Discounted-Sum Automata – Reductions between the Universality Problems

Udi Boker
Reichman University, Herzliya, Israel
udiboker@runi.ac.il

A nondeterministic discounted-sum automaton (NDA) values runs by the discounted sum of their transition weights: in a λ-NDA, the weight in position i is divided by λ^i, for a rational discount factor $\lambda > 1$. The value of a finite or infinite word is the infimum over the values of all runs on that word. This formalism captures systems in which present events have greater impact than future ones, a standard paradigm in economics and other disciplines.

The universality problem of an NDA is often considered via its negation, the nonuniversality problem, which asks whether there exists some word with value above 0; and the exact-value problem asks whether some word has value exactly 0. Considering both finite and infinite words, together with strict and non-strict comparisons, yields six natural decision problems. None of them is currently known to be decidable or undecidable, and their mutual relationships have remained unclear.

We study the computational relationships among these six problems, and show that the exact-value problem on infinite words is at least as hard as the $\geq$-nonuniversality problem on infinite words, while the four other problems are interreducible: exact-value, $\geq$-nonuniversality, and $>$-nonuniversality on finite words, and $>$-nonuniversality on infinite words.

1 Introduction

Discounted summation is a standard paradigm in economics and various other disciplines, and consequently plays a prominent role in computational models such as Markov decision processes (e.g., [31, 33, 25]), reinforcement learning (e.g., [35, 5]), games (e.g., [37, 6, 2, 32]), and automata (e.g., [28, 21, 22, 23, 34]). It formalizes the intuition that immediate rewards outweigh delayed rewards, while distant costs matter less than immediate ones.

A nondeterministic discounted-sum automaton (NDA, formally defined in Section 2) has transition weights and a fixed discount factor $\lambda > 1$. The value of a (finite or infinite) run is the discounted summation of the weights along the transitions, where the weight in the ith position of the run is divided by λ^i. The value of a (finite or infinite) word is the infimum of the values of the automaton runs on it.

The nonemptiness, nonuniversality, and exact-value problems of an NDA $\mathscr{A}$ are central in the usage of NDAs for quantitative verification [23, 25, 24, 29, 18, 7, 19], and can be framed together as questions on the existence of a nonempty finite (subscript F) or infinite (subscript I) word w on which $\mathscr{A}$ has a value below 0, equal to 0, or above 0:

- $<$-$\mathsf{NonEmptiness}_{F/I}$: $\mathscr{A}(w) < 0$?
- $\leq$-$\mathsf{NonEmptiness}_{F/I}$: $\mathscr{A}(w) \leq 0$?
- $\mathsf{ExactValue}_{F/I}$: $\mathscr{A}(w) = 0$?
- $\geq$-$\mathsf{NonUniversality}_{F/I}$: $\mathscr{A}(w) \geq 0$?
- $>$-$\mathsf{NonUniversality}_{F/I}$: $\mathscr{A}(w) > 0$?

Submitted to:
Express/SOS 2026

In addition to the [in]equality type $(<, \leq, =, \geq, >)$ and word type (finite, infinite), one also considers the automaton type – whether it is deterministic or nondeterministic.

Since NDAs interpret nondeterminism as infimum, the nonemptiness problems are easy:

- $<$-$\mathsf{NonEmptiness}_I$ and $\leq$-$\mathsf{NonEmptiness}_I$: There is a polynomial solution, by reduction to one-player discounted-payoff games. See [23, Theorem 3] for the nonemptiness solution and [6] for the algorithm of solving discounted-payoff games.
- $<$-$\mathsf{NonEmptiness}_F$: At first glance, the problem cannot be reduced to discounted-payoff games, as the latter consider infinite plays. Yet, there is a polynomial solution by reduction to the $<$-$\mathsf{NonEmptiness}_I$ problem [11, Theorem 5.5].
- $\leq$-$\mathsf{NonEmptiness}_F$: There is no explicit solution for it in the literature, yet a PSPACE solution can be derived from the results in [14], as shown in Proposition 1.

Considering deterministic NDAs, the nonuniversality problems are also easy, as they can be reduced to the nonemptiness problems, by multiplying all transition weights by (-1). As for the exact-value problem of deterministic NDAs, it is in PSPACE with respect to finite words [14, Theorem 26], while its decidability with respect to infinite words is a notoriously difficult open problem, already for a single-state automaton, in which setting it is called the target discounted-sum (TDS) problem [14, 24].

Considering nondeterministic NDAs, the decidability of the exact-value and nonuniversality problems is open since NDAs were introduced in 2008 [23, conference version], for both finite and infinite words. It is only known to be decidable when restricting the discount factor to be an integer [13, 8, 34, 4]. These six open problems are related to various open problems in computer science, concerning dynamical systems [15, 30], Markov decision processes [26, 25], games [17, 16], reinforcement learning [5], and more. In particular, if the $\geq$-nonuniversality problem on infinite words is decidable, then so is the TDS problem and the membership problem in the middle-kth Cantor set [14, Theorems 29, 34].

The connections between the different variants of the exact-value and nonuniversality problems are also unclear, which is the focus of this work.

We show that the exact-value problem is at least as hard as the $\geq$-nonuniversality problem on infinite words, while the other four problems, namely exact-value, $\geq$-nonuniversality and $>$-nonuniversality on finite words, and $>$-nonuniversality on infinite words are interreducible, in the sense that if one of them is decidable then so are all the others. (See Fig. 1.)

$$\begin{array}{ccccc}
\mathsf{ExactValue}_F & & & & \mathsf{ExactValue}_I \\
\wr\!\vee_{\text{Thm.2}} \quad \wedge\!\wr_{\text{Thm.5}} & & & & \wr\!\vee_{\text{Thm.2}} \\
\geq\text{-}\mathsf{NonUniversality}_F & \underset{\text{Thm.4}}{\sim} & >\text{-}\mathsf{NonUniversality}_F \underset{\text{Thm.3}}{\sim} & >\text{-}\mathsf{NonUniversality}_I & \geq\text{-}\mathsf{NonUniversality}_I
\end{array}$$

Figure 1: The reductions we show between the nonuniversality and exact-value problems of discounted-sum automata on finite (subscript F) and infinite (subscript I) words. $P \gtrsim P'$ marks that P' is reducible to P, namely if P is decidable then so is P', and $P \sim P'$ marks interreducibility.

Key Ideas and Novelty in the Reductions The reduction of the $\geq$-nonuniversality problem to the exact-value problem, with respect to both finite and infinite words, is obtained by a simple construction, as detailed in Section 4.1 and illustrated in Fig. 3.

For the other reductions, we first characterize the problems by the *gap-tree* of the considered NDA $\mathcal{A}$ (Section 3). Intuitively, the gap of a run r of $\mathcal{A}$ on a finite word u stands for the weight that a continuation r' of r should save, such that the total value of $r \cdot r'$ will be 0. For each state q, the gap with respect to u is the minimum such gap among runs reaching q (or ∞ if no such run exists). The gap-tree of $\mathcal{A}$ has a node for every finite word u, storing the tuple of gaps of $\mathcal{A}$'s states with respect to u. The provided characterization directly implies that all of the problems are in RE or co-RE, and is instrumental in the reductions. While it is customary to use gaps in works on NDAs, our usage of the gap-tree and its analysis are novel.

A central insight to our reductions is that one can alter the gaps of selected states while preserving the gaps of others and, when needed, preserving the values of infinite runs. Transition weight-shifting is considered in previous works, however it results in changing run values. The novelty here is in our surgical manipulation of gaps. This insight together with the gap-tree characterization of the problems allows us to interreduce the $>$-nonuniversality problems with respect to finite and infinite words (Section 4.2).

For the interreductions between the $\geq$-nonuniversality and $>$-nonuniversality problems on finite words (Section 4.3), and for the reduction from the exact-value problem to the $\geq$-nonuniversality problem on finite words (Section 4.4), we further provide an insight on how partial information on 0-valued *words*, the recognition of 0-valued finite *runs*, can be leveraged to relate the universality problems. On top of it, we perform some more involved gap manipulations, combined with an interplay among deterministic automata, nondeterministic automata, and their products.

2 Preliminaries

An *alphabet* Σ is an arbitrary finite set, and a *word* over Σ is a finite or infinite sequence of letters in Σ, with ε for the empty word. We write Σ^* and Σ^ω for the sets of all finite words and infinite words, respectively, over Σ, and Σ^+ for the set of all finite words except for the empty word, i.e., $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$. We denote the concatenation of a finite word u and a finite or infinite word w by $u \cdot w$, or simply by uw. For a word $w = w[0]w[1]w[2]\dots$, we denote the sequence of its letters starting at index i and ending at index j as $w[i..j] = w[i]w[i+1]\dots w[j]$.

A *nondeterministic discounted-sum automaton* (NDA) with a rational discount factor[1] $\lambda > 1$ (called λ-NDA) on finite or infinite words, is a tuple $\mathcal{A} = \langle \Sigma, Q, \iota, \delta, \gamma \rangle$ over an alphabet Σ, with a finite set of states Q, an initial state $\iota \in Q$, a transition relation $\delta \subseteq Q \times \Sigma \times Q$, and a weight function $\gamma : \delta \to \mathbb{Q}$. We often write $q \xrightarrow{\sigma:x} q'$ for the transition $t = (q, \sigma, q')$, where $\gamma(t) = x$.

A *run* of $\mathcal{A}$ on a (finite or infinite) word w is a sequence r of transitions that starts in the initial state and respects the transition relation; that is $r = t_0, t_1, t_2, \dots$, such that for every i, we have $t_i = q \xrightarrow{w[i]:x} q'$ and $t_{i+1} = q' \xrightarrow{w[i+1]:x'} q''$, where $t_i, t_{i+1} \in \delta$. We write $|r|$ for the length of r (where $|r| = \infty$ for an infinite run r). The *value* of r is $\mathcal{A}(r) = \sum_{i=0}^{|r|-1} \frac{\gamma(r(i))}{\lambda^i}$. The *value* of $\mathcal{A}$ on a word w is $\mathcal{A}(w) = \inf\{\mathcal{A}(r) \mid r \text{ is a run of } \mathcal{A} \text{ on } w\}$.

If for every $q \in Q$ and $\sigma \in \Sigma$, we have $|\{q' \mid (q, \sigma, q') \in \delta\}| \geq 1$, intuitively meaning that $\mathcal{A}$ cannot get stuck, we say that $\mathcal{A}$ is *complete*. It is natural to assume that discounted-sum automata are complete, and we adopt this assumption, as dead-end states, which are equivalent to states with infinite-weight transitions, break the property of the decaying importance of future events. An NDA is *deterministic*

[1] Discount factors are often defined as numbers between 0 and 1, under which setting, transition weights are multiplied by these factors rather than divided by them. Discount factors are usually assumed to be rational, while [9] studies their extension to real values. When the discount factor is an integer, all the considered problems are known to be decidable [13].

if its transition relation maps every state and letter to a singleton. Automata $\mathscr{A}$ and $\mathscr{A}'$ are *equivalent*, denoted by $\mathscr{A} \equiv \mathscr{A}'$, if for every word w, $\mathscr{A}(w) = \mathscr{A}'(w)$.

The *cost* of reaching a state q of an NDA $\mathscr{A}$ over a finite word u is $\mathsf{cost}(q,u) = \min\{\mathscr{A}(r) \mid r$ is a run of $\mathscr{A}$ on u ending in q $\}$, where $\min\emptyset = \infty$. The *gap* of a finite run r (for reaching the value 0)[2] is $\mathsf{gap}(r) = \lambda^{|r|}\mathscr{A}(r)$, and the *gap* of a state q over a finite word u is $\mathsf{gap}(q,u) = \lambda^{|u|}\mathsf{cost}(q,u) = \min\{\mathsf{gap}(r) \mid r$ is a run of $\mathscr{A}$ on u ending in $q\}$. Intuitively, the gap of a run r stands for the weight that a continuation r' of r should save, such that the total value of $r \cdot r'$ will be 0.

We end the section with the claim made in the Introduction about the $\leq$-$\mathsf{NonEmptiness}_F$ problem.

Proposition 1. *The $\leq$-$\mathsf{NonEmptiness}_F$ problem can be solved in* PSPACE.

Proof. Given an NDA $\mathscr{A}$, one can first consider the $<$-$\mathsf{NonEmptiness}_F$ problem, getting in polynomial time an answer for whether there exists a word w, such that $\mathscr{A}(w) < 0$. If yes, we are done.

Otherwise, we know that all of the automaton finite runs have value at least 0, and we should check whether there exists a run with value exactly 0. We may solve it by linear reduction to the 'constrained generalized target discounted-sum problem' on finite words, the CGTDS^F problem considered in [14], which has a PSPACE solution [14, Theorem 25].

The CGTDS^F problem asks, given a finite set S of letters, each having a rational weight, a regular language $L \subseteq S^*$, and a target rational value t, whether there exists a finite word $u \in L$, such that its discounted-sum is exactly t. In the reduction, we set t to be 0, S to be $\mathscr{A}$'s transitions, each weighted by its weight in $\mathscr{A}$, and L to be the safety language, obtained by considering $\mathscr{A}$ as a DFA over the alphabet S, whose states are all accepting. □

Notice that the above technique, of finding a 0-valued finite run, is not enough for solving the exact-value problem, since there might be many runs over a word, some of which with value 0 and some with negative values, in which case the value of the word will not be 0.

3 Characterizing the Decision Problems via Gap-Trees

We characterize the exact-value and nonuniversality problems of an NDA by the gap-tree of its computations over all finite words (Lemma 2). It will directly imply that all of these problems are in RE or coRE, and will be further used in Section 4 for the reduction proofs.

The *gap-tree* of an NDA $\mathscr{A}$ with n states over an alphabet Σ is the full Σ^*-tree (namely, the tree with a node for every word in Σ^*, whose root is ε, and the children of a node $u \in \Sigma^*$ are $u \cdot \sigma$, for every $\sigma \in \Sigma$, reached from u by an edge labeled σ), where the value of a node is the n-tuple of gaps of $\mathscr{A}$'s states. See Fig. 2 for an example.

Note that the gaps in the tree can be computed iteratively, based only on the gaps of the parent node and the weights of the last transitions. Indeed, recall that for a state q and a finite word $u = v \cdot \sigma$, we have $\mathsf{gap}(q,u) = \lambda^{|u|}\mathsf{cost}(q,u)$, where $\mathsf{cost}(q,u)$ is the value of the minimal-valued run r of $\mathscr{A}$ on u reaching q. Let q' be the state that r reaches reading v, and x the weight of r's last transition (over σ). Then

$$\begin{aligned}\mathsf{gap}(q,u) &= \lambda^{|u|}\mathsf{cost}(q,u) = \lambda \cdot \lambda^{|u|-1}(\mathsf{cost}(q',v) + \lambda^{1-|u|}x)\\ &= \lambda(\lambda^{|u|-1}\mathsf{cost}(q',v) + \lambda^{|u|-1}\lambda^{1-|u|}x) = \lambda(\mathsf{gap}(q',v) + x)\end{aligned}$$

[2] In some papers, e.g., [13], depending on the context, the gap is defined a little differently, with respect to an optimal run of $\mathscr{A}$ on u, rather than with respect to the value 0.

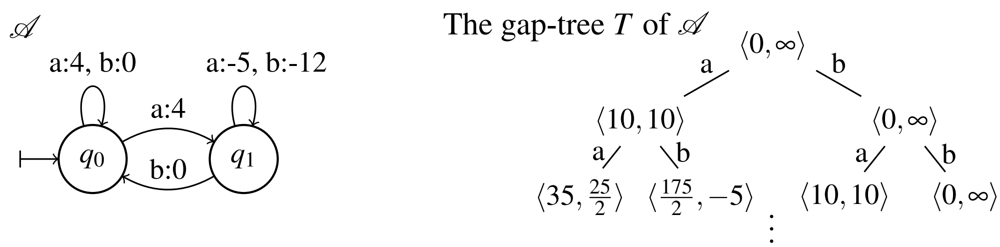

Figure 2: An NDA $\mathscr{A}$ with discount factor $\lambda = \frac{5}{2}$ and the first levels of its gap-tree T. The nodes in T are pairs, in which the left element corresponds to the gap of q_0 over the word leading from the root of T to that node, and the right element corresponds to the gap of q_1 over this word.

Observe that once a gap of a run is too big, positively or negatively, the gaps of all runs extending it will be, from that point on, positive or negative, respectively, growing toward $\pm\infty$. Formally, consider an NDA $\mathscr{A}$ with discount factor λ and maximal weight m in absolute value. The *recovery bound* of $\mathscr{A}$ is $B = \frac{m\cdot\lambda}{\lambda-1}$, and a gap $g \in \mathbb{Q}$ with respect to $\mathscr{A}$ is said to be *unrecoverable* if $|g| > B$ (cf. gap-unrecoverability in NDA determinization). A node in the gap-tree is said to be unrecoverable if the minimal gap in it is unrecoverable. A gap or a node that is not unrecoverable is called *recoverable*.

Lemma 1. *Consider an NDA $\mathscr{A}$ and a run r of $\mathscr{A}$ whose gap g is unrecoverable. Then all finite and infinite runs of $\mathscr{A}$ that extend r have positive values if g is positive and negative values if g is negative.*

Proof. Let m be $\mathscr{A}$'s maximal weight in absolute value, and let B be its recovery bound. Consider a run r with an unrecoverable gap $g = B+\delta$ for $\delta > 0$. (Wlog, we assume g to be positive.) Since the next gap g_{next} of a run extending r is at least $\lambda(g-m)$, we have:

$$\begin{aligned} g_{next} &\geq \lambda(B+\delta-m) = \lambda\Big(\frac{m\cdot\lambda}{\lambda-1}+\delta-m\Big) \\ &= \frac{\lambda}{\lambda-1}(m\cdot\lambda+\delta(\lambda-1)-m(\lambda-1)) = \frac{\lambda\cdot m}{\lambda-1}+\lambda\cdot\delta = B+\lambda\cdot\delta \end{aligned}$$

By induction on i, the gap g_i of a run r_i extending r by i transitions is at least $B+\lambda^i\delta$, implying that:

$$\mathscr{A}(r_i) = \frac{g_i}{\lambda^{|r|+i}} \geq \frac{B+\lambda^i\delta}{\lambda^{|r|+i}} = \frac{B}{\lambda^{|r|+i}} + \frac{\lambda^i\delta}{\lambda^{|r|+i}} \geq \frac{\lambda^i\delta}{\lambda^{|r|+i}} = \frac{\delta}{\lambda^{|r|}}$$

Hence, all prefixes of length at least $|r|$ of an infinite run r' extending r have a value at least the constant $\frac{\delta}{\lambda^{|r|}}$, implying that the value of r' is at least that positive constant. $\square$

Lemma 2. *Consider an NDA $\mathscr{A}$, and let T be its gap-tree. Then for $\mathscr{A}$:*

1. $\mathsf{ExactValue}_F$ *is True iff there is a node in T whose minimal gap is* 0.
2. $\geq$-$\mathsf{NonUniversality}_F$ *is True iff there is a node in T whose minimal gap is at least* 0.
3. $>$-$\mathsf{NonUniversality}_F$ *is True iff there is a node in T whose minimal gap is strictly positive.*
4. $\mathsf{ExactValue}_I$ *is False iff there is a level of T, such that for every node at that level it holds that its minimal gap is unrecoverable (the gap may be positive or negative).*
5. $\geq$-$\mathsf{NonUniversality}_I$ *is False iff there is a level of T, such that for every node at that level it holds that its minimal gap is negative and unrecoverable.*

6. *$>$-NonUniversality$_I$ is True iff there is a node in T whose minimal gap is positive and unrecoverable.*

Proof. Claims 1-3 are straightforward, as for every state q of $\mathscr{A}$ and finite word u, we have $\mathsf{gap}(q,u) = \lambda^{|u|}\mathsf{cost}(q,u)$, implying that a gap is negative or 0 or positive if and only if the corresponding best run of $\mathscr{A}$ on u reaching q is negative or 0 or positive, respectively. Then, since the minimal gap within a node u of the gap-tree T corresponds to the gap of the best run of $\mathscr{A}$ on u, the required result follows.

Claim 4: if there is a level in T all of whose nodes are unrecoverable, then by Lemma 1 all the finite and infinite runs extending them will not be equal to 0, implying the required result. As for the other direction, if in every level there is a recoverable node, then by Lemma 1 its parent node must also be recoverable, implying an infinite path of recoverable nodes. Thus, a run along this path will have gaps bounded by the constant recovery bound of $\mathscr{A}$, implying that its value will converge to 0, as the value in its ith position is the gap at that position divided by λ^i.

Claim 5: if there is a level all of whose nodes are unrecoverable with a negative gap, then by Lemma 1 all the infinite runs extending them will have negative values, implying the required result. If there is no such level, then either in every level there is a recoverable node, in which case we fall into Claim 4 proved above, or there is a level with a node whose minimal gap is positive and unrecoverable, in which case we fall into Claim 6 proved below.

Claim 6: if there is a node whose minimal gap is positive and unrecoverable, then by Lemma 1 all infinite runs extending it will have positive values, implying the required result. If there is no such level, then the gaps along every infinite run are either negative or bounded by $\mathscr{A}$'s recovery bound, implying that all runs have a non-positive value. □

Theorem 1. *The ExactValue$_F$, $\geq$-NonUniversality$_F$, $>$-NonUniversality$_F$, and $>$-NonUniversality$_I$ problems are in RE and the ExactValue$_I$ and $\geq$-NonUniversality$_I$ problems are in coRE.*

Proof. It directly follows from Lemma 2, as a positive or negative witness for each of the problems can be found by unfolding the gap-tree of the NDA, until reaching a node or a level of the tree that characterizes a witness. □

4 Reductions between the Problems

We begin in Section 4.1 with a simple construction that reduces the $\geq$-nonuniversality problem to the exact-value problem, with respect to both finite and infinite words.

Next, in Section 4.2, we establish interreducibility between the $>$-nonuniversality problems on finite and infinite words. We build on the gap-tree characterization of the problems (Lemma 2), and manipulate the automaton transition weights, such that the gap characterization of one problem reduces to the gap characterization of the other.

In Section 4.3, we provide interreductions between the $\geq$- and $>$-nonuniversality problems on finite words. Unlike the reductions of Section 4.2, here we cannot manipulate directly the transition weights of a given λ-NDA $\mathscr{A}$. We thus first construct a DFA $\mathscr{D}$ that recognizes the finite sequences of weights from $\mathscr{A}$ whose λ-discounted-sum is 0. We then construct a λ-NDA $\mathscr{H}$ from the product of $\mathscr{A}$ and $\mathscr{D}$, and manipulate its transition weights in two opposite ways into λ-NDAs $\mathscr{B}$ and $\mathscr{C}$, where both $\mathscr{B}$ and $\mathscr{C}$ do not have 0-valued runs, and such that the $>$-nonuniversality and $\geq$-nonuniversality problems of $\mathscr{A}$ reduce, respectively, to the $\geq$-nonuniversality problem of $\mathscr{B}$ and to the $>$-nonuniversality problem of $\mathscr{C}$.

Finally, in Section 4.4, we present the most involved reduction, from the exact-value problem to the $\geq$-nonuniversality problem on finite words. While in the reductions of Section 4.3, we manipulated 0-valued runs, in this case we need to manipulate runs on finite words u, such that there is no 0-valued run of $\mathscr{A}$ on u. That is, we cannot manipulate a run r according to its own value, but need to consider all runs of $\mathscr{A}$ on the word u on which r runs.

We accomplish it via a series of five constructions. We start by building the DFA $\mathscr{D}$, as done in Section 4.3, just without the dead-end state $(\cdot)$. We then take the product of $\mathscr{A}$ and $\mathscr{D}$, however this time as an NFA $\mathscr{J}$, recognizing words in Σ^* on which $\mathscr{A}$ has a 0-valued run. We further determinize $\mathscr{J}$ into a DFA $\mathscr{K}$ and then construct a λ-NDA by taking the product of $\mathscr{A}$ and $\mathscr{K}$. Finally, we modify the transition weights in this λ-NDA to obtain a λ-NDA $\mathscr{P}$, such that there is a finite word u with $\mathscr{A}(u)=0$ iff there is a finite word u with $\mathscr{P}(u)\geq 0$.

As all the considered problems are decidable when the discount factor λ is an integer [13, 8, 34, 4], we assume in our proofs that λ is a non-integer.

4.1 From $\geq$-Nonuniversality to Exact-Value

By a simple construction (Fig. 3), we obtain that the exact-value problem is at least as difficult as the $\geq$-nonuniversality problem, with respect to both finite and infinite words.

Theorem 2. *If ExactValue$_F$ (resp. ExactValue$_I$) is decidable then so is $\geq$-NonUniversality$_F$ (resp. $\geq$-NonUniversality$_I$).*

Proof. Given an NDA $\mathscr{A}$, we construct an NDA $\mathscr{A}'$ similar to $\mathscr{A}$, except for having an additional initial state with a 0-weighted self-loop over all alphabet letters; see Fig. 3. (Technically, we defined in Section 2 that an NDA has a single initial state; the NDA $\mathscr{A}'$ can have a single initial state, whose outgoing transitions are as of both of the considered initial states.) We claim that there is a word w s.t. $\mathscr{A}(w)\geq 0$ iff there is a word w s.t. $\mathscr{A}'(w)=0$.
$\Rightarrow$: Consider a word w, such that $\mathscr{A}(w)\geq 0$. Then all runs of $\mathscr{A}$ on w have value at least 0, and thus all the runs of $\mathscr{A}'$ in its part that is identical to $\mathscr{A}$ have value at least 0. Hence, due to the additional state of $\mathscr{A}'$, whose self-loop run on w has value 0, the value of $\mathscr{A}'$ on w is 0.
$\Leftarrow$: Consider a word w', s.t. $\mathscr{A}'(w')=0$. By the construction of $\mathscr{A}'$, it follows that all the runs of $\mathscr{A}'$ on w' in its part that is identical to $\mathscr{A}$ have value at least 0. Hence, the value of $\mathscr{A}$ on w' is at least 0. $\square$

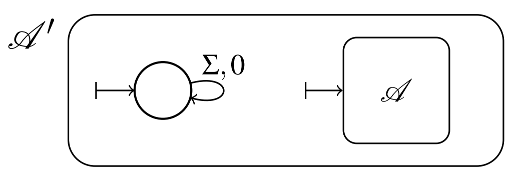

Figure 3: The $\geq$-nonuniversality problem of an NDA $\mathscr{A}$ reduces, by the simple construction above, as explained in the proof of Theorem 2, to the exact-value problem of the NDA $\mathscr{A}'$.

4.2 Between $>$-Nonuniversality on Finite and Infinite Words

The reduction from the finite-word setting to the infinite-word setting is straightforward: we adapt the original λ-NDA $\mathscr{A}$ by adding 0-valued transitions over a fresh letter # from all states to a new state with

0-weighted self-loop over all alphabet letters (see the NDA $\mathscr{B}$ in Fig. 4), thus "prolonging" the positive finite runs to infinite ones.

The reduction in the other direction is more involved. We build on the gap-tree characterization of the problems (Lemma 2), according to which there exists an infinite word on which $\mathscr{A}$ has a strictly positive value iff there is a node in its gap-tree whose minimal gap is above the recovery bound B. Then, we show how to construct a λ-NDA $\mathscr{C}$ whose gap-tree is similar to that of $\mathscr{A}$, except that all its gaps are smaller by B than those in $\mathscr{A}$. We achieve this by reducing $\frac{B}{\lambda}$ from the weight of the initial transitions of $\mathscr{A}$, and adding $B - \frac{B}{\lambda}$ to all other transition weights. (See the NDA $\mathscr{C}$ in Fig. 4.) Since there exists a finite word on which $\mathscr{C}$ has a strictly positive value iff there is a node in its gap-tree whose minimal gap is strictly positive, we get the required reduction.

Theorem 3. *$>$-NonUniversality$_F$ is decidable if and only if $>$-NonUniversality$_I$ is.*

Proof. I. Reducing $>$-NonUniversality$_F$ to $>$-NonUniversality$_I$ is straightforward: Given an NDA $\mathscr{A}$ over an alphabet Σ, we construct from it an NDA $\mathscr{B}$, by adding a new sink state q_{new} with a 0-weighted self-loop over all alphabet letters, and 0-valued transitions to q_{new} from all states over a fresh letter #. (See an example in Fig. 4.)

Then, a positive-valued optimal run r in $\mathscr{A}$ over a finite word u implies a positive-valued optimal run r' in $\mathscr{B}$ over the infinite word $u\#^\omega$. As for the other direction, consider an infinite word w, for which $\mathscr{B}(w) > 0$. If w is in the form $u\#v$, for some finite word $u \in \Sigma^*$ and infinite word $v \in (\Sigma \cup \{\#\})^*$, then the value of $\mathscr{B}$ on u and on w is the same, and since u does not have the letter #, so is the value of $\mathscr{A}$ on it. Otherwise, namely when w does not have the letter #, we have that $\mathscr{A}(w) > 0$. By the characterization of the problems provided in Lemma 2, it follows that there is a node in the gap-tree of $\mathscr{A}$ whose minimal gap is positive and unrecoverable, and therefore the value of $\mathscr{A}$ on the finite word leading to this node is strictly positive.

II. Reducing $>$-NonUniversality$_I$ to $>$-NonUniversality$_F$ is obtained by shifting-down all gaps by the recovery bound: Given an NDA $\mathscr{A}$ with recovery bound B, we construct from it an NDA $\mathscr{C}$ (see an example in Fig. 4), by

- adding a special initial state $start$, and copying all the transitions going out of the original initial state of $\mathscr{A}$ to go out also from $start$;
- subtracting $\frac{B}{\lambda}$ from the weights of the transitions going out of $start$; and
- adding $B - \frac{B}{\lambda}$ to the weights of all other transitions.

Observe that $\mathscr{A}$ and $\mathscr{C}$ have the same finite and infinite runs (except for the initial state in $\mathscr{C}$ being $start$), and while their values and gaps are different, since the weight changes across the board, the value of a run r is smaller than or equal to or bigger than a value of a run r' in $\mathscr{A}$ iff it is the case in $\mathscr{C}$.

We claim that for every finite run r with gap g in $\mathscr{A}$, we have that its gap in $\mathscr{C}$ is $g - B$. Indeed, we prove it by induction on the length of r. The base case is immediate by the construction of $\mathscr{C}$, as the gap in $\mathscr{A}$ of a run with a single transition with weight x is λx, and in $\mathscr{C}$ it is $\lambda(x - \frac{B}{\lambda}) = \lambda x - B$. For the induction step, recall that if the gap of a run r is g then the gap of a run r' that extends r with a single transition of weight x is $g' = \lambda(g+x)$. Thus, by the induction assumption, the gap of r in $\mathscr{C}$ is $g - B$, and by the construction of $\mathscr{C}$, the gap of r' in $\mathscr{C}$ is $\lambda(g - B + x + B - \frac{B}{\lambda}) = \lambda(g + x - \frac{B}{\lambda}) = \lambda(g+x) - B = g' - B$.

Thus, $\mathscr{A}$ and $\mathscr{C}$ have the same gap-trees, just that all gaps in $\mathscr{C}$'s tree are smaller by B than the gaps in $\mathscr{A}$'s tree. Now, by the characterization provided in Lemma 2, if $>$-NonUniversality$_F$ is True for $\mathscr{C}$ then there is a node in its gap-tree whose minimal gap is positive, implying that it is above the recovery bound B in $\mathscr{A}$'s gap-tree, implying that $>$-NonUniversality$_I$ is True for $\mathscr{A}$. Analogously, if $>$-NonUniversality$_F$

is False for $\mathscr{C}$ then all nodes in its gap-tree have minimal gap at most 0, implying that all nodes in $\mathscr{A}$'s gap-tree have minimal gap at most B, implying that $>$-$\mathsf{NonUniversality}_I$ is False for $\mathscr{A}$. $\square$

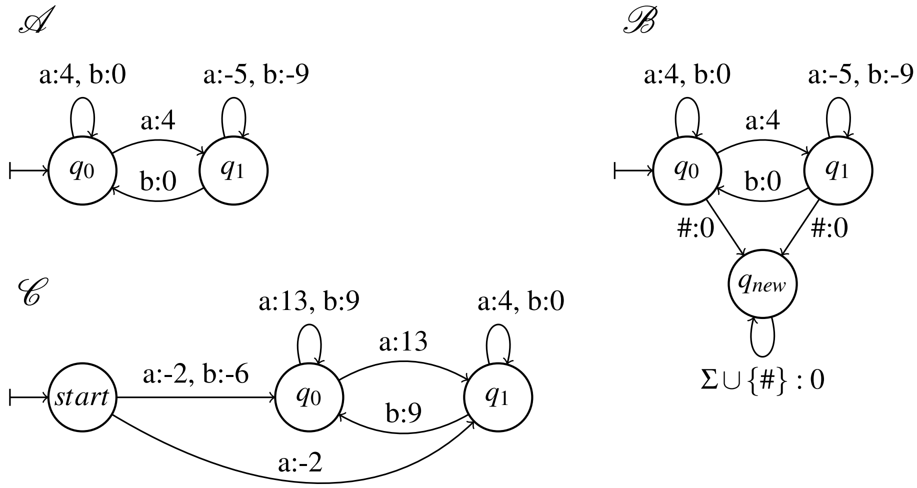

Figure 4: An NDA $\mathscr{A}$ with discount-factor $\lambda = \frac{5}{2}$; the λ-NDA $\mathscr{B}$ obtained from $\mathscr{A}$ by adding 0-valued transitions over a fresh letter # from all states to a new state with 0-weighted self-loop over all alphabet letters; and the λ-NDA $\mathscr{C}$, obtained from $\mathscr{A}$ by subtracting $\frac{B}{\lambda}$, where B is $\mathscr{A}$'s recovery bound (15 in this case), from the weights of the transitions going out of a new initial state $start$, and adding $B - \frac{B}{\lambda}$ to the weights of all other transitions. By Theorem 3, the $>$-nonuniversality problems of $\mathscr{A}$ on finite and infinite words reduce, respectively, to the $>$-nonuniversality problem of $\mathscr{B}$ on infinite words, and to the $>$-nonuniversality problem of $\mathscr{C}$ on finite words.

4.3 Between $\geq$-Nonuniversality and $>$-Nonuniversality on Finite Words

A first key observation for the considered reductions is that, given a rational discount factor $\lambda > 1$ and a finite set W of rational weights, it is decidable in PSPACE whether there exists a non-empty finite sequence of weights from W such that its λ-discounted sum is 0 [14, Theorem 25]. Following the arguments in [14], given a λ-NDA $\mathscr{A}$, we construct a DFA $\mathscr{D}$ whose alphabet is the set of weights of $\mathscr{A}$, such that $\mathscr{D}$ accepts a finite sequence of weights if and only if its λ-discounted-sum is 0 (see Lemma 3 below and an example in Fig. 5).

The states of $\mathscr{D}$ correspond to the relevant gaps along a finite sequence of weights in W whose λ-discounted-sum is 0. The idea behind the existence of $\mathscr{D}$ is that after a bounded number of steps along a sequence, the gap must either return to a previously visited gap, be too big or too small for allowing for a 0-valued sequence, or have a denominator that will forever grow in additional steps, not allowing for a 0-valued sequence. $\mathscr{D}$'s only accepting state is its initial state (whose gap is 0), and if we need $\mathscr{D}$ to be complete, as in the proof of Theorem 4, we add to it a "dead-end" state $(\cdot)$ to which all dead-end transitions go.

Lemma 3. *Given a discount factor $\lambda \in \mathbb{Q} \cap (1, \infty)$ and a finite set $W \subseteq \mathbb{Q}$ of weights, we can build a DFA $\mathscr{D}$ recognizing the finite sequences in W^* whose λ-discounted-sum is* 0.

Proof of Lemma 3. Consider a discount factor $\lambda = \frac{p}{q}$ in reduced form, and a finite set $W = \{w_1, \ldots, w_k\}$ of rational weights, for which $m \in \mathbb{Q}$ is the maximal weight in absolute value, and $d \in \mathbb{N} \setminus \{0\}$ is the minimal common denominator. Let $d = q^e \cdot c$, for $e \in \mathbb{N}$ and $c \in \mathbb{N} \setminus \{0\}$, such that c is not divisible by q.

We build the DFA $\mathscr{D}$ by starting to explore the, possibly infinite, tree T of gaps of W-weighted sequences, as detailed below, while turning it into a DFA, as detailed next. See an example in Fig. 5.

The tree T.

The root of T has gap 0, and every node x of T with gap g has the children nodes $x_1, \ldots, x_k$ with gaps $g_i = \lambda(g + w_i)$, for $i \in [1..k]$.

Observe that the gaps of T have the following properties: The gap g of a child x of the root is $g = \frac{p}{q}(0 + \frac{a}{q^e \cdot c}) = \frac{p \cdot a}{q^{e+1} \cdot c}$ (not necessarily in reduced form), for some $a \in \mathbb{N}$.

A child x' of x will have a gap $g' = \lambda(g + v')$ for some weight $v' = \frac{a'}{d} \in W$, implying that $g' = \frac{p}{q}(\frac{p \cdot a}{q^{e+1} \cdot c} + \frac{a'}{q^e \cdot c}) = \frac{p}{q}(\frac{p \cdot a + q \cdot a'}{q^{e+1} \cdot c}) = \frac{p(p \cdot a + q \cdot a')}{q^{e+2} \cdot c}$ (not necessarily in reduced form).

By induction on i, we get that a node at level i of T has a gap that can be represented in the form $\frac{v}{q^{e+i} \cdot c}$ (not necessarily in reduced form), for some $v \in \mathbb{N}$.

The DFA $\mathscr{D}$.

While generating T, we remove a node x (and thus do not explore its continuation) if its gap g falls into at least one of the following cases:

1. g is the same as the gap of an already generated node x'. In this case, the parent of x has a transition to x' instead of to x.
2. g is unrecoverable, namely $|g| > \frac{m \cdot \lambda}{\lambda - 1}$.
3. The denominator of g's reduced form is divisible by q^{e+1}.

We make the resulting graph a DFA $\mathscr{D}$ by setting the root node to be the initial and only accepting state, and removing all dead-end states. When we need $\mathscr{D}$ to be complete, as in the proof of Theorem 4, we add to it a "dead-end" state $(\cdot)$ to which all dead-end transitions go, and when we do not need it, as in the proof of Theorem 5, it is omitted.

Notice that $\mathscr{D}$ is finite, since there are finitely many rational numbers between $-\frac{m \cdot \lambda}{\lambda - 1}$ and $\frac{m \cdot \lambda}{\lambda - 1}$ that can be represented with denominator $q^e \cdot c = d$.

Correctness.

We show below that each of the three changes above to T does not influence the sequences of W-weights whose λ-discounted-sum is 0:

1. Since two nodes with the same gap have the same subtree, a transition to x' instead of to x makes no change in the generated gap sequences.
2. The absolute value of any gap continuing g will be bigger than the absolute value of g (see the proof of Lemma 1), so no continuation of it can lead to the gap 0.
3. Consider a node with a gap g in reduced form whose denominator is divisible by q^{e+1}. Then it is in the form $\frac{v}{q^{e+i} \cdot b}$ for some $i > 0$ and b that divides c (and is thus not divisible by q).

   We claim that every child of that node will have a gap g' in reduced form whose denominator is divisible by q^{e+1} (from which it follows that no continuation of that node can lead to the gap 0).

   Indeed, $g' = \lambda(g + \frac{a}{d}) = \lambda(\frac{v}{q^{e+i} \cdot b} + \frac{a}{q^e \cdot c})$, for some $\frac{a}{d} = \frac{a}{q^e \cdot c} \in W$. Hence, $g' = \frac{p}{q}(\frac{v}{q^{e+i} \cdot b} + \frac{a}{q^e \cdot c}) = \frac{p}{q}(\frac{(c/b)v + q^i \cdot a}{q^{e+i} \cdot c})$.

   Now, since v and (c/b) are not divisible by q, while $(q^i \cdot a)$ is, it follows that $((c/b)v + q^i \cdot a)$ is not divisible by q. Thus, the denominator of g' in reduced form is divisible by q^{e+i+1}.

□

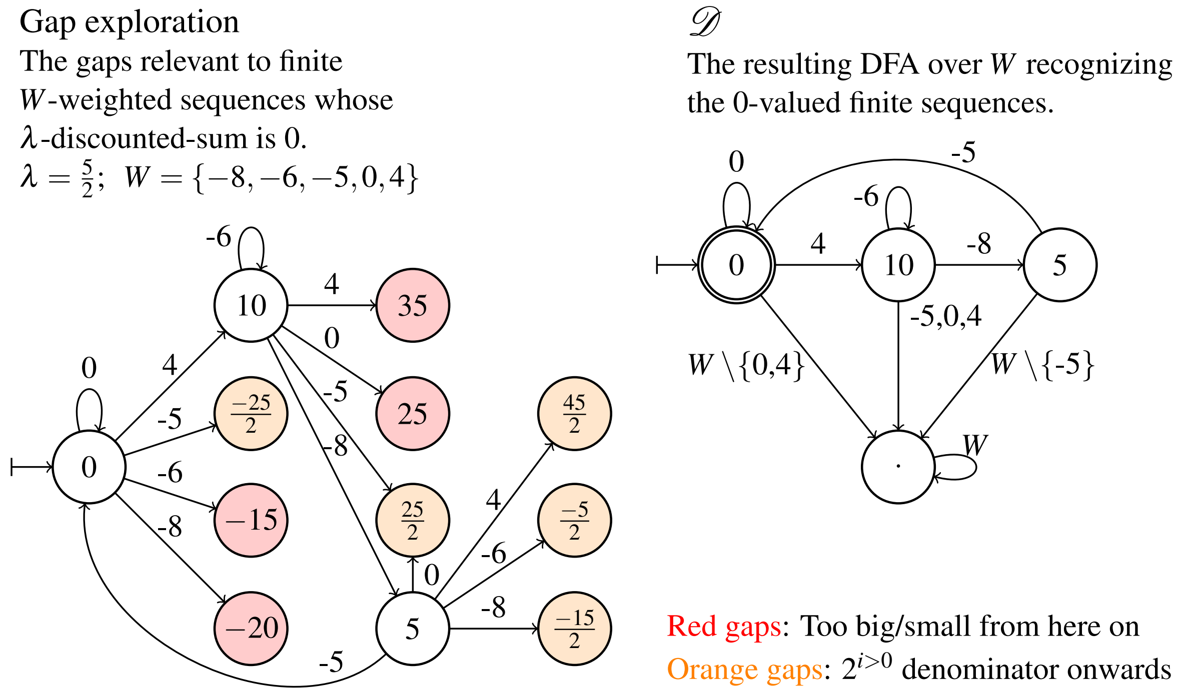

Figure 5: An example for the construction of the DFA $\mathscr{D}$ along the proof of Lemma 3. The $(\cdot)$ state is a dead-end state, added if we need the automaton to be complete (as in the proof of Theorem 4), and omitted otherwise (as in the proof of Theorem 5.)

Given an NDA $\mathscr{A}$, let $\mathscr{D}$ be the corresponding DFA as per Lemma 3. Theorem 4 below shows that the $\geq$-NonUniversality$_F$ and $>$-NonUniversality$_F$ problems are interreducible, by manipulating the product $\mathscr{H}$ of $\mathscr{A}$ and $\mathscr{D}$. Notice that $\mathscr{H}$ augments the states of $\mathscr{A}$ with the gaps that might occur along finite runs of $\mathscr{A}$ whose λ-discounted-sum is 0.

Observe that our usage of $\mathscr{H}$ does not aim at deciding whether there exists a *word* w, such that $\mathscr{A}(w) = 0$, since there might be several runs of $\mathscr{A}$ on w, some of which with value 0 and some with value below 0, in which case $\mathscr{A}(w)$ will be negative. We use it in a somewhat opposite way, for getting NDAs $\mathscr{B}$ and $\mathscr{C}$ very similar to $\mathscr{A}$, but in which there cannot be a finite run with value 0.

We obtain $\mathscr{B}$ and $\mathscr{C}$ from $\mathscr{H}$, by "bypassing" states with gap 0, in two opposite ways (see an example in Fig. 6). In $\mathscr{B}$, we reduce by 1 the weights of all transitions that enter a state q of $\mathscr{H}$ with gap 0, and compensate on it by increasing by λ the weights of all transitions going out of q; in $\mathscr{C}$, we increase by 1 the weights of the incoming transitions and reduce by λ the weights of the outgoing transitions.

Observe that both $\mathscr{B}$ and $\mathscr{C}$ do not have the value 0 on any word. In $\mathscr{C}$, the value of a word w increases, compared to its value in $\mathscr{A}$, if and only if $\mathscr{A}(w) = 0$, while $\mathscr{C}(w) = \mathscr{A}(w)$ if $\mathscr{A}(w) < 0$ or $\mathscr{A}(w) > 0$. In $\mathscr{B}$, the value of a word w decreases if $\mathscr{A}(w) = 0$ and might decrease if $\mathscr{A}(w) < 0$, while $\mathscr{B}(w) = \mathscr{A}(w)$ if $\mathscr{A}(w) > 0$. As a result, the $>$-nonuniversality and $\geq$-nonuniversality problems of $\mathscr{A}$ reduce, respectively, to the $\geq$-nonuniversality problem of $\mathscr{B}$ and to the $>$-nonuniversality problem of $\mathscr{C}$.

Theorem 4. *$\geq$-NonUniversality$_F$ is decidable if and only if $>$-NonUniversality$_F$ is.*

Proof. Consider an NDA $\mathscr{A}$ over an alphabet Σ with states Q and discount factor λ, and let W be the set of its transition weights. We construct a DFA $\mathscr{D}$, as per Lemma 3, recognizing the finite sequences in W^* whose λ-discounted-sum is 0. Let D denote $\mathscr{D}$'s states, and $\iota_{\mathscr{D}}$ its initial, and only accepting, state.

We then construct a λ-NDA $\mathscr{H}$ over Σ by taking the product of $\mathscr{A}$ and $\mathscr{D}$, having the states $Q \times D$, and transitions $\langle q, d\rangle \xrightarrow{\sigma:x} \langle q', d'\rangle$, such that there are transitions $q \xrightarrow{\sigma:x} q'$ in $\mathscr{A}$ and $d \xrightarrow{x} d'$ in $\mathscr{D}$. Notice that $\mathscr{H}$ augments the states of $\mathscr{A}$ with the gaps relevant to finite runs leading to gap (and value) 0, or if there is no such gap, with the 'dead-end' state $(\cdot)$ of $\mathscr{D}$. We further slightly change $\mathscr{H}$, such that its initial state is not returned to, by having a special initial state *start*, and copying all the transitions going out of the original initial state to go out also from *start*.

Now, we construct from $\mathscr{H}$ two λ-NDAs, $\mathscr{B}$ and $\mathscr{C}$, both "bypassing" the 0-gaps of $\mathscr{H}$, where $\mathscr{B}$ does it with temporarily-lower-weight transitions and $\mathscr{C}$ with temporarily-higher-weight transitions (see Fig. 6). Formally, $\mathscr{B}$ and $\mathscr{C}$ have the same states and transitions as $\mathscr{H}$, while having the following weight changes to transitions going into and out of states $\langle q, \iota_{\mathscr{D}}\rangle$ of $\mathscr{H}$ (namely, states of $\mathscr{H}$ having a 0-gap):

- For every transition $\langle q', d \neq \iota_{\mathscr{D}}\rangle \xrightarrow{\sigma:x} \langle q, \iota_{\mathscr{D}}\rangle$ and $start \xrightarrow{\sigma:x} \langle q, \iota_{\mathscr{D}}\rangle$, we change in $\mathscr{B}$ the weight from x to $x-1$ and in $\mathscr{C}$ to $x+1$.
- For every transition $\langle q, \iota_{\mathscr{D}}\rangle \xrightarrow{\sigma:x} \langle q', d \neq \iota_{\mathscr{D}}\rangle$, we change in $\mathscr{B}$ the weight from x to $x+\lambda$ and in $\mathscr{C}$ to $x-\lambda$.
- For every transition $\langle q, \iota_{\mathscr{D}}\rangle \xrightarrow{\sigma:0} \langle q', \iota_{\mathscr{D}}\rangle$ (notice that the weight of a transition that keeps the gap 0 must be 0), we change in $\mathscr{B}$ the weight from 0 to $+\lambda-1$ and in $\mathscr{C}$ to $-\lambda+1$.

We claim that for every word w, we have:

1. If $\mathscr{A}(w) > 0$ then $\mathscr{A}(w) = \mathscr{B}(w) = \mathscr{C}(w)$.
2. If $\mathscr{A}(w) = 0$ then $\mathscr{B}(w) < 0$ and $\mathscr{C}(w) > 0$.
3. If $\mathscr{A}(w) < 0$ then $\mathscr{B}(w) \leq \mathscr{A}(w)$ and $\mathscr{C}(w) = \mathscr{A}(w)$.

Before proving these three claims, notice that they imply the required results: Checking whether there exists a word w such that $\mathscr{B}(w) \geq 0$ provides the answer to whether there exists a word w, such that $\mathscr{A}(w) > 0$; and checking whether there exists a word w such that $\mathscr{C}(w) > 0$ provides the answer to whether there exists a word w, such that $\mathscr{A}(w) \geq 0$.

For the correctness of Claims 1–3, observe that $\mathscr{H}$, $\mathscr{B}$, and $\mathscr{C}$ have the same runs, while:

- Every finite run that does not end in a state $\langle q, \iota_{\mathscr{D}}\rangle$ has the same value in $\mathscr{H}$, $\mathscr{B}$ and $\mathscr{C}$. Indeed, if the run does not go through a $\langle q, \iota_{\mathscr{D}}\rangle$ state then it has the same sequence of weights in $\mathscr{H}$, $\mathscr{B}$, and $\mathscr{C}$. If it does go through a $\langle q, \iota_{\mathscr{D}}\rangle$ state then the -1 weight decrease in $\mathscr{B}$'s transition entering $\langle q, \iota_{\mathscr{D}}\rangle$ is compensated by the addition of $+\lambda$ in the outgoing transition, and analogously, in the other direction, in $\mathscr{C}$. As for transitions between two states having the 0-gap (or a self-loop over such a state), the -1 gap is preserved in $\mathscr{B}$ by the $+\lambda-1$ weight of the self-transition, since the $+\lambda$ part compensates on the previous -1 gap, and the -1 part creates a new -1 gap, and analogously for the preservation of a $+1$ gap in $\mathscr{C}$.
- Every finite run that ends in a state $\langle q, \iota_{\mathscr{D}}\rangle$ has the value 0 in $\mathscr{H}$ (since the gap is 0), a negative value in $\mathscr{B}$ (since the last weight along the run is smaller in $\mathscr{B}$ than in $\mathscr{H}$), and a positive value in $\mathscr{C}$ (since the last weight along the run is bigger in $\mathscr{C}$ than in $\mathscr{H}$).

Now, proving the claims:

1. If $\mathscr{A}(w) > 0$ then all runs of $\mathscr{H}$ on w end in a positive-gap state, thus by the two observations above they have the same values in $\mathscr{H}$, $\mathscr{B}$ and $\mathscr{C}$, so $\mathscr{A}(w) = \mathscr{B}(w) = \mathscr{C}(w)$.
2. If $\mathscr{A}(w) = 0$ then some runs of $\mathscr{H}$ on w end in 0-gap-states and possibly some in positive-gap state. Thus, by the two observations above, some runs of $\mathscr{B}$ on w have negative values, implying that $\mathscr{B}(w) < 0$, while all runs of $\mathscr{C}$ on w have positive values, implying that $\mathscr{C}(w) > 0$.
3. If $\mathscr{A}(w) < 0$ then some runs of $\mathscr{H}$ on w end in negative-gap-states, possibly some in 0-gap-states and possibly some in positive-gap states. As both $\mathscr{B}$ and $\mathscr{C}$ do not change the values of runs ending in negative-gap-states and positive-gap-states, while $\mathscr{B}$ decreases the values of runs ending in 0-gap-states and $\mathscr{C}$ increases them, the value of $\mathscr{C}$ on w will be the same as $\mathscr{A}(w)$, while the value of $\mathscr{B}$ on w might be smaller than $\mathscr{A}(w)$ (if the decreased run turned to be a minimal one) or equal to $\mathscr{A}(w)$ (otherwise).

$\square$

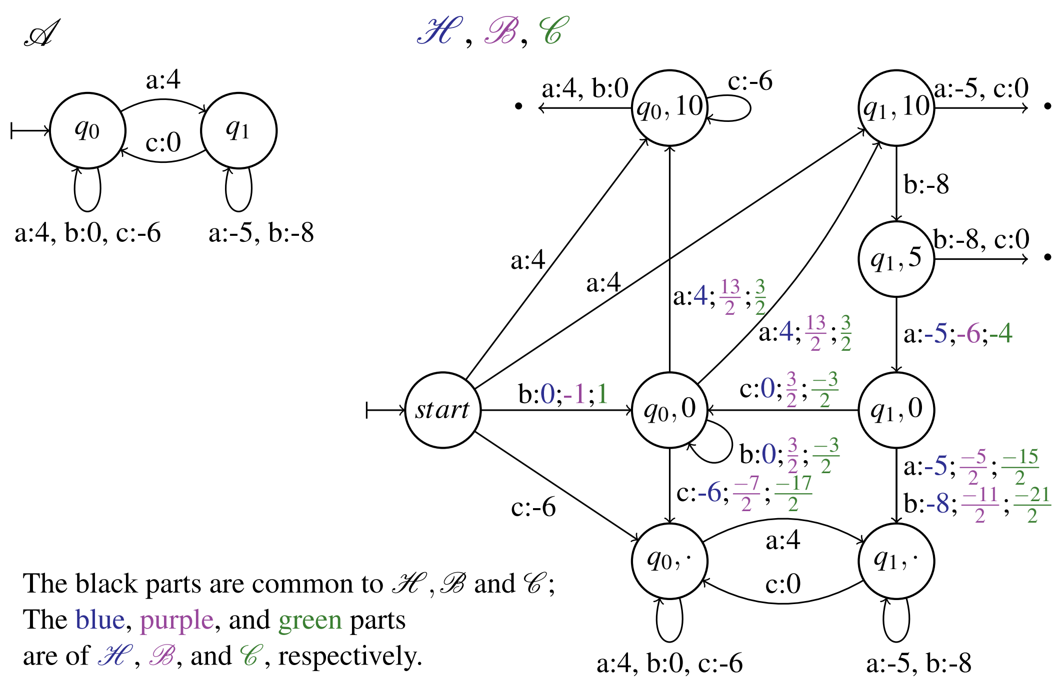

Figure 6: An NDA $\mathscr{A}$ with discount-factor $\lambda = \frac{5}{2}$ and set of weights $W = \{-8,-6,-5,0,4\}$, the λ-NDA $\mathscr{H}$ obtained by the product of $\mathscr{A}$ and the DFA $\mathscr{D}$ of Fig. 5, as per Lemma 3, and the adaptation of $\mathscr{H}$ to λ-NDAs $\mathscr{B}$ and $\mathscr{C}$ "bypassing" the 0-valued finite runs, as per Theorem 4. The edges to the dots ($\cdot$) continue to the state $(q_0,\cdot)$ or to the state $(q_1,\cdot)$; we trim them for the drawing clarity. By Theorem 4, the $>$-nonuniversality and $\geq$-nonuniversality problems of $\mathscr{A}$ reduce, respectively, to the $\geq$-nonuniversality problem of $\mathscr{B}$ and to the $>$-nonuniversality problem of $\mathscr{C}$.

4.4 From Exact-Value to $\geq$-Nonuniversality on Finite Words

Given the exact-value problem with respect to an NDA $\mathscr{A}$, we aim at changing $\mathscr{A}$ to an NDA $\mathscr{P}$, such that there is a finite word u with $\mathscr{A}(u) = 0$ iff there is a finite word u with $\mathscr{P}(u) \geq 0$. Hence, in the modification of $\mathscr{A}$ to $\mathscr{P}$, we need to change the value of $\mathscr{P}$ to be negative on every finite word u for which $\mathscr{A}(u) > 0$. However, we do not know how to recognize such words; we only know to recognize, by Lemma 3, *weight sequences* whose value is 0. Nevertheless, it turns out to suffice for making the modification, as briefly explained below, formally proved in Theorem 5, and exemplified in Fig. 7.

We first build an NFA $\mathscr{J}$ that recognizes words on which $\mathscr{A}$ has a 0-valued *run*, by taking the product of $\mathscr{A}$ and the DFA $\mathscr{D}$ of Lemma 3. (Notice that $\mathscr{A}$ may have the value 0 or negative values on such words.) We then determinize $\mathscr{J}$ into a DFA $\mathscr{K}$, and construct a λ-NDA by the product of $\mathscr{A}$ and $\mathscr{K}$. Observe that this product NDA augments every state q of $\mathscr{A}$ with information on whether words leading to q have 0-valued runs in $\mathscr{A}$ (we mark them by type $\mathsf{Has0}$), can be extended to words having 0-valued runs (marked by $\mathsf{CanLeadTo0}$), or cannot be continued to such words (marked by $\mathsf{CannotLeadTo0}$).

We then modify this product λ-NDA into a λ-NDA $\mathscr{P}$, by making changes to the transition weights, depending on the type of states they go into and out of. The modifications do not change the gaps of $\mathsf{Has0}$ states, while decreasing by the recovery bound B of $\mathscr{A}$ the gaps of $\mathsf{CanLeadTo0}$ states, analogously to the way it is done in the reductions of Section 4.2. As for states of type $\mathsf{CannotLeadTo0}$, we replace them all with a sink state $(\emptyset)$ with a 0-weighted self loop, to which all transitions have weight (-1).

Intuitively, $\mathscr{P}$ fits the bill due to the following: If for a finite word u, we have $\mathscr{A}(u) > 0$, it may either be that u can be continued to a word with a 0-valued run or not. In the former case, there is some run r of $\mathscr{A}$ on u whose gap is recoverable, reaching in $\mathscr{P}$ a state of type $\mathsf{CanLeadTo0}$, in which the gap is reduced by B, making it negative, and witnessing that $\mathscr{P}(u) < 0$. In the latter case, let p be the maximal prefix of u that can be continued to a word with a 0-valued run. Then there is a run r of $\mathscr{A}$ on u, such that on its $|p|$ step, it has a recoverable value, while reaching in $\mathscr{P}$ a state of type $\mathsf{Has0}$ or $\mathsf{CanLeadTo0}$. Hence, its gap in $\mathscr{P}$ is either 0 or negative, and on the next step it goes via a (-1)-weighted transition to the sink state, hence having a negative value, which will forever remain so due to the 0-weighted self loop of the sink state.

Theorem 5. *If $\geq$-$\mathsf{NonUniversality}_F$ is decidable then so is $\mathsf{ExactValue}_F$.*

Proof. Consider an NDA $\mathscr{A}$ over an alphabet Σ with discount factor λ, states Q, initial state $\iota_{\mathscr{A}}$, and recovery bound B, and let W be the set of its transition weights.

We will construct a λ-NDA $\mathscr{P}$ over Σ, such that there is a finite word on which $\mathscr{A}$ has value 0 iff there is a finite word on which $\mathscr{P}$ has value at least 0. For building $\mathscr{P}$, we first construct a DFA $\mathscr{D}$, an NFA $\mathscr{J}$, and a DFA $\mathscr{K}$, as described below and demonstrated in Figs. 5 and 7.

The DFA $\mathscr{D}$.

We construct a DFA $\mathscr{D}$, as per Lemma 3, recognizing the finite sequences in W^* whose λ-discounted-sum is 0. Let D denote $\mathscr{D}$'s states and $\iota_{\mathscr{D}}$ its initial, and only accepting, state. (See an example in Fig. 5, without the dead-end state $(\cdot)$.) If $\mathscr{D}$ only accepts the empty word, we do not need to continue the construction and reduction, as there is no finite word on which $\mathscr{A}$ has value 0.

The NFA $\mathscr{J}$.

We then construct an NFA $\mathscr{J} = \langle \Sigma, J, \iota_J, \Delta_J, F_J \rangle$ by taking the product of $\mathscr{A}$ and $\mathscr{D}$, having the states $J = Q \times D$, initial state $\iota_J = \langle \iota_q, \iota_d \rangle$, transitions $\Delta_J = \{ \langle q, d \rangle \to \langle q', d' \rangle \mid$ there are transitions $q \xrightarrow{\sigma:x} q'$ in $\mathscr{A}$ and $d \xrightarrow{x} d'$ in $\mathscr{D}\}$, and final states $F_J = Q \times \{\iota_{\mathscr{D}}\}$. Notice that $\mathscr{J}$ recognizes the words in Σ^*, for which there exist runs of $\mathscr{A}$ with value 0. (The value of $\mathscr{A}$ on these words need not be 0, as it may have other runs on them with smaller values.)

The DFA $\mathscr{K}$.

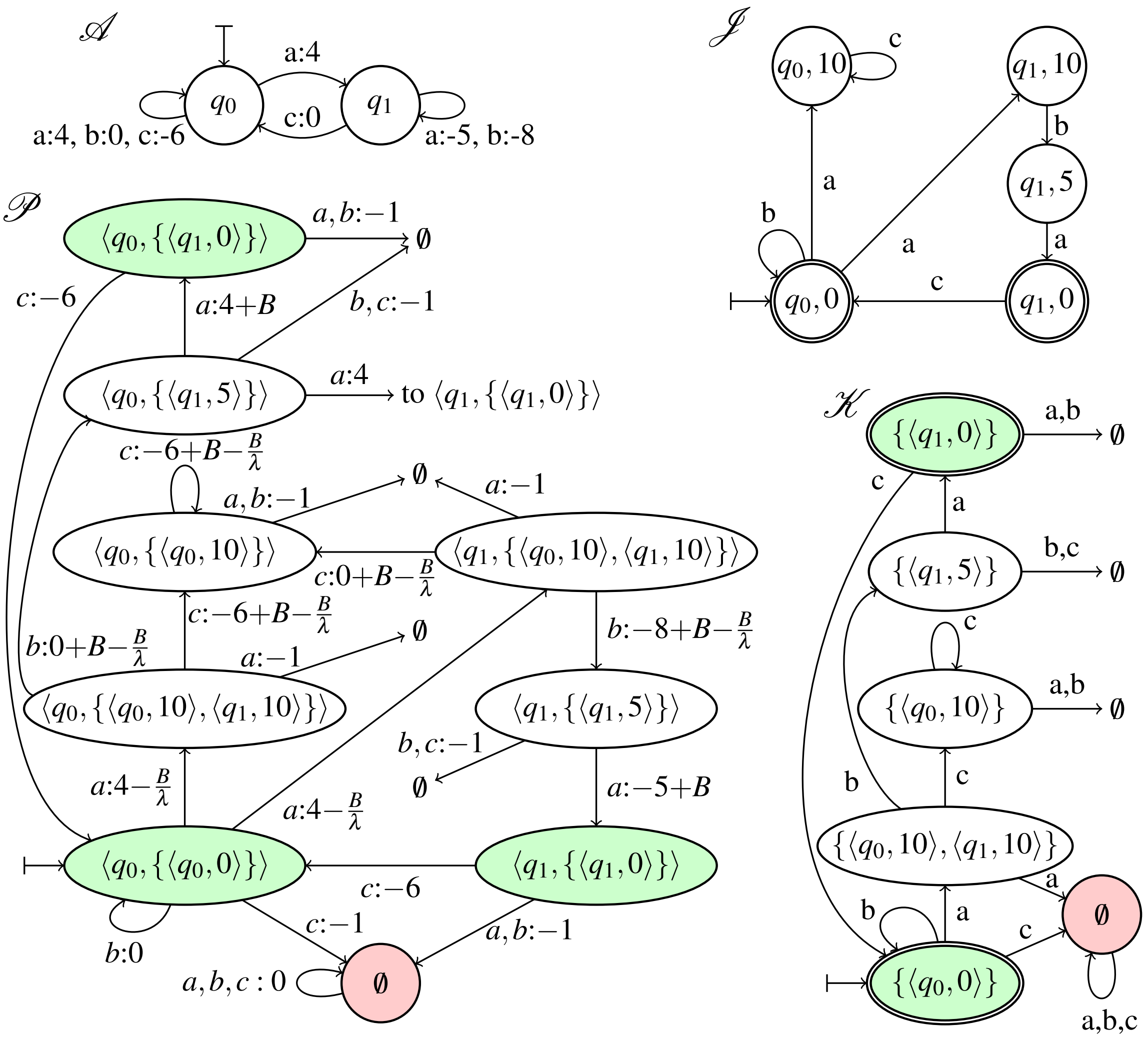

Figure 7: Automata constructions along the proof of Theorem 5. Given a $\frac{5}{2}$-NDA $\mathscr{A}$ over an alphabet Σ with set of weights W and recovery bound B ($13\frac{1}{3}$ in this case), we construct a DFA $\mathscr{D}$ as in Fig. 5 (without the dead-end state $(\cdot)$), and take its product with $\mathscr{A}$ to obtain the NFA $\mathscr{J}$, recognizing words in Σ^* on which $\mathscr{A}$ has a 0-valued run. We determinize it to the DFA $\mathscr{K}$, and obtain the λ-NDA $\mathscr{P}$ by certain modifications of the product of $\mathscr{A}$ and $\mathscr{K}$. The green states in $\mathscr{K}$ and $\mathscr{P}$ are of type $\mathsf{Has0}$, the uncolored ones of type $\mathsf{CanLeadTo0}$ and the red states of type $\mathsf{CannotLeadTo0}$. By Theorem 5, there is a finite word u such that $\mathscr{A}(u) = 0$ iff there is a finite word u such that $\mathscr{P}(u) \geq 0$.

Next, we construct from $\mathscr{J}$ an equivalent DFA $\mathscr{K}$ with states K and initial state ι_K along the subset construction. Observe that every state of $\mathscr{K}$ is a set of pairs $\langle q, d\rangle$, where q is a state of $\mathscr{A}$ and d is a state of $\mathscr{D}$. We divide the states of $\mathscr{K}$ into three disjoint types, where a state S is of type

- $\mathsf{Has0}$ if it is accepting;
- $\mathsf{CanLeadTo0}$ if it is not accepting and not the empty-set state;
- $\mathsf{CannotLeadTo0}$ if it is the empty-set state.

Observe that a state k of $\mathscr{K}$ is of type

- $\mathsf{Has0}$ iff for every word $u \in \Sigma^*$, such that $\mathscr{K}(u) = S$, there exists a 0-valued run of $\mathscr{A}$ on u;
- $\mathsf{CanLeadTo0}$ iff for every word $u \in \Sigma^*$, such that $\mathscr{K}(u) = S$, there is no 0-valued run of $\mathscr{A}$ on u, while there exists a word $v \in \Sigma^+$, such that there is a 0-valued run of $\mathscr{A}$ on $u \cdot v$;
- $\mathsf{CannotLeadTo0}$ iff for every word $u \in \Sigma^*$, such that $\mathscr{K}(u) = S$, there is no word $v \in \Sigma^*$, such that there is a 0-valued run of $\mathscr{A}$ on $u \cdot v$.

The NDA $\mathscr{P}$.

Finally, we construct a λ-NDA $\mathscr{P}$ over Σ by taking the product of $\mathscr{A}$ and $\mathscr{K}$, and making some changes in it. In the product of $\mathscr{A}$ and $\mathscr{K}$, we have the states $Q \times K$, initial state $\langle \iota_{\mathscr{A}}, \iota_K \rangle$, and transitions $\langle q,k \rangle \xrightarrow{\sigma:x} \langle q',k' \rangle$, such that there are transitions $q \xrightarrow{\sigma:x} q'$ in $\mathscr{A}$ and $k \xrightarrow{\sigma} k'$ in $\mathscr{K}$. We consider the type of a state ($\mathsf{Has0}$, $\mathsf{CanLeadTo0}$, $\mathsf{CannotLeadTo0}$) according to the type of its K's component.

We then modify the transitions, such that the gaps of some states will change, depending on their types. For states of type $\mathsf{Has0}$ the gaps will not change, for states of type $\mathsf{CanLeadTo0}$, their gaps will be B below their original gaps, and states of type $\mathsf{CannotLeadTo0}$ will be unified into a sink state, denoted by $(\emptyset)$, into which the transitions will have weight (-1). Formally, for every transition $\langle q,k \rangle \xrightarrow{\sigma:x} \langle q',k' \rangle$:

- If $k \in \mathsf{Has0}$ and $k' \in \mathsf{CanLeadTo0}$, we change the weight from x to $x - \frac{B}{\lambda}$.
- If $k,k' \in \mathsf{CanLeadTo0}$, we change the weight from x to $x + B - \frac{B}{\lambda}$.
- If $k \in \mathsf{CanLeadTo0}$ and $k' \in \mathsf{Has0}$, we change the weight from x to $x + B$.
- If $k' \in \mathsf{CannotLeadTo0}$, we replace that transition with the transition $\langle q,k \rangle \xrightarrow{\sigma:-1} (\emptyset)$.

Correctness.

Observe first that since the $\mathscr{K}$ component of $\mathscr{P}$ is deterministic, the runs of $\mathscr{P}$ on any word u are, state-wise, the same as $\mathscr{A}$'s runs, when projecting the states of $\mathscr{P}$ to their $\mathscr{A}$ component. By arguments analogous to those given in the proof of Theorem 3, the gaps along runs of $\mathscr{P}$ are the same as their counterparts in $\mathscr{A}$, when reaching a state of type $\mathsf{Has0}$, and are smaller by B when reaching a state of type $\mathsf{CanLeadTo0}$.

We claim that there is a finite word on which $\mathscr{A}$ has value 0 iff there is a finite word on which $\mathscr{P}$ has value at least 0. Indeed:

$\Rightarrow$: Consider a word $u \in \Sigma^*$, such that $\mathscr{A}(u) = 0$. Let r be a run of $\mathscr{A}$ on u with value 0, and let $\rho \in W^*$ be the sequence of weights resulting from r. By the constuction of $\mathscr{D}$, we have that it accepts ρ. Since $\mathscr{J}$ is the product of $\mathscr{A}$ and $\mathscr{D}$, and $\mathscr{K}$ is equivalent to $\mathscr{J}$, we get that $\mathscr{K}$ accepts u.

Hence, all runs of $\mathscr{P}$ on u ends in states of type $\mathsf{Has0}$, and by the arguments above, have the same values as in $\mathscr{A}$. Thus, $\mathscr{P}(u) = 0$, and in particular $\mathscr{P}(u) \geq 0$.

$\Leftarrow$: Consider a word $u \in \Sigma^*$, such that $\mathscr{A}(u) \neq 0$. We split the arguments into disjoint cases, depending on the type of state in $\mathscr{K}$ that u leads to:

- $\mathsf{Has0}$: Since there is a run of $\mathscr{A}$ on u with value 0, but $\mathscr{A}(u) \neq 0$, it follows that $\mathscr{A}(u) < 0$. By the arguments above, the runs of $\mathscr{P}$ and of $\mathscr{A}$ on u have the same values, thus $\mathscr{P}(u) < 0$.
- $\mathsf{CanLeadTo0}$: Let r be a run of $\mathscr{A}$ on u that can be continued to have value 0. Then by the arguments above, the gap (and value) of the corresponding run of $\mathscr{P}$ on u is negative (regardless of whether the value of r is positive or negative). Hence, $\mathscr{P}(u) < 0$.

- $\mathsf{CannotLeadTo0}$: Let p be the longest prefix of u that has a 0-valued run in $\mathscr{A}$ or can be continued to a word on which $\mathscr{A}$ has a 0-valued run. Notice that since the empty word falls into this definition, there exists such a prefix p, and since u does not have this property, p is a proper prefix of u. Let r be a 0-valued run of $\mathscr{A}$ on p in the former case, and a recoverable run in the latter case. Then the corresponding run $\hat{r}$ of $\mathscr{P}$ ends in a state of type $\mathsf{Has0}$ and has a 0 gap in the former case, and ends in a state of type $\mathsf{CanLeadTo0}$ and has a negative gap in the latter case. Hence, its continuation on the suffix of u from p onwards starts with the (-1)-transition to the sink state ($\emptyset$), and continues there with the 0-weighted self loop. Hence, the resulting run of $\mathscr{P}$ on u has a negative value, implying that $\mathscr{P}(u) < 0$.

$\square$

5 Conclusions

In quantitative verification, the two most central measures are limit-average (mean payoff) and discounted-sum, used for system properties [2, 20, 33], automata valuation schemes [23, 12, 13, 8], game conditions [37, 6, 27], and temporal specifications [1, 36, 3, 10].

For limit-average automata on infinite words, the strict nonuniversality problem is undecidable [27], whereas the decidability of the non-strict variant remains open. For discounted-sum automata, decidability is open for both the nonuniversality and exact-value problems, under strict as well as non-strict comparison, on both finite and infinite words.

Our reductions may help clarify the source of difficulty underlying these decision problems, and possibly contribute to resolving the four mutually interreducible cases: exact-value, $\geq$-nonuniversality, and $>$-nonuniversality on finite words, together with $>$-nonuniversality on infinite words. These problems appear to differ in nature from the exact-value and $\geq$-nonuniversality problems on infinite words, to which the TDS problem reduces.

References

[1] Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar & Mariëlle Stoelinga (2005): *Model checking discounted temporal properties*. Theor. Comput. Sci. 345(1), pp. 139–170, doi:10.1016/j.tcs.2005.07.033.

[2] Luca de Alfaro, Thomas A. Henzinger & Rupak Majumdar (2003): *Discounting the Future in Systems Theory*. In: proceedings of ICALP, 2719, pp. 1022–1037, doi:10.1007/3-540-45061-0_79.

[3] Shaull Almagor, Udi Boker & Orna Kupferman (2014): *Discounting in LTL*. In: proceedings of TACAS, LNCS 8413, pp. 424–439, doi:10.1007/978-3-642-54862-8_37.

[4] Shaull Almagor & Neta Dafni (2024): *Determinization of Integral Discounted-Sum Automata is Decidable*. In: proceedings of FoSSaCS, Springer, pp. 191–211, doi:10.1007/978-3-031-57228-9_10.

[5] Rajeev Alur, Suguman Bansal, Osbert Bastani & Kishor Jothimurugan (2022): *A Framework for Transforming Specifications in Reinforcement Learning*. In Jean-François Raskin, Krishnendu Chatterjee, Laurent Doyen & Rupak Majumdar, editors: Principles of Systems Design - Essays Dedicated to Thomas A. Henzinger on the Occasion of His 60th Birthday, Lecture Notes in Computer Science 13660, Springer, pp. 604–624, doi:10.1007/978-3-031-22337-2_29.

[6] Daniel Andersson (2006): *An improved algorithm for discounted payoff games*. In: proceedings of ESSLLI Student Session, pp. 91–98.

[7] Suguman Bansal (2023): *Automata-based quantitative reasoning*. ACM SIGLOG News 10(3), pp. 4–19, doi:10.1145/3617576.3617578.

[8] Suguman Bansal, Swarat Chaudhuri & Moshe Y. Vardi (2022): *Comparator automata in quantitative verification*. Log. Methods Comput. Sci. 18(3), doi:10.46298/LMCS-18(3:13)2022.

[9] Udi Boker (2024): *Discounted-Sum Automata with Real-Valued Discount Factors*. In: Proc. of LICS, ACM, pp. 15:1–15:14.

[10] Udi Boker, Krishnendu Chatterjee, Thomas A. Henzinger & Orna Kupferman (2014): *Temporal Specifications with Accumulative Values*. ACM Trans. Comput. Log. 15(4), pp. 27:1–27:25, doi:10.1145/2629686.

[11] Udi Boker & Guy Hefetz (2025): *Discounted-Sum Automata with Multiple Discount Factors*. Log. Methods Comput. Sci. 21(2), doi:10.46298/LMCS-21(2:18)2025.

[12] Udi Boker & Thomas A. Henzinger (2012): *Approximate Determinization of Quantitative Automata*. In: proceedings of FSTTCS, LIPIcs 18, pp. 362–373, doi:10.4230/LIPIcs.FSTTCS.2012.362.

[13] Udi Boker & Thomas A. Henzinger (2014): *Exact and Approximate Determinization of Discounted-Sum Automata*. Log. Methods Comput. Sci. 10(1), doi:10.2168/LMCS-10(1:10)2014.

[14] Udi Boker, Thomas A. Henzinger & Jan Otop (2015): *The Target Discounted-Sum Problem*. In: Proc. of LICS, pp. 750–761. An updated version in `https://arxiv.org/abs/2511.22979`.

[15] Olivier Bournez, Oleksiy Kurganskyy & Igor Potapov (2018): *Reachability Problems for One-Dimensional Piecewise Affine Maps*. Int. J. Found. Comput. Sci. 29(4), pp. 529–549, doi:10.1142/S0129054118410046.

[16] Patricia Bouyer, Mickael Randour & Pierre Vandenhove (2023): *Characterizing omega-regularity through finite-memory determinacy of games on infinite graphs*. TheoretiCS 2, doi:10.46298/THEORETICS.23.1.

[17] Léonard Brice, Jean-François Raskin & Marie van den Bogaard (2023): *Rational Verification for Nash and Subgame-Perfect Equilibria in Graph Games*. In Jérôme Leroux, Sylvain Lombardy & David Peleg, editors: proceedings of MFCS, LIPIcs 272, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 26:1–26:15, doi:10.4230/LIPICS.MFCS.2023.26.

[18] Véronique Bruyère (2021): *Synthesis of equilibria in infinite-duration games on graphs*. ACM SIGLOG News 8(2), pp. 4–29, doi:10.1145/3467001.3467003.

[19] Marek Chalupa, Thomas A Henzinger, Nicolas Mazzocchi & N Ege Saraç (2025): *Automating the analysis of quantitative automata with quak*. In: proceedings of TACAS, Springer, pp. 303–312, doi:10.1007/978-3-031-90643-5_16.

[20] Krishnendu Chatterjee (2007): *Markov Decision Processes with Multiple Long-Run Average Objectives*. In: proceedings of FSTTCS, LNCS 4855, Springer, pp. 473–484, doi:10.1007/978-3-540-77050-3_39.

[21] Krishnendu Chatterjee, Laurent Doyen & Thomas A. Henzinger (2009): *Alternating Weighted Automata*. In: proceedings of FCT, LNCS 5699, pp. 3–13, doi:10.1007/978-3-642-03409-1_2.

[22] Krishnendu Chatterjee, Laurent Doyen & Thomas A. Henzinger (2010): *Expressiveness and Closure Properties for Quantitative Languages*. Log. Methods Comput. Sci. 6(3). Available at `http://arxiv.org/abs/1007.4018`.

[23] Krishnendu Chatterjee, Laurent Doyen & Thomas A. Henzinger (2010): *Quantitative languages*. ACM Trans. Comput. Log. 11(4), pp. 23:1–23:38, doi:10.1145/1805950.1805953.

[24] Krishnendu Chatterjee, Laurent Doyen & Thomas A. Henzinger (2017): *The Cost of Exactness in Quantitative Reachability*. In: Models, Algorithms, Logics and Tools - Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday, Lecture Notes in Computer Science 10460, Springer, pp. 367–381, doi:10.1007/978-3-319-63121-9_18.

[25] Krishnendu Chatterjee, Vojtech Forejt & Dominik Wojtczak (2013): *Multi-objective Discounted Reward Verification in Graphs and MDPs*. In: proceedings of LPAR, LNCS 8312, pp. 228–242, doi:10.1007/978-3-642-45221-5_17.

[26] Krishnendu Chatterjee, Petr Novotnỳ, Guillermo Pérez, Jean-François Raskin & Đorđe Žikelić (2017): *Optimizing expectation with guarantees in POMDPs*. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 31.

[27] Aldric Degorre, Laurent Doyen, Raffaella Gentilini, Jean-François Raskin & Szymon Toruńczyk (2010): *Energy and Mean-Payoff Games with Imperfect Information*. In: *proceedings of CSL*, *LNCS* 6247, pp. 260–274, doi:10.1007/978-3-642-15205-4_22.

[28] Manfred Droste & Dietrich Kuske (2006): *Skew and infinitary formal power series*. *Theor. Comput. Sci.* 366(3), pp. 199–227, doi:10.1016/j.tcs.2006.08.024.

[29] Emmanuel Filiot, Christof Löding & Sarah Winter (2020): *Synthesis from Weighted Specifications with Partial Domains over Finite Words*. In: *Proc. of (FSTTCS 2020)*, pp. 46:1–46:16, doi:10.4230/LIPIcs.FSTTCS.2020.46.

[30] Faraz Ghahremani, Edon Kelmendi & Joël Ouaknine (2023): *Reachability in Injective Piecewise Affine Maps*. In: *proceedings of LICS*, pp. 1–11, doi:10.1109/LICS56636.2023.10175723.

[31] Hugo Gimbert & Wieslaw Zielonka (2007): *Limits of Multi-Discounted Markov Decision Processes*. In: *proceedings of LICS*, pp. 89–98, doi:10.1109/LICS.2007.28.

[32] Paul Hunter, Guillermo A. Pérez & Jean-François Raskin (2016): *Minimizing Regret in Discounted-Sum Games*. In Jean-Marc Talbot & Laurent Regnier, editors: *CSL*, *LIPIcs* 62, pp. 30:1–30:17, doi:10.4230/LIPIcs.CSL.2016.30.

[33] Omid Madani, Mikkel Thorup & Uri Zwick (2010): *Discounted deterministic Markov decision processes and discounted all-pairs shortest paths*. *ACM Trans. Algorithms* 6(2), pp. 33:1–33:25, doi:10.1145/1721837.1721849.

[34] Senthil Rajasekaran, Suguman Bansal & Moshe Y. Vardi (2023): *Multi-Agent Systems with Quantitative Satisficing Goals*. In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pp. 280–288, doi:10.24963/IJCAI.2023/32.

[35] Richard S. Sutton & Andrew G.Barto (1998): *Introduction to Reinforcement Learning*. MIT Press. Available at `http://dl.acm.org/doi/book/10.5555/551283`.

[36] Takashi Tomita, Shin Hiura, Shigeki Hagihara & Naoki Yonezaki (2012): *A Temporal Logic with Mean-Payoff Constraints*. In: *proceedings of ICFEM*, *LNCS* 7635, Springer, pp. 249–265, doi:10.1007/978-3-642-34281-3_19.

[37] Uri Zwick & Mike Paterson (1996): *The Complexity of Mean Payoff Games on Graphs*. *Theor. Comput. Sci.* 158, pp. 343–359, doi:10.1016/0304-3975(95)00188-3.