The Interdisciplinary Center, Herzlia
Efi Arazi School of Computer Science
M.Sc. program - Research Track

Discounted-Sum Automata with Multiple Discount Factors

by
Guy Hefetz

M.Sc. dissertation, submitted in partial fulfillment of the requirements for the M.Sc. degree, research track, School of Computer Science
The Interdisciplinary Center, Herzliya

June 2020

This work was carried out under the supervision of Prof. **Udi Boker** from the Efi Arazi School of Computer Science, The Interdisciplinary Center, Herzliya.

Abstract

Discounting the influence of future events is a key paradigm in economics and it is widely used in computer-science models, such as games, Markov decision processes (MDPs), and automata. While a single game or MDP may allow for several different discount factors, discounted-sum automata (NDAs) were only studied with respect to a single discount factor. For every integer $\lambda \in \mathbb{N} \setminus \{0,1\}$, as opposed to every $\lambda \in \mathbb{Q} \setminus \mathbb{N}$, the class of NDAs with discount factor λ (λ-NDAs) has good computational properties: it is closed under determinization and under the algebraic operations min, max, addition, and subtraction, and there are algorithms for its basic decision problems, such as automata equivalence and containment.

We define and analyze discounted-sum automata in which each transition can have a different integral discount factor (integral *NMDAs*). We show that integral NMDAs with an arbitrary choice of discount factors are not closed under determinization and under algebraic operations. We then define and analyze a restricted class of integral NMDAs, which we call *tidy NMDAs*, in which the choice of discount factors depends on the prefix of the word read so far. Some of their special cases are NMDAs that correlate discount factors to actions (alphabet letters) or to the elapsed time. We show that for every function θ that defines the choice of discount factors, the class of θ-NMDAs enjoys all of the above good properties of integral NDAs, as well as the same complexity of the required decision problems. Tidy NMDAs are also as expressive as deterministic integral NMDAs with an arbitrary choice of discount factors. We conclude with analyzing the relation between the different classes of tidy NMDAs, demonstrating the importance of each of the classes: No class is strictly contained in another, union of any two inequivalent classes ruins closure under algebraic operations, and the intersection of all classes is exactly the set of eventually constant functions.

All of our results hold for both automata on finite words and automata on infinite words.

Contents

List of Figures

1 Introduction

Discounted summation is a central valuation function in various computational models, such as games (e.g., [37, 2, 23]), Markov decision processes (e.g, [28, 31, 18]), and automata (e.g, [25, 14, 16, 17]), as it formalizes the concept that an immediate reward is better than a potential one in the far-away future, as well as that a potential problem (such as a bug in a reactive system) in the far away future is less troubling than a current one.

A Nondeterministic Discounted-sum Automaton (NDA) is an automaton with rational weights on the transitions, and a fixed rational discount factor $\lambda > 1$. The value of a (finite or infinite) run is the discounted summation of the weights on the transitions, such that the weight in the ith position of the run is divided by λ^i. The value of a (finite or infinite) word is the minimal value of the automaton runs on it. An NDA $\mathcal{A}$ realizes a function from words to real numbers, and we write $\mathcal{A}(w)$ for the value of $\mathcal{A}$ on a word w.

In the Boolean setting, where automata realize languages, closure under the basic Boolean operations of union, intersection, and complementation is desirable, as it allows to use automata in formal verification, logic, and more. In the quantitative setting, where automata realize functions from words to numbers, these Boolean operations are naturally generalized to algebraic ones: union to min, intersection to max, and complementation to multiplication by -1 (depending on the function's co-domain). Likewise, closure under these algebraic operations, as well as under addition and subtraction, is desirable for quantitative automata, serving for quantitative verification. Determinization is also very useful in automata theory, as it gives rise to many algorithmic solutions, and is essential for various tasks, such as synthesis and probabilistic model checking[1].

NDAs cannot always be determinized [17], they are not closed under basic algebraic operations [9], and basic decision problems on them, such as universality, equivalence, and containment, are not known to be decidable and relate to various longstanding open problems [10]. However, restricting NDAs to an integral discount factor $\lambda \in \mathbb{N}$ provides a robust class of automata that is closed under determinization and under the algebraic operations, and for which the decision problems of universality equivalence, and containment are decidable [9].

Various variants of NDAs are studied in the literature, among which are *functional*, *k-valued*, *probabilistic*, and more [27, 26, 15]. Yet, to the best of our knowledge, all of these models are restricted to have a single discount factor in an automaton. This is a significant restriction of the general discounted-summation paradigm, in which multiple discount factors are considered. For example, Markov decision processes and discounted-sum games allow for multiple discount factors within the same entity [28, 2]. As automata are aimed at modeling systems, allowing different discount factors on different transitions extends the system behaviors that can be modeled. For example, it allows to

[1] In some cases, automata that are "almost deterministic", such as limit-deterministic [36] or good-for-games automata [29, 11] suffice.

better model how the value of used vehicles changes over time, having a bigger discount factor in the first year, slightly smaller in the next couple of years, and significantly smaller in further years.

A natural extension to NDAs is to allow for different discount factors over the transitions, providing the ability to model systems in which each action (alphabet letter in the automaton) causes a different discounting, systems in which the discounting changes over time, and more. As integral NDAs provide robust automata classes, whereas non-integral NDAs do not, we look into extending integral NDAs into integral *NMDAs* (Definition 1), allowing multiple integral discount factors in a single automaton.

We start with analyzing NMDAs in which the integral discount factors can be chosen arbitrarily. Unfortunately, we show that this class of automata does not allow for determinization and is not closed under the basic algebraic operations.

For more restricted generalizations of integral NDAs, in which the discount factor depends on the transition's letter (*letter-oriented* NMDAs) or on the elapsed time (*time-oriented* NMDAs), we show that the corresponding automata classes do enjoy all of the good properties of integral NDAs, while strictly extending their expressiveness.

We further analyze a rich class of integral NMDAs that extends both letter-oriented and time-oriented NMDAs, in which the choice of discount factor depends on the word-prefix read so far (*tidy* NMDAs). We show that their expressiveness is as of deterministic integral NMDAs with an arbitrary choice of discount factors and that for every choice function $\theta : \Sigma^+ \to \mathbb{N} \setminus \{0, 1\}$, the class of θ-NMDAs enjoys all of the good properties of integral NDAs. (See Fig. 1.)

Looking into the structure of the family of tidy NMDAs, we provide evidence to support the importance of each of the classes: No class is strictly contained in another, union of any two inequivalent classes ruins closure under algebraic operations, and the intersection of all classes is exactly the set of eventually constant functions.

As general choice functions need not be finitely represented, it might upfront limit the usage of tidy NMDAs. Yet, we show that finite transducers (Mealy machines) suffice, in the sense that they allow to represent every choice function θ that can serve for a θ-NMDA. We provide a PTIME algorithm to check whether a given NMDA is tidy, as well as if it is a $\mathcal{T}$-NMDA for a given transducer $\mathcal{T}$.

Considering the decision problems of tidy NMDAs, we provide a PTIME algorithm for emptiness and PSPACE algorithms for the other problems of exact-value, universality, equivalence, and containment. The complexities are with respect to the automaton (or automata) size, which is considered as the maximum between the number of transitions and the maximal binary representation of any discount factor or weight in it. For rational weights, we assume all of them to have the same denominator. (Omitting this assumption changes in the worst case the PSPACE algorithms into EXPSPACE ones.)

We show all of our results for both automata on finite words and automata on infinite words. Whenever possible, we provide a single proof for both settings.

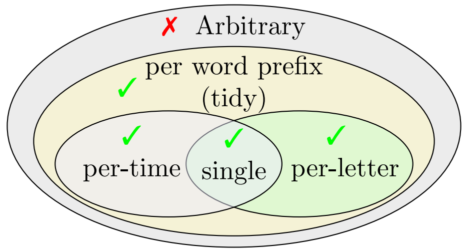

Figure 1: Classes of integral NMDAs, defined according to the flexibility of choosing the discount factors. The class of NMDAs with arbitrary integral factors is not closed under algebraic operations and under determinization. The other classes are (for a specific choice function). Tidy NMDAs are as expressive as deterministic NMDAs with arbitrary integral discount factors.

2 Discounted-Sum Automata with Multiple Integral Discount Factors

We define a discounted-sum automaton with arbitrary discount factors, abbreviated NMDA, by adding to an NDA a discount factor in each of its transitions. An NMDA is defined on either finite or infinite words. The formal definition is given in Definition 1, and an example in Fig. 2.

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 denote the concatenation of a finite word u and a finite or infinite word w by $u \cdot w$, or simply by uw. We define Σ^+ to be the set of all finite words except the empty word, i.e., $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$. 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 $w[i..j] = w(i)w(i+1)\dots w(j)$.

Definition 1. *A nondeterministic discounted-sum automaton with multiple discount factors (NMDA), on finite or infinite words, is a tuple $\mathcal{A} = \langle \Sigma, Q, \iota, \delta, \gamma, \rho \rangle$ over an alphabet Σ, with a finite set of states Q, an initial set of states $\iota \subseteq Q$, a transition function $\delta \subseteq Q \times \Sigma \times Q$, a weight function $\gamma : \delta \to \mathbb{Q}$, and a discount-factor function $\rho : \delta \to \mathbb{Q} \cap (1, \infty)$, assigning to each transition its discount factor, which is a rational greater than 1.* [2]

- *A* walk *in $\mathcal{A}$ from a state p_0 is a sequence of states and alphabet letters, $p_0, \sigma_0, p_1, \sigma_1, p_2, \cdots$, such that for every i, $(p_i, \sigma_i, p_{i+1}) \in \delta$.*
  For example, $\psi = q_1, a, q_1, b, q_2$ is a walk of the NMDA $\mathcal{A}$ of Fig. 2 on the word ab from the state q_1 .

- *A run of $\mathcal{A}$ is a walk from an initial state.*

- *The length of a walk ψ, denoted by $|\psi|$, is n for a finite walk $\psi = p_0, \sigma_0, p_1, \cdots, \sigma_{n-1}, p_n$, and ∞ for an infinite walk.*

[2] Discount factors are sometimes defined in the literature as numbers between 0 and 1, under which setting weights are multiplied by these factors rather than divided by them.

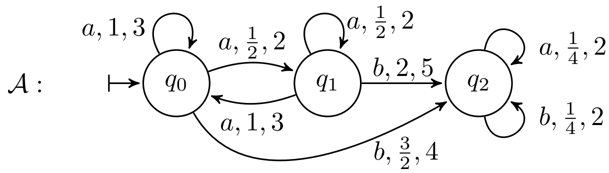

Figure 2: An NMDA $\mathcal{A}$. The labeling on the transitions indicate the alphabet letter, the weight of the transition, and its discount factor.

- *The i-th transition of a walk* $\psi = p_0, \sigma_0, p_1, \sigma_1, \cdots$ *is denoted by* $\psi(i) = (p_i, \sigma_i, p_{i+1})$.
- *The* value *of a finite or an infinite walk* ψ *is* $\mathcal{A}(\psi) = \sum_{i=0}^{|\psi|-1} \left(\gamma\big(\psi(i)\big) \cdot \prod_{j=0}^{i-1} \frac{1}{\rho\big(\psi(j)\big)} \right)$.
  For example, the value of the walk $r_1 = q_0, a, q_0, a, q_1, b, q_2$ *(which is also a run) of* $\mathcal{A}$ *from Fig. 2 is* $\mathcal{A}(r_1) = 1 + \frac{1}{2} \cdot \frac{1}{3} + 2 \cdot \frac{1}{2 \cdot 3} = \frac{3}{2}$.
- *The* value *of* $\mathcal{A}$ *on a finite or infinite word* w *is* $\mathcal{A}(w) = \inf\{\mathcal{A}(r) \mid r \text{ is a run of } \mathcal{A} \text{ on } w\}$.
- *In the case where* $|\iota| = 1$ *and for every* $q \in Q$ *and* $\sigma \in \Sigma$*, we have* $|\{q' \mid (q, \sigma, q') \in \delta\}| \leq 1$*, we say that* $\mathcal{A}$ *is* deterministic*, denoted by DMDA.*

  In this case we use $\delta(q, \sigma)$ *for the target state of the transition from* q *over the* σ *letter, and use* $\gamma(q, \sigma)$ *and* $\rho(q, \sigma)$ *for the weight and discount factor of that transition.*
- *When all the discount factors are integers, we say that* $\mathcal{A}$ *is an* integral *NMDA.*

In the case where 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*. In this work, we only consider complete automata. 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.

Automata $\mathcal{A}$ and $\mathcal{A}'$ are *equivalent*, denoted by $\mathcal{A} \equiv \mathcal{A}'$, if for every word w, $\mathcal{A}(w) = \mathcal{A}'(w)$.

For every finite (infinite) walk $\psi = p_0, \sigma_0, p_1, \cdots, \sigma_{n-1}, p_n$ ($\psi = p_0, \sigma_0, p_1, \cdots$), and all integers $0 \leq i \leq j \leq |\psi| - 1$ ($0 \leq i \leq j$), we define the finite sub-walk from i to j as $\psi[i..j] = p_i, \sigma_i, p_{i+1}, \cdots, \sigma_j, p_{j+1}$. For an infinite walk, we also define $\psi[i..\infty] = p_i, \sigma_i, p_{i+1}, \cdots$, namely the infinite suffix from position i. For a finite walk, we also define the target state as $\delta(\psi) = p_n$ and the accumulated discount factor as $\rho(\psi) = \prod_{i=0}^{n-1} \rho\big(\psi(i)\big)$.

We extend the transition function δ to finite words in the regular manner: For a word $u \in \Sigma^*$ and a letter $\sigma \in \Sigma$, $\delta(\varepsilon) = \iota; \delta(u \cdot \sigma) = \bigcup_{q \in \delta(u)} \delta(q, \sigma)$. When $\mathcal{A}$ is deterministic, we refer to $\delta(u)$ as a state and not as singleton set, and to $\rho(u)$ as $\rho(r)$, where r is the single run of $\mathcal{A}$ on u. For a state q of $\mathcal{A}$, we denote by $\mathcal{A}^q$ the automaton that is identical to $\mathcal{A}$, except for having q as its single initial state.

An NMDA may have rational weights, yet it is often convenient to consider an analogous NMDA with integral weights, achieved by multiplying all weights by their common denominator.

Proposition 2. *For all constant $0 < m \in \mathbb{Q}$, NMDA $\mathcal{A} = \langle \Sigma, Q, \iota, \delta, \gamma, \rho \rangle$, NMDA $\mathcal{A}' = \langle \Sigma, Q, \iota, \delta, m \cdot \gamma, \rho \rangle$ obtained from $\mathcal{A}$ by multiplying all its weights by m, and a finite or infinite word w, we have $\mathcal{A}'(w) = m \cdot \mathcal{A}(w)$.*

Proof. Let $0 < m \in \mathbb{Q}$, $\mathcal{A} = \langle \Sigma, Q, \iota, \delta, \gamma, \rho \rangle$ and $\mathcal{A}' = \langle \Sigma, Q, \iota, \delta, m \cdot \gamma, \rho \rangle$ NMDAs, and w a finite or infinite word.

For every run r of $\mathcal{A}$ on w, we have that the same run in $\mathcal{A}'$ has the value of

$$\begin{aligned}\mathcal{A}'(r) &= \sum_{i=0}^{|w|-1} \left(m \cdot \gamma(r(i)) \cdot \prod_{j=0}^{i-1} \frac{1}{\rho(r(j))} \right) \\ &= m \cdot \sum_{i=0}^{|w|-1} \left(\gamma(r(i)) \cdot \prod_{j=0}^{i-1} \frac{1}{\rho(r(j))} \right) = m \cdot \mathcal{A}(r)\end{aligned}$$

Hence for every run of $\mathcal{A}$ with value v_0 we have a run of $\mathcal{A}'$ for the same word with value of $m \cdot v_0$. Symmetrically for every run of $\mathcal{A}'$ with value v_1 we have a run of $\mathcal{A}$ for the same word with value of $\frac{1}{m} \cdot v_1$. So,

$$\begin{aligned}\mathcal{A}'(w) &= \inf\{\mathcal{A}'(r) \mid r \text{ is a run of } \mathcal{A}' \text{ on } w\} \\ &\geq \inf\{m \cdot \mathcal{A}(r) \mid r \text{ is a run of } \mathcal{A} \text{ on } w\} \\ &= m \cdot \inf\{\mathcal{A}(r) \mid r \text{ is a run of } \mathcal{A} \text{ on } w\} = m \cdot \mathcal{A}(w)\end{aligned}$$

and

$$\begin{aligned}\mathcal{A}(w) &= \inf\{\mathcal{A}(r) \mid r \text{ is a run of } \mathcal{A} \text{ on } w\} \\ &\geq \inf\left\{\frac{1}{m} \cdot \mathcal{A}'(r) \mid r \text{ is a run of } \mathcal{A}' \text{ on } w\right\} = \frac{1}{m} \cdot \mathcal{A}'(w)\end{aligned}$$

which leads to $\mathcal{A}'(w) = m \cdot \mathcal{A}(w)$. $\square$

Size. We define the size of $\mathcal{A}$, denoted by $|\mathcal{A}|$, as the maximum between the number of transitions and the maximal binary representation of any discount factor or weight in it. For rational weights, we assume all of them to have the same denominator. The motivation for a common denominator stems from the determinization algorithm (Theorem 11). Omitting this assumption will still result in a deterministic automaton whose size is only single exponential in the size of the original automaton, yet storing its states will require a much bigger space, changing our PSPACE algorithms (Section 4) into EXPSPACE ones.

Algebraic operations. Given automata $\mathcal{A}$ and $\mathcal{B}$ over the same alphabet, and a non-negative scalar $c \in \mathbb{Q}$, we define

- $\mathcal{C} \equiv \min(\mathcal{A}, \mathcal{B})$ if $\forall w.\ \mathcal{C}(w) = \min\big(\mathcal{A}(w), \mathcal{B}(w)\big)$
- $\mathcal{C} \equiv \max(\mathcal{A}, \mathcal{B})$ if $\forall w.\ \mathcal{C}(w) = \max\big(\mathcal{A}(w), \mathcal{B}(w)\big)$
- $\mathcal{C} \equiv \mathcal{A} + \mathcal{B}$ if $\forall w.\ \mathcal{C}(w) = \mathcal{A}(w) + \mathcal{B}(w)$
- $\mathcal{C} \equiv \mathcal{A} - \mathcal{B}$ if $\forall w.\ \mathcal{C}(w) = \mathcal{A}(w) - \mathcal{B}(w)$
- $\mathcal{C} \equiv c \cdot \mathcal{A}$ if $\forall w.\ \mathcal{C}(w) = c \cdot \mathcal{A}(w)$
- $\mathcal{C} \equiv -\mathcal{A}$ if $\forall w.\ \mathcal{C}(w) = -\mathcal{A}(w)$

Decision problems. Given automata $\mathcal{A}$ and $\mathcal{B}$ and a threshold $\nu \in \mathbb{Q}$, we consider the following properties, with strict (or non-strict) inequalities:

- *Nonemptiness.* There exists a word w, s.t. $\mathcal{A}(w) < \nu$ (or $\mathcal{A}(w) \leq \nu$).
- *Exact-value.* There exists a word w, s.t. $\mathcal{A}(w) = \nu$.
- *Universality.* For all words w, $\mathcal{A}(w) < \nu$ (or $\mathcal{A}(w) \leq \nu$).
- *Equivalence.* For all words w, $\mathcal{A}(w) = \mathcal{B}(w)$.
- *Containment.* For all words w, $\mathcal{A}(w) > \mathcal{B}(w)$ (or $\mathcal{A}(w) \geq \mathcal{B}(w)$).

Notice that considering quantitative containment as a generalization of language containment, and defining the "acceptance" of a word w as having a small enough value on it, we define that $\mathcal{A}$ is contained in $\mathcal{B}$ if for every word w, $\mathcal{A}$'s value on w is at least as big as $\mathcal{B}$'s value. (Observe the $>$ and $\geq$ signs in the definition.)

Finite and infinite words. Results regarding NMDAs on finite words that refer to the existence of an equivalent automaton ("positive results") can be extended to NMDAs on infinite words due to Lemma 3 below. Likewise, results that refer to non-existence of an equivalent automaton ("negative results") can be extended from NMDAs on infinite words to NMDAs on finite words. Accordingly, if not stated otherwise, we prove the positive results for automata on finite words and the negative results for automata on infinite words, getting the results for both settings.

Lemma 3. *For all NMDAs $\mathcal{A}$ and $\mathcal{B}$, if for every finite word $u \in \Sigma^+$, we have $\mathcal{A}(u) = \mathcal{B}(u)$, then also for every infinite word $w \in \Sigma^\omega$, we have $\mathcal{A}(w) = \mathcal{B}(w)$.*

The proof is a simple extension of the proof of a similar lemma in [9] with respect to NDAs.

Notice that the converse does not hold, namely there are automata equivalent w.r.t. infinite words, but not w.r.t. finite words. A simple example for this case is the automata $\mathcal{A}$ and $\mathcal{B}$ depicted in Fig. 5.

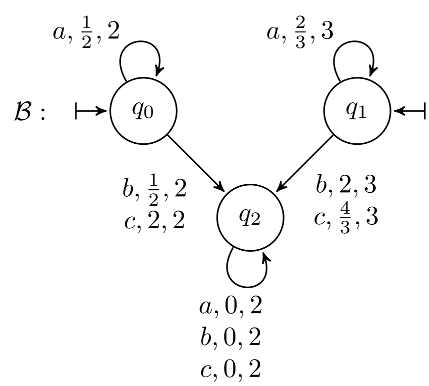

Figure 3: An integral NMDA $\mathcal{B}$ on infinite words that cannot be determinized.

3 Arbitrary Integral NMDAs

Unfortunately, we show that the family of integral NMDAs in which discount factors can be chosen arbitrarily is not closed under determinization and under basic algebraic operations.

Theorem 4. *There exists an integral NMDA that no integral DMDA is equivalent to.*

Proof. Let $\mathcal{B}$ be the integral NMDA depicted in Fig. 3 over the alphabet $\Sigma = \{a, b, c\}$. We show that for every $n \in \mathbb{N}$, $\mathcal{B}(a^n b^\omega) = 1 - \frac{1}{2^{n+1}}$ and $\mathcal{B}(a^n c^\omega) = 1 + \frac{1}{3^{n+1}}$.

An integral DMDA $\mathcal{D}$ that is equivalent to $\mathcal{B}$ will intuitively need to preserve an accumulated discount factor Π_n and an accumulated weight W_n on every a^n prefix, such that both suffixes of b^ω and c^ω will match the value of $\mathcal{B}$. Since the difference between the required value of each pair $\langle a^n b^\omega, a^n c^\omega \rangle$ is "relatively large", Π_n must have "many" small discount factors of 2 to compensate this difference. But too many discount factors of 2 will not allow to achieve the "delicate" values of $1 + \frac{1}{3^{n+1}}$. We will formally analyze the mathematical properties of Π_n, showing that its prime-factor decomposition must indeed contain mostly 2's, "as well as" mostly 3's, leading to a contradiction.

Note that the only nondeterminism in $\mathcal{B}$ is in the initial state. Intuitively, for an infinite word for which the first non-a letter is b, the best choice for $\mathcal{B}$ would be to start in q_0, while if the first non-a letter is c, the best choice would be to start in q_1.

Formally, for each $n \in \mathbb{N} \setminus \{0\}$, observe that for the finite word a^n, the run r_1 starting at q_0 will have the accumulated value of $\mathcal{B}(r_1) = \sum_{k=0}^{n-1} \frac{1}{2} \cdot \frac{1}{2^k} = \frac{1}{2} \cdot \frac{1-\frac{1}{2^n}}{1-\frac{1}{2}} = 1 - \frac{1}{2^n}$, and an accumulated discount factor of 2^n; the run r_2 starting at q_1 the value $\mathcal{B}(r_2) = \sum_{k=0}^{n-1} \frac{2}{3} \cdot \frac{1}{3^k} = \frac{2}{3} \cdot \frac{1-\frac{1}{3^n}}{1-\frac{1}{3}} = 1 - \frac{1}{3^n}$, and an accumulated

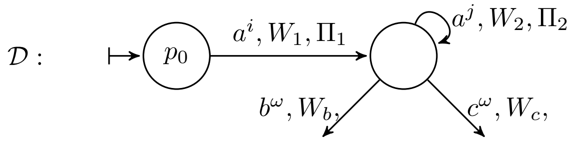

Figure 4: Partial structure of the DMDA $\mathcal{D}$ in the proof of Theorem 4.

discount factor of 3^n; and thus the value of $\mathcal{B}$, which is the minimum value of the two runs, $\mathcal{B}(a^n) = \min\left\{1 - \frac{1}{2^n}, 1 - \frac{1}{3^n}\right\} = 1 - \frac{1}{2^n}$.

Accordingly, we have that for every $n \in \mathbb{N}$,

$$\mathcal{B}(a^n b^\omega) = \min\left\{1 - \frac{1}{2^n} + \frac{1}{2} \cdot \frac{1}{2^n}, 1 - \frac{1}{3^n} + 2 \cdot \frac{1}{3^n}\right\} = 1 - \frac{1}{2^{n+1}} \tag{1}$$

$$\mathcal{B}(a^n c^\omega) = \min\left\{1 - \frac{1}{2^n} + 2 \cdot \frac{1}{2^n}, 1 - \frac{1}{3^n} + \frac{4}{3} \cdot \frac{1}{3^n}\right\} = 1 + \frac{1}{3^{n+1}} \tag{2}$$

We will show a contradiction regarding the accumulated discount factor on a cycle in the alleged equivalent DMDA. Assume toward contradiction that there exists an integral DMDA $\mathcal{D} = \langle \Sigma, Q_\mathcal{D}, p_0, \delta_\mathcal{D}, \gamma_\mathcal{D}, \rho_\mathcal{D} \rangle$ such that $\mathcal{B} \equiv \mathcal{D}$. Since $Q_\mathcal{D}$ is finite, there exist $i \in \mathbb{N}$ and $j \in \mathbb{N} \setminus \{0\}$ such that $\delta_\mathcal{D}(a^i) = \delta_\mathcal{D}(a^{i+j})$. Let r be the run of $\mathcal{D}$ on a^{i+j}, and denote the weight and discount factor of the prefix of r on a^i as $W_1 = \mathcal{D}(a^i) = \mathcal{D}(r[0..i-1])$ and $\Pi_1 = \rho(r[0..i-1])$, and the weight and discount factor of the suffix of r on the a^j cycle as $W_2 = \mathcal{D}(r[i..i+j-1])$ and $\Pi_2 = \rho(r[i..i+j-1])$. Let $W_b = \left[\mathcal{D}(a^i b^\omega) - \mathcal{D}(a^i)\right] \cdot \Pi_1$, be the weight of a b^ω word starting from $\delta_\mathcal{D}(a^i)$, and similarly $W_c = \left[\mathcal{D}(a^i c^\omega) - \mathcal{D}(a^i)\right] \cdot \Pi_1$. The partial structure of $\mathcal{D}$ with respect to those symbols is depicted in Fig. 4. For every $k \in \mathbb{N}$ we have

$$\mathcal{D}(a^{i+j\cdot k} b^\omega) = W_1 + \left(\sum_{t=0}^{k-1} \frac{W_2}{\Pi_1 \cdot \Pi_2^t}\right) + \frac{W_b}{\Pi_1 \cdot \Pi_2^k} \tag{3}$$

$$\mathcal{D}(a^{i+j\cdot k} c^\omega) = W_1 + \left(\sum_{t=0}^{k-1} \frac{W_2}{\Pi_1 \cdot \Pi_2^t}\right) + \frac{W_c}{\Pi_1 \cdot \Pi_2^k} \tag{4}$$

By the assumption that $\mathcal{B} \equiv \mathcal{D}$, subtracting Eq. (1) from Eq. (2) and Eq. (3) from Eq. (4), we get

$$\frac{1}{3^{i+j\cdot k+1}} + \frac{1}{2^{i+j\cdot k+1}} = \frac{W_c - W_b}{\Pi_1 \cdot \Pi_2^k}$$

Let M be the maximal weight in absolute value in $\mathcal{D}$. Since 2 is the minimal integral discount factor, we have that the value of $\mathcal{D}$ on any infinite word is no more than $2M$ in absolute value. Hence $|W_b| \leq 2M$ and $|W_c| \leq 2M$, which lead to

$$\frac{1}{2^{i+j\cdot k+1}} < \frac{1}{3^{i+j\cdot k+1}} + \frac{1}{2^{i+j\cdot k+1}} \leq \frac{2\cdot 2M}{\Pi_1}\cdot\frac{1}{\Pi_2^k}$$

and therefore, $\frac{1}{2^{j\cdot k}} < \frac{2\cdot 2M\cdot 2^{i+1}}{\Pi_1}\cdot\frac{1}{\Pi_2^k}$ and $\left(\frac{\Pi_2}{2^j}\right)^k < \frac{2\cdot 2M\cdot 2^{i+1}}{\Pi_1}$.

The above holds for every $k \in \mathbb{N}$. Observe that $\frac{2\cdot 2M\cdot 2^{i+1}}{\Pi_1}$ is a constant and $\lim_{k\to\infty}\left(\frac{\Pi_2}{2^j}\right)^k = \infty$ if and only if $\frac{\Pi_2}{2^j} > 1$, to conclude that $\Pi_2 \leq 2^j$. But Π_2 is a product of j integers bigger than 1, hence $\Pi_2 = 2^j$.

Let m be the least common denominator of W_c and W_2, and construct a DMDA $\mathcal{D}' = \langle\Sigma, Q_\mathcal{D}, p_0, \delta_\mathcal{D}, m\cdot\gamma_\mathcal{D}, \rho_\mathcal{D}\rangle$ created from $\mathcal{D}$ by multiplying all its weights by m. According to Proposition 2 and Lemma 3, for every $w \in \Sigma^\omega$ we have

$$\mathcal{D}'(w) = m\cdot\mathcal{D}(w) = m\cdot\mathcal{B}(w) \tag{5}$$

Let W'_1, W'_2 and W'_c be the values of $\mathcal{D}'$ on the a^i prefix, the following a^j cycle and the final c^ω respectively. Observe that $W'_1 = m\cdot W_1$, $W'_2 = m\cdot W_2$ and $W'_c = m\cdot W_c$, and that W'_2 and W'_c are integers.

For every $k \in \mathbb{N}$, similarly to Eq. (4), we have

$$\begin{aligned}\mathcal{D}'(a^ic^\omega) - \mathcal{D}'(a^{i+k\cdot j}c^\omega) &= \frac{W'_c}{\Pi_1} - \Big(\sum_{t=0}^{k-1}\frac{W'_2}{\Pi_1\cdot 2^{t\cdot j}}\Big) - \frac{W'_c}{\Pi_1\cdot 2^{k\cdot j}}\\ &= \frac{W'_c(2^{k\cdot j}-1) - \sum_{t=1}^{k}2^{t\cdot j}W'_2}{\Pi_1\cdot 2^{k\cdot j}}\end{aligned} \tag{6}$$

Define $X(k) = W'_c(2^{k\cdot j}-1) - \sum_{t=1}^{k}2^{t\cdot j}W'_2$ and observe that $X(k)$ is integer. Combine Eqs. (2), (5) and (6) to $m + \frac{m}{3^{i+1}} - \left(m + \frac{m}{3^{i+k\cdot j+1}}\right) = \mathcal{D}'(a^ic^\omega) - \mathcal{D}'(a^{i+k\cdot j}c^\omega) = \frac{X(k)}{\Pi_1\cdot 2^{k\cdot j}}$, simplified to $\frac{m\cdot(3^{k\cdot j}-1)}{3^{i+k\cdot j+1}} = \frac{X(k)}{\Pi_1\cdot 2^{k\cdot j}}$. But both m and Π_1 are constants and each of them has a finite number of prime factors of 3. Since $(3^{k\cdot j}-1)$ is not divisible by 3, and $X(k)$ is integer, when k gets bigger, eventually the denominator of the left side will have more prime factors of 3 than the denominator of the right side, which leads to a contradiction.

Hence, no DMDA is equivalent to $\mathcal{B}$ with respect to infinite words. According to Lemma 3, we also conclude that no DMDA is equivalent to $\mathcal{B}$ with respect to finite words. □

In the following proof that integral NMDAs are not closed under algebraic operations, we cannot assume toward contradiction a candidate deterministic automaton, and thus, as opposed to the proof of Theorem 4, we cannot assume a specific accumulative discount factor for each word prefix. Yet, we analyze the behavior of a candidate nondeterministic automaton on an infinite series of

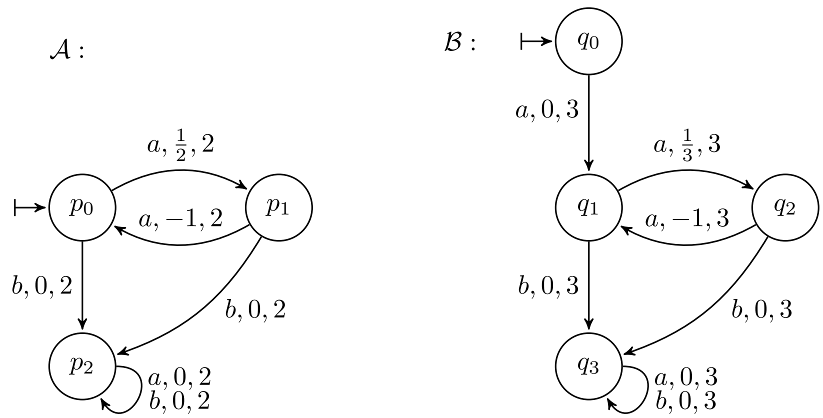

Figure 5: Deterministic integral NDAs that no integral NMDA is equivalent to their max or addition.

words, and build on the observation that there must be a state that appears in "the same position of the run" in infinitely many optimal runs of the automaton on these words.

Theorem 5. *There exist integral NMDAs (even deterministic integral NDAs) $\mathcal{A}$ and $\mathcal{B}$ over the same alphabet, such that no integral NMDA is equivalent to* $\max(\mathcal{A}, \mathcal{B})$*, and no integral NMDA is equivalent to $\mathcal{A} + \mathcal{B}$.*

Proof. Consider the NMDAs $\mathcal{A}$ and $\mathcal{B}$ over the alphabet $\Sigma = \{a, b\}$ depicted in Fig. 5. Observe that for every $n \in \mathbb{N}$,
$\mathcal{A}(a^n b^\omega) = \begin{cases} \frac{1}{2^n} & n \text{ is odd} \\ 0 & n \text{ is even} \end{cases}$, $\mathcal{B}(a^n b^\omega) = \begin{cases} 0 & n \text{ is odd} \\ \frac{1}{3^n} & n \text{ is even} \end{cases}$
Hence $\max(\mathcal{A}, \mathcal{B})(a^n b^\omega) = \Big(\mathcal{A} + \mathcal{B}\Big)(a^n b^\omega) = \begin{cases} \frac{1}{2^n} & n \text{ is odd} \\ \frac{1}{3^n} & n \text{ is even} \end{cases}$.

Intuitively, the target function has relatively large jumps in its value between every $a^n b^\omega$ and $a^{n+1} b^\omega$ words. For even values, as n gets bigger, when considering some prefix of a run r_n on $a^n b^\omega$ that entails a minimum value, two different suffixes can cause relatively "far" values. Hence, a relatively small accumulated discount factor is required for allowing these large jumps. For that property to hold, the accumulated discount factor must be no more than a constant product of 2^n. Meaning, each relevant prefix of r_n in the $\max(\mathcal{A}, \mathcal{B})$ (or $\mathcal{A} + \mathcal{B}$) automaton cannot have more than a constant number of discount factors bigger than 2.

On the other hand, the value given by $\mathcal{B}$ to $a^{n+1} b^\omega$ is very "delicate", meaning very close to 0 (and yet positive). Thus, as n gets bigger, a "fine grained" discount factor is required, contradicting the "coarse" accumulated factor of mostly 2's.

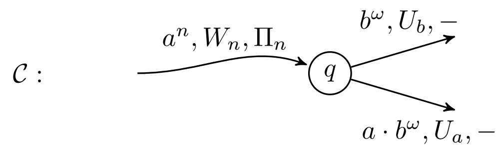

Figure 6: The state q and the notations from the proof of Theorem 5, for two different even $n \in \mathbb{N}$ such that $\delta(r_n[1..n]) = q$. The labels on the walks indicate the input word and the accumulated weight and discount factors.

Formally, assume toward contradiction that there exists an integral NMDA $\mathcal{C}'$, such that $\mathcal{C}' \equiv \max(\mathcal{A}, \mathcal{B}) \equiv \mathcal{A} + \mathcal{B}$, and let $d \in \mathbb{N}$ be the least common denominator of the weights in $\mathcal{C}'$.

Consider the NMDA $\mathcal{C} = \langle \Sigma, Q, \iota, \delta, \gamma, \rho \rangle$ created from $\mathcal{C}'$ by multiplying all its weights by d. Observe that all the weights in $\mathcal{C}$ are integers. According to Proposition 2, for every $n \in \mathbb{N}$, we have $\mathcal{C}(a^n b^\omega) = d \cdot \mathcal{C}'(a^n b^\omega) = \begin{cases} \frac{d}{2^n} & n \text{ is odd} \\ \frac{d}{3^n} & n \text{ is even} \end{cases}$

For every even $n \in \mathbb{N}$, let $w_n = a^n b^\omega$, and r_n a run of $\mathcal{C}$ on w_n that entails the minimal value of $\frac{d}{3^n}$. There exists a state $q \in Q$ such that for infinitely many even $n \in \mathbb{N}$, the target state of r_n after n steps is q, i.e, $\delta(r_n[0..n-1]) = q$. Let $U_b = \mathcal{C}^q(b^\omega)$ and $U_a = \mathcal{C}^q(a \cdot b^\omega)$, and for every such $n \in \mathbb{N}$, let $W_n = \mathcal{C}(r_n[0..n-1])$, and $\Pi_n = \rho(r_n[0..n-1])$ (See Fig. 6).

For every such $n \in \mathbb{N}$, since $\mathcal{C}(r_n) = \frac{d}{3^n}$, we have

$$W_n + \frac{U_b}{\Pi_n} = \frac{d}{3^n} \tag{7}$$

and since the value of every run of $\mathcal{C}$ on $a^{n+1}b^\omega$ is at least $\frac{d}{2^{n+1}}$, we have $W_n + \frac{U_a}{\Pi_n} \geq \frac{d}{2^{n+1}}$. Combining them both to get $\frac{d}{3^n} - \frac{U_b}{\Pi_n} + \frac{U_a}{\Pi_n} \geq \frac{d}{2^{n+1}}$ resulting in

$$\frac{U_a - U_b}{\Pi_n} \geq d \cdot \left(\frac{1}{2^{n+1}} - \frac{1}{3^n} \right) \tag{8}$$

But for large enough n, we have $3^n > 2^{n+2}$, hence we get $\frac{1}{2^{n+2}} > \frac{1}{3^n}$ and $\frac{1}{2^{n+1}} - \frac{1}{3^n} > \frac{1}{2^{n+1}} - \frac{1}{2^{n+2}} = \frac{1}{2^{n+2}}$. Assign this into Eq. (8) to get $\frac{U_a - U_b}{d} \cdot 2^{n+2} \geq \Pi_n$. Hence, there exists a positive constant $m_1 = \frac{U_a - U_b}{d} \cdot 2^2$ such that

$$m_1 \cdot 2^n \geq \Pi_n \tag{9}$$

Now, U_b is a rational constant, otherwise Eq. (7) cannot hold, as the other elements are rationals. Hence, there exist $x \in \mathbb{Z}$ and $y \in \mathbb{N}$ such that $U_b = \frac{x}{y}$. Assign it into Eq. (7) to get

$$\frac{1}{3^n} = \frac{W_n \cdot \Pi_n + U_b}{d \cdot \Pi_n} = \frac{W_n \cdot \Pi_n + \frac{x}{y}}{d \cdot \Pi_n} = \frac{W_n \cdot \Pi_n \cdot y + x}{d \cdot y \cdot \Pi_n}$$

But since the denominator and the numerator of the right-hand side are integers, we conclude that there exists a positive constant $m_2 = d \cdot y$, such that $m_2 \cdot \Pi_n \geq 3^n$. Combined with Eq. (9), we get $m_1 \cdot m_2 \cdot 2^n \geq 3^n$, for some positive constants m_1 and m_2, and for infinitely many $n \in \mathbb{N}$. But this stands in contradiction with $\lim_{n\to\infty} \left(\frac{2}{3}\right)^n = 0$. □

4 Tidy NMDAs

We present the family of "tidy NMDAs" and show that it is as expressive as deterministic NMDAs with arbitrary integral discount factors. Intuitively, an integral NMDA is tidy if the choice of discount factors depends on the word prefix read so far. We further show that for every choice function θ, the class of all θ-NMDAs is closed under determinization and algebraic operations, and satisfies the requirement of having decidable algorithms for its decision problems.

The family of tidy NMDAs contains various other natural subfamilies, such as integral NMDAs in which the discount factors are chosen per letter (action) or per the elapsed time, on which we elaborate in Section 4.4. Each of these subfamilies strictly extends the expressive power of integral NDAs.

We conclude with analyzing the structure of the family of tidy NMDAs.

Definition 6. *An integral NMDA $\mathcal{A}$ over an alphabet Σ and with discount-factor function ρ is* tidy *if there exists a function $\theta : \Sigma^+ \to \mathbb{N} \setminus \{0, 1\}$, such that for every finite word $u = \sigma_1 \dots \sigma_n \in \Sigma^+$, and every run $q_0, \sigma_1, \cdots, q_n$ of $\mathcal{A}$ on u, we have $\rho(q_{n-1}, \sigma_n, q_n) = \theta(u)$.*

In this case we say that $\mathcal{A}$ is a θ-NMDA.

Simple examples of tidy NMDAs are given in Figs. 11 and 14.

Definition 7. *For an alphabet Σ, a function $\theta : \Sigma^+ \to \mathbb{N} \setminus \{0, 1\}$ is a* choice function *if there exists an integral NMDA that is a θ-NMDA.*

For choice functions θ_1 and θ_2, the classes of θ_1-NMDAs and of θ_2-NMDAs are *equivalent* if they express the same functions, namely if for every θ_1-NMDA $\mathcal{A}$, there exists a θ_2-NMDA $\mathcal{B}$ equivalent to $\mathcal{A}$ and vice versa.

For every tidy NMDA $\mathcal{A}$ and finite word u, all the runs of $\mathcal{A}$ on u entail the same accumulated discount factor. We thus use the notation $\rho(u)$ to denote $\rho(r)$, where r is any run of $\mathcal{A}$ on u.

Observe that a general function $\theta : \Sigma^+ \to \mathbb{N} \setminus \{0, 1\}$ might require an infinite representation. Yet, we will show in Theorem 12 that every choice function has a finite representation.

4.1 Determinizability

We determinize a tidy NMDA by generalizing the determinization algorithm presented in [9] for NDAs. The basic idea in that algorithm is to extend the subset construction, by not only storing in each state of the deterministic automaton whether or not each state q of the original automaton $\mathcal{A}$ is reachable, but also the "gap" that q has from the currently optimal state q' of $\mathcal{A}$. This gap stands for the difference between the accumulated weights for reaching q and for reaching q', multiplied by the accumulated discounted factor.

Since we consider tidy NMDAs, we can generalize this view of gaps to the setting of multiple discount factors, as it is guaranteed that the run to q and the run to q' accumulated the same discount factor. (For non-tidy integral NMDAs, this is not the case, and they indeed need not be determinizable, as shown in Theorem 4.)

The construction. Consider a tidy NMDA $\mathcal{A} = \langle \Sigma, Q, \iota, \delta, \gamma, \rho \rangle$.

For every finite word $u \in \Sigma^*$ and state $q \in Q$, we define $S(q, u)$ to be the set of runs of $\mathcal{A}$ on u with q as the target state, and $r_{(q,u)}$ to be a *preferred run* that entails the minimal value among all the runs in $S(q, u)$. Observe that every prefix of a preferred run is also a preferred run. Hence given the values of all the preferred runs on a certain finite word u, i.e., $\mathcal{A}(r_{(q,u)})$ for every $q \in Q$, we can calculate the values of the preferred runs on every $u \cdot \sigma$ word by $\mathcal{A}(r_{(q',u\cdot\sigma)}) = \min\big\{\mathcal{A}(r_{(q,u)}) + \gamma(t) \bigm| t = (q, \sigma, q') \in \delta\big\}$.

Intuitively, every state of $\mathcal{D}$ that was reached after reading u, will store for each $q \in Q$ its "gap", which is the difference between $\mathcal{A}(u)$ and $\mathcal{A}(r_{(q,u)})$, "normalized" by multiplying it with the accumulated discount factor $\rho(u)$, and "truncated" if reached a threshold value (which can no longer be recovered).

Formally, for a state $q \in Q$, and a finite word u, we define

- The *cost* of reaching q over u as

$$\begin{aligned} \texttt{cost}(q, u) &= \min\big\{\mathcal{A}(r) \bigm| r \text{ is a run of } \mathcal{A} \text{ on } u \text{ s.t. } \delta(r) = q\big\} \\ &= \min\big\{\mathcal{A}(r) \bigm| r \in S(q, u)\big\} \end{aligned}$$

  where $\min \emptyset = \infty$.

- The *gap* of q over u as $\texttt{gap}(q, u) = \rho(u)\big(\texttt{cost}(q, u) - \mathcal{A}(u)\big)$. Intuitively, the gap stands for the value that a walk starting in q should have, compared to a walk starting in u's optimal ending state, in order to make a run through q optimal.

Let T be the maximum difference between the weights in $\mathcal{A}$, That is, $T = \max\big(|x - y| \bigm| x, y \in \texttt{range}(\gamma)\big)$. Since for every infinite run r of $\mathcal{A}$ we have $\sum_{i=0}^{\infty} \frac{1}{\prod_{j=0}^{i-1} r(j)} \leq \sum_{i=0}^{\infty} \frac{1}{2^i} = 2$, we define the set of possible *recoverable-gaps* $G = \big\{v \bigm| v \in \mathbb{Q} \text{ and } 0 \leq v < 2T\big\} \cup \{\infty\}$. The ∞ element denotes a non-recoverable gap, and behaves as the standard infinity element in the algebraic operations that we will be using. Note that our NMDAs do not have infinite weights and the infinite element is only used as an internal component of the construction.

We will inductively construct $\mathcal{D} = \langle \Sigma, Q', q'_{in}, \delta', \gamma', \rho' \rangle$ as follows. A state of $\mathcal{D}$ extends the standard subset construction by assigning a gap to each state of $\mathcal{A}$. That is, for $Q = \{q_1, \cdots, q_n\}$, a state $p \in Q'$ is a tuple $\langle g_1, \cdots, g_n \rangle$, where

$g_h \in G$ for every $1 \leq h \leq n$. Once a gap is obviously not recoverable, by being larger than or equal to $2T$, it gets truncated by setting it to be ∞.

In the integral ρ function case, the construction only requires finitely many elements of G, as shown in Lemma 8, and thus it is guaranteed to terminate.

For simplicity, we assume that $\iota = \{q_1, q_2, \cdots, q_{|\iota|}\}$ and extend γ with $\gamma(q_i, \sigma, q_j) = \infty$ for every $(q_i, \sigma, q_j) \notin \delta$. The initial state of $\mathcal{D}$ is $q'_{in} = \langle 0, \cdots, 0, \infty, \cdots, \infty\rangle$, in which the left $|\iota|$ elements are 0, meaning that the initial states of $\mathcal{A}$ have a 0 gap and the others are currently not relevant.

We inductively build the desired automaton $\mathcal{D}$ using the intermediate automata $\mathcal{D}_i = \langle \Sigma, Q'_i, q'_{in}, \delta'_i, \gamma'_i, \rho'_i\rangle$. We start with $\mathcal{D}_1$, in which $Q'_1 = \{q'_{in}\}$, $\delta'_1 = \emptyset$, $\gamma'_1 = \emptyset$ and $\rho'_1 = \emptyset$, and proceed from $\mathcal{D}_i$ to $\mathcal{D}_{i+1}$, such that $Q'_i \subseteq Q'_{i+1}$, $\delta'_i \subseteq \delta'_{i+1}$, $\gamma'_i \subseteq \gamma'_{i+1}$ and $\rho'_i \subseteq \rho'_{i+1}$. The construction is completed once $\mathcal{D}_i = \mathcal{D}_{i+1}$, finalizing the desired deterministic automaton $\mathcal{D} = \mathcal{D}_i$.

In the induction step, $\mathcal{D}_{i+1}$ extends $\mathcal{D}_i$ by (possibly) adding, for every state $q' = \langle g_1, \cdots, g_n\rangle \in Q'_i$ and letter $\sigma \in \Sigma$, a state $q'' := \langle x_1, \cdots, x_n\rangle$, and a transition $t := (q', \sigma, q'')$ as follows:

- Weight: For every $1 \leq h \leq n$ define,
  $c_h := \min\left\{g_j + \gamma(q_j, \sigma, q_h) \;\middle|\; 1 \leq j \leq n\right\}$, and add a new weight, $\gamma'_{i+1}(t) = \min_{1\leq h\leq n}(c_h)$.
- Discount factor: By the induction construction, if $\mathcal{D}_i$ running on a finite word u ends in q', there is a run of $\mathcal{A}$ on u ending in q_h, for every $1 \leq h \leq n$ for which the gap g_h in q' is not ∞. Since $\mathcal{A}$ is tidy, all the transitions from every such state q_h over σ have the same discount factor, which we set to the new transition $\rho'_{i+1}(t)$.
- Gap: For every $1 \leq h \leq n$, set $x_h := \rho'_{i+1}(t) \cdot \big(c_h - \gamma'_{i+1}(t)\big)$. If $x_h \geq 2T$ then set $x_h := \infty$.

See Fig. 7 for an example of the determinization process.

We prove below that the procedure always terminates for a tidy NMDA, and that every state of the generated DMDA can be represented in PSPACE. The proof is similar to the corresponding proof in [9] with respect to NDAs, adding the necessary extensions for tidy NMDAs.

Lemma 8. *The above determinization procedure always terminates for a tidy NMDA $\mathcal{A}$. Every state of the resulting deterministic automaton $\mathcal{D}$ can be represented in space polynomial in $|\mathcal{A}|$, and $|\mathcal{D}| \in 2^{O(|\mathcal{A}|)}$.*

Proof. The induction step of the construction, extending $\mathcal{D}_i$ to $\mathcal{D}_{i+1}$, only depends on $\mathcal{A}$, Σ and Q'_i. Furthermore, for every $i \geq 0$, we have that $Q'_i \subseteq Q'_{i+1}$. Thus, for showing the termination of the construction, it is enough to show that there is a general bound on the size of the sets Q'_i. We do it by showing that the inner values, $g_1, \ldots, g_n$, of every state q' of every set Q'_i are from the finite set $\bar{G}$, defined below.

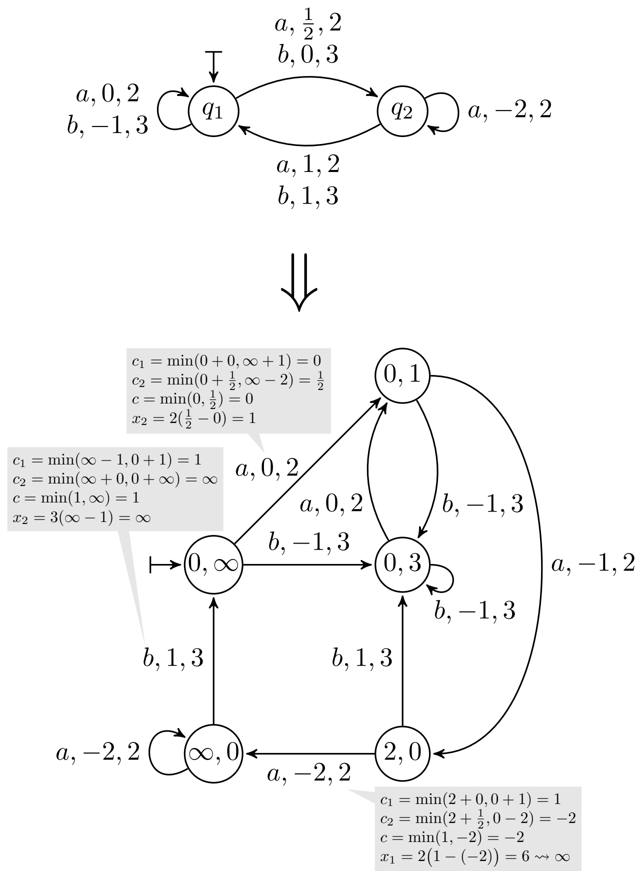

Figure 7: An example of the determinization procedure, as per Theorem 11. The gray rectangles detail some of the intermediate calculations.

Let $d \in \mathbb{N}$ be the least common denominator of the weights in $\mathcal{A}$, and let $T \in \mathbb{N}$ be the maximal difference between the weights. We define the set $\bar{G}$ as

$$\bar{G} = \left\{\frac{k}{d} \mid k \in \mathbb{N} \text{ and } \frac{k}{d} < 2T\right\} \cup \{\infty\}$$

We start with the first set of states Q'_1, which satisfies the property that the inner values, $g_1, \ldots, g_n$, of every state $q' \in Q'_1$ are from $\bar{G}$, as $Q'_1 = \{\langle 0, \cdots, 0, \infty, \cdots, \infty\rangle\}$. We proceed by induction on the construction steps, assuming that Q'_i satisfies the property. By the construction, an inner value of a state q'' of Q'_{i+1} is derived by four operations on elements of $\bar{G}$: addition, subtraction ($x - y$, where $x \geq y$), multiplication by $\lambda \in \texttt{range}(\rho) \subset \mathbb{N}$, and taking the minimum.

One may verify that applying these four operations on ∞ and numbers of the form $\frac{k}{d}$, where $k \in \mathbb{N}$, results in ∞ or in a number $\frac{k'}{d}$, where $k' \in \mathbb{N}$. Recall that once an inner value exceeds $2T$, it is replaced by the procedure with ∞, meaning that $\frac{k'}{d} < 2T$, or the calculated inner value is ∞. Concluding that all the inner values are in $\bar{G}$.

Observe that $|\bar{G}| \leq 2 \cdot T \cdot d + 1$. Meaning that every state in the resulting DMDA has up to $2 \cdot T \cdot d + 1$ possible values for each of the $|Q|$ inner elements. Hence we have no more than $(2 \cdot T \cdot d + 1)^{|Q|}$ possibilities for the states of $\mathcal{D}$, proving the termination claim.

Recall that in our definition for $|\mathcal{A}|$, we mention that we assume that all of the weights are given with the same denominator, which is d in our notations. Hence the space required for $|Q|$ elements with up to $2 \cdot T \cdot d + 1$ possible values each, which is the space required for every state in $\mathcal{D}$, is polynomial with respect to $|\mathcal{A}|$. Also the total size of $\mathcal{D}$ is in $2^{O(|\mathcal{A}|)}$. □

We will now show the correctness of the determinization procedure. According to Lemma 3, it is enough to show the equivalence $\mathcal{D} \equiv \mathcal{A}$ with respect to finite words.

Lemma 9. *Consider a tidy NMDA $\mathcal{A}$ over Σ^+ and a DMDA $\mathcal{D}$, constructed from $\mathcal{A}$ by the above determinization procedure. Then, for every $u \in \Sigma^+$, we have $\mathcal{A}(u) = \mathcal{D}(u)$.*

Proof. Let $\mathcal{A} = \langle \Sigma, Q, \iota, \delta, \gamma, \rho\rangle$ be the input NMDA, $\mathcal{D} = \langle \Sigma, Q', \iota', \delta', \gamma', \rho'\rangle$ the DMDA constructed from $\mathcal{A}$, and T be the maximal difference between the weights in $\mathcal{A}$.

For a finite word u, let $\delta'(u) = \langle g_1, \cdots, g_n\rangle \in Q'$ be the target state of $\mathcal{D}$'s run on u. We show by induction on the length of the input word u that:

i. $\mathcal{A}(u) = \mathcal{D}(u)$.

ii. For every $1 \leq h \leq n$, $g_h = \texttt{gap}(q_h, u)$ if $\texttt{gap}(q_h, u) < 2T$ and ∞ otherwise.

The assumptions obviously hold for the initial step, where u is the empty word. As for the induction step, we assume they hold for u and show that for every $\sigma \in \Sigma$, they hold for $u \cdot \sigma$. Let $\delta'(u \cdot \sigma) = \langle x_1, \cdots, x_n \rangle \in Q'$ be the target state of $\mathcal{D}$'s run on $u \cdot \sigma$.

We start by proving the claim with respect to an *infinite-state* automaton $\mathcal{D}'$ that is constructed as in the determinization procedure, except for not changing any gap to ∞. Afterwards, we shall argue that changing all gaps that exceed $2T$ to ∞ does not harm the correctness.

i. By the definitions of `cost` and `gap`, we have for every $1 \leq h \leq n$,

$$\begin{aligned}
\texttt{cost}(q_h, u \cdot \sigma) &= \min_{1 \leq j \leq n} \left(\texttt{cost}(q_j, u) + \frac{\gamma(q_j, \sigma, q_h)}{\rho(u)} \right) \\
&= \min_{1 \leq j \leq n} \left(\frac{\texttt{gap}(q_j, u)}{\rho(u)} + \mathcal{A}(u) + \frac{\gamma(q_j, \sigma, q_h)}{\rho(u)} \right) \\
&= \mathcal{A}(u) + \frac{\min_{1 \leq j \leq n} \Big(\texttt{gap}(q_j, u) + \gamma(q_j, \sigma, q_h) \Big)}{\rho(u)} \qquad (10) \\
&= \text{By the induction assumption} \\
&= \mathcal{D}'(u) + \frac{\min_{1 \leq j \leq n} \Big(g_j + \gamma(q_j, \sigma, q_h) \Big)}{\rho(u)} \qquad (11)
\end{aligned}$$

By the construction of $\mathcal{D}'$, the transition weight $\gamma'_i(t)$ assigned on the $i = |u| + 1$ step is
$\gamma'_{|u|+1}(t) = \min_{1 \leq h \leq n} \Big(\min_{1 \leq j \leq n} (g_j + \gamma(q_j, \sigma, q_h)) \Big)$. Therefore,

$$\begin{aligned}
\mathcal{D}'(u \cdot \sigma) &= \mathcal{D}'(u) + \frac{\gamma'_{|u|+1}(t)}{\rho(u)} \\
&= \mathcal{D}'(u) + \frac{\min_{1 \leq h \leq n} \min_{1 \leq j \leq n} \Big(g_j + \gamma(q_j, \sigma, q_h) \Big)}{\rho(u)} \\
&= \min_{1 \leq h \leq n} \left(\mathcal{D}'(u) + \frac{\min_{1 \leq j \leq n} \Big(g_j + \gamma(q_j, \sigma, q_h) \Big)}{\rho(u)} \right) \\
&= \min_{1 \leq h \leq n} \texttt{cost}(q_h, u \cdot \sigma) = \mathcal{A}(u \cdot \sigma)
\end{aligned}$$

ii. By Eq. (11), we get that for every $1 \leq h \leq n$:

$$\min_{1 \leq j \leq n} (g_j + \gamma(q_j, \sigma, q_h)) = \rho(u) \Big(\texttt{cost}(q_h, u \cdot \sigma) - \mathcal{D}'(u) \Big)$$

Let t be the transition that was added in the $i = |u| + 1$ step of the algorithm from the state $\delta'(u)$ over the σ letter.

For every $1 \leq h \leq n$, we have

$$\begin{aligned}
x_h &= \rho'_i(t) \cdot (c_h - \gamma'_i(t)) \\
&= \rho'_i(t)\Big(\min_{1\leq j\leq n}(g_j + \gamma(q_j,\sigma,q_h)) - \gamma'_i(t)\Big) \\
&= \rho'_i(t)\left(\min_{1\leq j\leq n}(g_j{+}\gamma(q_j,\sigma,q_h)) - \rho(u)\Big(\mathcal{D}'(u\cdot\sigma) - \mathcal{D}'(u)\Big)\right) \\
&= \rho'_i(t)\left(\rho(u)\Big(\texttt{cost}(q_h, u\cdot\sigma) - \mathcal{D}'(u)\Big) - \rho(u)\Big(\mathcal{D}'(u\cdot\sigma) - \mathcal{D}'(u)\Big)\right) \\
&= \rho'_i(t)\cdot\rho(u)\Big(\texttt{cost}(q_h, u\cdot\sigma) - \mathcal{D}'(u\cdot\sigma)\Big) \\
&= \rho(u\cdot\sigma)\cdot\Big(\texttt{cost}(q_h, u\cdot\sigma) - \mathcal{D}'(u\cdot\sigma)\Big)
\end{aligned}$$

And by the induction assumption we have

$$x_h = \rho(u\cdot\sigma)\cdot\Big(\texttt{cost}(q_h, u\cdot\sigma) - \mathcal{A}(u\cdot\sigma)\Big) = \texttt{gap}(q_h, u\cdot\sigma)$$

It is left to show that the induction is also correct for the *finite-state* automaton $\mathcal{D}$. The only difference between the construction of $\mathcal{D}$ and of $\mathcal{D}'$ is that the former changes all gaps (g_j) above $2T$ to ∞. We should thus show that if the gap g_j, for some $1 \leq j \leq n$, exceeds $2T$ at a step i of the construction, and this g_j influences the next gap of some state h (we denoted this gap in the construction as x_h) then $x_h \geq 2T$. This implies that $\mathcal{D}(u) = \mathcal{D}'(u)$, since at every step of the construction there is at least one $1 \leq h \leq n$, such that $x_h = 0$, corresponding to an optimal run of $\mathcal{A}$ on u ending in state q_h.

Formally, we should show that if $g_j \geq 2T$ and $x_h = \rho'_{i+1}(t)\cdot\Big(g_j{+}\gamma(q_j,\sigma,q_h){-}\gamma'_{i+1}(t)\Big)$, where t is the transition added in the construction on step i as defined in part (ii.) above, then $x_h \geq 2T$. Indeed, according to the construction exists an index $1 \leq k \leq n$ such that $g_k = 0$ and since $\mathcal{A}$ is complete, there is a transition from q_k to some state q_m, implying that $\gamma'_{i+1}(t) \leq g_k + \gamma(q_k,\sigma,q_m) = \gamma(q_k,\sigma,q_m)$. Hence

$$\begin{aligned}
x_h &\geq \rho'_{i+1}(t)\cdot\Big(2T + \gamma(q_j,\sigma,q_h) - \gamma'_{i+1}(t)\Big) \geq 2\cdot\Big(2T + \gamma(q_j,\sigma,q_h) - \gamma'_{i+1}(t)\Big) \\
&\geq 2\cdot\Big(2T + \gamma(q_j,\sigma,q_h) - \gamma(q_k,\sigma,q_m)\Big) \geq 2\cdot(2T + (-T)) = 2T
\end{aligned}$$

□

We show next that the DMDA created by the determinization procedure is indeed a θ-DMDA.

Lemma 10. *Consider a θ-NMDA $\mathcal{A}$ over Σ^+ and a DMDA $\mathcal{D}$, constructed from $\mathcal{A}$ by the determinization procedure above. Then $\mathcal{D}$ is a θ-DMDA.*

Proof. Consider a tidy NMDA $\mathcal{A} = \langle \Sigma, Q, \iota, \delta, \gamma, \rho \rangle$, and the DMDA $\mathcal{D} = \langle \Sigma, Q', \iota', \delta', \gamma', \rho' \rangle$ constructed from $\mathcal{A}$.

We show by induction on the length of an input word that for every finite word $u \in \Sigma^*$, we have $\rho'(u) = \rho(u)$. The base case regarding the empty word obviously holds. As for the induction step, we assume the claim holds for u and show that it also holds for $u \cdot \sigma$, for every $\sigma \in \Sigma$.

Let t be the final transition of $\mathcal{D}$'s run on $u \cdot \sigma$. Due to the construction of $\mathcal{D}$, there exist $q, q' \in Q$ such that $\mathtt{gap}(q, u) \neq \infty$, $\mathtt{gap}(q', u \cdot \sigma) \neq \infty$, and $\rho'(t) = \rho(q, \sigma, q')$.

Hence, $\rho'(u \cdot \sigma) = \rho'(u) \cdot \rho'(t) = \rho(u) \cdot \rho'(t) = \rho(u) \cdot \rho(q, \sigma, q')$ and since $\mathtt{gap}(q, u) \neq \infty$, we get that $q \in \delta(u)$, and $\rho'(u \cdot \sigma) = \rho(u) \cdot \rho(q, \sigma, q') = \rho(u \cdot \sigma)$. □

And finally, as a direct consequence of the above construction and Lemmas 8 to 10:

Theorem 11. *For every choice function θ and a θ-NMDA $\mathcal{A}$, there exists a θ-DMDA $\mathcal{D} \equiv \mathcal{A}$ of size in $2^{O(|\mathcal{A}|)}$. Every state of $\mathcal{D}$ can be represented in space polynomial in $|\mathcal{A}|$.*

4.2 Representing Choice Functions

We show that, as opposed to the case of a general function $f : \Sigma^+ \to \mathbb{N} \setminus \{0, 1\}$, every choice function θ can be finitely represented by a transducer (Mealy machine).

A transducer $\mathcal{T}$ (Mealy machine) is a 6-tuple $\langle P, \Sigma, \Gamma, p_0, \delta, \rho \rangle$, where P is a finite set of states, Σ and Γ are finite sets called the input and output alphabets, $p_0 \in P$ is the initial state, $\delta : P \times \Sigma \to P$ is the total transition function and $\rho : P \times \Sigma \to \Gamma$ is the total output function.

A transducer $\mathcal{T}$ represents a function, to which for simplicity we give the same name $\mathcal{T} : \Sigma^+ \to \Gamma$, such that for every word w, the value $\mathcal{T}(w)$ is the output label of the last transition taken when running $\mathcal{T}$ on w. The size of $\mathcal{T}$, denoted by $|\mathcal{T}|$, is the maximum between the number of transitions and the maximal binary representation of any output in the range of ρ.

Since in this work we only consider transducers in which the output alphabet Γ is the natural numbers $\mathbb{N}$, we omit Γ from their description, namely write $\langle P, \Sigma, p_0, \delta, \rho \rangle$ instead of $\langle P, \Sigma, \mathbb{N}, p_0, \delta, \rho \rangle$. An example of a transducer $\mathcal{T}$ and a $\mathcal{T}$-NMDA is given in Fig. 8.

As the structure of an NMDA is finite, we get that transducers are enough for representing any choice function of a tidy NMDA.

Theorem 12. *For every function $\theta : \Sigma^+ \to \mathbb{N} \setminus \{0, 1\}$, θ is a choice function, namely there exists a θ-NMDA, if and only if there exists a transducer $\mathcal{T}$ such that $\theta \equiv \mathcal{T}$.*

Proof. Consider a function $\theta : \Sigma^+ \to \mathbb{N} \setminus \{0, 1\}$. For the first direction, observe that given a transducer $\mathcal{T} = \langle P, \Sigma, p_0, \delta, \rho \rangle$ representing θ, it holds that the NMDA $\mathcal{T}' = \langle \Sigma, P, \{p_0\}, \delta, \gamma, \rho \rangle$, for every weight function γ, is a θ-NMDA.

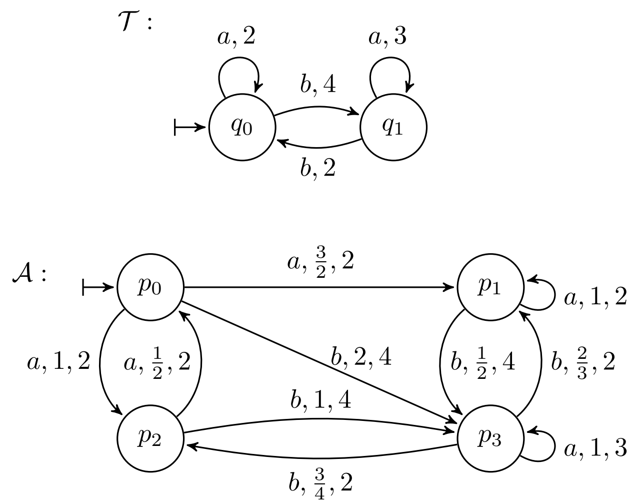

Figure 8: A transducer $\mathcal{T}$ and a $\mathcal{T}$-NMDA.

For the other direction, consider a θ-NMDA $\mathcal{A}'$. According to Theorem 11, there exists a θ-DMDA $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \gamma, \rho \rangle$ equivalent to $\mathcal{A}'$. Since the image of ρ is a subset of $\mathbb{N}$, we have that θ can be represented by the transducer $\mathcal{T} = \langle Q, \Sigma, q_0, \delta, \rho \rangle$. □

For a given choice function θ, we refer to the class of all θ-NMDAs. Observe that when considering such class, only the choice function is relevant, regardless of the transducer defining it.

4.3 Closure under Algebraic Operations

Theorem 13. *For every choice function θ, the set of θ-NMDAs is closed under the operations of min, max, addition, subtraction, and multiplication by a rational constant.*

Proof. Consider a choice function θ and θ-NMDAs $\mathcal{A}$ and $\mathcal{B}$.

- *Multiplication by constant $c \geq 0$*: A θ-NMDA for $c \cdot \mathcal{A}$ is straightforward from Proposition 2.
- *Multiplication by -1*: A θ-NMDA for $-\mathcal{A}$ can be achieved by first determinizing $\mathcal{A}$, as per Theorem 11, into a θ-DMDA $\mathcal{D}$ and then multiplying all the weights in $\mathcal{D}$ by -1.
- *Addition*: Considering the θ-NMDAs $\mathcal{A} = \langle \Sigma, Q_1, \iota_1, \delta_1, \gamma_1, \rho_1 \rangle$ and $\mathcal{B} = \langle \Sigma, Q_2, \iota_2, \delta_2, \gamma_2, \rho_2 \rangle$, a θ-NMDA for $\mathcal{A}+\mathcal{B}$ can be achieved by constructing the product automaton $\mathcal{C} = \langle \Sigma, Q_1 \times Q_2, \iota_1 \times \iota_2, \delta, \gamma, \rho \rangle$ such that:

Operation ╲ Family	Tidy DMDAs	Tidy NMDAs
$c \cdot \mathcal{A}$ (for $c \geq 0$)	Linear	Linear
$-\mathcal{A}$	Linear	Single Exponential
$\mathcal{A} + \mathcal{B}$	Quadratic	Quadratic
$\mathcal{A} - \mathcal{B}$	Quadratic	Single Exponential
$\min(\mathcal{A}, \mathcal{B})$	Single Exponential	Linear
$\max(\mathcal{A}, \mathcal{B})$	Single Exponential	Single Exponential

Table 1: The size blow-up involved in the algebraic operations.

- $\delta = \Big\{ \big((q_1, q_2), \sigma, (p_1, p_2)\big) \;\big|\; (q_1, \sigma, p_1) \in \delta_1 \text{ and } (q_2, \sigma, p_2) \in \delta_2 \Big\}$.
- $\gamma\big((q_1, q_2), \sigma, (p_1, p_2)\big) = \gamma_1(q_1, \sigma, p_1) + \gamma_2(q_2, \sigma, p_2)$.
- $\rho\big((q_1, q_2), \sigma, (p_1, p_2)\big) = \rho_1(q_1, \sigma, p_1) = \rho_2(q_2, \sigma, p_2)$. Note that the latter must hold since both ρ_1 and ρ_2 are compliant with θ.

- *Subtraction*: A θ-NMDA for $\mathcal{A} - \mathcal{B}$ can be achieved by i) Determinizing $\mathcal{B}$ to $\mathcal{B}'$; ii) Multiplying $\mathcal{B}'$ by -1, getting $\mathcal{B}''$; and iii) Constructing a θ-NMDA for $\mathcal{A} + \mathcal{B}''$.
- *min*: A θ-NMDA for $\min(\mathcal{A}, \mathcal{B})$ is straightforward by the nondeterminism on their union.
- *max*: A θ-NMDA for $\max(\mathcal{A}, \mathcal{B})$ can be achieved by i) Determinizing $\mathcal{A}$ and $\mathcal{B}$ to $\mathcal{A}'$ and $\mathcal{B}'$, respectively; ii) Multiplying $\mathcal{A}'$ and $\mathcal{B}'$ by -1, getting $\mathcal{A}''$ and $\mathcal{B}''$, respectively; iii) Constructing a θ-NMDA $\mathcal{C}''$ for $\min(\mathcal{A}'', \mathcal{B}'')$; iv) Determinizing $\mathcal{C}''$ into a θ-DMDA $\mathcal{D}$; and v) Multiplying $\mathcal{D}$ by -1, getting θ-NMDA $\mathcal{C}$, which provides $\max(\mathcal{A}, \mathcal{B})$.

□

We analyze next the size blow-up involved in algebraic operations. In addition to the general classes of θ-NMDAs, we also consider the case where both input and output automata are deterministic. Summation of the results can be seen in Table 1.

Most results in Table 1 are straightforward from the constructions presented in the proof of Theorem 13: multiplying all the weights by a constant is linear, creating the product automaton is quadratic, and whenever determinization is required, we get an exponential blow-up. However, the result of the size blow-up for the max operation on tidy NMDAs is a little more involved. At a first glance, determinizing back and forth might look like a doubly-exponential blow-up, however in this case an optimized determinization procedure can achieve a singly-exponential blow-up: Determinizing a tidy NMDA that is the union of two DMDAs, in which the transition weights are polynomial in the number of states, is shown to only involve a polynomial size blow-up.

Theorem 14. *The size blow-up involved in the* max *operation on tidy NMDAs is at most single-exponential.*

Proof. Consider a choice function θ, θ-NMDAs $\mathcal{A}$ and $\mathcal{B}$, and the automata $\mathcal{A}'', \mathcal{B}'', \mathcal{C}'', \mathcal{D}$ and $\mathcal{C}$, as constructed in the 'max' part of the proof of Theorem 13. Observe that $\mathcal{C}''$ is the the union of two θ-DMDAs. As so, for every word u, there are only two possible runs of $\mathcal{C}''$ on u. In order to determinize $\mathcal{C}''$ into $\mathcal{D}$ we present a slightly modified procedure compared to the one presented in Section 4.1. Instead of the basic subset construction, we use the product automaton of $\mathcal{A}''$ and $\mathcal{B}''$ and instead of saving in every state of $\mathcal{D}$ the gap from the preferred state for every state of $\mathcal{C}''$, we only save the gap between the two runs of $\mathcal{C}''$. Combined with the observation we showed in the proof of Lemma 9 that the weights of $\mathcal{A}''$ and $\mathcal{B}''$ are bounded by the weights of $\mathcal{A}$ and $\mathcal{B}$, we are able to reduce the overall blow-up to be only single-exponential.

The procedure presented in Section 4.1 requires the following modifications:

- Every state of $\mathcal{D}$ is a tuple $\langle q_1, q_2, g_1, g_2\rangle$ where q_1 is a state of $\mathcal{A}''$, q_2 is a state of $\mathcal{B}''$, and $g_1, g_2 \in G$ are the gaps from the preferred run.
- The initial state of $\mathcal{D}$ is $\langle q_{\mathcal{A}}, q_{\mathcal{B}}, 0, 0\rangle$ where $q_{\mathcal{A}}$ and $q_{\mathcal{B}}$ are the initial states of $\mathcal{A}''$ and $\mathcal{B}''$, respectively.
- In the induction step, $\mathcal{D}_{i+1}$ extends $\mathcal{D}_i$ by (possibly) adding for every state $p = \langle q_1, q_2, g_1, g_2\rangle$ and letter $\sigma \in \Sigma$, a state $p' := \langle q'_1, q'_2, g'_1, g'_2\rangle$ and a transition $t := \langle p, \sigma, p'\rangle$ such that for every $1 \leq h \leq 2$:
  - $c_h := g_h + \gamma\big(q_h, \sigma, \delta(q_h, \sigma)\big)$
  - $\gamma'_{i+1}(t) = \min(c_1, c_2)$
  - $\rho'_{i+1}(t) = \rho\big(q_1, \sigma, \delta(q_1, \sigma)\big)$
  - $x_h := \rho'_{i+1}(t) \cdot \big(c_h - \gamma'_{i+1}(t)\big)$. If $x_h \geq 2T$ then set $x_h := \infty$

With the above modifications, similarly to Lemma 8, we get that the number of possible gaps is $2 \cdot T \cdot d_{\mathcal{A}} \cdot d_{\mathcal{B}} + 1$ where $d_{\mathcal{A}}$ and $d_{\mathcal{B}}$ are the denominators of weights in $\mathcal{A}''$ and $\mathcal{B}''$, respectively. Hence, there are no more than $(2 \cdot T \cdot d_{\mathcal{A}} \cdot d_{\mathcal{B}} + 1)^2 \cdot N_{\mathcal{A}} \cdot N_{\mathcal{B}}$ possibilities for the states of $\mathcal{D}$, where $N_{\mathcal{A}}$ and $N_{\mathcal{B}}$ are the number of states in $\mathcal{A}''$ and $\mathcal{B}''$, respectively.

According to the determinization procedure showed in Section 4.1 and as explained in the proofs of Lemmas 8 and 9, the following observations hold:

- $d_{\mathcal{A}}$ and $d_{\mathcal{B}}$ are also the denominators of weights in $\mathcal{A}$ and $\mathcal{B}$, respectively, and since we use binary representation of weights, $d_{\mathcal{A}} \cdot d_{\mathcal{B}}$ is up to single-exponential in $|\mathcal{A}| + |\mathcal{B}|$.
- All the weights in $\mathcal{A}''$ and $\mathcal{B}''$ are bounded by the weights of $\mathcal{A}$ and $\mathcal{B}$, hence T is also up to single-exponential in $|\mathcal{A}| + |\mathcal{B}|$.
- $N_{\mathcal{A}}$ and $N_{\mathcal{B}}$ are up to single-exponential in $|\mathcal{A}| + |\mathcal{B}|$.

Concluding that the number of states in $\mathcal{D}$ is up to single-exponential in $|\mathcal{A}|+|\mathcal{B}|$, and since the number of states in $\mathcal{C}$ is equal to the number of states in $\mathcal{D}$, we get a single-exponential blow-up. ☐

Observe that if weights are represented in unary, we can achieve a quartic blow-up for the min and max operations on tidy-DMDAs, by using the above determinization procedure, and since T is linear in unary representation.

We are not aware of prior lower bounds on the size blow-up involved in algebraic operations on NDAs. For achieving such lower bounds, we develop a general scheme to convert every NFA to a λ-NDA of linearly the same size that defines the same language, with respect to a threshold value 0, and to convert some specific λ-NDAs back to corresponding NFAs.

The conversion of an NFA to a corresponding λ-NDA is quite simple. It roughly uses the same structure of the original NFA, and assigns four different transitions weights, depending on whether each of the source and target states is accepting or rejecting.

Lemma 15. *For every $\lambda \in \mathbb{N} \setminus \{0,1\}$ and NFA $\mathcal{A}$ with n states, there exists a λ-NDA $\tilde{\mathcal{A}}$ with $n+2$ states, such that for every word $u \in \Sigma^+$, we have $u \in L(\mathcal{A})$ iff $\tilde{\mathcal{A}}(u) < 0$. That is, the language defined by $\mathcal{A}$ is equivalent to the language defined by $\tilde{\mathcal{A}}$ and the threshold 0.*

Proof. Given an NFA $\mathcal{A} = \langle \Sigma, Q, \iota, \delta, F\rangle$ and a discount factor $\lambda \in \mathbb{N} \setminus \{0,1\}$, we construct a λ-NDA $\tilde{\mathcal{A}} = \langle \Sigma, Q', \{p_0\}, \delta', \gamma'\rangle$ for which there exists a bijection f between the runs of $\mathcal{A}$ and the runs of $\tilde{\mathcal{A}}$ such that for every run r of $\tilde{\mathcal{A}}$ on a word u,

- r is an accepting run of $\mathcal{A}$ iff $f(r)$ is a run of $\tilde{\mathcal{A}}$ on u with the value $\tilde{\mathcal{A}}\big(f(r)\big) = -\frac{1}{\lambda^{|r|}}$.
- r is a non-accepting run of $\mathcal{A}$ iff $f(r)$ is a run of $\tilde{\mathcal{A}}$ on u with the value $\tilde{\mathcal{A}}\big(f(r)\big) = \frac{1}{\lambda^{|r|}}$.

We first transform $\mathcal{A}$ to an equivalent NFA $\mathcal{A}' = \langle \Sigma, Q', \{p_0\}, \delta', F\rangle$ that is complete and in which there are no transitions entering its initial state, and later assign weights to its transitions to create $\tilde{\mathcal{A}}$.

To construct $\mathcal{A}'$ we add two states to Q, having $Q' = Q \cup \{p_0, q_{hole}\}$, duplicate all the transitions from ι to start from p_0, and add a transition from every state to q_{hole}, namely $\delta' = \delta \cup \big\{(p_0,\sigma,q) \;\big|\; \exists p \in \iota, (p,\sigma,q) \in \delta\big\} \cup \big\{(q,\sigma,q_{hole}) \;\big|\; q \in Q', \sigma \in \Sigma\big\}$. Observe that $|Q'| = |Q| + 2$, and $L(\mathcal{A}) = L(\mathcal{A}')$. Next, we assign the following transition weights:

- For every $t = (p_0,\sigma,q) \in \delta'$, $\gamma'(t) = -\frac{1}{\lambda}$ if $q \in F$ and $\gamma'(t) = \frac{1}{\lambda}$ if $q \notin F$.
- For every $t = (p,\sigma,q) \in \delta'$ such that $p \neq p_0$, $\gamma'(t) = \frac{\lambda-1}{\lambda}$ if $p,q \in F$; $\gamma'(t) = \frac{\lambda+1}{\lambda}$ if $p \in F$ and $q \notin F$; $\gamma'(t) = -\frac{\lambda+1}{\lambda}$ if $p \notin F$ and $q \in F$; and $\gamma'(t) = -\frac{\lambda-1}{\lambda}$ if $p,q \notin F$.

By induction on the length of the runs on an input word u, one can show that for every $u \in \Sigma^+$, $\tilde{\mathcal{A}}(u) = -\frac{1}{\lambda^{|u|}}$ if $u \in L(\mathcal{A})$ and $\tilde{\mathcal{A}}(u) = \frac{1}{\lambda^{|u|}}$ if $u \notin L(\mathcal{A})$. $\square$

Converting an NDA to a corresponding NFA is much more challenging, since a general NDA might have arbitrary weights. We develop a conversion scheme, whose correctness proof is quite involved, from every NDA $\dot{\mathcal{B}}$ that is equivalent to $-\tilde{\mathcal{A}}$, where $\tilde{\mathcal{A}}$ is generated from an arbitrary NFA as per Lemma 15, to a corresponding NFA $\mathcal{B}$. Notice that the assumption that $\dot{\mathcal{B}} \equiv -\tilde{\mathcal{A}}$ gives us some information on $\dot{\mathcal{B}}$, yet $\dot{\mathcal{B}}$ might a priori still have arbitrary transition weights. Using this scheme, we provide an exponential lower bound on the size blow-up involved in multiplying an NDA by (-1). The theorem holds with respect to both finite and infinite words.

Theorem 16. *For every $n \in \mathbb{N}$ and $\lambda \in \mathbb{N} \setminus \{0,1\}$, there exists a λ-NDA $\mathcal{A}$ with n states over a fixed alphabet, such that every λ-NDA that is equivalent to $-\mathcal{A}$, w.r.t. finite or infinite words, has $\Omega(2^n)$ states.*

Proof. Consider $n \in \mathbb{N}$ and $\lambda \in \mathbb{N} \setminus \{0,1\}$. By [34, 30] there exists an NFA $\mathcal{A}$ with n states over a fixed alphabet of two letters, such that any NFA for the complement language $\overline{L(\mathcal{A})}$ has at least 2^n states.

Finite words.

Let $\tilde{\mathcal{A}}$ be a λ-NDA that is correlated to $\mathcal{A}$ as per Lemma 15, and assume towards contradiction that there exists a λ-NDA $\dot{\mathcal{B}} = \langle \Sigma, Q_{\dot{\mathcal{B}}}, \iota_{\dot{\mathcal{B}}}, \delta_{\dot{\mathcal{B}}}, \gamma_{\dot{\mathcal{B}}} \rangle$ with less than $\frac{2^n}{4}$ states such that $\dot{\mathcal{B}} \equiv -\tilde{\mathcal{A}}$.

We provide below a conversion opposite to Lemma 15, leading to an NFA for $\overline{L(\mathcal{A})}$ with less than 2^n states, and therefore to a contradiction. The conversion of $\dot{\mathcal{B}}$ back to an NFA builds on the specific values that $\dot{\mathcal{B}}$ is known to assign to words, as opposed to the construction of Lemma 15, which works uniformly for every NFA, and is much more challenging, since $\dot{\mathcal{B}}$ might have arbitrary transition weights. This conversion scheme can only work for λ-NDAs whose values on the input words converge to some threshold as the words length grow to infinity.

For simplification, we do not consider the empty word, since one can easily check if the input NFA accepts it, and set the complemented NFA to reject it accordingly.

By Lemma 15 we have that for every word $u \in \Sigma^+$, $\tilde{\mathcal{A}}(u) = -\frac{1}{\lambda^{|u|}}$ if $u \in L(\mathcal{A})$ and $\tilde{\mathcal{A}}(u) = \frac{1}{\lambda^{|u|}}$ if $u \notin L(\mathcal{A})$. Hence, $\dot{\mathcal{B}}(u) = -\frac{1}{\lambda^{|u|}}$ if $u \notin L(\mathcal{A})$ and $\dot{\mathcal{B}}(u) = \frac{1}{\lambda^{|u|}}$ if $u \in L(\mathcal{A})$. We will show that there exists an NFA $\mathcal{B}$, with less than 2^n states, such that $u \in L(\mathcal{B})$ iff $\dot{\mathcal{B}}(u) = -\frac{1}{\lambda^{|u|}}$, implying that $L(B) = \overline{L(\mathcal{A})}$.

We first construct a λ-NDA $\mathcal{B}' = \langle \Sigma, Q_{\mathcal{B}'}, \iota, \delta, \gamma \rangle$ that is equivalent to $\dot{\mathcal{B}}$, but has no transitions entering its initial states. This construction eliminates the possibility that one run is a suffix of another, allowing to simplify some of our arguments. Formally, $Q_{\mathcal{B}'} = Q_{\dot{\mathcal{B}}} \cup \iota$, $\iota = \iota_{\dot{\mathcal{B}}} \times \{1\}$, $\delta = \delta_{\dot{\mathcal{B}}} \cup \big\{ \big((p,1), \sigma, q\big) \;\big|\; (p,\sigma,q) \in \delta_{\dot{\mathcal{B}}} \big\}$, and weights $\gamma(t) = \gamma_{\dot{\mathcal{B}}}(t)$ if $t \in \delta_{\dot{\mathcal{B}}}$ and $\gamma\big((p,1),\sigma,q\big) = \gamma_{\dot{\mathcal{B}}}(p,\sigma,q)$ otherwise.

Let R^- be the set of all the runs of $\mathcal{B}'$ that entail a minimal value which is less than 0, i.e., $R^- = \{r \mid r \text{ is a minimal run of } \mathcal{B}' \text{ on some word and } \mathcal{B}'(r) < 0\}$.

Let $\hat{\delta} \subseteq \delta$ be the set of all the transitions that take part in some run in R^-, meaning $\hat{\delta} = \{r(i) \mid r \in R^- \text{ and } 0 \leq i < |r|\}$, and $\hat{\hat{\delta}} \subseteq \delta$ the set of all transitions that are the last transition of those runs, meaning $\hat{\hat{\delta}} = \{r(|r|-1) \mid r \in R^-\}$.

We construct next the NFA $\mathcal{B} = \langle \Sigma, Q_\mathcal{B}, \iota, \delta_\mathcal{B}, F_\mathcal{B} \rangle$. Intuitively, $\mathcal{B}$ has the states of $\mathcal{B}'$, but only the transitions from $\hat{\delta}$. Its accepting states are clones of the target states of the transitions in $\hat{\hat{\delta}}$, but without outgoing transitions. We will later show that the only runs of $\mathcal{B}$ that reach these clones are those that have an equivalent run in R^-. Formally, $Q_\mathcal{B} = Q'_\mathcal{B} \cup F_\mathcal{B}$, $F_\mathcal{B} = \{(q,1) \mid \exists p, q \in Q'_\mathcal{B} \text{ and } (p,\sigma,q) \in \hat{\hat{\delta}}\}$, and $\delta_\mathcal{B} = \hat{\delta} \cup \{(p,\sigma,(q,1)) \mid (p,\sigma,q) \in \hat{\hat{\delta}}\}$.

Observe that the number of states in $\mathcal{B}$ is at most 3 times the number of states in $\dot{\mathcal{B}}$, and thus less than 2^n. We will now prove that for every word u, $\mathcal{B}$ accepts u iff $\mathcal{B}'(u) = -\frac{1}{\lambda^{|u|}}$.

The first direction is easy: if $\mathcal{B}'(u) = -\frac{1}{\lambda^{|u|}}$, we get that all the transitions of a minimal run of $\mathcal{B}'$ on u are in $\hat{\delta}$, and its final transition is in $\hat{\hat{\delta}}$, hence there exists a run of $\mathcal{B}$ on u ending at an accepting state.

For the other direction, assume towards contradiction that there exists a word u, such that $\mathcal{B}'(u) = \frac{1}{\lambda^{|u|}}$, while there is an accepting run r_u of $\mathcal{B}$ on u.

Intuitively, we define the "normalized value" of a run r' of $\mathcal{B}'$ as the value of $\mathcal{B}'$ multiplied by the accumulated discount factor, i.e., $\mathcal{B}'(r') \cdot \lambda^{|r'|}$. Whenever the normalized value reaches -1, we have an "accepting" run. We will show that r_u and the structure of $\mathcal{B}$ imply the existence of two "accepting" runs $r'_1, r'_2 \in R^-$ that intersect in some state q, such that taking the prefix of r'_1 up to q results in a normalized value $\lambda^k W_1$ that is strictly smaller than the normalized value $\lambda^j W_2$ of the prefix of r'_2 up to q. Since r'_2 is an "accepting" run, the suffix of r'_2 reduces $\lambda^j W_2$ to -1 and therefore it will reduce $\lambda^k W_1$ to a value strictly smaller than -1, and the total value of the run to a value strictly smaller than $-\frac{1}{\lambda^n}$, which is not a possible value of $\mathcal{B}'$.

Formally, let $r_u(|u|-1) = (p', u(|u|-1), (q',1))$ be the final transition of r_u. We replace it with the transition $t' = (p', u(|u|-1), q')$. The resulting run $r'_u = r_u[0..|u|-2] \cdot t$ is a run of $\mathcal{B}'$ on u, and therefore $\mathcal{B}'(r'_u) \geq \frac{1}{\lambda^{|u|}}$. Since $(q',1)$ is an accepting state, we get by the construction of $\mathcal{B}$ that t' is in $\hat{\hat{\delta}}$. Consider a run $r'_1 \in R^-$ that shares the maximal suffix with r'_u, meaning that if there exist $r' \in R^-$ and $x > 0$ such that $r'[|r'|-x..|r'|-1] = r'_u[|u|-x..|u|-1]$ then also $r'_1[|r'_1|-x..|r'_1|-1] = r'_u[|u|-x..|u|-1]$.

Recall that all the initial states of $\mathcal{B}'$ have no transitions entering them and $\mathcal{B}'(r'_1) \neq \mathcal{B}'(r'_u)$, hence r'_1 is not a suffix of r'_u and r'_u is not a suffix of r'_1. Let i be the maximal index of r'_u such that $r'_u[i..|u|-1]$ is a suffix of r'_1, but $r'_u[i-1..|u|-1]$ is not a suffix of r'_1. Let k be the index in r'_1 such that $r'_1[k..|r'_1|-1] = r_u[i..|u|-1]$, and let $x = |r'_1| - k$ (see Fig. 9).

Since $r'_u(i-1) \in \hat{\delta}$, there exists $r'_2 \in R^-$ and index j such that $r'_2(j-1) = r'_u(i-1)$. Let $y = |r'_2| - j$ (see Fig. 9). Consider the run $r'_3 = r'_2[0..j-1] \cdot r'_u[i..|u|-1]$, starting with the prefix of r'_2 up to the shared transition with r'_u, and then continuing with the suffix of r'_u. Observe that $\mathcal{B}'(r'_3) > -\frac{1}{\lambda^{|r'_3|}}$ as

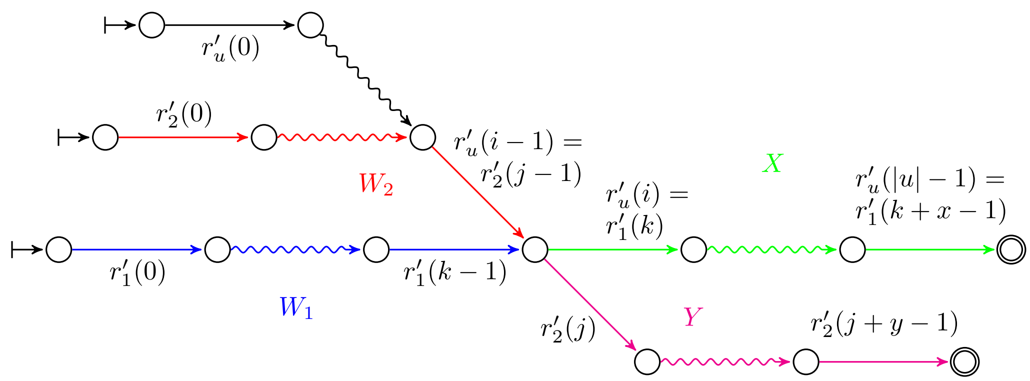

Figure 9: The runs and notations used in the proof of Theorem 16.

otherwise $r'_3 \in R^-$ and has a larger suffix with r'_u than r'_1 has.

Let $W_1 = \mathcal{B}'\big(r'_1[0..k-1]\big)$, $W_2 = \mathcal{B}'\big(r'_2[0..j-1]\big)$, $X = \mathcal{B}'\big(r'_1[k..k+x-1]\big)$ (which is also $\mathcal{B}'\big(r'_u[i..|u|-1]\big)$), and $Y = \mathcal{B}'\big(r'_2[j..j+y-1]\big)$ (see Fig. 9). The following must hold:

1. $W_1 + \frac{X}{\lambda^k} = \mathcal{B}'(r'_1) = -\frac{1}{\lambda^{k+x}}$. Hence, $\lambda^k W_1 = -\frac{1}{\lambda^x} - X$.
2. $W_2 + \frac{X}{\lambda^j} = \mathcal{B}'(r'_3) > -\frac{1}{\lambda^{j+x}}$. Hence, $\lambda^j W_2 > -\frac{1}{\lambda^x} - X$, and after combining with the previous equation, $\lambda^j W_2 > \lambda^k W_1$.
3. $W_2 + \frac{Y}{\lambda^j} = \mathcal{B}'(r'_2) = -\frac{1}{\lambda^{j+y}}$. Hence, $\lambda^j W_2 + Y = -\frac{1}{\lambda^y}$

Consider now the run $r'_4 = r'_1[0..k-1] \cdot r'_2[j..j+y-1]$, and combine Items 2 and 3 above to get that $\lambda^k W_1 + Y < -\frac{1}{\lambda^y}$. But this leads to $\mathcal{B}'(r'_4) = W_1 + \frac{Y}{\lambda^k} < -\frac{1}{\lambda^{k+y}} = -\frac{1}{\lambda^{|r'_4|}}$, and this means that there exists a word w of length $k+y$ such that $\mathcal{B}'(w) < -\frac{1}{\lambda^{k+y}}$, contradicting the assumption that $\mathcal{B}' \equiv \dot{\mathcal{B}} \equiv -\tilde{\mathcal{A}}$.

Infinite words.

For showing the lower bound for the state blow-up involved in multiplying an NDA by (-1) w.r.t. infinite words, we add a new letter $\#$ to the alphabet, and correlate every finite word u to an infinite word $u \cdot \#^\omega$. The proof is similar, applying the following modifications:

- The scheme presented in the proof of Lemma 15 now constructs a λ-NDA $\tilde{\mathcal{A}}$ over the alphabet $\Sigma \cup \{\#\}$, adding a 0-weighted transition from every state of $\tilde{\mathcal{A}}$ to q_{hole}. The function f that correlates between the runs of $\mathcal{A}$ and $\tilde{\mathcal{A}}$ is still a bijection, but with a different co-domain, correlating every run r of $\mathcal{A}$ on a finite word $u \in \Sigma^+$ to the run $f(r)$ of $\tilde{\mathcal{A}}$ on the word $u \cdot \#^\omega$.
- With this scheme, we get that $\dot{\mathcal{B}}(u \cdot \#^\omega) = -\frac{1}{\lambda^{|u|}}$ if $u \notin L(A)$ and $\dot{\mathcal{B}}(u \cdot \#^\omega) = \frac{1}{\lambda^{|u|}}$ if $u \in L(A)$, hence replacing all referencing to $\mathcal{B}'(u)$ with referencing to $\mathcal{B}'(u \cdot \#^\omega)$.

- R^- is defined with respect to words of the form $u \cdot \#^\omega$, namely $R^- = \{r \mid u \in \Sigma^+, r$ is a minimal run of $\mathcal{B}'$ on $u \cdot \#^\omega$ and $\mathcal{B}'(r) < 0\}$.
- R^-_p is a new set of all the maximal (finite) prefixes of the runs of R^- without any transitions for the $\#$ letter, meaning $R^-_p = \{r[0..i-1] \mid r \in R^-, r(i-1) = (p, \sigma, q)$ for some $\sigma \in \Sigma,$ and $r(i) = (q, \#, s)\}$. $\hat{\delta}$ and $\hat{\hat{\delta}}$ are defined with respect to R^-_p instead of R^-.
- Defining r'_u, we consider a run $r'_t \in R^-$ that is a witness for $t' \in \hat{\hat{\delta}}$, meaning there exists $i \in \mathbb{N}$ for which $r'_t(i) = t'$, and $r'_t(i+1)$ is a transition for the $\#$ letter. Then $r'_u = r_u[0..|u|-2] \cdot t \cdot r'[i+1..\infty] = r_u[0..|u|-2] \cdot r'[i..\infty]$, is a run of $\mathcal{B}'$ on $u \cdot \#^\omega$.
- For choosing r'_1 that "shares the maximal suffix" with r'_u, we take $r'_1 \in R^-$ such that for every $r' \in R^-$ and $x > 0$, if $r'_u[i..\infty]$ is a suffix of r' then it is also a suffix of r'_1.
- For the different runs and their parts, we set $X = \mathcal{B}'\big(r'_1[k..\infty]\big)$, $Y = \mathcal{B}'\big(r'_2[j..\infty]\big)$, $r'_3 = r'_2[0..j-1] \cdot r'_u[i..\infty]$ and $r'_4 = r'_1[0..k-1] \cdot r'_2[j..\infty]$.

□

4.4 Basic Subfamilies

Tidy NMDAs constitute a rich family that also contains some basic subfamilies that are still more expressive than integral NDAs. Two such subfamilies are integral NMDAs in which the discount factors depend on the transition letter or on the elapsed time.

Notice that closure of tidy NMDAs under determinization and under algebraic operations is related to a specific choice function θ, namely every class of θ-NMDAs enjoys these closure properties (Theorems 11 and 13). Since the aforementioned subfamilies of tidy NMDAs also consist of θ-NMDA classes, their closure under determinization and under algebraic operations follows. For example, the class of NMDAs that assigns a discount factor of 2 to the letter 'a' and of 3 to the letter 'b' enjoys these closure properties.

4.4.1 Letter-Oriented Discount Factors

A θ-NMDA over an alphabet Σ is *letter oriented* if all transitions over the same alphabet letter share the same discount factor; that is, if $\theta : \Sigma^+ \to \mathbb{N} \setminus \{0, 1\}$ coincides with a function $\Lambda : \Sigma \to \mathbb{N} \setminus \{0, 1\}$, in the sense that for every finite word u and letter σ, we have $\theta(u\sigma) = \Lambda(\sigma)$. (See an example in Fig. 10.)

Notice that every choice function θ for a letter-oriented θ-NMDA can be defined via a simple transducer of a single state, having a self loop over every letter with its assigned discount factor.

We show that letter-oriented NMDAs indeed add expressiveness over NDAs.

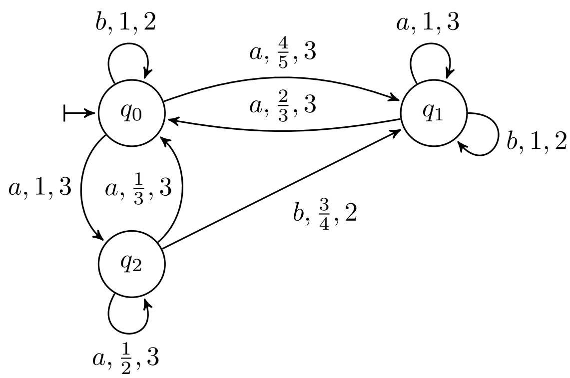

Figure 10: A letter-oriented discounted-sum automaton, for the discount factor function $\Lambda(a) = 3$; $\Lambda(b) = 2$.

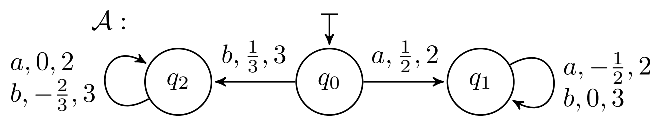

Figure 11: A letter-oriented discounted-sum automaton, for the discount factor function $\Lambda(a) = 2$; $\Lambda(b) = 3$, that no integral NDA is equivalent to.

Theorem 17. *There exists a letter-oriented NMDA that no integral NDA is equivalent to.*

Proof. Consider the NMDA $\mathcal{A}$ depicted in Fig. 11. Assume toward contradiction that there exists an integral NDA $\mathcal{B}'$ such that $\mathcal{B}' \equiv \mathcal{A}$. According to [9], there exists an integral deterministic NDA (integral DDA) $\mathcal{B}$ with transition function $\delta_\mathcal{B}$ and discount factor λ, such that $\mathcal{B} \equiv \mathcal{B}' \equiv \mathcal{A}$.

Observe that for every $n \in \mathbb{N} \backslash \{0\}$, we have $\mathcal{B}(a^n b^\omega) = \mathcal{A}(a^n b^\omega) = \frac{1}{2^n}$. As $\mathcal{B}$ has finitely many states, there exists a state q in $\mathcal{B}$ and $i, j \in \mathbb{N} \setminus \{0\}$ such that $\delta_\mathcal{B}(a^i) = \delta_\mathcal{B}(a^{i+j}) = q$. Let $W_1 = \mathcal{B}^q(a^j)$ and $W_2 = \mathcal{B}^q(b^\omega)$.

Observe that

$$\frac{1}{2^i} = \mathcal{B}(a^i b^\omega) = \mathcal{B}(a^i) + \frac{W_2}{\lambda^i} \tag{12}$$

$$\frac{1}{2^{i+j}} = \mathcal{B}(a^{i+j} b^\omega) = \mathcal{B}(a^i) + \frac{W_1}{\lambda^i} + \frac{W_2}{\lambda^{i+j}} \tag{13}$$

$$\frac{1}{2^{i+2j}} = \mathcal{B}(a^{i+2j} b^\omega) = \mathcal{B}(a^i) + \frac{W_1}{\lambda^i} + \frac{W_1}{\lambda^{i+j}} + \frac{W_2}{\lambda^{i+2j}} \tag{14}$$

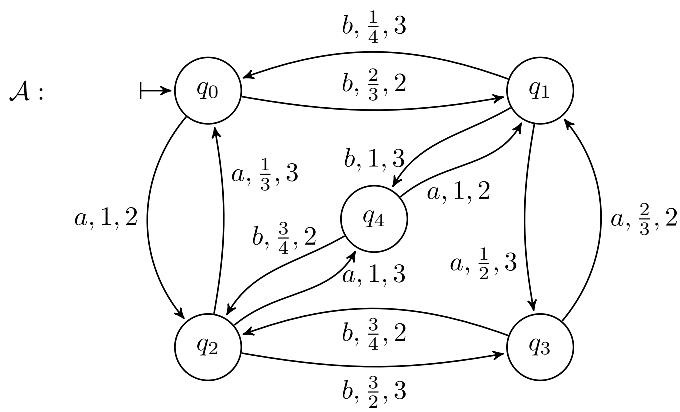

Figure 12: A time-oriented discounted-sum automaton $\mathcal{A}$.

Subtract Eq. (12) from Eq. (13), and Eq. (13) from Eq. (14) to get

$$\frac{1}{2^{i+j}} - \frac{1}{2^i} = \frac{W_1 - W_2}{\lambda^i} + \frac{W_2}{\lambda^{i+j}} \tag{15}$$

$$\frac{1}{2^{i+2j}} - \frac{1}{2^{i+j}} = \frac{W_1 - W_2}{\lambda^{i+j}} + \frac{W_2}{\lambda^{i+2j}} = \frac{1}{\lambda^j}\Big(\frac{W_1 - W_2}{\lambda^i} + \frac{W_2}{\lambda^{i+j}}\Big) \tag{16}$$

and combine Eqs. (15) and (16) to get $\frac{1}{2^j}\Big(\frac{1}{2^{i+j}} - \frac{1}{2^i}\Big) = \frac{1}{2^{i+2j}} - \frac{1}{2^{i+j}} = \frac{1}{\lambda^j}\Big(\frac{1}{2^{i+j}} - \frac{1}{2^i}\Big)$, which implies $\lambda = 2$.

Observe that for every $n \in \mathbb{N}\backslash\{0\}$, we have $\mathcal{B}(b^n a^\omega) = \mathcal{A}(b^n a^\omega) = \frac{1}{3^n}$. Symmetrically to the above, but with respect to 'b' instead of 'a' and '3' instead of '2', results in $\lambda = 3$, leading to a contradiction. □

4.4.2 Time-Oriented Discount Factors

A θ-NMDA over an alphabet Σ is *time oriented* if the discount factor on a transition is determined by the distance of the transition from an initial state; that is, if $\theta : \Sigma^+ \to \mathbb{N} \setminus \{0,1\}$ coincides with a function $\Lambda : \mathbb{N} \setminus \{0\} \to \mathbb{N} \setminus \{0,1\}$, in the sense that for every finite word u, we have $\theta(u) = \Lambda\big(|u|\big)$.

For example, the NMDA $\mathcal{A}$ of Fig. 12 is time-oriented, as all transitions taken at odd steps, in any run, have discount factor 2, and those taken at even steps have discount factor 3. The transducer $\mathcal{T}$ of Fig. 13 represents its choice function.

Time-oriented NMDAs also add expressiveness over NDAs.

Theorem 18. *There exists a time-oriented NMDA that no integral NDA is equivalent to.*

Proof. Let $\mathcal{A}$ be the time-oriented NMDA depicted in Fig. 14. Observe that $\mathcal{A}(a^n b^\omega) = \frac{1}{6^{\lceil \frac{n}{2} \rceil}}$. Analogously to the proof of Theorem 17, but with respect to

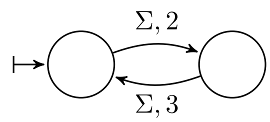

Figure 13: A transducer that represents the discount-factor choice function for the NMDA $\mathcal{A}$ of Fig. 12.

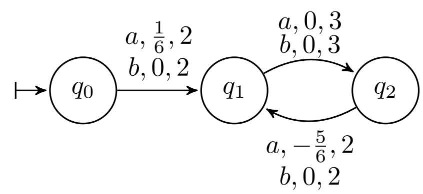

Figure 14: A time-oriented NMDA that no integral NDA is equivalent to.

"$\sqrt{6}$" instead of "2", we have that the discount factor of an equivalent DDA, if such exists, is $\lambda = \sqrt{6}$, hence no integral NDA can be equivalent to $\mathcal{A}$. □

4.5 The Structure of the Family

Every choice function θ defines a class of θ-NMDAs which is closed under algebraic operations (Theorem 13). In this section we show that this is not the case for the entire family of tidy-NMDAs. We show in Section 4.5.1 that the union of any two different classes of θ-NMDA is not closed under algebraic operations, meaning that we cannot extend any of them in the trivial way of taking the union with another. We further analyze the relations between the different θ-NMDA classes, and demonstrate the importance of each class. We show in Section 4.5.3 that no θ_1-NMDA class is a strict subset of any other θ_2-NMDA class, meaning that every class stands for itself. In Section 4.5.2 we show that the intersection of all θ-NMDA classes can only express a set of basic functions, the set of eventually constant functions, meaning that every such class has a significant contribution to the expressiveness of the family.

We start with identifying a *similarity* property between transducers, and showing that for every choice functions θ_1 and θ_2, the class of θ_1-NMDAs is equivalent to the class of θ_2-NMDAs iff θ_1 and θ_2 can be defined by similar transducers.

Definition 19. *Transducers $\mathcal{T}_1$ and $\mathcal{T}_2$ over the same alphabet and with transition functions $\delta_{\mathcal{T}_1}$ and $\delta_{\mathcal{T}_2}$ respectively, are* similar *if for every finite words v and w, such that $\delta_{\mathcal{T}_1}(v) = \delta_{\mathcal{T}_1}(v \cdot w)$ and $\delta_{\mathcal{T}_2}(v) = \delta_{\mathcal{T}_2}(v \cdot w)$, we have* $\prod_{k=0}^{|w|-1} \mathcal{T}_1\Big(v \cdot w[0..k]\Big) = \prod_{k=0}^{|w|-1} \mathcal{T}_2\Big(v \cdot w[0..k]\Big)$.

That is, $\mathcal{T}_1$ and $\mathcal{T}_2$ are similar if for every word $v \cdot w$ that causes a cycle over the suffix w both in $\mathcal{T}_1$ and $\mathcal{T}_2$, the discount factors accumulated in these two

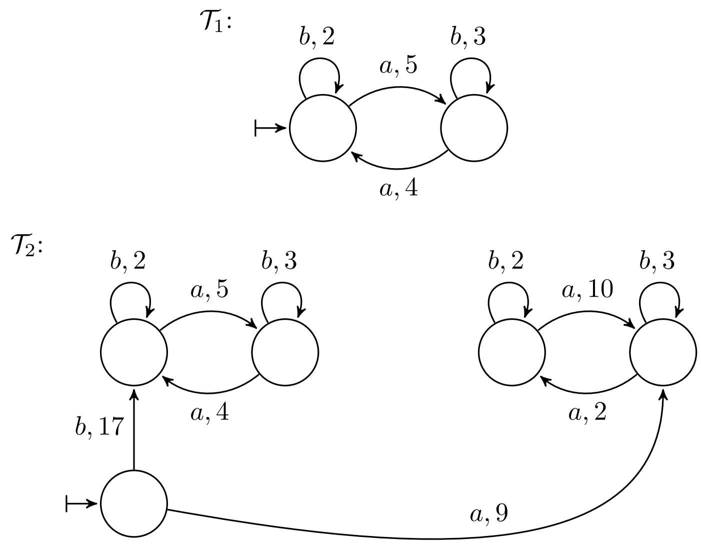

Figure 15: An example of similar transducers.

cycles are equal. For example, the transducers $\mathcal{T}_1$ and T_2 depicted in Fig. 15 are similar. Observe that similar transducers can have different structures, and are allowed to disagree on the accumulated discount factor achieved for some words, as long as these words do not cause a cycle in both of them.

Using similar transducers, we also define similar choice functions:

Definition 20. *Choice functions θ_1 and θ_2 are* similar *if there exist similar transducers $\mathcal{T}_1$ and $\mathcal{T}_2$ such that $\mathcal{T}_1 \equiv \theta_1$ and $\mathcal{T}_2 \equiv \theta_2$.*

As one would expect, the choice of transducers to represent the choice functions does not matter.

Corollary 21. *For every choice functions θ_1 and θ_2 for which there exist non-similar transducers $\mathcal{T}_1$ and $\mathcal{T}_2$ such that $\mathcal{T}_1 \equiv \theta_1$ and $\mathcal{T}_2 \equiv \theta_2$, we have that θ_1 and θ_2 are not similar, meaning there are no similar transducers $\mathcal{T}_1'$ and $\mathcal{T}_2'$ such that $\mathcal{T}_1' \equiv \theta_1$ and $\mathcal{T}_2' \equiv \theta_2$.*

Proof. This is a direct corollary of Theorem 29 and Corollary 30 (which we will prove in Section 4.5.3). According to Corollary 30, the non-similarity of $\mathcal{T}_1$ and $\mathcal{T}_2$ leads to the existence of a θ_1-NMDA for which no θ_2-NMDA is equivalent to. On the other hand, if there exist similar transducers $\mathcal{T}_1'$ and $\mathcal{T}_2'$ such that $\mathcal{T}_1' \equiv \theta_1$ and $\mathcal{T}_2' \equiv \theta_2$, according to Theorem 29, every θ_1-NMDA has an equivalent θ_2-NMDA. □

4.5.1 Class union

Observe that for every choice function θ, the class of θ-NMDAs is closed under algebraic operations (Theorem 13). Yet, this is not the case for the entire family of tidy NMDAs. The automata $\mathcal{A}$ and $\mathcal{B}$ provided in the proof of Theorem 5 are tidy, and it is shown there that no integral NMDA is equivalent to their max or addition.

Corollary 22. *There exist choice functions θ_1 and θ_2 over the same alphabet, a θ_1-NMDA $\mathcal{A}$, and a θ_2-NMDA $\mathcal{B}$, such that for every choice function θ_3, there is no θ_3-NMDA equivalent to $\mathcal{A}+\mathcal{B}$ and no θ_3-NMDA equivalent to* $\max(\mathcal{A},\mathcal{B})$.

Moreover, we show that the union of every two non-similar classes of θ-NMDAs in not closed under algebraic operations.

Theorem 23. *For every two non-similar choice function θ_1 and θ_2 over a non-singleton alphabet, there exist a θ_1-NMDA $\mathcal{A}_1$ and a θ_2-NMDA $\mathcal{A}_2$ such that no integral NMDA (even non-tidy one), is equivalent to* $\max\{\mathcal{A}_1,\mathcal{A}_2\}$ *or to* $\mathcal{A}_1+\mathcal{A}_2$.

Proof. Consider non-similar choice functions, θ_1 and θ_2, a transducer for θ_1, $\mathcal{T}_1 = \langle P_1, \Sigma, p_1, \delta_{\mathcal{T}_1}, \rho_{\mathcal{T}_1}\rangle$ and a transducer for θ_2, $\mathcal{T}_2 = \langle P_2, \Sigma, p_2, \delta_{\mathcal{T}_2}, \rho_{\mathcal{T}_2}\rangle$. By Definition 20, $\mathcal{T}_1$ and $\mathcal{T}_2$ can not be similar. Intuitively, when w denotes the word on a cycle that witnesses the non-similarity of the transducers $\mathcal{T}_1$ and $\mathcal{T}_2$, we construct automata such that the maximum (or the addition) of their values on words containing w^n for an increasing n, alternates between $\frac{1}{F_1^n}$ and $\frac{1}{F_2^n}$, where F_1 and F_2 are the accumulated discount factors over the w cycle in $\mathcal{T}_1$ and $\mathcal{T}_2$ respectively. We then continue, as in the proof of Theorem 5, to show that no integral NMDA can express the relatively large jumps in this function.

Let v and w be finite words such that $\delta_{\mathcal{T}_1}(v) = \delta_{\mathcal{T}_1}(v\cdot w)$; $\delta_{\mathcal{T}_2}(v) = \delta_{\mathcal{T}_2}(v\cdot w)$; and

$$\prod_{k=0}^{|w|-1} \mathcal{T}_1\Big(v\cdot w[0..k]\Big) \neq \prod_{k=0}^{|w|-1} \mathcal{T}_2\Big(v\cdot w[0..k]\Big)$$

Denote:

- The accumulated discount factors over the runs of $\mathcal{T}_1$ and $\mathcal{T}_2$ on v as $E_1 = \prod_{k=0}^{|v|-1} \mathcal{T}_1\Big(v[0..k]\Big)$ and $E_2 = \prod_{k=0}^{|v|-1} \mathcal{T}_2\Big(v[0..k]\Big)$ respectively.
- The accumulated discount factors over the w cycle in $\mathcal{T}_1$ and $\mathcal{T}_2$ as $F_1 = \prod_{k=0}^{|w|-1} \mathcal{T}_1\Big(v\cdot w[0..k]\Big)$ and $F_2 = \prod_{k=0}^{|w|-1} \mathcal{T}_2\Big(v\cdot w[0..k]\Big)$ respectively. Without loss of generality, we have

$$F_2 > F_1 \tag{17}$$

- The first letter of w as $\tau' = w(0)$.

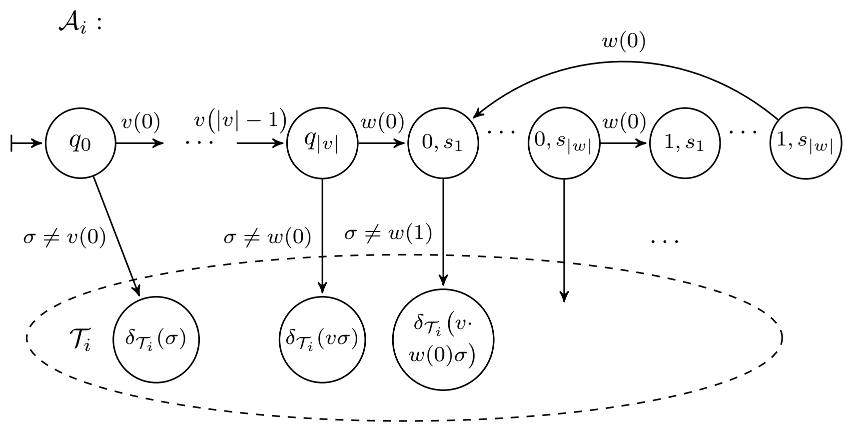

Figure 16: A partial sketch of the θ_1-DMDA and the θ_2-DMDA for which no integral NMDA is equivalent to their max or addition. The labels on the transitions indicate the input letter.

Let $\tau \in \Sigma$ such that $\tau \neq \tau'$. We will show a θ_1-DMDA $\mathcal{A}_1$ and a θ_2-DMDA $\mathcal{A}_2$, such that for every $n \in \mathbb{N}$,

$$\mathcal{A}_1(v \cdot w^n \cdot \tau^\omega) = \begin{cases} \frac{1}{F_1^n} & n \text{ is odd} \\ 0 & n \text{ is even} \end{cases} \quad \mathcal{A}_2(v \cdot w^n \cdot \tau^\omega) = \begin{cases} 0 & n \text{ is odd} \\ \frac{1}{F_2^n} & n \text{ is even} \end{cases} \tag{18}$$

Intuitively, $\mathcal{A}_1$ and $\mathcal{A}_2$ will each look like a $|v|$ sized list, walked upon the v word, followed by two $|w|$ sized lists, walked upon the w^2 word, having a back transition from the end of the third list to the beginning of the second list, to cause a cycle walked upon reading w^2 again and again. Each of them will also have a copy of the representing transducer, with transitions into it from all the states of the lists, such that whenever reading the first letter that breaks the pattern of $v \cdot w^n$, one of the transitions out of the lists will be taken, and the rest of the run will continue only in the states of the transducer.

All the transitions besides two will be zero weighted: In $\mathcal{A}_1$ the final transition in the second list, will have a weight that will assure a value of $\frac{1}{F_1^n}$, for odd number of occurrences of w. The transition from the second list to the third list will have a negative weight that will compensate on the previous value back to 0, in case another instance of w is present. Similarly, $\mathcal{A}_2$ will have a non-zero weighted transition as the final transition of the third list, and a negative weighted compensating transition from the third list back to the second list. A schematic sketch of $\mathcal{A}_1$ and $\mathcal{A}_2$ is given in Fig. 16.

Formally, for $i \in \{1, 2\}$, let $\mathcal{A}_i = \langle \Sigma, Q_i, \iota_i, \delta_i, \gamma_i, \rho_i \rangle$ be an NMDA such that:

- $Q_i = \{q_j \mid j \in \{0, 1, \ldots, |v|\}\} \cup \{0, 1\} \times \{s_j \mid j \in \{1, \ldots, |w|\}\} \cup P_i$.
- $\iota_i = \{q_0\}$.

- $\delta_i = \{(q_j, v(j), q_{j+1}) \mid j \in \{0,1,\ldots,|v|-1\}\} \cup$
$\Big\{\big(q_j, \sigma, \delta_{\mathcal{T}_i}(v[0..j-1]\cdot\sigma)\big) \mid j \in \{0,1,\ldots,|v|-1\}, \sigma \neq v(j)\Big\} \cup$
$\{\big(q_{|v|}, w(0), (0, s_0)\big)\} \cup$
$\Big\{\big(q_{|v|}, \sigma, \delta_{\mathcal{T}_i}(v\cdot\sigma)\big) \mid \sigma \neq w(0)\Big\} \cup$
$\Big\{\big((k, s_j), w(j), (k, s_{j+1})\big) \mid k \in \{0,1\}, j \in \{1,\ldots,|w|-1\}\Big\} \cup$
$\left\{\big((k, s_j), \sigma, \delta_{\mathcal{T}_1}(v\cdot w[0..j-1]\cdot\sigma)\big) \middle| \begin{matrix} k \in \{0,1\}, \\ j \in \{1,\ldots,|w|-1\}, \sigma \neq w(j) \end{matrix}\right\} \cup$
$\Big\{\big((k, s_{|w|-1}), w(0), (t, s_1)\big) \mid k \in \{0,1\}, t = k+1 \mod 2\Big\} \cup$
$\Big\{\big((k, s_{|w|-1}), \sigma, \delta_{\mathcal{T}_1}(v\cdot w\cdot\sigma)\big) \mid k \in \{0,1\}, \sigma \neq w(0)\Big\} \cup$
$\delta_{\mathcal{T}_i}$
- γ_i:
  - $\gamma_i\Big(\big(i-1, s_{|w|-1}\big), w\big(|w|-1\big), \big(i-1, s_{|w|}\big)\Big) = \frac{E_i}{\mathcal{T}_i(v\cdot w)}$.
  - $\gamma_i\Big(\big(i-1, s_{|w|}\big), w(0), \big(|i-2|, s_1\big)\Big) = -E_i$.
  - zero weights for all the other transitions.
- ρ_i:
  - For every $j \in \{0,1,\ldots,|v|\}$, letter $\sigma \in \Sigma$ and state $q \in Q_i$ such that $(q_j, \sigma, q) \in \delta_i$, we have $\rho_i(q_j, \sigma, q) = \mathcal{T}_i\big(v[0..j-1]\sigma\big)$.
  - For every $k \in \{0,1\}$, $j \in \{1,\ldots,|w|\}$, $\sigma \in \Sigma$ and $q \in Q_i$ such that $\big((k, s_j), \sigma, q\big) \in \delta_i$, we have $\rho_i\big((k, s_j), \sigma, q\big) = \mathcal{T}_i\big(v\cdot w[0..j-1]\sigma\big)$.
  - For every transition $t \in \delta_{\mathcal{T}_i}$, we have $\rho_i(t) = \rho_{\mathcal{T}_i}(t)$.
  - Observe that the above indeed assures that $\mathcal{A}_i$ is a θ_i-NMDA.

Denote the non-zero weighted transitions as $t_1 = \Big(\big(i-1, s_{|w|-1}\big), w\big(|w|-1\big), \big(i-1, s_{|w|}\big)\Big)$ and $t_2 = \Big(\big(i-1, s_{|w|}\big), w(0), \big(|i-2|, s_1\big)\Big)$. In all the runs of $\mathcal{A}_i$, either t_2 is always taken immediately after t_1, or the run moves to a state in P_i, and continues to stay in P_i. Observe that $\frac{\gamma_i(t_2)}{\gamma_i(t_1)} = -\mathcal{T}_i(v\cdot w)$. Also, in every run that takes t_2 immediately after t_1, the discount of $\gamma_i(t_2)$, divided by the discount of $\gamma_i(t_1)$ is $\mathcal{T}_i(v\cdot w)$. Hence, the value of A_i on every word $u \in \Sigma^*$, such that $\delta_i(u) \notin P_i$ and $\delta_i(u) \neq \big(i-1, s_{|w|}\big)$ is $\mathcal{A}_i(u) = 0$.

By the construction of $\mathcal{A}_1$ and $\mathcal{A}_2$, for $u \in \Sigma^*$, we have $\delta_1(u) = \big(0, s_{|w|}\big)$ if and only if there exists odd $n \in \mathbb{N}$ such that $u = v\cdot w^n$, and $\delta_2(u) = \big(1, s_{|w|}\big)$ if and only if there exists even $n \in \mathbb{N}$ such that $u = v\cdot w^n$. Hence for odd $n \in \mathbb{N}$, we have $\mathcal{A}_1(v\cdot w^n) = \frac{\gamma_i(t_1)}{E_1\cdot F_1^{n-1}\cdot\frac{F_1}{\mathcal{T}_i(v\cdot w)}} = \frac{1}{F_1^n}$. Similarly, for even $n \in \mathbb{N}$ we have $\mathcal{A}_2(v\cdot w^n) = \frac{1}{F_2^n}$.

Finally, a transition from $\big(i-1, s_{|w|}\big)$ on the letter $\tau \neq w(0)$, will force the run to continue in the zero weighted copy of $\mathcal{T}_i$, so Eq. (18) indeed holds.

The rest of the proof stands on the basis of the proof of Theorem 5, with the following modifications:

- In this case we don't necessary have $\max(\mathcal{A}_1, \mathcal{A}_2) \equiv \mathcal{A}_1 + \mathcal{A}_2$, but they only identify on every $v \cdot w^n \cdot \tau^\omega$ word.
- Replacing the discount factor 2 with F_1 and the discount factor 3 with F_2.
- Referring to the words $v \cdot w^n \cdot \tau^\omega$ instead of the words $a^n b^\omega$.
- Referring to the run prefix $r_n[0..|v| + n|w| - 1]$ instead of $r_n[0..n-1]$.
- Replacing the suffix b^ω with τ^ω and the suffix $a \cdot b^\omega$ with $w \cdot \tau^\omega$.

Implicitly, the modified proof follows. By Eq. (18), we have

$$\max(\mathcal{A}_1, \mathcal{A}_2)(v \cdot w^n \cdot \tau^\omega) = \Big(\mathcal{A}_1 + \mathcal{A}_2\Big)(v \cdot w^n \cdot \tau^\omega) = \begin{cases} \frac{1}{F_1^n} & n \text{ is odd} \\ \frac{1}{F_2^n} & n \text{ is even} \end{cases}$$

Assume toward contradiction that there exists an integral NMDA $\mathcal{C}'$, such that $\mathcal{C}' \equiv \max(\mathcal{A}_1, \mathcal{A}_2)$. Let $d \in \mathbb{N}$ be the least common denominator of the weights in $\mathcal{C}'$.

Consider the NMDA $\mathcal{C} = \langle \Sigma, Q, \iota, \delta, \gamma, \rho \rangle$ created from $\mathcal{C}'$ by multiplying all its weights by d. Observe that all the weights in $\mathcal{C}$ are integers. According to Proposition 2, for every $n \in \mathbb{N}$,

$$\mathcal{C}(v \cdot w^n \cdot \tau^\omega) = \begin{cases} \frac{d}{F_1^n} & n \text{ is odd} \\ \frac{d}{F_2^n} & n \text{ is even} \end{cases}$$

For every even $n \in \mathbb{N}$, let $w_n = v \cdot w^n \cdot \tau^\omega$, and r_n a run of $\mathcal{C}$ on w_n that entails the minimal value of $\frac{d}{F_2^n}$.

There exists a state $q \in Q$ such that for infinitely many even $n \in \mathbb{N}$, the target state of r_n after n steps is q, i.e, $\delta\big(r_n[0..|v| + n|w| - 1]\big) = q$. Let $U_b = \mathcal{C}^q(\tau^\omega)$ and $U_a = \mathcal{C}^q(w \cdot \tau^\omega)$, and for every such $n \in \mathbb{N}$, let $W_n = \mathcal{C}\big(r_n[0..|v| + n|w| - 1]\big)$, and $\Pi_n = \rho\big(r_n[0..|v| + n|w| - 1]\big)$.

For every such $n \in \mathbb{N}$, since $\mathcal{C}(r_n) = \frac{d}{F_2^n}$, we have

$$W_n + \frac{U_b}{\Pi_n} = \frac{d}{F_2^n} \tag{19}$$

and since the value of every run of $\mathcal{C}$ on $v \cdot w^{n+1} \cdot \tau^\omega$ is at least $\frac{d}{F_1^{n+1}}$, we have $W_n + \frac{U_a}{\Pi_n} \geq \frac{d}{F_1^{n+1}}$. Combining them both to get $\frac{d}{F_2^n} - \frac{U_b}{\Pi_n} + \frac{U_a}{\Pi_n} \geq \frac{d}{F_1^{n+1}}$ resulting in

$$\frac{U_a - U_b}{\Pi_n} \geq d \cdot \left(\frac{1}{F_1^{n+1}} - \frac{1}{F_2^n} \right) \tag{20}$$

Since $\left(\frac{F_2}{F_1}\right)^n \xrightarrow[n\to\infty]{} \infty$, we have that for large enough n, $F_2^n > F_1^{n+2}$. Hence, $\frac{1}{F_1^{n+2}} > \frac{1}{F_2^n}$ and $\frac{1}{F_1^{n+1}} - \frac{1}{F_2^n} > \frac{1}{F_1^{n+1}} - \frac{1}{F_1^{n+2}} = \frac{F_1 - 1}{F_1^{n+2}} > \frac{1}{F_1^{n+2}}$. Assign this into

Eq. (20) to get $\frac{U_a - U_b}{d} \cdot F_1^{n+2} \geq \Pi_n$. Hence, there exists a positive constant $m_1 = \frac{U_a - U_b}{d} \cdot F_1^2$ such that

$$m_1 \cdot F_1^n \geq \Pi_n \tag{21}$$

Since U_b is a rational constant, there exist $x \in \mathbb{Z}$ and $y \in \mathbb{N}$ such that $U_b = \frac{x}{y}$. Assign it into Eq. (19) to get

$$\frac{1}{F_2^n} = \frac{W_n \cdot \Pi_n + U_b}{d \cdot \Pi_n} = \frac{W_n \cdot \Pi_n + \frac{x}{y}}{d \cdot \Pi_n} = \frac{W_n \cdot \Pi_n \cdot y + x}{d \cdot y \cdot \Pi_n}$$

But since the denominator and the numerator of the right-hand side are integers, we conclude that there exists a positive constant $m_2 = d \cdot y$, such that $m_2 \cdot \Pi_n \geq F_2^n$. Combined with Eq. (21), we get $m_1 \cdot m_2 \cdot F_1^n \geq F_2^n$, for some positive constants m_1 and m_2, and for infinitely many $n \in \mathbb{N}$. But this stands in contradiction with $\lim_{n\to\infty} \left(\frac{F_1}{F_2}\right)^n = 0$.

Note that all the above also hold when changing $\max(\mathcal{A}_1, \mathcal{A}_2)$ to $\mathcal{A}_1 + \mathcal{A}_2$. □

Observe that the limitation in Theorem 23 for a non-singleton alphabet is required only for the infinite-words case. For a singleton alphabet, only eventually constant functions exist in the infinite-words case, and all of them can be represented by a θ-NMDA for any choice function θ. For the finite-words case, this limitation is not required.

Theorem 24. *For every two non-similar choice functions θ_1 and θ_2, there exist a θ_1-NMDA $\mathcal{A}_1$ and a θ_2-NMDA $\mathcal{A}_2$ such that no integral NMDA (even non-tidy one), is equivalent to* $\max\{\mathcal{A}_1, \mathcal{A}_2\}$ *or to* $\mathcal{A}_1 + \mathcal{A}_2$.

Proof. Let θ_1 and θ_2 be non-similar choice functions, $\mathcal{A}_1$ a θ_1-NMDA and $\mathcal{A}_2$ a θ_2-NMDA as constructed in the proof of Theorem 23. As shown in the proof of Theorem 23, for every $n \in \mathbb{N}$,

$$\mathcal{A}_1(v \cdot w^n) = \begin{cases} \frac{1}{F_1^n} & n \text{ is odd} \\ 0 & n \text{ is even} \end{cases} \quad \mathcal{A}_2(v \cdot w^n) = \begin{cases} 0 & n \text{ is odd} \\ \frac{1}{F_2^n} & n \text{ is even} \end{cases}$$

where F_1 and F_2 defined as in the proof of Theorem 23.

The rest of the proof is similar to the proof of Theorem 23 with the following adjustments:

- Replacing $v \cdot w^n \cdot \tau^\omega$ with $v \cdot w^n$.
- Replacing U_b with 0 and defining $U_a = \mathcal{C}^q(w)$ instead of $U_a = \mathcal{C}^q(w \cdot \tau^\omega)$.

□

4.5.2 Class intersection

We show that for every alphabet Σ, the intersection of all θ-NMDA classes over Σ is exactly the set of eventually constant functions. We start with a formal definition of eventually constant functions, and continue with Lemmas 26 and 27 that provide the two directions of the proof.

Definition 25. *For an alphabet Σ and a number $n \in \mathbb{N}$,*

- *A function $f : \Sigma^+ \to \mathbb{Q}$ is n-constant if for every finite word w of length at least n, we have $f(w) = f(w[0..n-1])$.*
- *A function $f : \Sigma^\omega \to \mathbb{Q}$ is n-constant if for every $w, w' \in \Sigma^\omega$, we have $f(w) = f(w[0..n-1] \cdot w')$.*
- *A function $f : \Sigma^+ \to \mathbb{Q}$ or $f : \Sigma^\omega \to \mathbb{Q}$ is* eventually constant *if there exists $n \in \mathbb{N}$, such that f is n-constant.*

Lemma 26. *For every eventually constant function f and choice function θ over the same alphabet, there exists a θ-NMDA that represents f.*

Proof. Let $\Sigma = \{\sigma_1, \sigma_2, \cdots, \sigma_k\}$ be an alphabet, $n \in \mathbb{N}$, f an n-constant function, θ a choice function, and $\mathcal{T} = \langle P, \Sigma, p_0, \delta_\mathcal{T}, \rho_\mathcal{T} \rangle$ a transducer for θ.

Finite words.

Let $f : \Sigma^+ \to \mathbb{Q}$ be an n-constant function. We will describe a θ-DMDA $\mathcal{A} = \langle \Sigma, Q, \{q_\varepsilon\}, \delta, \gamma, \rho \rangle$ that represents f.

Schematic sketch of $\mathcal{A}$ can be found in Fig. 17. Intuitively, $\mathcal{A}$ will be a tree with exactly $|\Sigma|$ children for every node up to depth of n, followed by a copy of the transducer $\mathcal{T}$ with 0-weighted transitions.

We will iteratively add states to $\mathcal{A}$, level after level of the tree. With this construction, for every state there is a single word leading to it. The weight of every new transition from a state q will be calculated according to the corresponding discount factor in $\mathcal{T}$, and the accumulated weight up to q, such that the value after it will comply with f. This process will continue up to depth n.

Every state in level n will be connected to the state in the copy of $\mathcal{T}$ that synchronizes with the input word that caused reaching it. This will preserve the correct discount factors when the following transitions will be taken, to finally ensure that $\mathcal{A}$ is indeed a θ-NMDA. All the weights in the copy of $\mathcal{T}$ will be 0. Since f is an n-constant function, this will ensure equivalence to f for all words.

Formally, the states of $\mathcal{A}$ are

$$Q = \{q_w | w \in \Sigma^* \text{ and } |w| \leq n\} \cup P$$

For simplifying the definition of $\mathcal{A}$'s weight function, we extend f to the empty word, defining $f(\varepsilon) = 0$.

The transitions, weights and discount factors of $\mathcal{A}$ are:

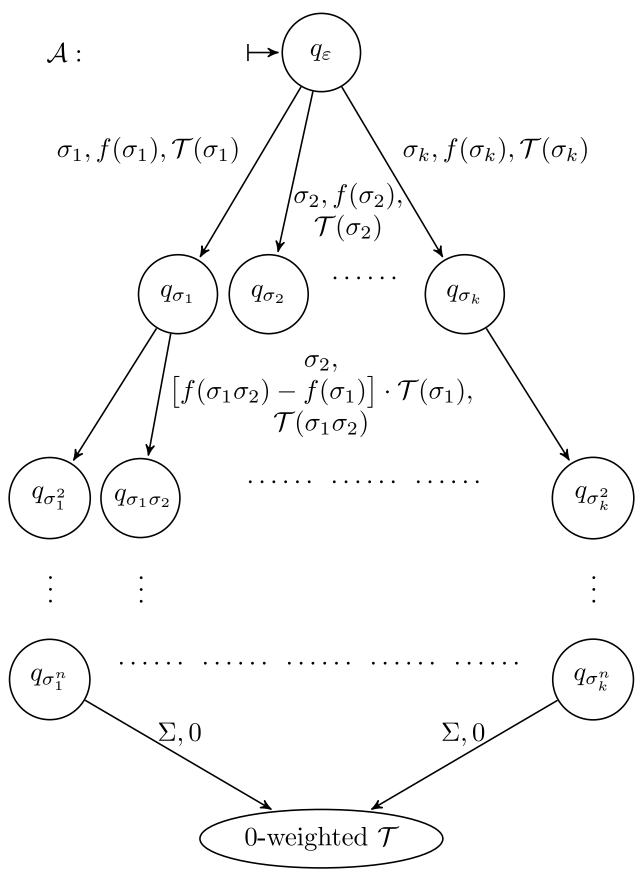

Figure 17: Partial sketch of the θ-NMDA for the n-constant function f in the proof of Lemma 26 for the finite-words case.

- For every $w \in \Sigma^*$ such that $0 \leq |w| < n$, and every $1 \leq j \leq k$,

$$t_{w,j} = (q_w, \sigma_j, q_{w\cdot\sigma_j}) \in \delta$$
$$\gamma(t_{w,j}) = \big[f(w\cdot\sigma_j) - f(w)\big] \cdot \prod_{i=1}^{j-1} \mathcal{T}\big(w[0..i]\big)$$
$$\rho(t_{w,j}) = \mathcal{T}(w\cdot\sigma_j)$$

- For every $p \in P$, and $1 \leq j \leq k$,

$$t_{p,j} = \big(p, \sigma_j, \delta_{\mathcal{T}}(p,\sigma_j)\big) \in \delta$$
$$\gamma(t_{p,j}) = 0$$
$$\rho(t_{p,j}) = \rho_{\mathcal{T}}\big(p, \delta_{\mathcal{T}}(p,\sigma_j)\big)$$

- For every $w \in \Sigma^n$, and $1 \leq j \leq k$,

$$t_{w,j} = \big(q_w, \sigma_j, \delta_{\mathcal{T}}(w\cdot\sigma_j)\big) \in \delta$$
$$\gamma(t_{w,j}) = 0$$
$$\rho(t_{w,j}) = \mathcal{T}(w\cdot\sigma_j)$$

Infinite words.
Let $f : \Sigma^\omega \to \mathbb{Q}$ be an n-constant function. Similarly to the construction in the finite-words case, the θ-DMDA $\mathcal{A} = \langle \Sigma, Q, \{q_\varepsilon\}, \delta, \gamma, \rho\rangle$ that represents f will be a tree of depth n followed by a copy of $\mathcal{T}$. Yet, in the infinite case, all the weights in the tree up to depth n will be zeroes, and the weights of the transitions from depth n to depth $n+1$ will be the value of f on the equivalent word. (See a sketch of $\mathcal{A}$ in Fig. 18.)

Formally, the transitions, weights and discount factors of $\mathcal{A}$ are:

- For every $w \in \Sigma^*$ such that $0 \leq |w| < n$, and every $1 \leq j \leq k$,

$$t_{w,j} = (q_w, \sigma_j, q_{w\cdot\sigma_j}) \in \delta$$
$$\gamma(t_{w,j}) = 0$$
$$\rho(t_{w,j}) = \mathcal{T}(w\cdot\sigma_j)$$

- For every $w \in \Sigma^n$, and $1 \leq j \leq k$,

$$t_{w,j} = \big(q_w, \sigma_j, \delta_{\mathcal{T}}(w\cdot\sigma_j)\big) \in \delta$$
$$\gamma(t_{w,j}) = f(w\cdot\sigma_1^\omega)$$
$$\rho(t_{w,j}) = \mathcal{T}(w\cdot\sigma_j)$$

- The rest of the transitions are identical to the finite-words case.

□

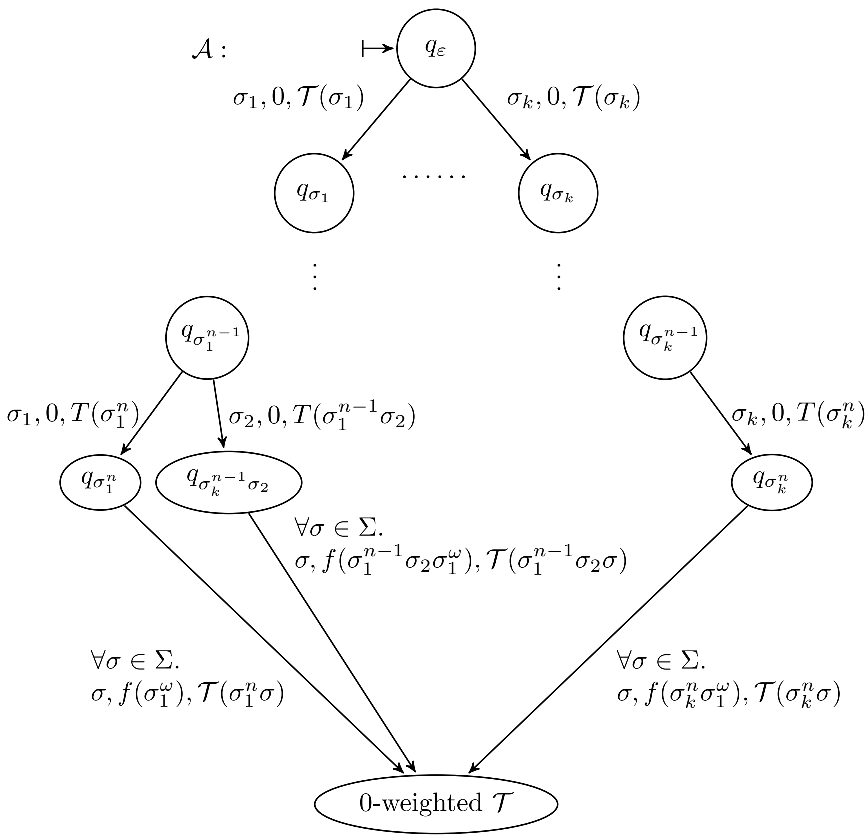

Figure 18: Partial sketch of the θ-NMDA for the n-constant function f in the proof of Lemma 26 for the infinite-words case.

We will now show that the intersection of all the θ-NMDA classes contains only eventually constant functions.

Lemma 27. *For every non-eventually constant function f over an alphabet Σ, there exists a choice function θ over Σ, such that no θ-NMDA represents f.*

Proof. Let f be a non-eventually constant function. We will show that there is no 2-NDA that represents f or no 3-NDA that represents f.

Assume toward contradiction that there exist a 2-NDA $\mathcal{A}'$ and 3-NDA $\mathcal{B}'$ such that $\mathcal{A}' \equiv \mathcal{B}' \equiv f$. Hence, according to [9] there exist a 2-DDA $\mathcal{A} = \langle \Sigma, Q_\mathcal{A}, \iota_\mathcal{A}, \delta_\mathcal{A}, \gamma_\mathcal{A} \rangle$, and 3-DDA $\mathcal{B} = \langle \Sigma, Q_\mathcal{B}, \iota_\mathcal{B}, \delta_\mathcal{B}, \gamma_\mathcal{B} \rangle$, such that

$$\mathcal{A} \equiv \mathcal{B} \equiv f \tag{22}$$

Finite words.

Intuitively, for a non-eventually constant function and a finite alphabet, there exist a letter σ and infinitely many words that their value is changed when appending σ to them. We find two such words, with different sizes n_1 and n_2, that cause the same final state both in $\mathcal{A}$ and in $\mathcal{B}$. We then show that it is impossible to reach the desired value after appending σ, in all the cases of accumulated discount factors of 2^{n_1}, 2^{n_2}, 3^{n_1} and 3^{n_2}.

Since f is not eventually constant, for every $0 < k \in \mathbb{N}$, there exist $\sigma \in \Sigma$, $k \leq n \in \mathbb{N}$ and $u \in \Sigma^n$ such that $f(u) \neq f(u \cdot \sigma)$. $\mathcal{A}$ and $\mathcal{B}$ have finitely many states, implying that there exist states $q_\mathcal{A} \in Q_\mathcal{A}$ and $q_\mathcal{B} \in Q_\mathcal{B}$, a letter $\sigma \in \Sigma$, numbers $n_1 \neq n_2 \in \mathbb{N}$, and words $u_1 \in \Sigma^{n_1}$ and $u_2 \in \Sigma^{n_2}$, such that

$$f(u_1) \neq f(u_1 \cdot \sigma) \tag{23}$$
$$f(u_2) \neq f(u_2 \cdot \sigma) \tag{24}$$
$$\delta_\mathcal{A}(u_1) = \delta_\mathcal{A}(u_2) = q_\mathcal{A} \tag{25}$$

and

$$\delta_\mathcal{B}(u_1) = \delta_\mathcal{B}(u_2) = q_\mathcal{B} \tag{26}$$

Combine Eqs. (22), (25) and (26) to get

$$\frac{\gamma_\mathcal{A}(q_\mathcal{A}, \sigma)}{2^{n_1}} = \mathcal{A}(u_1 \cdot \sigma) - \mathcal{A}(u_1) = \mathcal{B}(u_1 \cdot \sigma) - \mathcal{B}(u_1) = \frac{\gamma_\mathcal{B}(q_\mathcal{B}, \sigma)}{3^{n_1}}$$

and since according to Eqs. (22) and (23), $\gamma_\mathcal{B}(q_\mathcal{B}, \sigma) \neq 0$, we get

$$\frac{\gamma_\mathcal{A}(q_\mathcal{A}, \sigma)}{\gamma_\mathcal{B}(q_\mathcal{B}, \sigma)} = \left(\frac{2}{3}\right)^{n_1}$$

Similarly, when combining Eqs. (22) and (24) to (26), we get

$$\frac{\gamma_\mathcal{A}(q_\mathcal{A}, \sigma)}{\gamma_\mathcal{B}(q_\mathcal{B}, \sigma)} = \left(\frac{2}{3}\right)^{n_2}$$

and a contradiction to $n_1 \neq n_2$.

Infinite words.

We use a similar idea as in the finite-words case, with replacing the single final transition with transitions over two different infinite suffix words.

Since f is not eventually constant, for every $0 < k \in \mathbb{N}$, there exist $w_1, w_2 \in \Sigma^\omega$, $k \leq n \in \mathbb{N}$ and $u \in \Sigma^n$ such that $f(u \cdot w_1) \neq f(u \cdot w_2)$. Hence, there exists a state $q_{\mathcal{A}} \in Q_{\mathcal{A}}$ such that for every $0 < k \in \mathbb{N}$, there exist $w_1, w_2 \in \Sigma^\omega$, $k \leq n \in \mathbb{N}$ and $u \in \Sigma^n$, such that $\delta_{\mathcal{A}}(u) = q_{\mathcal{A}}$ and $\mathcal{A}^{q_{\mathcal{A}}}(w_1) \neq \mathcal{A}^{q_{\mathcal{A}}}(w_2)$. Hence, there exist states $q_{\mathcal{A}} \in Q_{\mathcal{A}}$ and $q_{\mathcal{B}} \in Q_{\mathcal{B}}$, infinite words $w_1, w_2 \in \Sigma^\omega$, integral numbers $n_1 \neq n_2 \in \mathbb{N}$, and finite words $u_1 \in \Sigma^{n_1}$, $u_2 \in \Sigma^{n_2}$, such that

$$\mathcal{A}^{q_{\mathcal{A}}}(w_1) \neq \mathcal{A}^{q_{\mathcal{A}}}(w_2) \tag{27}$$
$$\delta_{\mathcal{A}}(u_1) = \delta_{\mathcal{A}}(u_2) = q_{\mathcal{A}} \tag{28}$$

and

$$\delta_{\mathcal{B}}(u_1) = \delta_{\mathcal{B}}(u_2) = q_{\mathcal{B}} \tag{29}$$

Combining Eqs. (22) and (27) to (29) to get

$$\begin{aligned}
\frac{\mathcal{A}^{q_{\mathcal{A}}}(w_2) - \mathcal{A}^{q_{\mathcal{A}}}(w_1)}{2^{n_1}} &= \mathcal{A}(u_1 \cdot w_2) - \mathcal{A}(u_1 \cdot w_1) \\
&= f(u_1 \cdot w_2) - f(u_1 \cdot w_1) \\
&= \mathcal{B}(u_1 \cdot w_2) - \mathcal{B}(u_1 \cdot w_1) \\
&= \frac{\mathcal{B}^{q_{\mathcal{B}}}(w_2) - \mathcal{B}^{q_{\mathcal{B}}}(w_1)}{3^{n_1}}
\end{aligned} \tag{30}$$

and similarly

$$\frac{\mathcal{A}^{q_{\mathcal{A}}}(w_2) - \mathcal{A}^{q_{\mathcal{A}}}(w_1)}{2^{n_2}} = \frac{\mathcal{B}^{q_{\mathcal{B}}}(w_2) - \mathcal{B}^{q_{\mathcal{B}}}(w_1)}{3^{n_2}} \tag{31}$$

Eqs. (30) and (31) are a contradiction to either Eq. (27) or to $n_1 \neq n_2$. □

Theorem 28. *For every alphabet Σ, the intersection of all θ-NMDA classes over Σ is exactly the set of all eventually constant functions over Σ.*

Proof. Follows directly from Lemmas 26 and 27. □

4.5.3 Class inclusion

It turns out that for every choice functions θ_1 and θ_2, the class of θ_1-NMDAs is not more expressive than the class of θ_2-NMDAs. For showing it, we provide two complementary results: The first proves that the classes of θ_1-NMDAs and of θ_2-NMDAs define the same set of functions if θ_1 and θ_2 are similar. The second proves that if θ_1 and θ_2 can be represented by non-similar transducers then there exists a θ_1-NMDA, such that no θ_2-NMDA is equivalent to. (Notice that it also implies that two transducers representing the same choice function must be similar.)

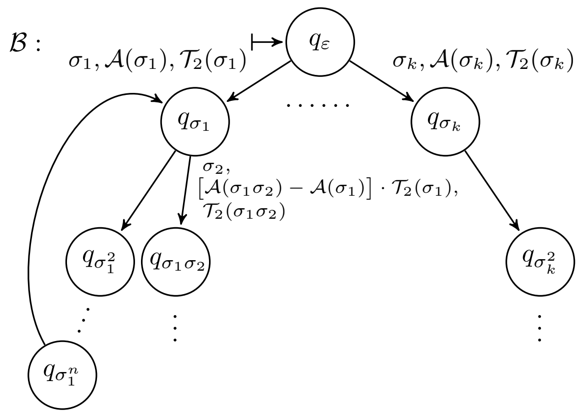

Figure 19: A sketch of the θ_2-DMDA constructed in the proof of Theorem 29.

Theorem 29. *For every similar choice functions θ_1 and θ_2, every θ_1-NMDA has an equivalent θ_2-NMDA.*

Proof. Consider similar choice functions θ_1 and θ_2 over an alphabet Σ and a θ_1-NMDA $\mathcal{A}'$. Let $\mathcal{T}_1 = \langle P_{\mathcal{T}_1}, \Sigma, \mathbb{N}, p_{\mathcal{T}_1}, \delta_{\mathcal{T}_1}, \rho_{\mathcal{T}_1} \rangle$ and $\mathcal{T}_2 = \langle P_{\mathcal{T}_2}, \Sigma, \mathbb{N}, p_{\mathcal{T}_2}, \delta_{\mathcal{T}_2}, \rho_{\mathcal{T}_2} \rangle$ be transducers for θ_1 and θ_2 respectively. By Theorem 11, there exists a θ_1-DMDA $\mathcal{A} = \langle \Sigma, Q_\mathcal{A}, \{q_0\}, \delta_\mathcal{A}, \gamma_\mathcal{A}, \rho_\mathcal{A} \rangle$ equivalent to $\mathcal{A}'$. We will construct a θ_2-DMDA $\mathcal{B} = \langle \Sigma, Q_\mathcal{B}, q_\varepsilon, \delta_\mathcal{B}, \gamma_\mathcal{B}, \rho_\mathcal{B} \rangle$ equivalent to $\mathcal{A}$.

Intuitively, $\mathcal{B}$ will be built as a tree while ensuring that the discount factors agree with $\mathcal{T}_2$, and the values agree with $\mathcal{A}$. Yet, whenever reaching a word that has a prefix that causes a "similar cycle" in $\mathcal{A}$, $\mathcal{T}_1$, and $\mathcal{T}_2$, we will have a transition to the appropriate ancestor in the tree instead of a transition to a new child. (See a sketch of $\mathcal{B}$ in Fig. 19.) We shall then prove that this process eventually terminates in all the branches and yields a finite automaton $\mathcal{B}$ equivalent to $\mathcal{A}$.

We create $\mathcal{B}$ through the following procedure:

1. Initialize $Q_\mathcal{B} = \{q_\varepsilon\}, \delta_\mathcal{B} = \emptyset, \gamma_\mathcal{B} = \emptyset, \rho_\mathcal{B} = \emptyset$, and a "stack" $S = \{\langle q_\varepsilon, \varepsilon, 1 \rangle\}$.

   Each tuple in S represents a state we need to continue the build of $\mathcal{B}$ from, the word on which $\mathcal{B}$ reached that state, and the accumulated discount factor in $\mathcal{B}$ running on that word.

2. Loop while S is not empty:

   (a) Pop $\langle q, u, \Pi \rangle$ from S.

   (b) For every $\sigma \in \Sigma$:

      i. If u has a prefix v such that $\delta_\mathcal{A}(v) = \delta_\mathcal{A}(u \cdot \sigma)$, $\delta_{\mathcal{T}_1}(v) = \delta_{\mathcal{T}_1}(u \cdot \sigma)$ and $\delta_{\mathcal{T}_2}(v) = \delta_{\mathcal{T}_2}(u \cdot \sigma)$, add a "back transition", $t = \big(q, \sigma, \delta_\mathcal{B}(v)\big)$, to $\delta_\mathcal{B}$.

ii. Else add a new state $q_{u\cdot\sigma}$ to $Q_\mathcal{B}$, a transition $t = (q, \sigma, q_{u\cdot\sigma})$ to $\delta_\mathcal{B}$, and push the tuple $\langle q_{u\cdot\sigma}, u\cdot\sigma, \Pi\cdot\mathcal{T}_2(u\cdot\sigma)\rangle$ to S.

(c) Define the weight and discount factor of the transition t added in the previous step to be
$\gamma_\mathcal{B}(t) = \big[\mathcal{A}(u\cdot\sigma) - \mathcal{A}(u)\big]\cdot\Pi$, $\rho_\mathcal{B}(t) = \mathcal{T}_2(u\cdot\sigma)$

In step 2(b)i, we take a prefix v, such that $\delta_\mathcal{A}(v) = \delta_\mathcal{A}(u\cdot\sigma)$, $\delta_{\mathcal{T}_1}(v) = \delta_{\mathcal{T}_1}(u\cdot\sigma)$ and $\delta_{\mathcal{T}_2}(v) = \delta_{\mathcal{T}_2}(u\cdot\sigma)$. Observe that such a prefix v, when exists, is unique: Assume toward contradiction that there exists another prefix $v' \neq v$ such that $\delta_\mathcal{A}(v') = \delta_\mathcal{A}(u\cdot\sigma)$, $\delta_{\mathcal{T}_1}(v') = \delta_{\mathcal{T}_1}(u\cdot\sigma)$ and $\delta_{\mathcal{T}_2}(v') = \delta_{\mathcal{T}_2}(u\cdot\sigma)$. Without loss of generality, we assume that v' is a prefix of v. But we have $\delta_\mathcal{A}(v') = \delta_\mathcal{A}(v)$, $\delta_{\mathcal{T}_1}(v') = \delta_{\mathcal{T}_1}(v)$ and $\delta_{\mathcal{T}_2}(v') = \delta_{\mathcal{T}_2}(v)$. This stands in contradiction to a previous iteration of the algorithm when v was examined and the 2(b)i step did not find the v' prefix of v to satisfy this property.

We prove that the procedure eventually terminates by showing that each branch of the tree (after which there is only the back-transition) is not longer than $X = \Big(|Q_\mathcal{A}|+1\Big)\cdot|P_{T_1}|\cdot|P_{T_2}|$. For every word w of length X, there exists a state $p_2 \in P_{T_2}$ that appears at least $Y = \Big(|Q_\mathcal{A}|+1\Big)\cdot|P_{T_1}|$ times in the run of $\mathcal{T}_2$ on w. Therefore, there exist prefixes $w_1, w_2, \cdots, w_Y$ of w, such that for every $0 < i \leq Y$, we have $\delta_{\mathcal{T}_2}(w_i) = p_2$. Also, there exist a state $p_1 \in P_{T_1}$ and at least $Z = |Q_\mathcal{A}| + 1$ prefixes of w out of the above Y prefixes, named $w_{i_1}, w_{i_2}, \cdots, w_{i_Z}$, such that for every $0 < j \leq Z$, we have $\delta_{\mathcal{T}_1}(w_{i_j}) = p_1$ and $\delta_{\mathcal{T}_2}(w_{i_j}) = p_2$. Finally, there exist prefixes w' and w'' of w out of the above Z prefixes, such that $\delta_\mathcal{A}(w') = \delta_\mathcal{A}(w'')$, $\delta_{\mathcal{T}_1}(w') = \delta_{\mathcal{T}_1}(w'')$ and $\delta_{\mathcal{T}_2}(w') = \delta_{\mathcal{T}_2}(w'')$, satisfying the condition of step 2(b)i. Hence, for each tree branch, step 2(b)i will be taken before reaching length of X.

Observe that due to the construction of $\mathcal{B}$, every state in $\mathcal{B}$ corresponds to a state in $\mathcal{A}$, a state in $\mathcal{T}_1$ and a state in $\mathcal{T}_2$. That is, for every $q_\mathcal{B} \in Q_\mathcal{B}$, there exist $q_\mathcal{A} \in Q_\mathcal{A}$, $q_{\mathcal{T}_1} \in P_1$ and $q_{\mathcal{T}_2} \in P_2$, such that for every $w \in \Sigma^*$ for which $\delta_\mathcal{B}(w) = q_\mathcal{B}$, we have $\delta_\mathcal{A}(w) = q_\mathcal{A}$, $\delta_{\mathcal{T}_1}(w) = q_{\mathcal{T}_1}$ and $\delta_{\mathcal{T}_2}(w) = q_{\mathcal{T}_2}$. This holds since all the transitions to an ancestor (in step 2(b)i) were added only if the same prefix caused a cycle in $\mathcal{A}$, in $\mathcal{T}_1$ and in $\mathcal{T}_2$ back to the corresponding states.

Therefore, since all the discount factors in $\mathcal{B}$ were added with values that match $\mathcal{T}_2$, we conclude that $\mathcal{B}$ is indeed a θ_2-NMDA.

It is left to show that $\mathcal{B}$ is equivalent to $\mathcal{A}$. Let $f_\mathcal{A} : Q_\mathcal{B} \to Q_\mathcal{A}$ be the function that returns the corresponding state in $\mathcal{A}$ with respect to the property mentioned above. Observe that for every $y \in \Sigma^+$, we have $f_\mathcal{A}\Big(\delta_\mathcal{B}(y)\Big) = \delta_\mathcal{A}(y)$.

Consider a state $q_u \in Q_\mathcal{B}$ and a letter $\sigma \in \Sigma$. The transition t from q_u on σ has, by the construction of $\mathcal{B}$, weight $\gamma_\mathcal{B}(t) = \Big[\mathcal{A}(u\cdot\sigma) - \mathcal{A}(u)\Big]\cdot\prod_{i=0}^{m-1}\mathcal{T}_2(u[0..i])$. Let $W_\mathcal{A}$ be the weight of the σ transition from $f_\mathcal{A}(q_u)$ in $\mathcal{A}$, i.e., $W_\mathcal{A} =$

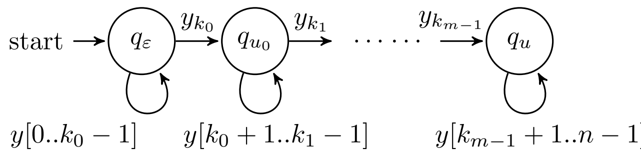

Figure 20: An illustration of the structure of the run r of $\mathcal{B}$ on y from the proof of Theorem 29. The labels on the transitions stand for the input letter/word for it.

$\gamma_{\mathcal{A}}\Big(f_{\mathcal{A}}(q_u), \sigma\Big)$. Hence,

$$\begin{aligned}\gamma_{\mathcal{B}}(t) &= \Big[\mathcal{A}(u) + \frac{W_{\mathcal{A}}}{\prod_{i=0}^{m-1} \mathcal{T}_1(u[0..i])} - \mathcal{A}(u)\Big] \cdot \prod_{i=0}^{m-1} \mathcal{T}_2(u[0..i]) \\ &= W_{\mathcal{A}} \cdot \prod_{i=0}^{m-1} \frac{\mathcal{T}_2(u[0..i])}{\mathcal{T}_1(u[0..i])} \end{aligned} \tag{32}$$

Let $y \in \Sigma^*$ be a word such that $\delta_{\mathcal{B}}(y) = q_u$ and r be the run of $\mathcal{B}$ on y. Due to the construction of $\mathcal{B}$, r identifies with the run of $\mathcal{B}$ on u (which is simply going along the tree), with possibly having some cycles in the middle. Formally, for every $0 \le j \le m-1$, we have

- $y(k_j) = u(j)$.
- $r(k_j) = \big(q_{u[0..j-1]}, u(j), q_{u[0..j]}\big)$, where for $j = 0$ we define $u[0..-1] = \varepsilon$.
- Each of the subwords $y[0..k_0-1]$, $y[k_{m-1}+1..n-1]$, and $y[k_j+1..k_{j+1}-1]$ for every $0 \le j < m-1$ is either empty, or the equivalent sub-walk of r on it contains a cycle.

The described structure can be seen in Fig. 20.

Due to the construction of $\mathcal{B}$, all the above subwords also cause a cycle in $\mathcal{T}_1$ and in $\mathcal{T}_2$. Since $\mathcal{T}_1$ and $\mathcal{T}_2$ are similar, we have

$$\prod_{i=0}^{k_0-1} \frac{\mathcal{T}_2(y[0..i])}{\mathcal{T}_1(y[0..i])} = 1 \tag{33}$$

$$\forall_{0 \le j < m-1} \prod_{i=k_j+1}^{k_{j+1}-1} \frac{\mathcal{T}_2(y[0..i])}{\mathcal{T}_1(y[0..i])} = 1 \tag{34}$$

$$\prod_{i=k_{m-1}+1}^{n-1} \frac{\mathcal{T}_2(y[0..i])}{\mathcal{T}_1(y[0..i])} = 1 \tag{35}$$

We will show by induction on j that for every $0 \le j \le m-1$,

$$\prod_{i=0}^{j} \frac{\mathcal{T}_2(u[0..i])}{\mathcal{T}_1(u[0..i])} = \prod_{i=0}^{k_j} \frac{\mathcal{T}_2(y[0..i])}{\mathcal{T}_1(y[0..i])} \tag{36}$$

For the base case of $j = 0$, since $y[0..k_0 - 1]$ causes a cycle in $\mathcal{B}$ or was empty, $\delta_{\mathcal{T}_1}(y[0..k_0 - 1]) = p_{\mathcal{T}1}$ and $\delta_{T_2}(y[0..k_0 - 1]) = p_{\mathcal{T}_2}$. Hence,

$$\mathcal{T}_1(y[0..k_0]) = \rho_{\mathcal{T}_1}(p_{\mathcal{T}1}, y_{k_0}) = \rho_{\mathcal{T}_1}(p_{\mathcal{T}1}, u_0) = \mathcal{T}_1(u_0)$$

and similarly $\mathcal{T}_2(y[0..k_0]) = \mathcal{T}_2(u_0)$. Combine this with Eq. (33) to get $\frac{\mathcal{T}_2(u_0)}{\mathcal{T}_1(u_0)} = \prod_{i=0}^{k_0} \frac{\mathcal{T}_2(y[0..i])}{\mathcal{T}_1(y[0..i])}$.

For the induction step, assume Eq. (36) holds and combine it with Eq. (34) to get

$$\begin{aligned}\prod_{i=0}^{k_{j+1}} \frac{\mathcal{T}_2(y[0..i])}{\mathcal{T}_1(y[0..i])} &= \prod_{i=k_j+1}^{k_{j+1}} \frac{\mathcal{T}_2(y[0..i])}{\mathcal{T}_1(y[0..i])} \cdot \prod_{i=0}^{k_j} \frac{\mathcal{T}_2(y[0..i])}{\mathcal{T}_1(y[0..i])} \\ &= \frac{\mathcal{T}_2(y[0..k_{j+1}])}{\mathcal{T}_1(y[0..k_{j+1}])} \cdot \prod_{i=0}^{j} \frac{\mathcal{T}_2(u[0..i])}{\mathcal{T}_1(u[0..i])}\end{aligned}$$

But similarly to the base case, since $y[0..k_{j+1}]$ and $u[0..j]$ both lead to the same state also in $\mathcal{T}_1$, we have $\mathcal{T}_1(y[0..k_{j+1}]) = \mathcal{T}_1(u[0..j])$ and since they lead to the same state in $\mathcal{T}_2$ we have $\mathcal{T}_2(y[0..k_{j+1}]) = \mathcal{T}_2(u[0..j])$. Combined with the above, we get $\prod_{i=0}^{j+1} \frac{\mathcal{T}_2(u[0..i])}{\mathcal{T}_1(u[0..i])} = \prod_{i=0}^{k_{j+1}} \frac{\mathcal{T}_2(y[0..i])}{\mathcal{T}_1(y[0..i])}$.

Now combine Eqs. (32), (35) and (36) to get that for every state $q_u \in Q_{\mathcal{B}}$, letter $\sigma \in \Sigma$, and word $y \in \Sigma^*$ such that $\delta_{\mathcal{B}}(y) = q_u$, we have

$$\gamma_{\mathcal{B}}(q_u, \sigma) = \gamma_{\mathcal{A}}\Big(f_{\mathcal{A}}(q_u), \sigma\Big) \cdot \prod_{i=0}^{|y|-1} \frac{\mathcal{T}_2(y[0..i])}{\mathcal{T}_1(y[0..i])} \tag{37}$$

Finally, we will show by induction on the size of the word that for every $y \in \Sigma^*$, we have $\mathcal{A}(y) = \mathcal{B}(y)$.

We extend $\mathcal{A}$ and $\mathcal{B}$ to have the value of 0 on empty words, so the claim will hold for the base case of an empty word. Now assume the claim holds for a word y of length l, and let $\sigma \in \Sigma$. We will show that the claim holds for $y \cdot \sigma$. Let t be the transition in $\mathcal{B}$ from $\delta_{\mathcal{B}}(y)$ on σ. We have

$$\begin{aligned}\mathcal{B}(y \cdot \sigma) &= \mathcal{B}(y) + \frac{\gamma_{\mathcal{B}}(t)}{\prod_{i=0}^{l-1} \mathcal{T}_2(y[0..i])} \\ &= \mathcal{A}(y) + \frac{\gamma_{\mathcal{B}}(t)}{\prod_{i=0}^{l-1} \mathcal{T}_2(y[0..i])}\end{aligned}$$

Combined with Eq. (37), we get

$$\begin{aligned}\mathcal{B}(y \cdot \sigma) &= \mathcal{A}(y) + \frac{\gamma_{\mathcal{A}}\Big(f_{\mathcal{A}}(\delta_{\mathcal{B}}(y)), \sigma\Big) \cdot \prod_{i=0}^{l-1} \frac{\mathcal{T}_2(y[0..i])}{\mathcal{T}_1(y[0..i])}}{\prod_{i=0}^{l-1} \mathcal{T}_2(y[0..i])} \\ &= \mathcal{A}(y) + \frac{\gamma_{\mathcal{A}}\Big(\delta_{\mathcal{A}}(y), \sigma\Big)}{\prod_{i=0}^{l-1} \mathcal{T}_1(y[0..i])} \\ &= \mathcal{A}(y \cdot \sigma)\end{aligned}$$

Concluding that for every finite word $w \in \Sigma^+$, we have $\mathcal{B}(w) = A'(w)$. Also, according to Lemma 3, for every infinite word $u \in \Sigma^\omega$, we have $\mathcal{B}(u) = A'(u)$. □

The complementary result is a direct corollary of Theorems 13 and 23: For two non-similar choice functions θ_1 and θ_2, if every function represented by a θ_1-NMDA can also be represented by a θ_2-NMDA, we get a contradiction between Theorems 13 and 23.

Corollary 30. *For every non-singleton alphabet Σ and non-similar choice functions θ_1 and θ_2 over Σ, there exists a θ_1-NMDA such that no θ_2-NMDA is equivalent to.*

From Corollary 30 and Theorem 29, we directly get the following:

Corollary 31. *For every choice functions θ_1 and θ_2, the class of θ_1-NMDAs is not more expressive than the class of θ_2-NMDAs*

Proof. Assume that every θ_1-NMDA has an equivalent θ_2-NMDA. According to Corollary 30, θ_1 is similar to θ_2. Hence, according to Theorem 29, and since the equivalence relation is symmetric, we conclude that the class of θ_2-NMDAs is equivalent to the class of θ_1-NMDAs. □

Observe that we state Corollary 30 with respect to a non-singleton alphabet, as in the setting of infinite words, there exists only a single word over a singleton alphabet. When restricting attention to finite words, the result is also relevant to a singleton alphabet, and it indeed holds, as a direct corollary of Theorems 13 and 24.

Corollary 32. *For every non-similar choice functions θ_1 and θ_2, there exists a θ_1-NMDA $\mathcal{A}$ such that no θ_2-NMDA is equivalent to $\mathcal{A}$ w.r.t. finite words.*

Corollaries 31 and 32 and the results shown in Section 4.5.2 complete our analysis, showing that every class of θ-NMDAs is important by itself and contributes to the expressiveness of the family of tidy-NMDAs.

5 Tidy NMDAs – Decision Problems

We show that all of the decision problems of tidy NMDAs are in the same complexity classes as the corresponding problems for discounted-sum automata with a single discount factor. That is, the nonemptiness problem is in PTIME, and the exact-value, universality, equivalence, and containment problems are PSPACE-complete (see Table 2). In the equivalence and containment problems, we consider θ-NMDAs with the same choice function θ. When integral DMDAs are considered, we show that all of the problems are in PTIME. In addition, the problem of checking whether a given NMDA is tidy, as well as whether it is a θ-NMDA, for a given choice function θ, is decidable in PTIME. The complexities are w.r.t. the automata size (as defined in Section 2), and when considering a threshold ν, w.r.t. its binary representation.

5.1 Tidiness

Given an NMDA $\mathcal{A}$, one can check in PTIME whether $\mathcal{A}$ is tidy. The algorithm follows by solving a reachability problem in a Cartesian product of $\mathcal{A}$ with itself, to verify that for every word, the last discount factors are identical in all runs.

Theorem 33. *Checking if a given NMDA $\mathcal{A}$ is tidy is decidable in time $O(|\mathcal{A}|^2)$.*

Proof. Consider an input NMDA $\mathcal{A} = \langle \Sigma, Q, \iota, \delta, \gamma, \rho \rangle$. Observe that $\mathcal{A}$ is tidy iff there does not exist a finite word $u \in \Sigma^+$ of length $n = |u|$ and runs r_1 and r_2 of $\mathcal{A}$ on u, such that $\rho(r_1(n-1)) \neq \rho(r_2(n-1))$. Intuitively, we construct the Cartesian product of $\mathcal{A}$ with itself, associating the weight of every transition in the product to the difference of the two discount factors of the transitions causing it. The problem then reduces to reachabilty in this product automaton of a transition with weight different from 0.

Formally, construct a weighted automaton $P = \langle \Sigma, Q \times Q, \iota \times \iota, \delta', \gamma' \rangle$ such that

- $\delta' = \Big\{ \big((s_0, s_1), \sigma, (t_0, t_1)\big) \bigm| \sigma \in \Sigma \text{ and } (s_0, \sigma, t_0), (s_1, \sigma, t_1) \in \delta \Big\}$.
- $\gamma'\big((s_0, s_1), \sigma, (t_0, t_1)\big) = \rho(s_0, \sigma, t_0) - \rho(s_1, \sigma, t_1)$.

Every run in P for a finite word u corresponds to two runs in $\mathcal{A}$ for the same word u. A non-zero weighted transition in P corresponds to two transitions in $\mathcal{A}$ for the same letter, but with different discount factors. Hence, $\mathcal{A}$ is tidy if and only if no run in P takes a non-zero weighted transition.

The graph underlying P can be constructed in time quadratic in the size of $\mathcal{A}$, and the reachability check on it can be performed in time linear in the size of this graph. □

Given also a transducer $\mathcal{T}$, one can check in polynomial time whether $\mathcal{A}$ is a $\mathcal{T}$-NMDA.

Theorem 34. *Checking if a given NMDA $\mathcal{A}$ is a $\mathcal{T}$-NMDA, for a given transducer $\mathcal{T}$, is decidable in time $O(|\mathcal{A}| \cdot |\mathcal{T}|)$.*

Proof. We show the procedure. Let $\mathcal{A} = \langle \Sigma, Q_\mathcal{A}, \iota, \delta_\mathcal{A}, \gamma, \rho_\mathcal{A} \rangle$ be the input NMDA and $\mathcal{T} = \langle Q_\mathcal{T}, \Sigma, q_0, \delta_\mathcal{T}, \rho_\mathcal{T} \rangle$ the input transducer.

We construct a nondeterministic weighted automaton $\mathcal{A}'$ that resembles $\mathcal{A}$ and a deterministic weighted automaton $\mathcal{T}'$ that resembles $\mathcal{T}$, as follows. $\mathcal{A}' = \langle \Sigma, Q_\mathcal{A}, \iota, \delta_\mathcal{A}, \rho_\mathcal{A} \rangle$ is derived from $\mathcal{A}$ by taking the same basic structure of states, initial states and transition function, and having the discount factors of $\mathcal{A}$ as its weight function. $\mathcal{T}' = \langle \Sigma, Q_\mathcal{T}, q_0, \delta_\mathcal{T}, \rho_\mathcal{T} \rangle$ is derived from $\mathcal{T}$, by having the same structure as $\mathcal{T}$ and having the output function of $\mathcal{T}$ as the weight function of $\mathcal{T}'$.

Then, we construct the product automaton $\mathcal{B} = \mathcal{A}' \times \mathcal{T}'$, in which the weight on each transition is the weight of the corresponding transition in $\mathcal{A}'$ minus the weight of the corresponding transition in $\mathcal{T}'$.

	Finite words	Infinite words
Non-emptiness ($<$)	PTIME (Theorem 36)	PTIME (Theorem 35)
Non-emptiness ($\leq$)	PTIME (Theorem 37)	
Containment ($>$)	PSPACE-complete (Theorem 41)	PSPACE (Theorem 43)
Containment ($\geq$)		PSPACE-complete (Theorem 42)
Equivalence	PSPACE-complete (Corollary 44)	
Universality ($<$)	PSPACE-complete (Theorem 45)	PSPACE(Theorem 45)
Universality ($\leq$)		PSPACE-complete (Theorem 45)
Exact-value	PSPACE-complete (Theorem 46)	PSPACE (Theorem 46)

Table 2: The complexities of the decision problems of tidy NMDAs.

It is only left to check whether or not all the weights on the reachable transitions of $\mathcal{B}$ are zero. Indeed, $\mathcal{A}$ is a $\mathcal{T}$-NMDA iff all its reachable discount factors, which are the weights in $\mathcal{A}'$, correspond to the outputs of $\mathcal{T}$, which are the weights in $\mathcal{T}'$. □

5.2 Nonemptiness, Exact-Value, Universality, Equivalence, and Containment of tidy NMDAs

We start with the non-emptiness problems. For both strict and non-strict inequalities with respect to infinite words, there is a simple reduction to one-player discounted-payoff games that also applies to arbitrary NMDAs (which are not necessarily tidy, or even integral), showing that those problems are in PTIME. This result can also be generalized to the strict non-emptiness problem of arbitrary NMDAs w.r.t. finite words. The non-strict problem w.r.t. finite words is solved differently, and applies to integral NMDAs (which are not necessarily tidy).

Theorem 35. *The nonemptiness problem of NMDAs w.r.t. infinite words is in PTIME for both strict and non-strict inequalities.*

Proof. Let $\mathcal{A} = \langle \Sigma, Q, \iota, \delta, \gamma, \rho \rangle$ be an NMDA and $\nu \in \mathbb{Q}$ a threshold. Discounted-payoff games with multiple discount factors (DPGs) were defined in [2]. We will construct a one-player DPG $G = \langle V_{MAX}, V_{MIN}, E, \gamma_G, \rho_G \rangle$ such that every infinite walk ψ of $\mathcal{A}$ will have a corresponding infinite play π of G, such that $\mathcal{A}(\psi) = \mu(\pi)$, where $\mu(\pi)$ is the value of G on the play π as defined in [2]. Observe that our definition of the value of a walk is identical to the definition of μ in [2]. Hence we would like G to have the same states, transitions, weights and discount factors as $\mathcal{A}$, while omitting the letters on the transitions.

Formally, the sets of vertices belonging to the players are $V_{MIN} = Q$ and $V_{MAX} = \emptyset$. For every transition $t = (q, \sigma, p) \in \delta$ we add a corresponding edge (q, p) to E with weight and discount factor of $\gamma_G(q, p) = \gamma(t)$ and $\rho_G(q, p) = \rho(t)$.

Observe that $\mathcal{A}$ might have two transitions with the same source and destination but with different weight and/or discount factor for different letters, however according to [2], DPGs are allowed to have multiple edges between the same ordered pair of vertices. Let f be the function that matches a transition in $\mathcal{A}$ to the corresponding edge in G.

We can extend f to be a bijection between the set of walks of $\mathcal{A}$ and the set of plays of G. Observe that by the construction, for every walk ψ, we have $\mathcal{A}(\psi) = \mu\Big(f(\psi)\Big)$, and for every play π, we have $\mu(\pi) = \mathcal{A}\Big(f^{-1}(\pi)\Big)$. Recall that the value of a word is the minimal value of a run of $\mathcal{A}$ on it, to conclude that the minimum value of an infinite word, equals the minimum value of a play in G starting from a vertex that corresponds to an initial state.

The problem of solving G, i.e., for each vertex $v \in V_{MIN}$ finding the minimum value of any play starting from v, can be represented as a linear program, as suggested by [2]. With the feasible solutions for this problem, all left to do is to iterate all the vertices that correspond to an initial state in ι, to check if the minimum value of a play from any of them is lower (or lower or equal for the non-strict case) than ν. If such a play π exists, then $f^{-1}(\pi)$ is an infinite walk starting from an initial state whose value is lower (or equal) than ν, hence $\mathcal{A}$ is not empty w.r.t. infinite words. Otherwise, there is no infinite run with value lower (or equal) than ν, meaning that $\mathcal{A}$ is not empty w.r.t. infinite words. $\square$

For nonemptiness with respect to finite words, we cannot directly use the aforementioned game solution, as it relies on the convergence of the values in the limit. However, for the nonemptiness with respect to strict inequality, we can reduce the finite-words case to the infinite-words case: If there exists an infinite word w such that $\mathcal{A}(w)$ is strictly smaller than the threshold, the distance between them cannot be compensated in the infinity, implying the existence of a finite prefix that also has a value smaller than the threshold; As for the other direction, we add to every state a 0-weight self loop, causing a small-valued finite word to also imply a small-valued infinite word.

Theorem 36. *The nonemptiness problem of NMDAs w.r.t. finite words and strict inequality is in PTIME.*

Proof. Let $\mathcal{A} = \langle \Sigma, Q, \iota, \delta, \gamma, \rho \rangle$ be an NMDA and $\nu \in \mathbb{Q}$ a threshold. We will construct in polynomial time an NMDA $\mathcal{A}' = \langle \Sigma, Q \cup \iota \times \{1\} \cup \{q_\infty\}, \iota \times \{1\}, \delta \cup \delta' \cup \delta'', \gamma \cup \gamma' \cup \gamma'', \rho \cup \rho' \cup \rho'' \rangle$, such that $\mathcal{A}'$ is empty($<$) with respect to infinite words if and only if $\mathcal{A}$ is empty($<$) with respect to finite words, getting from Theorem 35 the required result.

The construction duplicates all the initial states of $\mathcal{A}$ and adds a new state q_∞. The new transitions are:

- $\delta' = \Big\{\big((q,1), \sigma, q'\big) \;\Big|\; q \in \iota, \sigma \in \Sigma, (q, \sigma, q') \in \delta\Big\}$;
  $\gamma' : \delta' \to \mathbb{Q}$ such that $\gamma'\big((q,1), \sigma, q'\big) = \gamma(q, \sigma, q')$;
  $\rho' : \delta' \to \mathbb{N} \setminus \{0, 1\}$ such that $\rho'\big((q,1), \sigma, q'\big) = \rho(q, \sigma, q')$.

- $\delta'' = \{(q, \tau, q_\infty) \mid q \in Q\} \cup \{(q_\infty, \sigma, q_\infty) \mid \sigma \in \Sigma\}$ for some letter $\tau \in \Sigma$;
  $\gamma'' : \delta'' \to \mathbb{Q}$ such that $\gamma'' \equiv 0$;
  $\rho'' : \delta'' \to \mathbb{N} \setminus \{0, 1\}$ for any arbitrary discount factors.

Observe that for every finite word $u \in \Sigma^+$ we have that $\mathcal{A}'(u \cdot \tau^\omega) \leq \mathcal{A}(u)$, since for every run of A on u there is an equivalent run of A' on u that has the same value.

If $\mathcal{A}$ is not empty($<$) w.r.t. finite words, there exists $u \in \Sigma^+$ such that $\mathcal{A}(u) < \nu$. Hence $\mathcal{A}'(u \cdot \tau^\omega) \leq \mathcal{A}(u) < \nu$. Concluding that $\mathcal{A}'$ is not empty($<$) w.r.t. infinite words.

For the other direction, if $\mathcal{A}'$ is not empty($<$) w.r.t. infinite words, there exists $w \in \Sigma^\omega$ such that $\mathcal{A}'(w) < \nu$. Let r be the run of $\mathcal{A}'$ on w that entails the minimum value. Assume r contains some transitions from δ''. Let r' be the maximal prefix run of r that contains only transitions form δ and δ'. Since all the transitions in δ'' are targeted in q_∞ and have a weight of 0, we get that $\mathcal{A}'(r') = \mathcal{A}'(r) < 0$. By changing the first transition of r' from $\big((q,1), \sigma, q'\big)$ to (q, σ, q') we get a run of A on a finite prefix of w with the same value of $\mathcal{A}'$ on r, which is a value strictly less than ν. Meaning that there exists $v \in \Sigma^+$ such that $\mathcal{A}(v) < \nu$, which is our claim. Otherwise, r contains only transitions from δ and δ'. changing its first transition $\big((q,1), \sigma, q'\big)$ to (q, σ, q') results in a run of A on w with the same value strictly less than ν.

We will now show that if the value of $\mathcal{A}$ on some infinite word w is less than ν then there exists a prefix of w for which the value of $\mathcal{A}$ is also less than ν. Denote $\epsilon = \nu - \mathcal{A}(w)$. Let W be the maximal absolute value of $\mathcal{A}$ on any infinite word, and λ the minimal discount factor in $\mathcal{A}$.

Observe that there exists $n_\epsilon \in \mathbb{N}$ such that $\frac{W}{\lambda^{n_\epsilon}} < \epsilon$ and consider the run $r_{n_\epsilon} = r[0..n_\epsilon - 1]$ of $\mathcal{A}$ on the finite word $u = w[0..n_\epsilon - 1]$. We will show that after reaching $\delta(r_{n_\epsilon})$, if $\mathcal{A}(r_{n_\epsilon})$ is not smaller than ν, then the weight of the suffix $\mathcal{A}(r[n_\epsilon..\infty])$ reduced by the accumulated discount factor $\rho(r_{n_\epsilon})$ will be too small to compensate, resulting in $\mathcal{A}(r) \geq \nu$.

Observe that $|\mathcal{A}^{\delta(u)}(w[n_\epsilon..\infty])| \leq W < \epsilon \cdot \lambda^{n_\epsilon}$ and $\rho(r_{n_\epsilon}) \geq \lambda^{n_\epsilon}$, resulting in $\frac{1}{\rho(r_{n_\epsilon})} \leq \frac{1}{\lambda^{n_\epsilon}}$ and $\frac{|\mathcal{A}^{\delta(u)}(w[n_\epsilon..\infty])|}{\rho(r_{n_\epsilon})} < \epsilon$.

And finally,

$$\begin{aligned}\nu - \epsilon = \mathcal{A}(w) = \mathcal{A}(r) = \mathcal{A}(r_n) + \frac{\mathcal{A}^{\delta(u)}\big(w[n_\epsilon..\infty]\big)}{\rho(r_n)} \\ \geq \mathcal{A}(r_n) - \frac{\big|\mathcal{A}^{\delta(u)}(w[n_\epsilon..\infty])\big|}{\rho(r_{n_\epsilon})} > \mathcal{A}(r_n) - \epsilon \geq \mathcal{A}(u) - \epsilon\end{aligned}$$

Meaning that $\nu > \mathcal{A}(u)$ and $\mathcal{A}$ is not empty($<$) with respect to finite words. □

For nonemptiness with respect to finite words and non-strict inequality, we cannot use the construction used in Theorem 36, since its final part is inadequate: It is possible to have an infinite word with value that equals the threshold, while every finite prefix of it has a value strictly bigger than the threshold. We thus use a different approach. Instead of using linear programming to calculate the minimal value of an infinite run *starting* in every state (as in [2]), we use linear programming to calculate the minimal value of a finite run *ending* in every state.

Theorem 37. *The nonemptiness problem of integral NMDAs w.r.t. finite words and non-strict inequality is in PTIME.*

Proof. Consider an integral NMDA $\mathcal{A} = \langle \Sigma, Q, \iota, \delta, \gamma, \rho \rangle$ and a threshold ν. For every finite run r of $\mathcal{A}$, we define its normalized difference from ν as the accumulated discount factor multiplied by the difference, meaning $\Delta(r) = \rho(r)\big(\mathcal{A}(r) - \nu\big)$. For every state $q \in Q$, we define its minimal normalized difference from ν as the minimal normalized difference among all finite runs that end in q, meaning, $\Delta(q) = \inf\{\Delta(r) \mid \delta(r) = q\} = \inf(D_q)$.

$\mathcal{A}$ is not empty w.r.t. finite words and non-strict inequality iff there exists a run r such that $\Delta(r) \leq 0$. We will show that for every state $q \in Q$ such that $\Delta(q) \leq 0$, there exists a finite run r of $\mathcal{A}$ ending in q such that $\Delta(r) \leq 0$, and combine it with the trivial opposite direction to conclude that $\mathcal{A}$ is not empty iff there exists $q \in Q$ such that $\Delta(q) \leq 0$. Consider a state $q \in Q$,

- If $\Delta(q) = -\infty$, then by the definition of $\Delta(q)$, for every $x < 0$ there exists a run r ending in q such that $\Delta(r) < x$.

- If $\Delta(q) = x \in \mathbb{Q}$, then for every $\epsilon > 0$ there exists a run r_ϵ ending in q such that $\epsilon > \Delta(r_\epsilon) - x \geq 0$. Since we are dealing with integral discount factors, every normalized difference of a run is of the form $\frac{k}{d}$, where $k \in \mathbb{N}$ and d is the common denominator of the weights in γ and ν. We will show that the infimum of the set D_q is its minimum, since every element of D_q can have only discrete values.

  Let $k_x \in \mathbb{N}$ be the minimal integer such that $\frac{k_x}{d} \geq x$, meaning $k_x = \lceil x \cdot d \rceil$, and observe that for every run r ending in q we have $\Delta(r) \geq \frac{k_x}{d}$, leading to $\Delta(r) - x \geq \frac{k_x}{d} - x$. Since this different needs to be arbitrary small, we get that $\frac{k_x}{d} - x = 0$. For every run r ending in q we have that $\Delta(r) - x$ is 0 or at least $\frac{1}{d}$. And since this difference needs to be arbitrary small, it must be 0 for some of those runs. Hence, there exists a run r ending in q such that $\Delta(r) = x$.

We will now show a linear program that calculates the value of $\Delta(q)$ for every $q \in Q$, or determines that there exists some $q \in Q$ such that $\Delta(q) < 0$. For simplicity, we assume that all the states in $\mathcal{A}$ are reachable (since otherwise, one can create in polynomial time an equivalent integral NMDA for which all states are reachable). Let Q_{in} be the set of all states that have an incoming transition, and n its size, meaning $Q_{in} = \{q \in Q \mid \exists (p, \sigma, q) \in \delta\} = \{q_1, \cdots, q_n\}$. Our linear program is over the variables $x_1, x_2, \cdots, x_n$, such that if there exists a feasible solution to the program, meaning a solution that satisfies all the constraints, then $\langle \Delta(q_1), \Delta(q_2), \ldots, \Delta(q_n) \rangle$ is its maximal solution, and otherwise there exists a state q such that $\Delta(q) < 0$. For the first case, after finding the minimal normalized difference from ν for every state in Q_{in}, we can check if any of them equals to 0, and for the other case we can immediately conclude that $\mathcal{A}$ is not empty.

For defining the linear program, we first make the following observations. For every $t = (q_i, \sigma, q_j) \in \delta$ s.t. $q_i \in \iota$, we have $\Delta(t) = \rho(t) \cdot \big(\gamma(t) - \nu\big)$, and for every run r of length $|r| = m > 1$ we have

$$\begin{aligned}\Delta(r) &= \rho(r) \cdot \big(\mathcal{A}(r) - \nu\big)\\ &= \rho\big(r[0..m-2]\big)\rho\big(r(m-1)\big) \cdot \left(\mathcal{A}\big(r[0..m-2]\big) + \frac{\gamma\big(r(m-1)\big)}{\rho\big(r[0..m-2]\big)} - \nu\right)\\ &= \rho\big(r(m-1)\big) \cdot \Big(\Delta\big(r[0..m-2]\big) + \gamma\big(r(m-1)\big)\Big)\end{aligned}$$

Hence, $\langle x_1, x_2, \ldots, x_n\rangle = \langle \Delta(q_1), \Delta(q_2), \ldots, \Delta(q_n)\rangle$ must satisfy the following system of equations:

1. $x_j \leq \rho(t) \cdot \big(\gamma(t) - \nu\big)$ for every $t = (q_i, \sigma, q_j) \in \delta$ s.t. $q_i \in \iota$.
2. $x_j \leq \rho(t) \cdot \big(\gamma(t) + x_i\big)$ for every $t = (q_i, \sigma, q_j) \in \delta$ s.t. $q_i \in Q_{in}$.

These equations have a single maximal solution $\langle x_1^*, \cdots, x_n^*\rangle$ such that for any solution $\langle a_1, \cdots, a_n\rangle$ and $1 \leq i \leq n$, we have $x_i^* \geq a_i$. To see that $\langle \Delta(q_1), \ldots, \Delta(q_n)\rangle$ is indeed the unique maximal solution, if such exists, consider a solution $\langle a_1, \cdots, a_n\rangle$, a state $q_i \in Q_{in}$ and a run r such that $\delta(r) = q_i$ and $\Delta(r) = \Delta(q_i)$. For every $0 \leq j < |r|$, let q_{i_j} be the target state after the j-sized prefix of r, meaning $q_{i_j} = \delta\big(r[0..j]\big)$. We will show by induction on j that $a_{i_j} \leq \Delta(r[0..j])$ to conclude that $a_i = a_{i_{|r|-1}} \leq \Delta(r[0..|r|-1]) = \Delta(r) = \Delta(q_i)$:

- For the base case, we have $a_{i_0} \leq \rho\big(r(0)\big)\big(\gamma(r(0)) - \nu\big) = \Delta\big(r(0)\big)$.
- For the induction step,

$$\begin{aligned}a_{i_j} &\leq \rho\big(r(j)\big) \cdot \Big(\gamma\big(r(j)\big) + a_{i_{j-1}}\Big)\\ &\leq \rho\big(r(j)\big) \cdot \Big(\gamma\big(r(j)\big) + \Delta\big(r[0..j-1]\big)\Big) = \Delta\big(r[0..j]\big)\end{aligned}$$

The implicit constraint of non-negative values for the variables of the linear program, meaning $x_i \geq 0$ for every $1 \leq i \leq n$, handles the case of a possible divergence to $-\infty$. With these constraints, if there exists $q \in Q$ such that $\Delta(q) < 0$, then the linear program has no feasible solution, and this case will be detected by the algorithm that solves the linear program.

Meaning that the problem can be stated as the linear program: maximize $\sum_{i=0}^{n} x_i$ subject to Items 1 and 2 and $x_i \geq 0$ for every $1 \leq i \leq n$.

□

We continue with the PSPACE-complete problems, to which we first provide hardness proofs, by reductions from the universality problem of NFAs, known

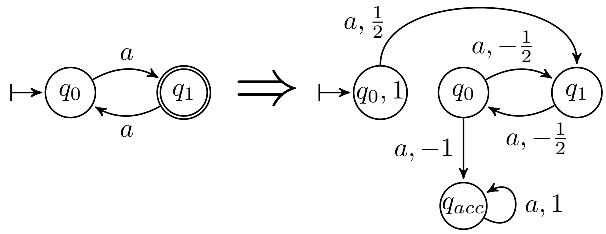

Figure 21: An example of the reduction defined in the proof of Lemma 38.

to be PSPACE-complete [32]. Notice that the provided hardness results already stand for integral NDAs, not only to tidy NMDAs.

PSPACE-hardness of the containment problem for NDAs with respect to infinite words and non-strict inequalities is shown in [4]. We provide below more general hardness results, considering the equivalence problem, first with respect to finite words and then with respect to infinite words, as well as the exact-value, universality($\leq$) and universality($<$) problems with respect to finite words.

Lemma 38. *The equivalence and universality($\leq$) problems of integral NDAs w.r.t. finite words are PSPACE-hard.*

Proof. Given an NFA $A = \langle \Sigma, Q, Q_0, \Delta, F \rangle$, we construct in polynomial time an NDA $\mathcal{B}$ with discount factor 2, such that $\mathcal{B}$ never gets a negative value, and $\mathcal{A}$ is universal if and only if $\mathcal{B}$ is equivalent to a 0 NDA, namely to an NDA that gets a value of 0 on all finite words. For simplicity, we ignore the empty word and words of length 1, whose acceptance is easy to check in $\mathcal{A}$.

Intuitively, $\mathcal{B}$ will have the same structure as $\mathcal{A}$, and the assigned weights on the transitions will guarantee that the value of $\mathcal{B}$ on every word u is $\frac{1}{2^{|u|}}$. In addition, we have in $\mathcal{B}$ a new "good" state q_{acc}, and for every original transition t to an accepting state $q \in F$, we add in $\mathcal{B}$ a new "good" transition t' to q_{acc}, such that the weight on t' allows $\mathcal{B}$ to have a value of 0 on a word u on which there is a run on u ending in q. Finally, we add a "bad" transition out of q_{acc}, such that its weight ensures a total positive value, in the case that $\mathcal{B}$ continues the run out of q_{acc}. (Example in Fig. 21.)

Formally, we construct a 2-NDA (with discount factor 2) $\mathcal{B} = \langle \Sigma, Q \cup Q_0 \times \{1\} \cup \{q_{acc}\}, Q_0 \times \{1\}, \Delta \cup \delta_{\mathcal{B}}, \gamma_{\mathcal{B}} \rangle$, where

- $\delta_{\mathcal{B}} = \big\{ \big((q,1), \sigma, q'\big)\} \mid (q, \sigma, q') \in \Delta \big\} \cup$
  $\big\{ (q, \sigma, q_{acc}) \mid \text{ exist } q' \in F \text{ and } (q, \sigma, q') \in \Delta \big\} \cup$
  $\big\{ (q_{acc}, \sigma, q_{acc}) \mid \sigma \in \Sigma \big\}$.

- $\gamma_{\mathcal{B}}$:
  - For every $t = \big((q,1), \sigma, q'\big) \in \delta_{\mathcal{B}}$, we have $\gamma_{\mathcal{B}}(t) = \frac{1}{2}$.
  - For every $t \in \Delta$, we have $\gamma_{\mathcal{B}}(t) = -\frac{1}{2}$.

– For every $t = (q, \sigma, q_{acc}) \in \delta_{\mathcal{B}}$, we have $\gamma_{\mathcal{B}}(t) = -1$.

– $\gamma_{\mathcal{B}}\big((q_{acc}, \sigma, q_{acc})\big) = 1$.

Observe that by the construction of $\mathcal{B}$, for every word w, $\mathcal{B}(w) \geq 0$. Hence, $\mathcal{B}$ is equivalent to a 0 NDA iff it is universal($\leq$) with respect to the threshold 0. Meaning that the same reduction shows the $PSPACE$-hardness of the universality($\leq$) with respect to infinite words. □

Lemma 39. *The equivalence and universality($\leq$) problems of integral NDAs w.r.t. infinite words are PSPACE-hard.*

Proof. Similarly to the proof of Lemma 38, we construct in polynomial time an NDA $\mathcal{B}$ with discount factor 2, such that the input NFA is universal if and only if $\mathcal{B}$ is equivalent to a 0 NDA with respect to infinite words. Also in this reduction, no negative values of words will be possible, so it is also valid for showing the PSPACE-hardness of the universality($\leq$) problem. The reduction is similar to the one provided in the proof of Lemma 38, with intuitively the following adaptations of the constructed NDA $\mathcal{B}$ to the case of infinite words: We add a new letter # to the alphabet, low-weighted #-transitions from the accepting states, and high-weighted #-transitions from the non-accepting states.

By this construction, the value of $\mathcal{B}$ on an infinite word $u \cdot \# \cdot w$, where u does not contain #, will be 0 if and only if $\mathcal{A}$ accepts u.

Notice that the value of $\mathcal{B}$ on an infinite word that does not contain # is also 0, as it is $\lim_{n\to\infty} \frac{1}{2^n}$.

Formally, given NFA $\mathcal{A} = \langle Q, \Sigma, \Delta, Q_0, F\rangle$, we construct a 2-NDA $\mathcal{B} = \langle \Sigma \cup \{\#\}, Q \cup Q_0 \times \{1\} \cup \{q_\infty\}, Q_0 \times \{1\}, \Delta \cup \delta_{\mathcal{B}}, \gamma_{\mathcal{B}}\rangle$ where

- $\# \notin \Sigma$ is a new letter.
- $\delta_{\mathcal{B}} = \big\{\big((q,1), \sigma, q')\big\} \mid (q, \sigma, q') \in \Delta\big\} \cup$
  $\big\{(q, \#, q_\infty) \mid q \in Q\big\} \cup$
  $\big\{\big((q,1), \#, q_\infty\big) \mid q \in Q\big\} \cup$
  $\big\{(q_\infty, \tau, q_\infty) \mid \tau \in \Sigma \cup \{\#\}\big\}$.
- $\gamma_{\mathcal{B}}$:
  – For every $t = \big((q,1), \sigma, q'\big) \in \delta_{\mathcal{B}}$, we have $\gamma_{\mathcal{B}}(t) = \frac{1}{2}$.
  – For every $t \in \Delta$, we have $\gamma_{\mathcal{B}}(t) = -\frac{1}{2}$.
  – For every $t_1 = (q, \#, q_\infty) \in \delta_{\mathcal{B}}$ or $t_2 = \big((q,1), \#, q_\infty\big) \in \delta_{\mathcal{B}}$, such that $q \in F$, we have $\gamma_{\mathcal{B}}(t_1) = -1$ and $\gamma_{\mathcal{B}}(t_2) = 0$. Those transitions assure that for every $u \in \Sigma^*$ that $\mathcal{A}$ accepts, there exists a run of $\mathcal{B}$ on $u\#$, ending in q_∞ with a value of 0.
  – For every $t_1 = (q, \#, q_\infty) \in \delta_{\mathcal{B}}$ or $t_2 = \big((q,1), \#, q_\infty\big) \in \delta_{\mathcal{B}}$, such that $q \in Q \setminus F$, we have $\gamma_{\mathcal{B}}(t_1) = 0$ and $\gamma_{\mathcal{B}}(t_2) = 1$.
  – $\gamma_{\mathcal{B}}\big((q_\infty, \tau, q_\infty)\big) = 0$.

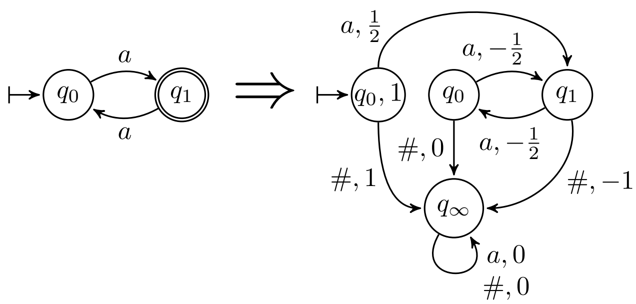

Figure 22: An example of the reduction defined in the proof of Lemma 39.

An example of the construction is given in Fig. 22.

Observe that for every infinite word $w \in \Sigma^\omega$, we have $\mathcal{B}(w) = \lim_{n\to\infty} \frac{1}{2^n} = 0$. In addition, for every finite word $u \in \Sigma^*$ and infinite word $w \in (\Sigma \cup \{\#\})^\omega$, we have $\mathcal{B}(u \cdot \# \cdot w) = 0 \Leftrightarrow$ there exists a run of $\mathcal{B}$ on $u \cdot \#$ with a final transition $(p, \#, q_\infty)$ or $\big((p, 1), \#, q_\infty\big)$ such that $p \in F \Leftrightarrow$ there exist $p \in F$ and a run of $\mathcal{A}$ on u with p as the final state $\Leftrightarrow u \in L(\mathcal{A})$. Hence $\mathcal{A}$ is universal iff $\mathcal{B} \equiv 0$.

Also, for every finite word $u \in \Sigma^*$ and infinite word $w \in (\Sigma \cup \{\#\})^\omega$, we have $\mathcal{B}(u \cdot \# \cdot w) \leq 0 \Leftrightarrow u \in L(\mathcal{A})$. Hence $\mathcal{A}$ is universal iff $\mathcal{B}$ is universal with respect to the threshold 0, non-strict inequality and infinite words. □

Lemma 40. *The universality(<) and exact-value problems of integral NDAs w.r.t. finite words are PSPACE-hard.*

Proof. Similarly to the proof of Lemma 38, we show a polynomial reduction from the problem of NFA universality to the problems of NDA universality and exact-value. The reduction is similar to the one provided in the proof of Lemma 38, yet changing the transition weights in the constructed NDA $\mathcal{B}$, such that for every finite word u, we have $\mathcal{B}(u) < 0$ if and only if $\mathcal{A}$ accepts u, and $\mathcal{B}(u) = 0$ otherwise. This provides reductions to both the universality(<) and exact-value problems.

Formally, given an NFA $\mathcal{A} = \langle Q, \Sigma, \Delta, Q_0, F\rangle$, we construct a 2-NDA $\mathcal{B} = \langle \Sigma, Q \cup \{q_{acc}, q_\infty\}, Q_0, \Delta \cup \delta_\mathcal{B}, \gamma_\mathcal{B}\rangle$ where:

- $\delta_\mathcal{B} = \big\{(q, \sigma, q_{acc}) \mid \text{ exist } q' \in F \text{ and } (q, \sigma, q') \in \Delta\big\} \cup$
  $\big\{(q_{acc}, \sigma, q_\infty) \mid \sigma \in \Sigma\big\} \cup$
  $\big\{(q_\infty, \sigma, q_\infty) \mid \sigma \in \Sigma\big\}$.
- $\gamma_\mathcal{B}$:
  - For every $t \in \Delta$, we have $\gamma_\mathcal{B}(t) = 0$.
  - For every $t = (q, \sigma, q_{acc}) \in \delta_\mathcal{B}$, we have $\gamma_\mathcal{B}(t) = -1$. These transitions ensure that if a word w is accepted in $\mathcal{A}$, then there exists a run of $\mathcal{B}$ on w with a negative value.

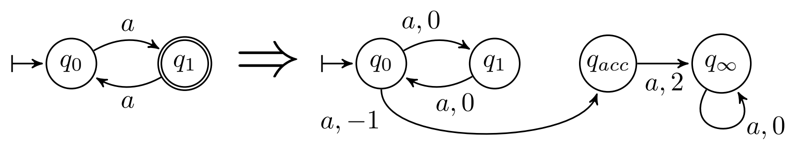

Figure 23: An example of the reduction defined in the proof of Lemma 40.

– For every $\sigma \in \Sigma$, we have $\gamma_{\mathcal{B}}\big((q_{acc}, \sigma, q_\infty)\big) = 2$.
  These transitions ensure that only runs that "exit" the original structure of $\mathcal{A}$ in the finial transition, will result in a negative value. The weight of 2, reduced by the fixed discount factor of 2, exactly compensates on the negative weight that was added in the transition that entered q_{acc}.
– For every $\sigma \in \Sigma$, we have $\gamma_{\mathcal{B}}\big((q_\infty, \sigma, q_\infty)\big) = 0$.
  These transitions ensure that a run entering q_∞ will maintain the exact same value for every suffix walk added to it.

An example of the construction is given in Fig. 23.

Observe that the only negative weights in $\mathcal{B}$ are on the transitions entering q_{acc}, and only a single one of them can be part of every run. All the runs not entering q_{acc} have a value of 0, and all the runs passing in q_{acc} in a transition that is not the final one will also have a value of 0.

For every finite word $w \in \Sigma^+$, we have that $w \in L(\mathcal{A}) \Leftrightarrow$ there exist $q \in Q$, $p \in F$ and a run r of $\mathcal{A}$ on w with a final transition $(q, \sigma, p) \Leftrightarrow$ there exist $q \in Q$ and a run r' of $\mathcal{B}$ on w with a final transition $(q, \sigma, q_{acc}) \Leftrightarrow$ there exists a run r' of $\mathcal{B}$ on w such that $\mathcal{B}(r') < 0 \Leftrightarrow \mathcal{B}(w) < 0 \Leftrightarrow \mathcal{B}(w) \neq 0$.

Hence $\mathcal{A}$ is universal iff $\mathcal{B}$ is universal($<$) with respect to finite words and the threshold $\nu = 0$. Also, $\mathcal{A}$ is universal iff there is no finite word w such that $\mathcal{B}(w) = 0$.

Another special case left to handle is the empty word ε, but this can be easily verified before constructing $\mathcal{B}$ by checking if $F \cap Q_0 \neq \emptyset$. □

Notice that the proof of Lemma 40 does not easily extend to the infinite-words setting, as Lemma 38 was adapted into Lemma 39, since the convergence of the values in the constructed NDA $\mathcal{B}$ to 0 interfere with the strict inequality.

We continue with the PSPACE upper bounds. The containment problem of NDAs was proved in [4] to be in PSPACE, using comparators to reduce the problem to language inclusion between Büchi automata. Our approach for the containment problem of NMDAs is different, and it also improves the complexity provided in [4] for NDAs (having a single discount factor), as we refer to binary representation of weights, while [4] assumes unary representation.[3]

[3]Rational weights are assumed to have a common denominator, both by us and by [4], where in the latter it is stated implicitly, by providing the complexity analysis with respect to transition weights that are natural numbers.

Our algorithm for solving the containment problem between θ-NMDAs $\mathcal{A}$ and $\mathcal{B}$ is based on performing the determinization of $\mathcal{B}$ on-the-fly into a DMDA $\mathcal{D}$, as suggested in [4], and simulating on the fly a θ-NMDA for the difference between $\mathcal{A}$ and $\mathcal{D}$. We then non-deterministically guess a run r that witnesses a negative value of the difference automaton, while ensuring that the entire process only uses space polynomial in the size of the input automata. For meeting this space requirement, after each step of the run r, the algorithm maintains a *local data* consisting of the current state of $\mathcal{A}$, the current state of $\mathcal{D}$ and a "normalized difference" between the values of the runs of $\mathcal{A}$ and $\mathcal{D}$ on the word generated so far. When the normalized difference goes below 0, we have that the generated word w is a witnesses for $\mathcal{A}(w) < \mathcal{D}(w)$, when it gets to 0 we have a witness for $\mathcal{A}(w) = \mathcal{D}(w)$, and when it exceeds a certain *maximal recoverable difference*, which is polynomial in $|\mathcal{A}| + |\mathcal{B}|$, no suffix can be added to w for getting a witness.

Theorem 41. *For every choice function θ, the containment problem of θ-NMDAs w.r.t. finite words is PSPACE-complete for both strict and non-strict inequalities.*

Proof. PSPACE hardness directly follows from Lemmas 38 and 40.

We provide a PSPACE upper bound. Consider a choice function θ, and θ-NMDAs $\mathcal{A} = \langle \Sigma, Q_{\mathcal{A}}, \iota, \delta_{\mathcal{A}}, \gamma_{\mathcal{A}}, \rho_{\mathcal{A}} \rangle$ and $\mathcal{B}$. We have that

$$\forall w.\mathcal{A}(w) > \mathcal{B}(w) \Leftrightarrow \nexists w.\mathcal{A}(w) \leq \mathcal{B}(w) \Leftrightarrow \nexists w.\mathcal{A}(w) - \mathcal{B}(w) \leq 0$$

and

$$\forall w.\mathcal{A}(w) \geq \mathcal{B}(w) \Leftrightarrow \nexists w.\mathcal{A}(w) < \mathcal{B}(w) \Leftrightarrow \nexists w.\mathcal{A}(w) - \mathcal{B}(w) < 0$$

We present a nondeterministic algorithm that determines the converse of containment, namely whether there exists a word w such that $\mathcal{A}(w) - \mathcal{B}(w) \leq 0$ for continament($>$) or $\mathcal{A}(w) - \mathcal{B}(w) < 0$ for continament($\geq$), while using polynomial space w.r.t. $|\mathcal{A}|$ and $|\mathcal{B}|$, to conclude that the problems are in co-NPSPACE and hence in PSPACE.

Let $\mathcal{D} = \langle \Sigma, Q_{\mathcal{D}}, \{p_0\}, \delta_{\mathcal{D}}, \gamma_{\mathcal{D}}, \rho_{\mathcal{D}} \rangle$ be a θ-DMDA equivalent to $\mathcal{B}$, as per Theorem 11. Observe that the size of $\mathcal{D}$ can be exponential in the size of $\mathcal{B}$, but we do not save it all, but rather simulate it on the fly, and thus only save a single state of $\mathcal{D}$ at a time. We will later show that indeed the intermediate data we use in each iteration of the algorithm only requires a space polynomial in $|\mathcal{A}|$ and $|\mathcal{B}|$.

Containment($\geq$).

For providing a word $w \in \Sigma^+$, such that $\mathcal{A}(w) - \mathcal{B}(w) < 0$, we nondeterministically generate on the fly a word w, a run r_w of $\mathcal{A}$ on w, and the single run of $\mathcal{D}$ on w, such that $\mathcal{A}(r_w) - \mathcal{B}(w) = \mathcal{A}(r_w) - \mathcal{D}(w) < 0$. Observe that $\mathcal{A}(w) \leq \mathcal{A}(r_w)$, hence the above condition is equivalent to $\mathcal{A}(w) - \mathcal{B}(w) < 0$.

Let $M_{\mathcal{A}}$, $M_{\mathcal{B}}$, and $M_{\mathcal{D}}$ be the maximal absolute weights in $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{D}$, respectively.

We start by guessing an initial state q_{in} of $\mathcal{A}$ and setting a *local data* storage of $\langle q_{in}, p_0, 0 \rangle$. The local data will maintain the current state of $\mathcal{A}$ and $\mathcal{D}$

respectively, and a "normalized difference" between the value of the run in $\mathcal{A}$ generated so far and the value of $\mathcal{D}$ on the word generated so far, as formalized below. The algorithm iteratively guesses, given a local data $\langle q, p, d \rangle$, a letter $\sigma \in \Sigma$ and a transition $t = (q, \sigma, q') \in \delta_{\mathcal{A}}(q, \sigma)$, and calculates the *normalized difference* $d' = \rho_{\mathcal{A}}(t)\big(d + \gamma_{\mathcal{A}}(t) - \gamma_{\mathcal{D}}(p, \sigma)\big)$ between the values $\mathcal{A}(r_w)$ and $\mathcal{B}(w)$, w.r.r. the word w and the run r_w generated so far. If d' is bigger than the *maximal recoverable difference* $2S$, where $S = M_{\mathcal{A}} + 3M_{\mathcal{B}}$, we abort, if $d' < 0$, we have that the generated word w indeed witnesses that $\mathcal{A}(w) < \mathcal{D}(w)$ (the *accept condition* holds), and otherwise we continue and update the local data to $\langle q', \delta(p, \sigma), d' \rangle$. Observe that by the construction in the proof of Theorem 11, for every weight W in $\mathcal{D}$ we have that $|W| \leq 2T + M_{\mathcal{B}} \leq 3M_{\mathcal{B}}$, where T is the maximal difference between the weights in $\mathcal{B}$. Hence $S > M_{\mathcal{A}} + M_{\mathcal{D}}$ is polynomial w.r.t. $|\mathcal{A}|$ and $|\mathcal{B}|$, and can be calculated in polynomial space w.r.t. $|\mathcal{A}|$ and $|\mathcal{B}|$.

We show by induction on the length of the word w that whenever a word w and a run r_w are generated, the value d in the corresponding local data $\langle q, p, d \rangle$ indeed stands for the normalized difference between $\mathcal{A}(r_w)$ and $\mathcal{D}(w)$, namely

$$d = \rho_{\mathcal{A}}(r_w)\big(\mathcal{A}(r_w) - \mathcal{D}(w)\big) \tag{38}$$

For the base case we have a single-letter word $w = \sigma$, and a single-transition run $r_w = t$. Hence, $d' = \rho_{\mathcal{A}}(t)\big(d + \gamma_{\mathcal{A}}(t) - \gamma_{\mathcal{D}}(p, \sigma)\big) = \rho_{\mathcal{A}}(r_w)\big(0 + \mathcal{A}(r_w) - \mathcal{D}(w)\big) = \rho_{\mathcal{A}}(r_w)\big(\mathcal{A}(r_w) - \mathcal{D}(w)\big)$.

For the induction step, consider an iteration whose initial local data is $\langle q, p, d \rangle$, for a generated word w and run r_w, that guessed the next letter σ and transition t, and calculated the next local data $\langle q', p', d' \rangle$. Then we have $d' = \rho_{\mathcal{A}}(t)\big(d + \gamma_{\mathcal{A}}(t) - \gamma_{\mathcal{D}}(p, \sigma)\big)$. By the induction assumption, we get:

$$\begin{aligned}
d' &= \rho_{\mathcal{A}}(t)\Big(\rho_{\mathcal{A}}(r_w)\big(\mathcal{A}(r_w) - \mathcal{D}(w)\big) + \gamma_{\mathcal{A}}(t) - \gamma_{\mathcal{D}}(p, \sigma)\Big) \\
&= \rho_{\mathcal{A}}(r_w)\rho_{\mathcal{A}}(t)\Big(\mathcal{A}(r_w) + \frac{\gamma_{\mathcal{A}}(t)}{\rho_{\mathcal{A}}(r_w)} - \mathcal{D}(w) - \frac{\gamma_{\mathcal{D}}(p, \sigma)}{\rho_{\mathcal{A}}(r_w)}\Big) \\
&= \rho_{\mathcal{A}}(r_w \cdot t)\Big(\mathcal{A}(r_w \cdot t) - \Big(\mathcal{D}(w) + \frac{\gamma_{\mathcal{D}}(p, \sigma)}{\rho_{\mathcal{A}}(r_w)}\Big)\Big),
\end{aligned}$$

and since the discount-factor functions of $\mathcal{A}$ and $\mathcal{D}$ both agree with θ, we have

$$d' = \rho_{\mathcal{A}}(r_w \cdot t)\Big(\mathcal{A}(r_w \cdot t) - \Big(\mathcal{D}(w) + \frac{\gamma_{\mathcal{D}}(p, \sigma)}{\rho_{\mathcal{D}}(w)}\Big)\Big) = \rho_{\mathcal{A}}(r_w \cdot t)\big(\mathcal{A}(r_w \cdot t) - \mathcal{D}(w \cdot \sigma)\big),$$

which provides the required result of the induction claim.

Next, we show that the accept condition holds iff there exist a finite word w and run r_w of $\mathcal{A}$ on w such that $\mathcal{A}(r_w) - \mathcal{D}(w) < 0$. Since for every finite word w we have $\rho_{\mathcal{A}}(w) > 0$, we conclude from Eq. (38) that if $d' < 0$ was reached for a generated word w and a run r_w, we have that $\mathcal{A}(r_w) - \mathcal{D}(w) < 0$. For the other direction, assume toward contradiction that there exist finite word w and run r_w of $\mathcal{A}$ on w such that $\mathcal{A}(r_w) - \mathcal{D}(w) < 0$, but the algorithm aborts after generating some prefixes $w[0..i]$ and $r_w[0..i]$. Meaning that

$\rho_{\mathcal{A}}(r_w[0..i])\big(\mathcal{A}(r_w[0..i])-\mathcal{D}(w[0..i])\big) > 2M_{\mathcal{A}}+2M_{\mathcal{D}}$. Let $W_1 = \mathcal{A}(r_w[i+1..|r_w|-1])$ and $W_2 = \mathcal{D}^{\delta_{\mathcal{D}}(w[0..i])}(w[i+1..|r_w|-1])$. Observe that

$$\begin{aligned} 0 &> \mathcal{A}(r_w) - \mathcal{D}(w) > \rho_{\mathcal{A}}\big(r_w[0..i]\big)\big(\mathcal{A}(r_w) - \mathcal{D}(w)\big) \\ &= \rho_{\mathcal{A}}\big(r_w[0..i]\big)\mathcal{A}\big(r_w[0..i]\big) + W_1 - \big(\rho_{\mathcal{A}}(r_w[0..i])\mathcal{D}(w[0..i]) + W2\big) \\ &> 2M_{\mathcal{A}} + 2M_{\mathcal{D}} + W_1 - W_2 \end{aligned}$$

But since all the discount factors applied by θ are greater or equal to 2, we have that $|W_1| \leq 2M_{\mathcal{A}}$ and $|W_2| \leq 2M_{\mathcal{B}}$, leading to a contradiction.

To see that the algorithm indeed only uses space polynomial in $|\mathcal{A}|$ and $|\mathcal{B}|$, observe that the first element of the data storage is a state of $\mathcal{A}$, only requiring a space logarithmic in $|\mathcal{A}|$, the second element is a state of $\mathcal{D}$, requiring by Theorem 11 a space polynomial in $\mathcal{B}$, and the third element is a non-negative rational number bounded by $2S$, whose denominator is the multiplication of the denominators of the weights in $\mathcal{A}$ and $\mathcal{D}$, and as shown in the proof of Theorem 11, also of the multiplication of the denominators of the weights in $\mathcal{A}$ and $\mathcal{B}$, thus requires a space polynomial in $|\mathcal{A}|$ and $|\mathcal{B}|$. Finally, in order to compute this third element, we calculated a weight of a transition in $\mathcal{D}$, which only requires, by the proof of Theorem 11, a space polynomial in $|\mathcal{B}|$.

Containment($>$).

The algorithm is identical to the one used for the containment($\geq$) problem with changing the accept condition $d' < 0$ to $d' \leq 0$. This condition is met iff there exists a finite word w such that $\mathcal{A}(w) - \mathcal{B}(w) \leq 0$. The proof is identical while modifying "< 0" to "≤ 0" in all of the equations. ☐

The algorithm for determining containment($\geq$) in the infinite-words settings is similar to the one presented for finite words, with the difference that rather than witnessing a finite word w, such that $\mathcal{A}(w) - \mathcal{B}(w) < 0$, we witness a finite prefix u (of an infinite word w), such that the normalized difference between $\mathcal{A}(u)$ and $\mathcal{B}(u)$ (taking into account the accumulated discount factor on u) is bigger than some fixed threshold.

Theorem 42. *For every choice function θ, the containment problem of θ-NMDAs w.r.t. infinite words and non-strict inequality is PSPACE-complete.*

Proof. PSPACE hardness directly follows from Lemma 39.

We provide a PSPACE upper bound. Consider a choice function θ, and θ-NMDAs $\mathcal{A}$ and $\mathcal{B}$. Analogously to the proof of Theorem 41, we present a nondeterministic algorithm that determines whether there exist a word w and a run r_w of $\mathcal{A}$ on w, such that $\mathcal{A}(r_w) - \mathcal{B}(w) < 0$, and thus $\mathcal{A}(w) - \mathcal{B}(w) < 0$. The algorithm uses polynomial space w.r.t. $|\mathcal{A}|$ and $|\mathcal{B}|$, which shows that the problem is in co-NPSPACE and hence in PSPACE.

The algorithm is identical to the one presented in the proof of Theorem 41, with the only difference that the condition for an infinite word w such that $\mathcal{A}(w) - \mathcal{B}(w) < 0$ is that we generated a finite word u and a run r_u of $\mathcal{A}$ on u, that resulted in a local data with normalized difference $d < -2S$. We will use the same notations as in the proof of Theorem 41.

Observe that for any infinite word w and infinite walks (ψ_1, ψ_2) of $(\mathcal{A}, \mathcal{D})$ on w from any state in $(\mathcal{A}, \mathcal{D})$, we have that $2S = 2M_\mathcal{A} + 2M_\mathcal{D} \geq \mathcal{A}(\psi_1) - \mathcal{D}(\psi_2)$.

If $-2S > d = \rho_\mathcal{A}(r_u)(\mathcal{A}(r_u) - \mathcal{D}(u))$ was reached for a generated finite word u, and a run r_u of $\mathcal{A}$ on u, then for any infinite suffix word w and a walk ψ_1 of $\mathcal{A}$ on w starting at $\delta_\mathcal{A}(r_u)$, we have that

$$0 = -2S + 2S > \Big(\rho_\mathcal{A}(r_u)\big(\mathcal{A}(r_u) - \mathcal{D}(u)\big)\Big) + \Big(\mathcal{A}(\psi_1) - \mathcal{D}(\psi_2)\Big)$$

where ψ_2 is the walk of $\mathcal{A}$ on w starting at $\delta_\mathcal{D}(u)$. Hence,

$$\begin{aligned}0 &> \mathcal{A}(r_u) + \frac{\mathcal{A}(\psi_1)}{\rho_\mathcal{A}(r_u)} - \Big(\mathcal{D}(u) + \frac{\mathcal{D}(\psi_2)}{\rho_\mathcal{A}(r_u)}\Big) \geq A(r_u \cdot \psi_1) - \mathcal{D}(u \cdot w)\\ &\geq \mathcal{A}(u \cdot w) - \mathcal{D}(u \cdot w)\end{aligned}$$

For the other direction, assume that there exists an infinite word $w \in \Sigma^\omega$ such that $\mathcal{A}(r_w) - \mathcal{D}(w) = -\epsilon < 0$, where r_w is a run of $\mathcal{A}$ on w that entails the minimum value. By an observation similar to the one presented in the proof of Theorem 41, we conclude that whenever a word prefix $w[0..i]$ and a run $r_w[0..i]$ are generated, the algorithm does not fulfill the abort condition.

It is only left to show that there exist prefixes of w and r_w that result with $d < -2S$. Indeed, we have that there exists $n_1 \in \mathbb{N}$ such that for every $i \geq n_1$ we have $\mathcal{A}(r_w[0..i]) - \mathcal{D}(w[0..i]) < -\frac{\epsilon}{2}$ and there exists $n_2 \in \mathbb{N}$ such that for every $i \geq n_2$ we have $-\frac{\epsilon}{2} < -\frac{2S}{\rho(w[0..i])}$. Hence for $n = \max\{n_1, n_2\}$ we have $\mathcal{A}(r_w[0..n]) - \mathcal{D}(w[0..n]) < -\frac{\epsilon}{2} < -\frac{2S}{\rho(w[0..n])}$, meaning that the algorithm will accept when $w[0..n]$ and $r_w[0..n]$ are generated.

As for the space analysis, the arguments presented in the proof of Theorem 41 also apply to the current algorithm, as the only relevant difference is that the third element in the data storage is now a rational number bounded by $2S$ and $-2S$, thus requiring double the space that was considered in the proof of Theorem 41, and hence remaining polynomial in $|A|$ and $|B|$. ☐

To find a witness for strict non-containment in the infinite-words setting, we adapt the above proof, by adding an accept condition for detecting convergence of the difference between the two automata values to the threshold value, which is the existence of a cycle with the same normalized difference.

Theorem 43. *For every choice function θ, the containment problem of θ-NMDAs w.r.t. infinite words and strict inequality is in PSPACE.*

Proof. We use the same algorithm as in Theorem 42 with adding a new accept condition, which will identify the existence of an infinite word w and a run r_w of $\mathcal{A}$ on w, such that $0 = \mathcal{A}(r_w) - \mathcal{B}(w)$. This new condition is reaching the same couple of states in $\mathcal{A}$ and $\mathcal{D}$ twice with the same value of normalized difference d. Our NPSPACE algorithm can check this condition by guessing states $q_{acc} \in Q_\mathcal{A}$, $p_{acc} \in Q_\mathcal{D}$ and a normalized difference d_{acc}, setting a flag when $\langle q_{acc}, p_{acc}, d_{acc}\rangle$ is reached while the flag was clean, and accepting if it is reached while the flag was set.

If the condition is met after generating some prefix word and a run of $\mathcal{A}$ on that word, we have cycles in both $\mathcal{A}$ and $\mathcal{D}$ for the same suffix word, leading to the same normalized difference. Meaning that there exist finite words u and v, a run r_u of $\mathcal{A}$ on u and a walk ψ_v of $\mathcal{A}$ on v starting at $\delta_{\mathcal{A}}(r_u)$, such that for every $i \in \mathbb{N}$, according to Eq. (38), we have $\frac{d_{acc}}{\rho(u \cdot v^i)} = \mathcal{A}(r_u \cdot \psi_v^i) - \mathcal{D}(u \cdot v^i)$. Hence

$$0 = \lim_{i\to\infty} \frac{d_{acc}}{\rho(u \cdot v^i)} = \lim_{i\to\infty} \mathcal{A}(r_u \cdot \psi_v^i) - \mathcal{D}(u \cdot v^i)$$

resulting in $\mathcal{A}(u \cdot v^\omega) \leq \mathcal{A}(r_u \cdot \psi_v^\omega) = \mathcal{B}(u \cdot v^\omega)$.

For the other direction, we show that if there exist an infinite word w and a run r_w of $\mathcal{A}$ on w such that $\mathcal{B}(w) = \mathcal{A}(r_w)$, then the new accept condition is met for some $\langle q_{acc}, p_{acc}, d_{acc}\rangle$. Consider such w and r_w and observe that similarly to the analysis shown in the proof of Theorem 41, the normalized difference between the value of every prefix of r_w and the value of the same sized prefix of the single run of $\mathcal{D}$ on w, never exceeds the maximal recoverable difference. Hence, for every finite prefix $w[0..i]$ of w, we have that $d_i = \rho(w[0..i])\big(\mathcal{A}(r_w[0..i]) - \mathcal{D}(w[0..i])\big)$. The representation of d is bounded by a polynomial value with respect to $|\mathcal{A}|$ and $|\mathcal{B}|$, hence it is finite. Also, $\mathcal{A}$ and $\mathcal{D}$ have finitely many states, meaning that there exist $j \neq k \in \mathbb{N}$, such that $\delta_{\mathcal{A}}(r_w[0..j]) = \delta_{\mathcal{A}}(r_w[0..k]) = q_{acc}$, $\delta_{\mathcal{D}}(w[0..j]) = \delta_{\mathcal{D}}(w[0..k]) = p_{acc}$, and $d_j = d_k = d_{acc}$. Hence the accept condition is met when the $(\max\{j,k\})$-sized prefixes of w and r_w are generated.

Combined with the results shown in the proof of Theorem 42, we conclude that there exist an infinite word w and a run r_w of $\mathcal{A}$ on w, such that $\mathcal{A}(r_w) - \mathcal{B}(w) \leq 0$ iff one of the accept conditions is met. □

A PSPACE algorithm for equivalence directly follows from the fact that $\mathcal{A} \equiv \mathcal{B}$ if and only if $\mathcal{A} \geq \mathcal{B}$ and $\mathcal{B} \geq \mathcal{A}$.

Corollary 44. *The equivalence problem of tidy NMDAs is PSPACE-complete.*

We continue with the universality problems which are special cases of the containment problems.

Theorem 45. *The universality problems of tidy NMDAs are in PSPACE. The universality(<) w.r.t. finite words, universality(≤) w.r.t. finite words, and universality(≤) w.r.t. infinite words are PSPACE-hard.*

Proof. We will show that the universality problems of tidy NMDAs are in PSPACE. Hardness directly follows from Lemma 40 for universality(<) with respect to finite words, from Lemma 38 for universality(≤) with respect to finite words, and from Lemma 39 for universality(≤) with respect to infinite words.

Consider a tidy NMDA $\mathcal{B}$, and a threshold ν. The universality(<) is a special case of the containment(>) problem, replacing the automaton $\mathcal{A}$ of the containment problem with a constant function that returns ν. Similarly, the non-strict universality is a special case of the non-strict containment. Accordingly, the algorithms for solving those problems are identical to the proofs of

Theorems 41 to 43, with changing all the references to the automaton $\mathcal{A}$ with a "virtual" automaton implementing the constant function ν. For that purpose, the local data will be initialized with a normalized difference of $d = \nu$ (instead of 0), and when updated, we replace the addition of $\gamma_{\mathcal{A}}(t)$ with 0, i.e., having $d' = \rho_{\mathcal{A}}(t)(d + 0 - \gamma_{\mathcal{D}}(p, \sigma))$. The maximal recoverable distance S will be calculated using $M_{\mathcal{A}} = 0$.

The space requirement analysis is identical to Theorem 41 with omitting the analysis of $\mathcal{A}$. □

Theorem 46. *The exact-value problem of tidy NMDAs is in PSPACE (and PSPACE-complete w.r.t. finite words).*

Proof. Consider a tidy NMDA $\mathcal{B}$ and a threshold ν. The procedures for checking the existance of a words w such that $\mathcal{B}(w) = \nu$ are similar to the procedures used in Theorems 41 and 43 for the containment(>) problems, with replacing the automaton $\mathcal{A}$ with a "virtual" NMDA for the constant function ν, as in the proof of Theorem 45, and using only the accept conditions that determines $\nu - \mathcal{B}(w) = 0$. For the fininte words case, the accept condition is generating a word that its normalized difference is $d = 0$. An analysis similar to the one showed in the proof of Theorem 41, with replacing "< 0" in the equations with "$= 0$", proves the correctness.

For the infinite words case, the accept condition is the one presentented in the proof of Theorem 43, which determines the convergence to ν. In the proof of Theorem 43 we showd that this accept condition determines the existence of an infininte word w such that $\nu - \mathcal{B}(w) = 0$.

In both problems we also abort if the normalized difference gets below $-2S$, to preserve the polynomial space usage.

Hardness with respect to finite words directly follows from Lemma 40. □

5.3 Nonemptiness, Universality, Equivalence, and Containment of integral DMDAs

We show that the nonemptiness, universality, equivalence, and containment problems of integral DMDAs are in PTIME, since we can reduce them to the non-emptiness problems of integral NMDAs which are in PTIME.

Theorem 47. *The non-emptiness, containment, equivalence and universality problems of integral DMDAs are in PTIME for both finite and infinite words.*

Proof. The complexity of the non-emptiness problem directly follows from Theorems 35 to 37.

We will now show that the containment problems are special cases of the emptiness problems when swapping the strictness of the problem ("$>$" becomes "$\leq$" and "$\geq$" becomes "$<$"). Consider integral DMDAs $\mathcal{A}$ and $\mathcal{B}$. According to Theorem 13, we can construct an integral DMDA $\mathcal{C} \equiv \mathcal{A} - \mathcal{B}$ in linear time. Observe that for all words w, $\mathcal{A}(w) > \mathcal{B}(w) \Leftrightarrow$ for all words w, $\mathcal{C}(w) > 0 \Leftrightarrow$ there is no word w s.t $\mathcal{C}(w) \leq 0$. Meaning that $\mathcal{A}$ is contained(>) in $\mathcal{B}$ iff $\mathcal{C}$ is empty($\leq$) with respect to the threshold 0. Similarly, $\mathcal{A}$ is contained($\geq$) in $\mathcal{B}$ iff $\mathcal{C}$ is empty($<$) with respect to the threshold 0.

Equivalence is a special case of containment($\geq$) as in Corollary 44, and the universality problems are special cases of the containment problems when setting $\mathcal{B}$ to be the input DMDA and $\mathcal{A}$ to be a constant DMDA that gets the value of the input threshold on every word. □

Observe that since Theorems 35 and 36 are also valid for general NMDAs, with not necessarily integral discount factors, the results of Theorem 47 are also valid for general DMDAs, in all the problems with respect to infinite words, and in the problems of non-emptiness($<$), containment($\geq$), universality($\leq$) and equivalence w.r.t. finite words.

6 Conclusions and Future Work

The measure functions most commonly used in the field of quantitative verification, whether for describing system properties [12, 23, 31], automata valuation schemes [8, 9, 17, 4], game winning conditions [2, 24, 37], or temporal specifications [1, 6, 22, 35], are the limit-average (mean payoff) and the discounted-sum functions.

Limit-average automata cannot always be determinized [17] and checking their (non-strict) universality is undecidable [24]. Therefore, the tendency is to only use deterministic such automata, possibly with the addition of algebraic operations on them [13].

Discounted-sum automata with arbitrary rational discount factors also cannot always be determinized [17] and are not closed under algebraic operations [9]. Yet, with integral discount factors, they do enjoy all of these closure properties and their decision problems are decidable [9]. They thus provide a very interesting automata class for quantitative verification. Yet, they have a main drawback of only allowing a single discount factor.

We define a rich class of discounted-sum automata with multiple integral factors (tidy NMDAs) that strictly extends the expressiveness of automata with a single factor, while enjoying all of the good properties of the latter, including the same complexity of the required decision problems. We thus believe that tidy NMDAs can provide a natural and useful generalization of integral discounted-sum automata in all fields and especially in quantitative verification.

We list below several natural directions for future research.

- *Containment of NMDAs with different choice functions:* Our algorithms for the containment problem of tidy NMDAs (Theorems 41 and 42) take advantage of the special configuration that results from the fact that both input automata have the same choice function. The algorithms cannot be directly extended to tidy NMDAs with different choice functions, and we conjecture that the solution for the latter might require a different approach and a much higher complexity.

- *Lower bounds:* Our lower bounds for the size blow-up involved in algebraic operations handled the subtraction and multiplying by -1 operations. Lower bound for the size blow-up involved in the max operation still remains an open question. Further, to the best of our knowledge, the exact-value problem of integral DMDAs is also currently an open question.

- *Non-integral discount factors:* We focused on integral NMDAs, extending the nice properties of integral NDAs. Studying non-integral NMDAs might shed some light on the open problems of non-integral NDAs, either toward finding a nice subclass that is closed under algebraic operations, or better understanding the difficulty in resolving the decidability of the universality, equivalence and containment problems, and whether it has some relation to the tidiness property that we introduced.

- *Comparator automata:* Comparator automata are introduced and studied in [4, 3, 5] for providing means to compare the aggregate values of two sequences of quantitative weights. They are instrumental in solving decision problems of quantitative automata and finding winning strategies in quantitative games. They are currently studied for several aggregation functions, among which are discounted summation with a single integral discount factor. It may be interesting to study if and how comparator automata can be extended to discounted summation with multiple discount factors.

- *Netsed automata:* In nested weighted automata, introduced in [20], a master automaton spins off and collects results from weighted slave automata, each of which computes a quantity along a finite portion of an infinite word. The master automaton is equipped with a value function on infinite sequences and the slave automata with a value function on finite sequences. Nested weighted automata were studied with respect to several value functions, such as limit-average for infinite sequences and summation for finite sequences [20, 19]. It is natural to enrich their value functions with discounted summation, with either a single or multiple discount factors, to both finite and infinite sequences. Notice that it can be combined with the other value functions, for example having discounted summation for the slave automata and LimSup for infinite sequences.

- *Probabilistic semantics:* In [21, 33], quantitative automata are studied under probabilistic semantics, where the probability distribution is given by means of a Markov chain. Under this setting, probabilistic questions are considered, such as the expected value of the automaton or its cummulative distribution under a given threshold, rather than the classic automata decision problems, such as emptiness and universality. Several types of quantitative automata were studied under this setting, among which are LimSup and LimAvg automata. It is natural to also study tidy NMDAs under the probabilistic semantics.

- *Implementation:* On the practical side, it will be interesting to implement our decision algorithms, and compare their performance with existing algorithms (which currently only address NDAs), like the ones presented in [3, 5].

References

[1] Shaull Almagor, Udi Boker, and Orna Kupferman. Discounting in LTL. In *proceedings of TACAS*, pages 424–439, 2014.

[2] Daniel Andersson. An improved algorithm for discounted payoff games. In *proceedings of ESSLLI Student Session*, pages 91–98, 2006.

[3] Suguman Bansal, Swarat Chaudhuri, and Moshe Y. Vardi. Automata vs linear-programming discounted-sum inclusion. In *proceedings of CAV*, pages 99–116, 2018.

[4] Suguman Bansal, Swarat Chaudhuri, and Moshe Y. Vardi. Comparator automata in quantitative verification. In *proceedings of FoSSaCS*, pages 420–437, 2018.

[5] Suguman Bansal and Moshe Y. Vardi. Safety and co-safety comparator automata for discounted-sum inclusion. In *proceedings of CAV*, pages 60–78, 2019.

[6] Udi Boker, Krishnendu Chatterjee, Thomas. A. Henzinger, and Orna Kupferman. Temporal specifications with accumulative values. *ACM Transactions on Computational Logic*, 15(4):27:1–27:25, 2014.

[7] Udi Boker and Guy Hefetz. Discounted-sum automata with multiple discount factors. submitted.

[8] Udi Boker and Thomas A. Henzinger. Approximate determinization of quantitative automata. In *proceedings of FSTTCS*, volume 18 of *LIPIcs*, pages 362–373, 2012.

[9] Udi Boker and Thomas A. Henzinger. Exact and approximate determinization of discounted-sum automata. *Logical Methods in Computer Science*, 10(1), 2014.

[10] Udi Boker, Thomas A. Henzinger, and Jan Otop. The target discounted-sum problem. In *proceedings of LICS*, pages 750–761, 2015.

[11] Udi Boker, Orna Kupferman, and Michal Skrzypczak. How deterministic are good-for-games automata? In *proceedings of FSTTCS*, pages 18:1–18:14, 2017.

[12] Krishnendu Chatterjee. Markov decision processes with multiple long-run average objectives. In *proceedings of FSTTCS*, volume 4855 of *LNCS*, pages 473–484. 2007.

[13] Krishnendu Chatterjee, Laurent Doyen, Herbert Edelsbrunner, Thomas A. Henzinger, and Philippe Rannou. Mean-payoff automaton expressions. In *proceedings of CONCUR*, pages 269–283, 2010.

[14] Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Alternating weighted automata. In *proceedings of FCT*, volume 5699 of *LNCS*, pages 3–13, 2009.

[15] Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Probabilistic weighted automata. In *proceedings of CONCUR*, pages 244–258, 2009.

[16] Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Expressiveness and closure properties for quantitative languages. *Logical Methods in Computer Science*, 6(3), 2010.

[17] Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Quantitative languages. *ACM Trans. Comput. Log.*, 11(4), 2010.

[18] Krishnendu Chatterjee, Vojtěch Forejt, and Dominik Wojtczak. Multi-objective discounted reward verification in graphs and MDPs. In *proceedings of LPAR*, volume 8312, pages 228–242, 2013.

[19] Krishnendu Chatterjee, Thomas A. Henzinger, and Jan Otop. Bidirectional nested weighted automata. In *proceedings of CONCUR*, pages 5:1–5:16, 2017.

[20] Krishnendu Chatterjee, Thomas A. Henzinger, and Jan Otop. Nested weighted automata. *ACM Transactions on Computational Logic*, 18(4):1–44, 2017.

[21] Krishnendu Chatterjee, Thomas A. Henzinger, and Jan Otop. Quantitative automata under probabilistic semantics. *Logical Methods in Computer Science*, 15(3), 2019.

[22] Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar, and Mariëlle Stoelinga. Model checking discounted temporal properties. *Theoretical Computer Science*, 345(1):139–170, 2005.

[23] Luca de Alfaro, Thomas A. Henzinger, and Rupak Majumdar. Discounting the future in systems theory. In *proceedings of ICALP*, pages 1022–1037, 2003.

[24] Aldric Degorre, Laurent Doyen, Raffaella Gentilini, Jean-François Raskin, and Szymon Torunczyk. Energy and mean-payoff games with imperfect information. In *proceedings of CSL*, volume 6247 of *LNCS*, pages 260–274, 2010.

[25] Manfred Droste and Dietrich Kuske. Skew and infinitary formal power series. *Theor. Comput. Sci.*, 366(3):199–227, 2006.

[26] Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. Finite-Valued Weighted Automata. In *proceedings of FSTTCS*, volume 29, pages 133–145, 2014.

[27] Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. Quantitative languages defined by functional automata. *Logical Methods in Computer Science*, 11(3), 2015.

[28] Hugo Gimbert and Wieslaw Zielonka. Limits of multi-discounted Markov decision processes. In *proceedings of LICS*, pages 89–98, 2007.

[29] Thomas A. Henzinger and Nir Piterman. Solving games without determinization. In *proceedings of CSL*, pages 394–410, 2006.

[30] Galina Jirásková. State complexity of some operations on binary regular languages. *Theoretical Compututer Science*, 330:287–298, 2005.

[31] Omid Madani, Mikkel Thorup, and Uri Zwick. Discounted deterministic Markov decision processes and discounted all-pairs shortest paths. *ACM Transactions on Algorithms*, 6(2), 2010.

[32] Albert R. Meyer and Larry J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *proceedings of 13th IEEE Symp. on Switching and Automata Theory*, pages 125–129, 1972.

[33] Jakub Michaliszyn and Jan Otop. Non-deterministic weighted automata evaluated over Markov chains. *Journal of Computer and System Sciences*, 108:118–136, 2020.

[34] William J. Sakoda and Michael Sipser. Nondeterminism and the size of two way finite automata. In *proceedings of STOC*, pages 275–286, 1978.

[35] Takashi Tomita, Shin Hiura, Shigeki Hagihara, and Naoki Yonezaki. A temporal logic with mean-payoff constraints. In *Formal Methods and Software Engineering*, volume 7635 of *LNCS*, pages 249–265. 2012.

[36] Moshe Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *proceedings of FOCS*, pages 327–338, 1985.

[37] Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158:343–359, 1996.

דטרמיניסטים שאינם מוגבלים בבחירת פרמטרי הפיחות. לסיום אנחנו מנתחים את הקשרים בין המחלקות השונות של NMDA-ים מסודרים, על מנת להציג את החשיבות של כל אחת מהן: אף אחת מהן אינה מוכלת ממש באחרת, איחוד שתי מחלקות שונות גורם לאיבוד תכונת הסגירות לפעולות אלגבריות, והחיתוך של כל המחלקות הללו הוא בדיוק קבוצת הפונקציות שערכן קבוע החל מאורך מילה סופי כלשהו.

כל התוצאות שלנו תקפות גם עבור אוטומטים למילים סופיות וגם עבור אוטומטים למילים אינסופיות.

תקציר

חישוב מופחת (Discounted sum) הוא כלי מרכזי בכלכלה ונמצא בשימוש נרחב במודלים במדעי המחשב כגון משחקים, תהליכי הכרעה מרקוביים (MDP) ואוטומטים, היות והוא מגדיר את הרעיון שרווח מיידי עדיף על רווח פוטנציאלי בעתיד הרחוק, וכן שבעיה פוטנציאלית בעתיד (למשל באג במערכות ראקטיביות), מדאיגה פחות מבעיה קיימת. בעוד שמשחק או MDP יכול להכיל מספר פרמטרי פיחות (Discount Factors), אוטומטי סכום מופחת (NDA – Nondeterministic Discounted-sum Automata) נחקרו רק בהקשרים בהם ישנו פרמטר פיחות יחיד. עבור כל מספר שלם $\lambda \in \mathbb{N}\backslash\{0,1\}$, למחלקה של כל הNDA-ים עם פרמטר הפיחות λ ישנן תכונות חישוביות טובות: המחלקה סגורה לדטרמיניזציה ותחת הפעולות האלגבריות של מינימום, מקסימום, חיבור וחיסור. בנוסף, ישנם אלגוריתמים לבעיות ההכרעה היסודיות, כגון הכלה ושוויון אוטומטים, עבור האוטומטים במחלקה. הנ״ל אינו מתקיים עבור פרמטרי פיחות שאינם שלמים ($\lambda \in \mathbb{Q} \setminus \mathbb{N}$).

בעבודה זו אנו מגדירים ומנתחים אוטומטי סכום מופחת בהם כל מעבר (transition) יכול להיות בעל פרמטר פיחות שלם שונה (NMDA אינטגרלים). אנחנו מראים שNMDA אינטגרלים עם בחירה שרירותית של פרמטרי פיחות אינם סגורים לדטרמיניזציה ולפעולות האלגבריות. בהמשך אנחנו מגדירים ומנתחים תת מחלקה של NMDA אינטגרליים, להם אנחנו קוראים NMDA-ים *מסודרים*, כך שבחירת פרמטרי הפיחות תלויה בתחילית של המילה שנקראה עד לנקודת הזמן הנוכחית. מקרים פרטיים של אוטומטים אלו הם NMDA-ים בהם פרמטרי הפיחות נקבעים בהתאם לפעולה (אות ב- א״ב), או בהתאם לזמן. אנו מראים שלכל פונקציה θ המגדירה את הבחירה של פרמטרי הפיחות, המחלקה של כל ה θ-NMDAs נהנית מכל התכונות החישוביות הטובות של NDA-ים אינטגרליים, ומאותה הסיבוכיות לבעיות ההכרעה המתאימות. בנוסף, NMDA-ים מסודרים הם בעלי כוח הבעה השווה לNMDA-ים אינטגליים

עבודה זו בוצעה בהדרכתו של פרופ׳ אודי בוקר מבי״ס אפי ארזי למדעי המחשב, המרכז הבינתחומי, הרצליה.

המרכז הבינתחומי הרצליה

בית-ספר אפי ארזי למדעי המחשב

התכנית לתואר שני (M.Sc.) - מסלול מחקרי

אוטומטי סכום מופחת עם פרמטרי פיחות מרובים

מאת

גיא חפץ

עבודת תזה המוגשת כחלק מהדרישות לשם קבלת תואר מוסמך M.Sc.
במסלול המחקרי בבית ספר אפי ארזי למדעי המחשב, המרכז הבינתחומי הרצליה

יוני 2020