word w is the supremum[7] of $\mathsf{Val}(\pi)$ over all runs π of $\mathcal{A}$ on w.

An *alternating quantitative automaton* is a complete alternating transition-labeled automaton with labels of real numbers and some value function Val.

The value of $\mathcal{A}$ on a word w is determined by the game $G_{\mathcal{A}}(w)$, which becomes a Val game: the value of a play (which is a path π of transitions) is $\mathsf{Val}(\gamma(\pi))$; Eve wants to maximize it and Adam wants to minimize it. When this game is determined, which is guaranteed for the considered value functions, the value of $\mathcal{A}$ on w is the value of $G_{\mathcal{A}}(w)$.

Two automata $\mathcal{A}$ and $\mathcal{A}'$ are *equivalent*, denoted by $\mathcal{A} \equiv \mathcal{A}'$, if they realize the same function[8].

2.2 Weighted Automata

A weighted automaton is defined with respect to a semiring or more generally with respect to an [ω-]valuation monoid[9].

A *semiring* is a structure $(D, \oplus, \otimes, \overline{0}, \overline{1})$, where $(D, \oplus, \overline{0})$ is a commutative monoid, $(D, \otimes, \overline{1})$ is a monoid, multiplication distributes over addition, and for every $x \in D$, $\overline{0} \otimes x = x \otimes \overline{0} = \overline{0}$.

6 Considering algorithmic aspects of quantitative automata, labels are usually rational numbers, concretely represented.

7 It is sometimes defined analogously with infimum instead of supremum. Considering alternating quantitative automata, infimum relates to universal transitions.

8 A function in this context is called in [23] a “quantitative language”.

9 Automata with multiple weights on transitions can be defined with respect to structure monoids [38].

A semiring is *complete* if $(D, \oplus, \overline{0})$ is a complete monoid (namely, equipped with a $\sum$ operation that properly extends $\oplus$ to infinite sets of elements), and it is *totally complete* if it is also equipped with a $\prod$ operation that properly extends $\otimes$ to infinite sequences of elements, while preserving distributivity over addition [39, 59].

Examples of common semirings.

- The Boolean $(\{0,1\}, \vee, \wedge, 0, 1)$
- The tropical (also known as min-plus) $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$
- The arctic (also known as max-plus) $(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$
- The natural numbers $(\mathbb{N}, +, \cdot, 0, 1)$ with the usual addition and multiplication
- The log semiring $(\mathbb{R} \cup \{-\infty, +\infty\}, \oplus, +, -\infty, 0)$ with $x \oplus y = \log(e^x + e^y)$

A *valuation monoid* is a tuple $(D, \oplus, \overline{0}, \mathsf{Val})$, where $(D, \oplus, \overline{0})$ is a commutative monoid and $\mathsf{Val} : D^* \to D$ is a function[10]. An *ω-valuation monoid* is defined analogously, while requiring that $\oplus$ is properly extended to $\sum$ over infinite sets of elements, and having $\mathsf{Val} : D^\omega \to D$.

Examples of common [ω-]valuation monoids.

- A semiring, taking its product $\otimes$ to be the valuation function Val.
- $(\mathbb{R} \cup \{-\infty\}, \sup, -\infty, \mathsf{Val})$, where Val is some value function as appears in Section 2.1 with a corresponding extension to $\mathbb{R} \cup \{-\infty\}$.

A *weighted automaton* on finite words (resp. infinite words) is a transition-labeled nondeterministic automaton together with a Boolean acceptance condition[11] and a semiring or a valuation-monoid (resp. a totally complete semiring or an ω-valuation-monoid).

A run is accepting if it satisfies the acceptance condition[12].

The value of an accepting run π is $\prod \gamma(\pi)$ with respect to a semiring and $\mathsf{Val}(\gamma(\pi))$ with respect to an [ω-]valuation monoid.

The value of $\mathcal{A}$ on a word w is the semiring's/[ω-]valuation-monoid's summation ($\sum$) of w's accepting-runs values or $\overline{0}$ if there are no accepting runs.

Two automata $\mathcal{A}$ and $\mathcal{A}'$ are *equivalent*, denoted by $\mathcal{A} \equiv \mathcal{A}'$, if they realize the same function[13].

[10] In [37], the original definition of a valuation monoid has additional restrictions that are loosened in [47].

[11] Semiring weighted automata are sometimes defined with an acceptance condition (e.g., [47]) and sometimes without it, while having instead labels on both transitions and states (e.g., [33]). However, considering infinite words or valuation monoids, weighted automata have an acceptance condition [37].

[12] For finite words, the acceptance condition is a set $F \subseteq Q$ of final states, and a run satisfies it if it ends in a final state. For infinite words there are many acceptance conditions, such as Büchi or Muller. (More details on the different acceptance conditions can be found, for example, in [12]).

[13] A function in this context is often viewed as a formal power series.

3 Related Notions and Entities

Automata are closely related to many notions and entities in computer science. We briefly look into how quantitative automata and weighted automata are, often differently, related to some of them, and how to possibly extend each such relation with respect to the "less related" automata family.

3.1 Alternation

Since alternating automata were introduced in [22, 21], they were extended to many models (e.g., [55, 31]) over various input structures (e.g., [62, 57, 58]), and shown to be closely related to logic and to games (e.g., [58, 74]), very useful in formal verification (e.g., [72, 31]), and in general to play a key role in automata theory.

Quantitative automata are naturally generalized from nondeterminism to alternation, having the dual roles of *choice* for nondeterminism and *obligation* for universality, and alternating quantitative automata are often more expressive than nondeterministic ones and allow for better closure properties [24]. (The title of [24] speaks of "weighted automata", but relates to quantitative automata.)

Considering weighted automata, on either a semirings or [ω-]valuation monoids, there is no natural interpretation of alternation, as nondeterminism is interpreted by a general commutative operation $\oplus$, which need not have a dual.

The $\otimes$ or Val functions that are used for valuing a path of transitions have an orthogonal role, and are not generally adequate for the 'universality role'. Yet, in some settings it is interesting to look into an interpretation of an alternating weighted automaton, in which $\otimes$ takes this role, as is done in [4] with respect to the tropical semiring and in [53] with respect to commutative semirings.

It may possibly be interesting to look into extensions of semirings and valuation monoids, as suggested in [53], that add another operator to take the role of universality.

It may also be interesting to look into restrictions of semirings and valuation monoids for which the $\oplus$ operation has a meaningful syntactic dual.

3.2 Games

Two-player turn-based *win-lose* games are closely related to logic and to *Boolean* automata, especially to alternating Boolean automata. For example, deciding the winner of an infinite game over an arena A is the same as deciding whether A, seen as a one-letter alternating automaton with a corresponding acceptance condition, is empty. Other examples are *good-for-games* automata [50, 15] (see Section 3.4), which are useful in solving games [50, 29], and the interplay between automata and games in formal verification and synthesis (see Section 3.6).

Analogously, two-player turn-based *zero-sum* games, which generalize win-lose games by having (possibly infinitely) many values to plays, and in which Eve wants to maximize the play's value and Adam wants to minimize it, are closely related to *quantitative* automata.

In particular, the above example of viewing a game as a special case of an alternating automaton over a singleton alphabet generalizes to the quantitative setting—the value of the game, which is the value that Eve can guarantee against any strategy of Adam, is the value of the automaton on the single input word. Along the same lines, the other examples above are also naturally generalized to the quantitative setting (e.g., [32, 49, 5, 45, 16])[14].

Weighted automata are connected to games via their connection to logics (see Section 3.3), which are connected to Ehrenfeucht-Fraïssé games. However, we are not aware of works that directly connect between general weighted automata and games.

3.3 Logic

Automata theory has evolved from logic and remained very related to it. In particular, Büchi, Elgot and Trakhtenbrot established the equivalence of (ω-regular complete) automata and monadic second order (MSO) logic (of order) on words [41, 71, 19], while a series of results provided the equivalence of counter-free (aperiodic) automata and both first order logic (FOL) and linear temporal logic (LTL) on finite and infinite words [68, 52, 61, 56, 69, 46, 70, 63] (see a detailed exploration of the latter in [48]).

The result on the equivalence of Boolean automata and MSO was extended by Manfred Droste and Paul Gastin to the equivalence of semiring weighted automata on finite words and a restricted version of a weighted MSO logic that they defined [33]. This result was then further extended to totally-complete semiring weighted automata on infinite words [39], to [ω-]valuation monoid weighted automata on finite and infinite words [37], and to various extensions of weighted automata on various input structures, each corresponding to a variant of weighted MSO. In [47], there is a unifying framework for the equivalence of weighted automata and weighted MSO on finite words.

The result on the equivalence of aperiodic Boolean automata and first-order logic was extended by Droste and Gastin to the equivalence of aperiodic polynomially ambiguous weighted automata on finite words and weighted FOL [34].

As for the connection of quantitative automata and logic, there are various extensions of temporal logic with value functions Val that are related to Val automata, for example [60, 10, 1, 17]. However, we are not aware of general equivalence theorems as in the case of weighted automata.

It may be interesting to look into adaptations of weighted MSO that are equivalent to nondeterministic and alternating quantitative automata, as well as on adaptations of weighted FOL that are equivalent to their aperiodic counterparts. Another interesting direction, in particular for formal verification (see Section 2.2), is to establish equivalence between quantitative automata and some weighted temporal logics.

[14] The "weighted automata" in the title of [49] refer to a variant of Sum automata, defined over infinite words with an acceptance condition on finite prefixes.

3.4 Between determinism and nondeterminism

In general, deterministic automata have better compositional properties than nondeterministic automata, making them better suited for applications such as synthesis and probabilistic model checking. Yet, deterministic automata are often exponentially bigger than equivalent nondeterministic automata and sometimes lack in expressive power.

This unpleasant trade-off between determinism and nondeterminism motivates formalisms that are in between them, aiming at enjoying, sometimes, the best of both worlds.

Dominant such formalisms are *unambiguity*, *determinism in the limit* (semi-determinism), *history determinism* [28], and *good for gameness* [50].

A Boolean automaton is unambiguous if there is at most one accepting run on each word; it is deterministic in the limit (for Büchi automata) if its continuation from every accepting state is deterministic; it is history deterministic if there is a strategy to resolve its nondeterminism by only considering the prefix of the word read so far (and getting an equivalent automaton); and it is good for games if its product with every game G whose winning condition is the automaton's language provides a game with the same winner as of G.

Unambiguity and determinism-in-the-limit are defined, as is, on weighted automata, based on their acceptance conditions. Observe that unambiguous weighted automata do not need the $\oplus$ operation, as there is at most one accepting run. Hence, the notion is relevant also to quantitative automata if adding an acceptance condition, which can also relate to a *threshold* (see Section 3.6 for threshold quantitative automata). For example, requiring at most one run on each word whose value is equal to or bigger than a threshold.

Unambiguity can also be generalized with respect to weighted automata (with no $\oplus$ operation) or quantitative automata (with acceptance conditions) by means of *functional automata* [43]—rather than having at most one accepting run on each word, all accepting runs on a word should have the same value.

History-determinism and good-for-gameness have natural generalizations to quantitative automata, due to the *choice* interpretation of nondeterminism. The definition of history determinism follows as is [28, 16], while good-for-gameness relates to zero-sum games rather than to win-lose games [16]. Interestingly, while history determinism and good for gameness are equivalent for Boolean automata [15], they are not equivalent for quantitative automata [16].

Though history-determinism and good-for-gameness look less natural for weighted automata, it might be interesting to analyze such notions. History determinism is technically defined also for a weighted automaton with an arbitrary $\oplus$ operation (requiring a strategy to generate for every word w a single run with the value of the automaton on w), though possibly not very meaningful for a general weighted automaton. As for good for gameness, it might be that certain types of games are adequate to an interesting product with weighted automata.

3.5 Approximations

Quantitative automata, having real values for words, naturally allow for approximations with respect to standard distance functions. Accordingly, there are many works on approximated solutions with quantitative automata (some of them have "weighted automata" in the title), either with respect to a specific distance function, such as difference (e.g., [11, 51, 44]) or ratio[15] (e.g., [6, 7]), or with respect to a general distance function d that respects the order on $\mathbb{R}$, namely having that for every $x \leq y \leq z \in \mathbb{R}$, we have $d(x,y) \leq d(x,z)$ and $d(y,z) \leq d(x,z)$ (e.g., [14]).

These approximated solutions provide a significant added value to the generalization of Boolean automata to quantitative automata, as often an exact solution is impossible or computationally very difficult.

Considering weighted automata, once the domain of values is arbitrary, there is a problem to consider meaningful distance functions. However, restricting the domain to $\mathbb{R}$, or to some other set with meaningful distance functions, allows for analogous approximated solutions.

3.6 Formal Verification and Synthesis

Verification (model checking) asks whether a given system satisfies a given specification, and synthesis asks to automatically generate a system that satisfies a given specification.

Verification and synthesis are traditionally Boolean, having a yes-no value to both the system properties (such as whether or not the system 'serves only coffee') and to the satisfaction level of the specification (for example, 'yes' if the system satisfies the specification of 'serving only coffee' or 'serving only tea').

This Boolean perspective falls short of many verification needs of contemporary systems, concerning performance, robustness, and resource-constraint requirements. One system is often preferred over another, even though they are both correct, since one is, for example, faster than the other, or, if they are both incorrect, one misbehaves less frequently than the other.

As a result, recent years have seen the emergence and rapid development of *quantitative formal verification and synthesis* in an attempt to cope with these needs (e.g., [2, 10, 32, 26, 42, 9, 18, 5, 51, 45]).

According to this approach, both the system properties and the satisfaction values are no longer Boolean. For example, a property of a system can be an 'average response time', and the system can get a satisfaction level of 0.7 to a specification that quantitatively combines requirements on the 'average response time' and the 'power consumption'.

Automata and game theory play a key role in verification and synthesis of reactive systems (see, e.g., [73, 54]) and both quantitative and weighted automata are valuable in generalizing them to the quantitative setting.

[15] As ratio does not satisfy the triangle inequality, it is formally not a distance function, and one may speak instead of $d(x,y) = |\log x - \log y|$.

Considering the generalization of the satisfaction level, quantitative automata are more natural, as satisfaction evaluates to a value from an ordered domain, and we want the system to get a value as high as possible. In this context, and others, it is also common to consider *threshold quantitative automata*, which return a yes-no answer for whether the value of the automaton on an input word is equal to or bigger than a threshold. This provides the flexibility of playing back and forth between Boolean and quantitative satisfaction values.

Also for synthesis, which is viewed as a two-player game between the environment, generating the inputs to the system, and the system, interactively responding to these inputs, quantitative automata are more natural (see Section 3.2).

As for generalization of system properties, both quantitative and weighted automata are suitable, as such properties might have very general aspects.

Acknowledgments

We thank Nathanaël Fijalkow for raising up the terminology mismatch in the literature, and Shaull Almagor and Karoliina Lehtinen for stimulating discussions on preliminary versions of the paper.

References

1. Almagor, S., Boker, U., Kupferman, O.: Discounting in LTL. In: Proc. of TACAS. LNCS, vol. 8413, pp. 424–439. Springer (2014)
2. Almagor, S., Boker, U., Kupferman, O.: Formalizing and reasoning about quality. Journal of the ACM **63**(3), 24:1–24:56 (2016)
3. Almagor, S., Boker, U., Kupferman, O.: What's decidable about weighted automata? In: Proc. of ATVA. LNCS, vol. 6996, pp. 482–491. Springer (2011)
4. Almagor, S., Kupferman, O.: Max and sum semantics for alternating weighted automata. In: Proceedings of ATVA. LNCS, vol. 6996, pp. 13–27. Springer (2011)
5. Almagor, S., Kupferman, O., Ringert, J.O., Velner, Y.: Quantitative assume guarantee synthesis. In: Proc. of CAV. pp. 353–374. Springer (2017)
6. Aminof, B., Kupferman, O., Lampert, R.: Reasoning about online algorithms with weighted automata. ACM Transactions on Algorithms **6**(2), 28:1–28:36 (2010)
7. Aminof, B., Kupferman, O., Lampert, R.: Rigorous approximated determinization of weighted automata. Theoretical Computer Science **480**, 104–117 (2013)
8. Babari, P.: Quantitative Automata and Logic for Pictures and Data Words. Ph.D. thesis, Leipzig University (2017)
9. Bloem, R., Chatterjee, K., Henzinger, T.A., Jobstmann, B.: Better quality in synthesis through quantitative objectives. In: Bouajjani, A., Maler, O. (eds.) Proc. of CAV. Lecture Notes in Computer Science, vol. 5643, pp. 140–156. Springer (2009)
10. Boker, U., Chatterjee, K., Henzinger, T.A., Kupferman, O.: Temporal specifications with accumulative values. ACM Trans. Comput. Log. **15**(4), 27:1–27:25 (2014)
11. Boker, U., Henzinger, T.A.: Exact and approximate determinization of discounted-sum automata. Logical Methods in Computer Science **10**(1), 1–33 (2014)
12. Boker, U.: Why these automata types? In: Proc. of LPAR. pp. 143–163 (2018)

13. Boker, U., Hefetz, G.: Discounted-sum automata with multiple discount factors. In: Proc. of CSL. LIPIcs, vol. 183, pp. 12:1–12:23 (2021)
14. Boker, U., Henzinger, T.A.: Approximate determinization of quantitative automata. In: Proc. of FSTTCS. LIPIcs, vol. 18, pp. 362–373 (2012)
15. Boker, U., Lehtinen, K.: Good for games automata: From nondeterminism to alternation. In: Fokkink, W.J., van Glabbeek, R. (eds.) Proc. of CONCUR. LIPIcs, vol. 140, pp. 19:1–19:16 (2019)
16. Boker, U., Lehtinen, K.: History determinism vs. good for gameness in quantitative automata. In: Proc. of FSTTCS. pp. 35:1–35:20 (2021)
17. Bouyer, P., Markey, N., Matteplackel, R.: Averaging in LTL. In: Proc. of CONCUR. pp. 266–280 (2014)
18. Brenguier, R., Clemente, L., Hunter, P., Pérez, G.A., Randour, M., Raskin, J.F., Sankur, O., Sassolas, M.: Non-zero sum games for reactive synthesis. In: Language and Automata Theory and Applications. pp. 3–23. Springer (2016)
19. Büchi, J.R.: On a decision method in restricted second order arithmetic. In: Proc. Int. Congress on Logic, Method, and Philosophy of Science. 1960. pp. 1–12. Stanford University Press (1962)
20. Buchsbaum, A.L., Giancarlo, R., Westbrook, J.: An approximate determinization algorithm for weighted finite-state automata. Algorithmica **30**(4), 503–526 (2001)
21. Chandra, A.K., Kozen, D.C., Stockmeyer, L.J.: Alternation. J. ACM **28**(1), 114–133 (Jan 1981)
22. Chandra, A.K., Stockmeyer, L.J.: Alternation. In: Proceedings of FOCS. pp. 98–108 (1976)
23. Chatterjee, K., Doyen, L., Henzinger, T.A.: Quantitative languages. In: Proc. of CSL, pp. 385–400. LNCS 5213, Springer (2008)
24. Chatterjee, K., Doyen, L., Henzinger, T.A.: Alternating weighted automata. In: Proceedings of FCT. pp. 3–13 (2009)
25. Chatterjee, K., Doyen, L., Henzinger, T.A.: Quantitative languages. ACM Trans. Comput. Log. **11**(4), 23:1–23:38 (2010)
26. Chatterjee, K., Henzinger, T.A., Jobstmann, B., Singh, R.: Quasy: Quantitative synthesis tool. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems. pp. 267–271. Springer (2011)
27. Chatterjee, K., Henzinger, T.A., Otop, J.: Nested weighted automata. ACM Trans. Comput. Log. **18**(4), 31:1–31:44 (2017)
28. Colcombet, T.: The theory of stabilisation monoids and regular cost functions. pp. 139–150 (2009)
29. Colcombet, T., Fijalkow, N.: Universal graphs and good for games automata: New tools for infinite duration games. In: Proc. of FOSSACS. Lecture Notes in Computer Science, vol. 11425, pp. 1–26. Springer (2019)
30. Culik, K., Karhumaki, J.: Finite automata computing real functions. SIAM J. Comput. **23**(4), 789–814 (1994)
31. Dickhfer, M., Wilke, T.: Timed alternating tree automata: the automata-theoretic solution to the TCTL model checking problem. In: Automata, Languages and Programming. LNCS, vol. 1644, pp. 281–290. Springer, Berlin (1999)
32. Doyen, L.: Games and automata: From Boolean to quantitative verification. Habilitation Thesis, École Normale Supérieure de Cachan (2011)
33. Droste, M., Gastin, P.: Weighted automata and weighted logics. Theoretical Computer Science **380**(1-2), 69–86 (2007)
34. Droste, M., Gastin, P.: Aperiodic weighted automata and weighted first-order logic. In: Proc. of MFCS. LIPIcs, vol. 138, pp. 76:1–76:15 (2019)

35. Droste, M., Kuske, D.: Skew and infinitary formal power series. In: Proc. of ICALP. Lecture Notes in Computer Science, vol. 2719, pp. 426–438. Springer (2003)
36. Droste, M., Meinecke, I.: Describing average- and longtime-behavior by weighted MSO logics. In: Proceedings of MFCS. pp. 537–548 (2010)
37. Droste, M., Meinecke, I.: Weighted automata and weighted MSO logics for average and long-time behaviors. Information and Computation **220-221**, 44–59 (2012)
38. Droste, M., Perevoshchikov, V.: Multi-weighted automata and MSO logic. In: Computer Science – Theory and Applications. pp. 418–430 (2013)
39. Droste, M., Rahonis, G.: Weighted automata and weighted logics on infinite words. In: Ibarra, O.H., Dang, Z. (eds.) Proc. of DLT. Lecture Notes in Computer Science, vol. 4036, pp. 49–58. Springer (2006)
40. Eilenberg, S.: Automata, Languages, and Machines. Academic Press, Inc., USA (1974)
41. Elgot, C.: Decision problems of finite-automata design and related arithmetics. Trans. Amer. Math. Soc. **98**, 21–51 (1961)
42. Faella, M., Legay, A., Stoelinga, M.: Model checking quantitative linear time logic. Electr. Notes Theor. Comput. Sci. **220**(3), 61–77 (2008)
43. Filiot, E., Gentilini, R., Raskin, J.: Quantitative languages defined by functional automata. Log. Methods Comput. Sci. **11**(3), 1–32 (2015)
44. Filiot, E., Jecker, I., Lhote, N., Pérez, G.A., Raskin, J.: On delay and regret determinization of max-plus automata. In: LICS. pp. 1–12 (2017)
45. Filiot, E., Löding, C., Winter, S.: Synthesis from weighted specifications with partial domains over finite words. In: Saxena, N., Simon, S. (eds.) FSTTCS. LIPIcs, vol. 182, pp. 46:1–46:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020)
46. Gabbay, D.M., Pnueli, A., Shelah, S., Stavi, J.: On the temporal analysis of fairness. In: Proc. of POPL. pp. 163–173 (1980)
47. Gastin, P., Monmege, B.: A unifying survey on weighted logics and weighted automata - core weighted logic: minimal and versatile specification of quantitative properties. Soft Comput. **22**(4), 1047–1065 (2018)
48. and Paul Gastin, V.D.: A combinatorial approach to the theory of omega-automata. Inf. Control. **48**(3), 261–283 (1981)
49. Halava, V., Harju, T., Niskanen, R., Potapov, I.: Weighted automata on infinite words in the context of attacker-defender games. In: Proc. of CIE. Lecture Notes in Computer Science, vol. 9136, pp. 206–215. Springer (2015)
50. Henzinger, T., Piterman, N.: Solving games without determinization. In: Proc. of CSLl. pp. 395–410 (2006)
51. Hunter, P., Pérez, G.A., Raskin, J.: Reactive synthesis without regret. Acta Informatica **54**(1), 3–39 (2017)
52. Kamp, J.: Tense Logic and the Theory of Order. Ph.D. thesis, UCLA (1968)
53. Kostolányi, P., Misún, F.: Alternating weighted automata over commutative semirings. Theor. Comput. Sci. **740**, 1–27 (2018)
54. Kupferman, O.: Automata theory and model checking. In: Handbook of Model Checking, pp. 107–151. Springer (2018)
55. Ladner, R., Lipton, R., Stockmeyer, L.: Alternating pushdown and stack automata. SIAM Journal on Computing **13**(1), 135–155 (1984)
56. Ladner, R.E.: Application of model theoretic games to discrete linear orders and finite automata. Inf. Control. **33**(4), 281–303 (1977)
57. Lindsay, P.A.: On alternating ω-automata. Theoretical Computer Science **43**, 107–116 (1988)

58. Loding, C., Thomas, W.: Alternating automata and logics over infinite words. In: Theoretical Computer Science: Exploring New Frontiers of Theoretical Informatics. pp. 521–535 (2000)
59. M. Droste, W.K., Vogler, H.: Handbook of Weighted Automata. Springer Publishing Company, Incorporated, 1st edn. (2009)
60. Mandrali, E.: Weighted LTL with discounting. In: Moreira, N., Reis, R. (eds.) Proc. of CIAA. pp. 353–360 (2012)
61. McNaughton, R., Papert, S.: Counter-free automata. M.I.T. Press (1971)
62. Muller, D., Schupp, P.: Alternating automata on infinite trees. In: Automata on Infinite Words. LNCS, vol. 192, pp. 100–107. Springer (1985)
63. Perrin, D., Pin, J.: First-order logic and star-free sets. J. Comput. Syst. Sci. **32**(3), 393–406 (1986)
64. Rabin, M.O.: Probabilistic automata. Information and Control **6**(3), 230–245 (1963)
65. Rabin, M.O., Scott, D.: Finite automata and their decision problems. IBM Journal of Research and Development **3**(2), 114–125 (1959)
66. Salomaa, A., Soittola, M.: Automata-Theoretic Aspects of Formal Power Series. Texts and Monographs in Computer Science, Springer (1978)
67. Schützenberger, M.P.: On the definition of a family of automata. Information and Control **4**(2), 245–270 (1961)
68. Schützenberger, M.P.: On finite monoids having only trivial subgroups. Inf. Control. **8**(2), 190–194 (1965)
69. Thomas, W.: Star-free regular sets of omega-sequences. Inf. Control. **42**(2), 148–156 (1979)
70. Thomas, W.: A combinatorial approach to the theory of omega-automata. Inf. Control. **48**(3), 261–283 (1981)
71. Trakhtenbrot, B.A.: Finite automata and logic of monadic predicates. Doklady Akademii Nauk SSSR. In Russian (1961)
72. Vardi, M.Y.: Alternating automata and program verification, pp. 471–485. Springer Berlin Heidelberg (1995)
73. Vardi, M.: Verification of concurrent programs: The automata-theoretic framework. pp. 167–176 (1987)
74. Wilke, T.: Alternating tree automata, parity games, and modal μ-calculus. Bulletin of the Belgian Mathematical Society Simon Stevin **8**(2), 359 (2001)