

On the Complexity of the Discussion-based Semantics in Abstract Argumentation

Lydia Blümel¹, Kai Sauerwald¹, Kenneth Skiba^{1,2} and Matthias Thimm¹

¹University of Hagen

²University of Luxembourg

Abstract

We show that deciding whether an argument a is stronger than an argument b with respect to the discussion-based semantics of Amgoud and Ben-Naim is decidable in polynomial time. At its core, this problem is about deciding whether, for two vertices in a graph, the number of walks of each length ending in those vertices is the same. We employ results from automata theory and reduce this problem to the equivalence problem for semiring automata. This offers a new perspective on the computational complexity of ranking semantics, an area in which the complexity of many semantics remains open.

Keywords

Ranking-based Semantics, Complexity, Number of walks, EqualWalkCount, Discussion-based Semantics, Semiring-automata, Equivalence, Abstract Argumentation

1. Introduction

Abstract argumentation frameworks (AAFs) [1] are an approach for modelling argumentative scenarios that represents arguments as vertices in a directed graph, where a directed edge represents an attack from one argument to another. Besides the classical approach for reasoning in AAFs consisting of identifying sets of arguments (*extensions*) that form a *plausible* outcome of the argumentation, another important approach are *ranking-based* semantics [2, 3, 4]. A ranking-based semantics gives a ranking (usually a total preorder) over the set of arguments of an AAF, with the intended meaning that higher ranks indicate stronger acceptability or plausibility. Using ranking-based semantics, a more finer-grained assessment of the strength of arguments can be achieved, in contrast to classical extension-based semantics. Several different approaches for ranking-based semantics (and the related *gradual semantics*) have been proposed over the years [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 2, 18], ranging from approaches founded in the topological analysis of the underlying graph structure [2, 15] to approaches generalising the extension-based perspective to rankings [10], see also [4] for a recent survey.

In this paper, we are addressing issues of computability and complexity of ranking-based semantics. Interestingly, very little work has been done on this aspect so far. Some works such as [19, 4] do present algorithms for some ranking-based semantics and conduct empirical evaluations, but the exact computational complexity of the even most basic ranking-based semantics from [2] has not been settled until now. This is not so surprising, after all, the problem of deciding whether some argument a is stronger than another argument b wrt., in particular, the *discussion-based semantics* [2], has some underlying combinatorial challenges when it comes to considering walks of infinite lengths in the argumentation framework. In fact, when abstracting from the application of ranking-based semantics in abstract argumentation, the underlying graph theoretical problem is quite intricate and has, to the best of our knowledge, not been considered in its generality so far.

We consider the *discussion-based semantics* [2] in this paper as an example for a simple ranking-based semantics. We will present the formal background in Section 3, but this semantics, roughly, assesses an argument a is stronger than an argument b if the number of odd (resp. even) walks ending in a are less (resp. more) than the corresponding number of walks ending in b . Implementations of this semantics, such as [4], usually set some maximal length of walks to consider in order to have a

terminating algorithm. However, no formal justification is known why any maximal length maintains the correctness of the algorithm. We will settle this question and show, in particular, that the problem of deciding whether an argument a is stronger than an argument b wrt. the discussion-based semantics is not only decidable but also decidable in polynomial time. In order to show this, we use results from automata theory and, in particular, about the equivalence of *automata over semirings* [20, 21]. Our result does not only shed light on computational questions regarding the discussion-based semantics, but also on all ranking-based semantics that are defined in terms of walks in an abstract argumentation framework. Moreover, our result also solves an interesting general graph-theoretical problem that has not been considered until now, i. e., deciding for two given vertices whether, for each length, the number of walks of that length is the same.

The main contributions of this paper are:

- [Interlinking Concepts] We introduce the notion of *agreement on the numbers of walks* for two vertices of a graph, which is for each length the number of walks of that length that end in these vertices is the same. This notion offers the following connections between areas:
 - [Ranking Semantics $\rightarrow$ Graph Theory] The agreement on the numbers of walks strongly corresponds to the equivalence of strength of arguments with respect to discussion-based semantics. (Section 4)
 - [Graph Theory $\rightarrow$ Semiring Automata] For two vertices v and u , there are two semiring automata over the rationals that are equivalent if and only if v and u agree on the number of walks. (Section 6)
- [Complexity of EQUALWALKCOUNT] Deciding if two vertices of a graph agree on the numbers of walks is shown to be in **PTIME**. (Section 6)
- [Complexity of Discussion-Based Semantics] Deciding **STRONGERDIS**, i. e., if an argument is stronger than another argument with respect to discussion-based semantics, and deciding **EQUIV-DIS**, i. e., if two arguments have the same strength with respect to discussion-based semantics, are shown to be in **PTIME**. (Section 7)

Notably, we will derive that the length of walks one must consider in each of the aforementioned problems is bounded. This paper is the first to establish such bounds, which are central to identifying the computational complexity.

The paper is organized as follows. The main results are presented in Section 4, Section 6 and Section 7 as indicated above. We are presenting in Section 2 (basic notions), Section 3 (abstract argumentation), and Section 5 (semiring automata) the required background and notions. We conclude in Section 8.

2. Background and Basic Notions

With $\mathbb{N} = \{1, 2, 3, \dots\}$ we denote the set of positive integers and denote by $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ the set of non-negative integers. The semiring of rational numbers is the tuple $\langle \mathbb{Q}, \cdot, +, 0, 1 \rangle$, where the product “ $\cdot$ ” of the semiring is the usual multiplication on $\mathbb{Q}$ and the sum “ $+$ ” is the usual addition on $\mathbb{Q}$, the neutral element of the product is 1 and 0 is the neutral element of summation (and null-element of the product). We will use notions and results from formal language and automata theory (see, e.g., [22]) and computational complexity (see, e.g., [23]). With ε we indicate the empty word. A decision problem P is a formal language over $\{0, 1\}$, and a complexity class is a set of decision problems. We write $P_1 \leq_p P_2$ if the decision problem P_1 can be reduced to P_2 in polynomial time. The following relationship between complexity classes is known:

$$\mathbf{NC} \subseteq \mathbf{PTIME} \subseteq \mathbf{PSPACE}$$

NC consists of all problems decidable in polylogarithmic time on a parallel computer with a polynomial number of processors, **PTIME** consists of all problems decidable in polynomial time by a deterministic-Turing machine, and **PSPACE** is the class of all problems that can be solved with polynomial space by a deterministic Turing machine.

3. Abstract Argumentation

An *abstract argumentation framework* (AAF) [1] is a directed graph $F = (A, R)$ where A is a (finite) set of *arguments* and R is a relation $R \subseteq A \times A$. An argument a is said to *attack* an argument b if $(a, b) \in R$. For a set $E \subseteq A$ we define

$$\begin{aligned} E_F^+ &= \{a \in A \mid \exists b \in E, bRa\} \\ E_F^- &= \{a \in A \mid \exists b \in E, aRb\} \end{aligned}$$

If E is a singleton set, i. e., $E = \{a\}$ for some $a \in A$, then we also just write a_F^+ (resp. a_F^-) for $\{a\}_F^+$ (resp. $\{a\}_F^-$). If the AF is clear in the context, we will omit the index.

A *ranking-based semantics* [2] is defined as follows.

Definition 3.1. A *ranking-based semantics* ρ is a function which maps an AAF $F = (A, R)$ to a preorder $\succeq_F^\rho$ on A .

Intuitively $a \succeq_F^\rho b$ means that a is at least as strong as b in F . We define the usual abbreviations as follows; $a \succ_F^\rho b$ denotes *strictly stronger*, i. e. $a \succeq_F^\rho b$ and $b \not\succeq_F^\rho a$. Moreover, $a \simeq_F^\rho b$ denotes *equally strong*, i. e. $a \succeq_F^\rho b$ and $b \succeq_F^\rho a$.

Many different ranking-based semantics exist, see [4] for a survey, but in this paper, we will only consider the *discussion-based semantics* from Amgoud and Ben-Naim [2]. Very central to this semantics is the notion of length of walks, from graph theory, which we define next.

Definition 3.2. Let $F = (A, R)$ be an AAF, $x \in A$ and $i > 0$. The set of walks of length i in F that end in x is denoted by $\mathcal{R}_i^F(x)$ and defined by:

$$\begin{aligned} \mathcal{R}_1^F(x) &= \{(y, x) \mid y \in x_F^-\} \\ \mathcal{R}_i^F(x) &= \bigcup_{y \in x_F^-} \{(z_1, \dots, z_{i-1}, y, x) \mid (z_1, \dots, z_{i-1}, y) \in \mathcal{R}_{i-1}^F(y)\} \end{aligned}$$

for all $i > 1$.

The intuition behind the discussion-based semantics is that an argument is stronger if it has less *attack walks*, i. e., walks of odd length that end in the argument, and more *defense walks*, i. e., walks of even length that end in the argument. This is formalised by Amgoud and Ben-Naim [2] as follows.

Definition 3.3. Let $F = (A, R)$ be an AAF, $x \in A$ and $i > 0$. The *discussion count* $Dis(x)$ is an infinite vector of natural numbers $Dis^F(x) = \langle Dis_1^F(x), Dis_2^F(x), \dots \rangle$ with

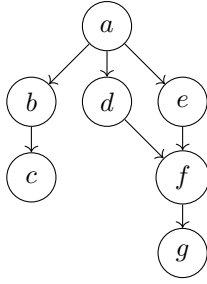
$$Dis_i^F(x) = \begin{cases} -|\mathcal{R}_i^F(x)| & \text{if } i \text{ is odd} \\ |\mathcal{R}_i^F(x)| & \text{if } i \text{ is even} \end{cases}$$

Now an argument a is stronger than an argument b if the discussion count of a is *lexicographically larger* than the discussion count of b . That is, for two (possibly infinite) vectors $\vec{x} = \langle x_1, x_2, \dots \rangle$ and $\vec{y} = \langle y_1, y_2, \dots \rangle$ we have $\vec{x} \geq^{\text{lex}} \vec{y}$ if either $x_i = y_i$ for all $i > 0$ or there is $k > 0$ with $x_k > y_k$ and $x_j = y_j$ for all $j = 1, \dots, k-1$.

Definition 3.4. Let $F = (A, R)$ be an AAF. The *discussion-based semantics* ρ^{dbs} is defined via $\succeq_F^{\text{dbs}} = \rho^{\text{dbs}}(F)$ with

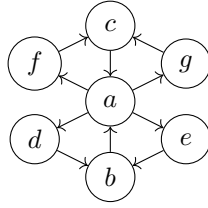
$$a \succeq_F^{\text{dbs}} b \text{ iff } Dis^F(a) \geq^{\text{lex}} Dis^F(b)$$

for all $a, b \in A$.



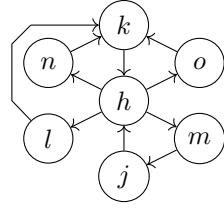
x	$Dis_1(x)$	$Dis_2(x)$	$Dis_3(x)$
a	0	0	0
b	-1	0	0
c	-1	1	0
d	-1	0	0
e	-1	0	0
f	-2	2	0
g	-1	2	-2

Figure 1: F_1 from Example 3.5 and the discussion count until 3.



x	$Dis_1(x)$	$Dis_2(x)$	$Dis_3(x)$
a	-2	4	-4
b	-2	2	-4
c	-2	2	-4
d	-1	2	-4
e	-1	2	-4
f	-1	2	-4
g	-1	2	-4

Figure 2: F_2 from Example 3.6 and the discussion count until 3.



x	$Dis_1(x)$	$Dis_2(x)$	$Dis_3(x)$
h	-2	4	-4
j	-1	1	-2
k	-3	3	-6
l	-1	2	-4
m	-1	2	-4
n	-1	2	-4
o	-1	2	-4

Figure 3: F_3 from Example 3.7 and the discussion count until 3.

Example 3.5. Let $F_1 = (A_1, R_1)$ with

$$A_1 = \{a, b, c, d, e, f, g\}$$

$$R_1 = \{(a, b), (a, d), (a, e), (b, c), (d, f), (e, f), (f, g)\}$$

be the AAF as depicted in Figure 1. Argument a is unattacked, so it is the strongest argument. The arguments b, d and e have the same attackers (argument a) and are therefore equally strong. The arguments c and g are both attacked by one argument each, however g is defended by two arguments, while c is only defended by one argument, thus $g \succ_{F_1}^{dbs} c$. Argument f is attacked by two arguments, rendering f the weakest one according to the discussion-based semantics. The ranking is:

$$a \succ_{F_1}^{dbs} g \succ_{F_1}^{dbs} c \succ_{F_1}^{dbs} e \simeq_{F_1}^{dbs} d \simeq_{F_1}^{dbs} b \succ_{F_1}^{dbs} f$$

In the remainder of this paper, we are interested in the following decision problems:

EQUIVDis

Input: AAF $F = (A, R)$ and $a, b \in A$.

Question: Does $a \simeq_F^{dbs} b$ hold?

STRONGERDis

Input: AAF $F = (A, R)$ and $a, b \in A$.

Question: Does $a \succeq_F^{dbs} b$ hold?

Note that the above questions can be easily answered if F is *acyclic*, since the number of all walks ending in arguments a and b are finite and their numbers can easily be compared in polynomial time. However, if F contains cycles, the set of walks ending in a and b , respectively, becomes infinite, with possibly a complex underlying structure.

Example 3.6. Let $F_2 = (A_2, R_2)$ with

$$A_2 = \{a, b, c, d, e, f, g\}$$

$$R_2 = \{(a, d), (a, e), (a, f), (a, g), (b, a), (c, a), (d, b), (e, b), (g, c), (f, c)\}$$

as depicted in Figure 2. Arguments d, g, f and g have the same attacker a , while the other three arguments have two attackers each. For argument a , there are four walks of length 2 and length 3 ending in that argument. For length 4, there are already eight walks of length 4 ending in argument a . These numbers increase significantly for longer lengths.

One might guess that $a \simeq_F^{dbs} b$ only holds if F is somewhat isomorphic about a and b , i. e., the topological structure of F is the same for both a and b . This is not the case as demonstrated next.

Example 3.7. Consider $F_3 = (A_3, R_3)$ with

$$A_3 = \{h, i, j, k, l, m, n, o\}$$

$$R_3 = \{(h, l), (h, m), (h, n), (h, o), (j, h), (k, h), (l, k), (m, j), (n, k), (o, k)\},$$

which is also presented in Figure 3, we compare F_3 with F_2 from Example 3.6. Note that there are two walks of length 2 to a over b and two walks of length 2 to a over c , but there is only one walk of length 2 to h over j and three walks of length 2 to h over k . Thus, the topological structure of F_3 and F_2 are different, but, nonetheless, as we show at the end of this paper, $a \simeq_{F_{\text{joint}}}^{\text{dbs}} h$ holds for the AAF $F_{\text{joint}} = (A_2 \cup A_3, R_2 \cup R_3)$.

The above examples may raise the question of whether there is a bound at all for a walk length i for which one has to consider $\text{Dis}_i^F(a)$ and $\text{Dis}_i^F(b)$ in order to be sure that $a \simeq_F^{\text{dbs}} b$ holds. One might argue that it should suffice to only consider walks up to length $|A| + 1$ since all longer walks contain cycles and are therefore “combinations” of walks already seen. However, there is no apparent formal reason for such an assumption, since it remains difficult (or even indeterminate), from a combinatorial point of view, to find the exact number of walks of a certain length, given that we already know the number of walks of smaller lengths. In fact, no closed-form solution is known for deriving the number of walks of length k for any k , given that we know the lengths of walks of length $1, \dots, k - 1$ [24].

4. Graph-Theoretical Perspective

We now take a more general perspective than for abstract argumentation frameworks and consider general directed graphs and graph theory notation, where we write $G = (V, E)$ for a directed graph. As for argumentation frameworks, we denote by $\mathcal{R}_i^G(v)$ the walks of length i in a graph $G = (V, E)$ that end in $v \in V$. If G is clear from the context, we may omit G .

We say that two vertices v, v' of G agree on the numbers of walks if for each length the number of walks of that length ending in these vertices is the same, i.e., $|\mathcal{R}_i(v)| = |\mathcal{R}_i(v')|$ holds for all $i \in \mathbb{N}$. Next, we define the corresponding decision problem as follows.

EQUALWALKCOUNT

Input: Graph $G = (V, E)$ and two vertices $v, v' \in V$.

Question: Do v and v' agree on the numbers of walks?

There is an immediate relationship between the problem **EQUALWALKCOUNT** and the problem **EQUIVDIS**.

Theorem 4.1. $\text{EQUIVDIS} \leq_p \text{EQUALWALKCOUNT}$

Proof. The reduction is the identity function $(F, a, b) \mapsto (F, a, b)$, which is easily computable in polynomial time. We have that $a \simeq_F^{\text{dbs}} b$ holds exactly whenever $\text{Dis}_n^F(a) = \text{Dis}_n^F(b)$ for all $n \in \mathbb{N}$. The latter is the case when $|\mathcal{R}_n(a)| = |\mathcal{R}_n(b)|$ hold for all $n \in \mathbb{N}$. $\square$

We establish connections between $|\mathcal{R}_i(v)|$ and other notions. With $N(v) = \{v' \in V \mid (v', v) \in E\}$ we denote the set of in-neighbours of v , i.e., those vertices that have an edge going out that ends in v . We define the notion

$$T_i : V \rightarrow \mathbb{N}_0 \quad T_1(v) = |N(v)| \quad T_{i+1}(v) = \sum_{v' \in N(v)} T_i(v'),$$

for which one can show that $T_i(v)$ coincides with the number of walks of length i ending in v .

Proposition 4.2. For all $v \in V$ and $u \in \mathbb{N}$ holds:

$$|\mathcal{R}_i(v)| = T_i(v)$$

Proof. For $i = 1$, this is immediate by the definitions. For $i \geq 1$, we have that

$$|\mathcal{R}_{i+1}(v)| = \left| \bigcup_{y \in N(v)} \{(z_1, \dots, z_i, y, v) \mid (z_1, \dots, z_i, y) \in \mathcal{R}_i(y)\} \right|$$

Because every tuple $(z_1, \dots, z_i, y)$ ends with the unique element y , the union is disjoint. Hence, we obtain the following:

$$|\mathcal{R}_{i+1}(v)| = \sum_{y \in N(v)} \left| \{(z_1, \dots, z_i, y, v) \mid (z_1, \dots, z_i, y) \in \mathcal{R}_i(y)\} \right| = \sum_{y \in N(v)} T_i(y) = T_{i+1}(v) \quad \square$$

Another representation of $|\mathcal{R}_i(v)|$ is via the adjacency matrix. The adjacency matrix $M \in \{0, 1\}^{|V| \times |V|}$ of G is the matrix with $M[v, v'] = 1$ if and only if $(v, v') \in E$; whereby, $M[v, v']$ stands for $M[i, j]$ presuming that $V = \{v_1, \dots, v_m\}$ and $v = v_i$ and $v' = v_j$. We write $M[a]$ as an abbreviation for $\sum_{b \in A} M[b, a]$. It is well-known, see, e.g., Tinhofer et al. [25], that $|\mathcal{R}_i(v)|$ can be represented as the sum of the column of v in the i -th power of the adjacency matrix M .

Proposition 4.3. *For all $v \in V$ and $i \in \mathbb{N}$ holds:*

$$|\mathcal{R}_i(v)| = M^i[v]$$

Proof. For $i = 1$, we have

$$\sum_{j=1}^m A[v_j, v] = \sum_{v' \in V, (v', v) \in E} 1 = |\mathcal{R}_1(v)|$$

For $i = n + 1$ we have

$$\sum_{j=1}^m A^i[v_j, v] = \sum_{j=1}^m (A^{i-1} \cdot A)[v_j, v] = \sum_{v' \in V, (v', v) \in E} \sum_{j=1}^m A^{i-1}[v_j, v'] = \sum_{v' \in V, (v', v) \in E} |\mathcal{R}_{i-1}(v')| = |\mathcal{R}_i(v)| \quad \square$$

5. $\mathbb{Q}$ -Automata and their Equivalence

We provide background on semiring-automata and the complexity of computing whether two respective automata over the semiring of rational numbers are equivalent.

5.1. $\mathbb{Q}$ -Automata

A generalization of finite (nondeterministic) automata is its extension to arbitrary semirings and its connection to formal power series [26, 27]. In the following, we introduce automata over the semiring of rational numbers $\mathbb{Q}$. For a finite alphabet Σ , a formal power series (over the semiring $\mathbb{Q}$ and Σ^*) is a function $S : \Sigma^* \rightarrow \mathbb{Q}$.

Semiring-automata follow the same intuition as nondeterministic finite automata [22, 28], but assign weights from an underlying semiring to the initial states, to the transitions and to the final states. The semantics is then that every run yields a weight, which is the product of the weights along the run.

Definition 5.1. A $\mathbb{Q}$ -automaton (over the finite alphabet Σ) is a tuple $\mathcal{A} = \langle Q, \Sigma, \delta, I, F \rangle$ such that

- Q a finite set (the states),
- Σ a finite alphabet,
- $\delta : Q \times \Sigma \times Q \rightarrow \mathbb{Q}$ the transition function,
- $I : Q \rightarrow \mathbb{Q}$ initial values, and
- $F : Q \rightarrow \mathbb{Q}$ final values.

For all states $q, p \in Q$, all $n > 1$ and all $a, a_1, \dots, a_n \in \Sigma$ define the extended transition function $\delta^* : Q \times \Sigma^* \times Q \rightarrow \mathbb{Q}$:

$$\begin{aligned} \delta^*(q, \varepsilon, p) &= \begin{cases} 1 & , \text{ if } p = q \\ 0 & , \text{ if } p \neq q \end{cases} & \delta^*(q, a, p) &= \delta(q, a, p) \\ \delta^*(q, a_1 \dots a_n, p) &= \sum_{r \in Q} \delta(q, a_1 \dots a_{n-1}, r) \cdot \delta(r, a_n, p) \end{aligned}$$

For each state $q \in Q$ and each word $w \in \Sigma^*$ define:

$$\mathcal{A}(q, w) = \sum_{p \in Q} I(p) \cdot \delta^*(p, w, q) \cdot F(q)$$

That is, $\mathcal{A}(q, w)$ is the sum over all runs for the word w that end in q , where weights are multiplied. Each automaton $\mathcal{A}$ induces a formal power series $\llbracket \mathcal{A} \rrbracket$ which is defined by:

$$\llbracket \mathcal{A} \rrbracket : \Sigma^* \rightarrow \mathbb{Q} \qquad \llbracket \mathcal{A} \rrbracket(w) = \sum_{q \in Q} \mathcal{A}(q, w)$$

Note that, when one replaces $\mathbb{Q}$ by the semiring of Booleans $\mathbb{B} = (\{0, 1\}, \wedge, \vee, 0, 1)$, one obtains nondeterministic finite automata. For instance, in $\mathbb{B}$ -automata $\mathcal{A}(q, w) = 1$ holds when there is an accepting run for the word w that starts in q .

5.2. Equivalence of $\mathbb{Q}$ -Automata

We say that two $\mathbb{Q}$ -automata are equivalent if they induce the same formal power series. The following problem of equivalence has been considered in automata theory.

$\mathbb{Q}$ -EQUIVALENCE

Input: $\mathbb{Q}$ -automata $\mathcal{A}_1$ and $\mathcal{A}_2$ over the same alphabet Σ .

Question: Does $\llbracket \mathcal{A}_1 \rrbracket = \llbracket \mathcal{A}_2 \rrbracket$ hold?

The general methodology for deciding the equivalence of semiring automata is attributed to Schützenberger [29]. Decidability for the semiring $\mathbb{Q}$ is given by Ésik and Maletti [30]. The detailed analysis of the computational complexity of $\mathbb{Q}$ -EQUIVALENCE is given by Tzeng [20].

Proposition 5.2 (Tzeng [20]). *$\mathbb{Q}$ -EQUIVALENCE is in NC.*

We will also make use of the following result by Kiefer [31] that shows when solving $\mathbb{Q}$ -EQUIVALENCE in **PTIME** (instead of **NC**), one obtains a witness whenever the input is non-equivalent.

Proposition 5.3 (Kiefer [31]). *There is an algorithm that decides $\mathbb{Q}$ -EQUIVALENCE in polynomial time for two given automata $\mathcal{A}_1 = \langle Q_1, \Sigma, \delta_1, I_1, F_1 \rangle$ and $\mathcal{A}_2 = \langle Q_2, \Sigma, \delta_2, I_2, F_2 \rangle$, and if $\mathcal{A}_1$ and $\mathcal{A}_2$ are not equivalent, then a word $w \in \Sigma^*$ with $\llbracket \mathcal{A}_1 \rrbracket(w) \neq \llbracket \mathcal{A}_2 \rrbracket(w)$ of maximal size $|Q_1| + |Q_2| - 1$ is provided.*

The proof of the above proposition is constructive and provides an algorithm. We will outline this algorithm now, but for a more deliberated and detailed presentation, we refer to Kiefer [31]. For the algorithm, one computes the *linear representation* $\langle \alpha, M, \eta \rangle$ of an $\mathbb{Q}$ -automaton $\mathcal{A} = \langle Q, \Sigma, \delta, I, F \rangle$ with $Q = \{q_0, \dots, q_{n-1}\}$, which is that

- $\alpha \in \mathbb{Q}^{1 \times n}$ is the initial vector, a row-vector over $\mathbb{Q}$, such that $\alpha[0, i] = I(q_i)$ for all $i \in \{0, \dots, n\}$,
- $M : \Sigma \rightarrow \mathbb{Q}^{n \times n}$ assigns to every sign a quadratic matrix (the transition matrix), such that $M(a)[i, j] = \delta(q_i, a, q_j)$ for all $i \in \{0, \dots, n\}$, and
- $\eta \in \mathbb{Q}^{n \times 1}$ is the final vector, a column-vector, such that $\eta[i, 0] = F(q_i)$ for all $i \in \{0, \dots, n\}$.

One can obtain the power series $\llbracket \mathcal{A} \rrbracket$ by means of the matrix multiplication, i. e., for all words $w = a_0 \dots a_m \in \Sigma^*$ it holds

$$\llbracket \mathcal{A} \rrbracket(w) = \alpha \cdot M(w) \cdot \eta$$

whereby $M(w) = M(a_0) \cdot \dots \cdot M(a_m)$. Because $\mathbb{Q}$ is also a field, $\mathbb{Q}^n$ is also a vector space over $\mathbb{Q}$ of dimension $\dim(\mathbb{Q}^n) = n$. For a set of vectors $S \subseteq \mathbb{Q}^n$, we denote by $\text{span}(S)$ the smallest subset of $\mathbb{Q}^n$ that contains S and is itself a vector space (with the same scalars and operators from $\mathbb{Q}^n$). The *forward-space* $\mathcal{F}_{\mathcal{A}}$ of a $\mathbb{Q}$ -automaton $\mathcal{A}$ is defined by

$$\mathcal{F}_{\mathcal{A}} = \text{span}(\{\alpha \cdot M(w) \mid w \in \Sigma^*\}) .$$

Because $\mathcal{F}_{\mathcal{A}}$ is a subspace of $\mathbb{Q}^n$, the dimension of the vector space $\mathcal{F}_{\mathcal{A}}$ is bound by $\dim(\mathcal{F}_{\mathcal{A}}) \leq n$. Hence, $\mathcal{F}_{\mathcal{A}}$ has a base $\mathcal{B}_{\mathcal{A}}$ consisting of $\dim(\mathcal{F}_{\mathcal{A}})$ many vectors. Notably, for each base $b \in \mathcal{B}_{\mathcal{A}}$, there is a word w_b of length $|w_b| \leq \dim(\mathcal{F}_{\mathcal{A}}) - 1$ such that $b = \alpha \cdot M(w_b)$. One computes a base $\mathcal{B}_{\mathcal{A}}$ of $\mathcal{F}_{\mathcal{A}}$ by initially set $\mathcal{B}_{\mathcal{A}} = \{\alpha\}$ and $W = \Sigma$. In each iteration dequeue $w_b \in W$ and if $b = \alpha \cdot M(w_b)$ is linear independent to $\mathcal{B}_{\mathcal{A}}$, add b to $\mathcal{B}_{\mathcal{A}}$ and set $W = W \cup \{w_b a \mid a \in \Sigma\}$. All operations are feasible in polynomial time; linear independence can be tested by computing the reduced row echelon form [32]. Given $\mathcal{B}_{\mathcal{A}}$, we decide in polynomial time the emptiness problem for $\mathcal{A}$, i. e., decide if $\llbracket \mathcal{A} \rrbracket(w) = 0$ holds for all $w \in \Sigma^*$. This can be done by checking for each vector $b \in \mathcal{B}_{\mathcal{A}}$ if $b \cdot \eta = 0$ holds, i. e., all bases are orthogonal to the final vector η . Because matrix multiplication can be computed in polynomial time and because $\mathcal{B}_{\mathcal{A}}$ has at most n elements, checking for emptiness is also in polynomial time.

For computing $\mathbb{Q}$ -EQUIVALENCE in **PTIME**, one employs the method of checking emptiness from above. First, construct (in polynomial time) from the input $\mathcal{A}_1$ and $\mathcal{A}_2$ an automaton $\mathcal{A}_-$ which runs $\mathcal{A}_1$ and $\mathcal{A}_2$ in parallel and negates the initial values of $\mathcal{A}_2$. When $\langle \alpha_i, M_i, \eta_i \rangle$ is the linear representation of $\mathcal{A}_i$, the linear representation $\langle \alpha_-, M_-, \eta_- \rangle$ of $\mathcal{A}_-$ is

$$\alpha_- = [\alpha_1 \quad -\alpha_2] \quad M_-(a) = \begin{bmatrix} M_1(a) & 0 \\ 0 & M_2(a) \end{bmatrix} \quad \eta_- = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}$$

where $\alpha_- \in \mathbb{Q}^{1 \times (|Q_1| + |Q_2|)}$, $\eta_- \in \mathbb{Q}^{(|Q_1| + |Q_2|) \times 1}$ and $M_-(a) \in \mathbb{Q}^{(|Q_1| + |Q_2|) \times (|Q_1| + |Q_2|)}$ for each $a \in \Sigma$. The automaton $\mathcal{A}_-$ has $|Q_1| + |Q_2|$ states. By construction, we have $\llbracket \mathcal{A}_1 \rrbracket = \llbracket \mathcal{A}_2 \rrbracket$ exactly when $\mathcal{A}_-$ is empty, i.e., $\llbracket \mathcal{A}_- \rrbracket(w) = 0$ holds for all $w \in \Sigma^*$. We check emptiness of $\mathcal{A}_-$ as described above, i.e., by computing the base $\mathcal{B}_-$ of the forward-space $\mathcal{F}_{\mathcal{A}_-}$ and check then orthogonality of $\mathcal{B}_-$ with η_- . When $\mathcal{A}_-$ is non-empty, above procedure guarantees a word $w_b \in \Sigma^*$ of length $|w_b| \leq |Q_1| + |Q_2| - 1$ that witnesses $\llbracket \mathcal{A}_1 \rrbracket \neq \llbracket \mathcal{A}_2 \rrbracket$.

Note that the **NC** complexity of $\mathbb{Q}$ -EQUIVALENCE is a rather surprising phenomenon, as similar problems are usually much harder, e. g., **PSPACE** [33, 20] or even undecidable [34].

6. Complexity of EQUALWALKCOUNT

In this section, we will show how the problem of deciding whether two vertices v, v' agree on the numbers of walks, i.e., if for each length the number of walks of that length that end in each vertex is the same (EQUALWALKCOUNT), can be reduced to the equivalence of semiring-automata over $\mathbb{Q}$ ($\mathbb{Q}$ -EQUIVALENCE), yielding a **PTIME** complexity.

The central insight of this section is that one can construct for a vertex $v \in V$ of a graph $G = (V, E)$ an $\mathbb{Q}$ -automaton $\mathcal{A}_G^v$ over a unary alphabet, e.g., $\Sigma = \{s\}$, such that the formal series $\llbracket \mathcal{A}_G^v \rrbracket$ yields the numbers of walks of length i that end in v for the word s^i . Formally, for a graph $G = (V, E)$ and a vertex $v \in V$, we define the following $\mathbb{Q}$ -automaton $\mathcal{A}_G^v = (V, \Sigma, \delta, I, F)$ with:

$$\begin{aligned} \Sigma &= \{s\} & I(q) &= 1 \\ \delta(q, a, p) &= \begin{cases} 1 & \text{if } (q, p) \in E \\ 0 & \text{if } (q, p) \notin E \end{cases} & F(q) &= \begin{cases} 1 & q = v \\ 0 & q \neq v \end{cases} \end{aligned}$$

The states of the automaton $\mathcal{A}_G^v$ are the vertices V , the alphabet $\Sigma = \{s\}$ is unary, and v is the only state to which we assign a non-zero weight. The transition function is (quintessentially) the edge relation of the graph G , i. e., the transition from v to v' when reading a has weight 1 when there is an edge from v to v' in G ; otherwise the weight is 0.

Lemma 6.1. *For each Graph $G = (V, E)$, each vertex $v \in V$ and each $i \in \mathbb{N}$ holds $T_i(v) = \mathcal{A}_G^v(v, s^i)$.*

Proof. We start by showing for each $q \in V$ and $i \in \mathbb{N}$ holds:

$$T_i(q) = \sum_{p \in V} \delta^*(p, s^i, q)$$

The proof is by induction. For these case of $i = 1$, we have

$$T_1(q) = |N(q)| = |\{p \in V \mid (p, q) \in E\}| = \sum_{p \in V} \begin{pmatrix} 1 & , \text{ if } (p, q) \in E \\ 0 & , \text{ if } (p, q) \notin E \end{pmatrix} = \sum_{p \in V} \delta(p, s, q) = \sum_{p \in V} \delta^*(p, s, q)$$

For the case of $i > 1$, we have

$$\begin{aligned} \sum_{p \in V} \delta^*(p, s^i, q) &= \sum_{p \in V} \sum_{r \in V} \delta^*(p, s^{i-1}, r) \cdot \delta(r, s, q) = \sum_{p \in V} \sum_{r \in V} \begin{pmatrix} \delta^*(p, s^{i-1}, r) & , (r, q) \in E \\ 0 & , (r, q) \notin E \end{pmatrix} \\ &= \sum_{p \in V} \sum_{r \in N(q)} \delta^*(p, s^{i-1}, r) = \sum_{r \in N(q)} \sum_{p \in V} \delta^*(p, s^{i-1}, r) = \sum_{r \in N(q)} T_{i-1}(r) = T_i(q) \end{aligned}$$

Finally, we show $T_i(v) = \mathcal{A}_G^v(v, s^i)$:

$$\mathcal{A}_G^v(v, s^i) = \sum_{p \in V} I(p) \cdot \delta^*(p, s^i, v) \cdot F(v) = \sum_{p \in V} 1 \cdot \delta^*(p, s^i, v) \cdot 1 = \sum_{p \in V} \delta^*(p, s^i, v) = T_i(v)$$

This completes the proof of Lemma 6.1. □

We obtain the following theorem by combining Proposition 5.2 and Lemma 6.1.

Theorem 6.2. *The following holds:*

(a) $\text{EQUALWALKCOUNT} \leq_p \mathbb{Q}\text{-EQUIVALENCE}$

(b) EQUALWALKCOUNT is in **P**TIME.

Proof. Clearly, (b) follows from (a) and Proposition 5.2. For (a), we consider the construction of $\mathcal{A}_G^v$. One constructs $\mathcal{A}_G^v$ easily in polynomial time in the size of the graph $G = (V, E)$. Lemma 6.1 guarantees the following property:

$$\left(T_i(v) = T_i(v') \text{ for all } i \in \mathbb{N} \right) \Leftrightarrow \left(\mathcal{A}_G^v(v, s^i) = \mathcal{A}_G^{v'}(v', s^i) \text{ for all } i \in \mathbb{N} \right)$$

Now note that the following property holds:

$$\left(\mathcal{A}_G^v(v, s^i) = \mathcal{A}_G^{v'}(v', s^i) \text{ for all } i \in \mathbb{N} \right) \Leftrightarrow \llbracket \mathcal{A}_G^v \rrbracket = \llbracket \mathcal{A}_G^{v'} \rrbracket$$

This is because u is in $\mathcal{A}_G^u$ the only state non-zero final value for $u \in \{v, v'\}$. Moreover, because $\mathcal{A}_G^v$ and $\mathcal{A}_G^{v'}$ operate on the unary alphabet $\Sigma = \{s\}$, for each $i \in \mathbb{N}_0$ the word s^i is the one and only word in Σ^i . Due to the above insights,

$$(G, v, v') \mapsto (\mathcal{A}_G^v, \mathcal{A}_G^{v'})$$

is a polynomial-time many-one reduction from EQUALWALKCOUNT to $\mathbb{Q}$ -EQUIVALENCE. □

By combining Theorem 6.2, respectively Lemma 6.1, with Proposition 5.3, we obtain the following central insight.

Theorem 6.3. *For every graph $G = (V, E)$ and all vertices $v, v' \in V$ that do not agree on the number of walks, it holds*

$$\min\{ i \in \mathbb{N} \mid |\mathcal{R}_i(v)| \neq |\mathcal{R}_i(v')| \} \leq 2|V| - 1.$$

Intuitively, Theorem 6.3 states that the size of the smallest witness for non-agreement on the number of walks is bound linear in the amount of vertices.

7. Complexity of Discussion-Based Semantics

In this section, we settle the complexity of the discussion-based semantics. Specifically, we show that EQUIVDis and STRONGERDis are decidable in polynomial time. The preceded sections provide the following sequence of reductions:

$$\text{EQUIVDis} \leq_p \text{EQUALWALKCOUNT} \leq_p \mathbb{Q}\text{-EQUIVALENCE}$$

By employing Proposition 5.2, we obtain from Theorem 4.1 and Theorem 6.2 decidability within polynomial time.

Theorem 7.1. *EQUIVDis is in PTIME.*

In the following example, we carry out the line of reductions and show how the final result is computed.

Example 7.2. Consider the AAF

$$F_{\text{joint}} = (A_2 \cup A_3, R_2 \cup R_3)$$

from Example 3.7, which is the union of $F_2 = (A_2, R_2)$ from Example 3.6 and $F_3 = (A_3, R_3)$ from Example 3.7. We show that $a \simeq_F^{\text{dbs}} h$ holds by carrying out the reduction of the problem to $\mathbb{Q}$ -EQUIVALENCE.

The input of EQUIVDis is (F_{joint}, a, h) and as described in Theorem 4.1 the reduction to EQUALWALKCOUNT is the identity function, yielding the same tuple. Following Theorem 6.2, the reduction $\text{EQUALWALKCOUNT} \leq_p \mathbb{Q}\text{-EQUIVALENCE}$ computes for (F_{joint}, a, h) the two $\mathbb{Q}$ -automata $\mathcal{A}_{\text{joint}}^a = \mathcal{A}_{F_{\text{joint}}}^a$ and $\mathcal{A}_{\text{joint}}^h = \mathcal{A}_{F_{\text{joint}}}^h$. As a is the only accepting state in $\mathcal{A}_{\text{joint}}^a$, it is sufficient to consider only the connected component of $\mathcal{A}_{\text{joint}}^a$ which contains a , which we denote by $\mathcal{A}_a$. Analogously, we denote with $\mathcal{A}_h$ the restriction of $\mathcal{A}_{\text{joint}}^h$ to the connected component which contains h . The automata $\mathcal{A}_a$ and $\mathcal{A}_h$ are presented in Figure 4. Following the line of reductions in Theorem 4.1 and Theorem 6.2, we have $a \simeq_F^{\text{dbs}} h$ exactly when $\mathcal{A}_a$ and $\mathcal{A}_h$ are equivalent.

Next, we decide equivalence of $\mathcal{A}_a$ and $\mathcal{A}_h$. For that, we compute an automaton $\mathcal{A}_-$, that is empty exactly when $\mathcal{A}_a$ and $\mathcal{A}_h$ are equivalent. The construction of $\mathcal{A}_-$ is given in Section 5.2, and the resulting linear representation $\langle \alpha_-, M_-, \eta_- \rangle$ of the automaton $\mathcal{A}_-$ is given in Figure 4. Basically, $\mathcal{A}_-$ is the union of $\mathcal{A}_a$ and $\mathcal{A}_h$ where the initial values of states of $\mathcal{A}_-$ is -1 instead of 1 . The base $\mathcal{B}_- = \{b_1, b_2, b_3, b_4\}$ of the forward space $\mathcal{F}_- = \mathcal{F}_{\mathcal{A}_-} = \text{span}(\{\alpha_- \cdot M_-(w) \mid w \in \Sigma^*\})$ is:

	a	b	c	d	e	f	g	h	j	k	l	m	n	o
$b_1 =$	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1
$b_2 =$	2	2	2	1	1	1	1	-2	-1	-3	-1	-1	-1	-1
$b_3 =$	4	2	2	2	2	2	2	-4	-1	-3	-2	-2	-2	-2
$b_4 =$	4	4	4	4	4	4	4	-4	-2	-6	-4	-4	-4	-4

For each base b_i , with $i \in \{1, 2, 3, 4\}$, the word $w_i = s^{i-1}$ is such that $b_i = \alpha_- \cdot M_-(w_i)$. Recall that η_- is orthogonal to $\mathcal{F}_-$ exactly when for all bases $b \in \mathcal{B}_-$ holds $b \cdot \eta_- = 0$. This is the case, as $b \cdot \eta_- = 0$ holds for all bases $b \in \mathcal{B}_-$. Because η_- is orthogonal to $\mathcal{F}_-$, we obtain that $\mathcal{A}_-$ is empty. Consequently, $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent and we obtain $a \simeq_F^{\text{dbs}} h$.

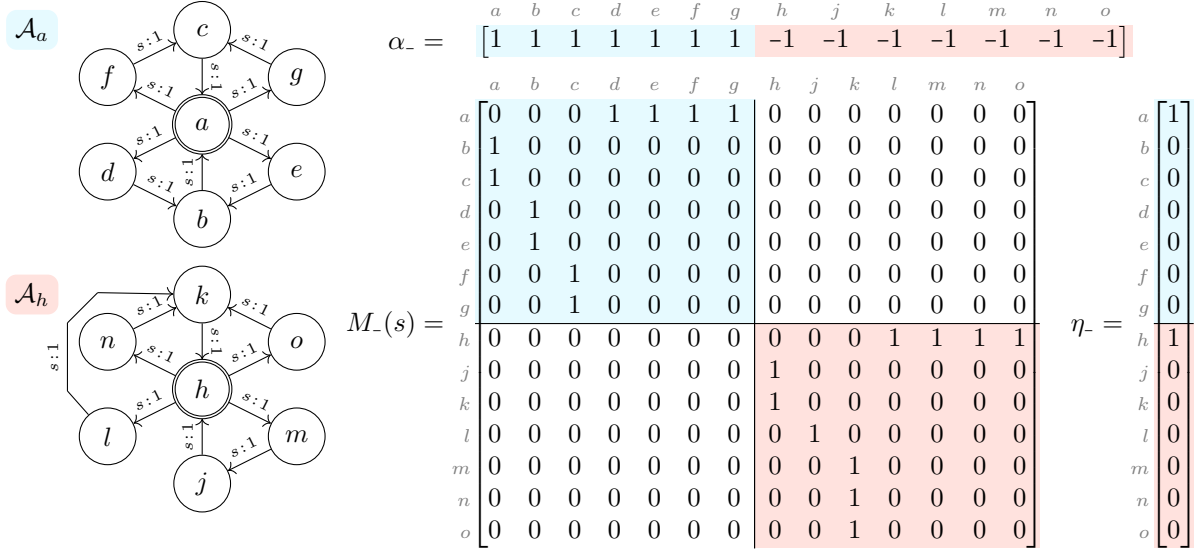


Figure 4: Automata $\mathcal{A}_a$ and $\mathcal{A}_h$ (left) and the linear representation $\langle \alpha_-, M_-, \eta_- \rangle$ of the automaton $\mathcal{A}_-$ (right) from Example 7.2. *Left side:* An edge with label $s:1$ indicates that transitioning from the source to the target when reading s has weight 1. Edges with weight zero are omitted. Initial weights and final weights of the states are not explicitly given; each state has an initial weight of 1, and each state has a final weight of 0, except for double-circled states, which have a final weight of 1 (“accepting state”).

A careful reconsideration of the above sequence of reductions from EQUIVDis to $\mathbb{Q}$ -EQUIVALENCE reveals that whenever for two arguments a, b their discussion count differs, then the smallest index in which the infinite vectors $Dis^F(a)$ and $Dis^F(b)$ are different is bound linear in the size of F .

Theorem 7.3. *Let $F = (A, R)$ be an AAF and let a, b be two arguments of F . If $a \not\sim_F^{dbs} b$ holds, then there is some $i \in \mathbb{N}$ with $i \leq 2|A| - 1$ such that $Dis_i^F(a) \neq Dis_i^F(b)$.*

Proof. Due to Theorem 4.1, we have that $a \not\sim_F^{dbs} b$ holds exactly when a and b do not agree on the number of walks. For the latter, due to Theorem 6.3, the smallest index i , such that $|\mathcal{R}_i(a)| \neq |\mathcal{R}_i(b)|$ holds is bound by $2|A| - 1$. $\square$

We consider Theorem 7.3 and Theorem 6.3 as one of the most central insights of this paper. First, because there seems to be no easy way to prove the bounds that are provided by these theorems. Moreover, Theorem 7.3 enables us to decide STRONGERDis in polynomial time. Such an algorithm for deciding STRONGERDis is sketched in Algorithm 1.

Theorem 7.4. *STRONGERDis is in PTIME.*

Proof. Matrix multiplication is known to be in polynomial time [35]. Hence, because M is of size $|A| \times |A|$, all checks and multiplications in Algorithm 1 are computable in polynomial time in $|A|$. The central loop is executed at most $2|A| - 1$ times, rendering the whole runtime of Algorithm 1 to be in polynomial time. Correctness of Algorithm 1 follows from Proposition 4.3 and Theorem 7.3. $\square$

Example 7.5. Let $F_4 = (A_4, R_4)$ be the AAF with

$$\begin{aligned} A_4 &= \{a, b, c, d\} \\ R_4 &= \{(a, d), (b, a), (b, d), (c, a), (c, b), (c, c), (d, b)\}, \end{aligned}$$

depicted in Figure 5 together with its adjacency matrix M . We show that $a \succeq_{F_4}^{dbs} b$ holds by executing Algorithm 1. Algorithm 1 computes the respective matrix power M^i of the adjacency matrix succinctly. This is because Proposition 4.3 guarantees that $M^i[v]$, i.e., the sum of the respective column vector

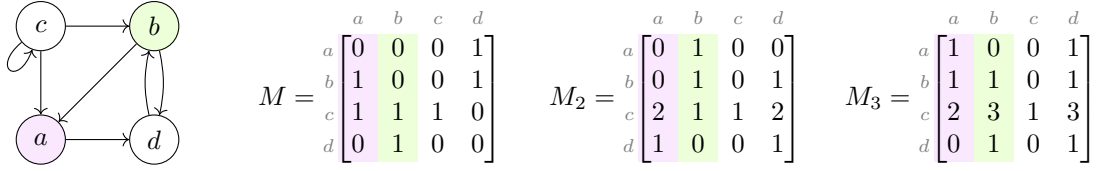


Figure 5: AAF F_4 from Example 7.5 and its adjacency matrix.

Algorithm 1: Algorithm for STRONGERDIS.

Input: AAF $F = (A, R)$ and arguments $a, b \in A$.
Output: Yes, if $a \succeq_F^{\text{dbs}} b$ holds; otherwise, No.
 $i \leftarrow 1$; $M \leftarrow \text{AdjacencyMatrix}(F)$;
while $i \leq 2|A| - 1$ **do**
 if $M[a] < M[b]$ **and** i **is odd** **then return** Yes;
 if $M[b] < M[a]$ **and** i **is even** **then return** Yes;
 if $M[b] \neq M[a]$ **then return** No;
 $M \leftarrow M \cdot \text{AdjacencyMatrix}(F)$;
 $i \leftarrow i + 1$;
return Yes;

Algorithm 2: Algorithm for EQUIVDis.

Input: AAF $F = (A, R)$ and arguments $a, b \in A$.
Output: Yes, if $a \simeq_F^{\text{dbs}} b$ holds; otherwise, No.
 $i \leftarrow 1$; $M \leftarrow \text{AdjacencyMatrix}(F)$;
while $i \leq 2|A| - 1$ **do**
 if $M[b] \neq M[a]$ **then return** No;
 $M \leftarrow M \cdot \text{AdjacencyMatrix}(F)$;
 $i \leftarrow i + 1$;
return Yes;

of v , equals $|\mathcal{R}_i(v)|$. The powers M^2 and M^3 are given in Figure 5. As one can check we have $M[a] = 2 = M[b]$ and $M^2[a] = 3 = M^2[b]$. When the central loop of Algorithm 1 is executed the third time, we obtain $M^3[a] = 4 < 5 = M^3[b]$ and, because 3 is odd, Algorithm 1 returns Yes. This corresponds to $a \succeq_{F_4}^{\text{dbs}} b$, which holds because we have:

$$\text{Dis}^{F_4}(a) = \langle -2, 3, -4, \dots \rangle \geq^{\text{lex}} \langle -2, 3, -5, \dots \rangle = \text{Dis}^{F_4}(b)$$

With Theorem 7.4 established, one easily adapts Algorithm 1 to Algorithm 2 which computes EQUIVDis. Finally, the reduction from Section 6 also shows $\text{EQUALWALKCOUNT} \leq_p \mathbb{Z}\text{-EQUIVALENCE}$, whereby the latter is the equivalence problem of semiring automata over the integers, which is also in PTIME [21]. However, there is no analogue to Proposition 5.3 known for $\mathbb{Z}$ and hence, using $\mathbb{Q}$ is vital to identify the complexity of STRONGERDIS.

8. Conclusion

We settled a computational question that had long been unanswered, namely the question of decidability and computational complexity of the discussion-based semantics in abstract argumentation. We showed that deciding whether two arguments are ranked equally or whether one argument is ranked strictly better than the other are both decidable in polynomial time. In particular, we showed that it suffices to explore walks up to length $2|A| - 1$ in order to correctly classify the relation of two arguments, thus providing the means to prove correctness for algorithms such as the one from [4]. Our result made use of insights from automata theory, in particular regarding the question of equivalence of automata over semirings. Thus, our result provides a means to link abstract argumentation research to graph theory and automata theory.

Although we focused our analysis on the discussion-based semantics of Amgoud and Ben-Naim [2], the results of this work are relevant for a broad range of ranking-based semantics, in particular those that are defined on lengths of walks. For example, the *burden-based* semantics [2], the larger family of *propagation-based* semantics [17], and the *tuples* approach [18] all are based on (information flowing through) walks and values determined from these walks. In particular, all these semantics highly depend on numbers of walks of different lengths. For future work, we aim to generalise the result from this paper to these approaches.

Acknowledgments

We thank Stefan Göller, who gave the initial impulse to consider the equivalence of formal power series to address the questions considered here. We also thank the grammar enthusiast Zeynep G. Saribatur for spotting an error.

Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

- [1] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence* 77 (1995) 321–358.
- [2] L. Amgoud, J. Ben-Naim, Ranking-based semantics for argumentation frameworks, in: *Proceedings of the 17th International Conference on Scalable Uncertainty Management (SUM 2013)*, volume 8078 of *LNCS*, Springer, 2013, pp. 134–147.
- [3] E. Bonzon, J. Delobelle, S. Konieczny, N. Maudet, A comparative study of ranking-based semantics for abstract argumentation, in: *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*, 2016, pp. 914–920.
- [4] E. Bonzon, J. Delobelle, S. Konieczny, N. Maudet, Explaining arguments', *Journal of Applied Non-Classical Logics* 33 (2023) 328–386.
- [5] X. Yin, N. Potyka, F. Toni, Explaining arguments' strength: Unveiling the role of attacks and supports, in: *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI 2024)*, 2024, pp. 3622–3630.
- [6] N. Potyka, Open-mindedness of gradual argumentation semantics, in: *Proceedings of the 13th International Conference on Scalable Uncertainty Management (SUM 2019)*, volume 11940 of *LNCS*, Springer, 2019, pp. 236–249.
- [7] P. Baroni, A. Rago, F. Toni, From fine-grained properties to broad principles for gradual argumentation: A principled spectrum, *International Journal of Approximate Reasoning* 105 (2019) 252–286.
- [8] D. Grossi, S. Modgil, On the graded acceptability of arguments in abstract and instantiated argumentation, *Artificial Intelligence* 275 (2019) 138–173.
- [9] M. Thimm, F. Cerutti, T. Rienstra, Probabilistic graded semantics, in: *Proceedings of the 17th International Conference on Computational Models of Argumentation (COMMA 2018)*, volume 305 of *Frontiers in Artificial Intelligence and Applications*, 2018, pp. 369–380.
- [10] L. Blümel, M. Thimm, A ranking semantics for abstract argumentation based on serialisability, in: *Proceedings of the 9th International Conference on Computational Models of Argument (COMMA 2022)*, 2022, pp. 104–115.
- [11] K. Skiba, T. Rienstra, M. Thimm, J. Heyninck, G. Kern-Isberner, Ranking extensions in abstract argumentation, in: *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI 2021)*, 2021, pp. 2047–2053.
- [12] K. Skiba, M. Thimm, Towards ranking-based semantics for abstract argumentation using conditional logic semantics, in: *Proceedings of the International Workshop on Computational Argumentation and Cognition (COGNITAR 2020)*, 2020.
- [13] J.-G. Mailly, J. Rossit, Argument, i choose you! preferences and ranking semantics in abstract argumentation, in: *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020)*, 2020, pp. 647–651.

- [14] E. Bonzon, J. Delobelle, S. Konieczny, N. Maudet, Combining extension-based semantics and ranking-based semantics for abstract argumentation, in: Proceedings of the Sixteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2018), 2018, pp. 118–127.
- [15] P. Dondio, Ranking semantics based on subgraphs analysis, in: Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), 2018, pp. 1132–1140.
- [16] B. Yun, S. Vesic, M. Croitoru, P. Bisquert, Viewpoints using ranking-based argumentation semantics, in: Proceedings of the 7th International Conference on Computational Models of Argumentation (COMMA 2018), 2018, pp. 381–392.
- [17] E. Bonzon, J. Delobelle, S. Konieczny, N. Maudet, Argumentation ranking semantics based on propagation, in: Proceedings of the 6th International Conference on Computational Models of Argument (COMMA 2016), 2016, pp. 139–150.
- [18] C. Cayrol, M.-C. Lagasquie-Schiex, Graduality in argumentation, *Journal of Artificial Intelligence Research* 23 (2005) 245–297.
- [19] J. Delobelle, Ranking-based Semantics for Abstract Argumentation, Ph.D. thesis, Centre de Recherche en Informatique de Lens, 2017.
- [20] W. Tzeng, On path equivalence of nondeterministic finite automata, *Information Processing Letters* 58 (1996) 43–46.
- [21] M. Béal, S. Lombardy, J. Sakarovitch, On the equivalence of $\mathbb{Z}$ -automata, in: Automata, Languages and Programming, 32nd International Colloquium (ICALP 2005), volume 3580 of *LNCS*, Springer, 2005, pp. 397–409.
- [22] J. E. Hopcroft, R. Motwani, J. D. Ullman, Introduction to automata theory, languages, and computation, 3rd Edition, Pearson international edition, Addison-Wesley, 2007.
- [23] S. Arora, B. Barak, Computational Complexity - A Modern Approach, Cambridge University Press, 2009.
- [24] P. Flajolet, R. Sedgewick, Analytic Combinatorics, Cambridge University Press, 2009.
- [25] G. Tinhofer, R. Albrecht, E. Mayr, H. Noltemeier, M. M. Syslo (Eds.), Computational Graph Theory, Computing Supplementa, Springer, 1990.
- [26] A. Salomaa, M. Soittola, Automata-Theoretic Aspects of Formal Power Series, Texts and Monographs in Computer Science, Springer, 1978.
- [27] J. Berstel, C. Reutenauer, Rational series and their languages, volume 12 of *EATCS monographs on theoretical computer science*, Springer, 1988.
- [28] M. O. Rabin, D. Scott, Finite automata and their decision problems, *IBM Journal of Research and Development* 3 (1959) 114–125.
- [29] M. P. Schützenberger, On the definition of a family of automata, *Inf. Control.* 4 (1961) 245–270.
- [30] Z. Ésik, A. Maletti, Simulation vs. equivalence, in: Proceedings of the 2010 International Conference on Foundations of Computer Science (FCS 2010), CSREA Press, 2010, pp. 119–124.
- [31] S. Kiefer, Notes on equivalence and minimization of weighted automata, CoRR abs/2009.01217 (2020). [arXiv:2009.01217](https://arxiv.org/abs/2009.01217).
- [32] J. Edmonds, Systems of distinct representatives and linear algebra, *Journal of Research of the National Bureau of Standards* 71B (1967) 241–245.
- [33] L. J. Stockmeyer, A. R. Meyer, Word problems requiring exponential time: Preliminary report, in: Proceedings of the 5th Annual ACM Symposium on Theory of Computing (STOC 1973), ACM, 1973, pp. 1–9.
- [34] J. Sakarovitch, Rational and Recognisable Power Series, Springer, 2009, pp. 105–174.
- [35] V. Strassen, Gaussian elimination is not optimal, *Numerische Mathematik* 13 (1969) 354–356.