Moscow Journal of Combinatorics and Number Theory

2020
vol. 9 no. 3

msp

Moscow Journal of Combinatorics and Number Theory

msp.org/moscow

See inside back cover or msp.org/moscow for submission instructions.

The subscription price for 2020 is US $310/year for the electronic version, and $365/year (+$20, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues and changes of subscriber address should be sent to MSP.

Moscow Journal of Combinatorics and Number Theory (ISSN 2640-7361 electronic, 2220-5438 printed) at Mathematical Sciences Publishers, 798 Evans Hall #3840, c/o University of California, Berkeley, CA 94720-3840 is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

MJCNT peer review and production are managed by EditFlow® from MSP.

PUBLISHED BY
mathematical sciences publishers
nonprofit scientific publishing
http://msp.org/

Moscow Journal of Combinatorics and Number Theory
Vol. 9, No. 3, 2020
https://doi.org/10.2140/moscow.2020.9.203

Provenance analysis for logic and games

Erich Grädel and Val Tannen

A model-checking computation checks whether a given logical sentence is true in a given finite structure. Provenance analysis abstracts from such a computation mathematical information on how the result depends on the atomic data that describe the structure. In database theory, provenance analysis by interpretations in commutative semirings has been rather successful for positive query languages (such as unions of conjunctive queries, positive relational algebra, and Datalog). However, it did not really offer an adequate treatment of negation or missing information. Here we propose a new approach for the provenance analysis of logics with negation, such as first-order logic and fixed-point logics. It is closely related to a provenance analysis of the associated model-checking games, and based on new semirings of dual-indeterminate polynomials or dual-indeterminate formal power series. These are obtained by taking quotients of traditional provenance semirings by congruences that are generated by products of positive and negative provenance tokens. Beyond the use for model-checking problems in logics, provenance analysis of games is of independent interest. Provenance values in games provide detailed information about the number and properties of the strategies of the players, far beyond the question whether or not a player has a winning strategy from a given position.

1. Introduction

Provenance analysis aims at understanding how the result of a computational process with a complex input, consisting of multiple items, depends on the various parts of this input. In database theory, provenance analysis based on interpretations in commutative semirings has been developed for positive database query languages, to understand which combinations of the atomic facts in a database can be used for deriving the result of a given query. In this approach, atomic facts are interpreted not just by true or false, but by values in an appropriate semiring, where 0 is the value of false statements, whereas any element $a \neq 0$ of the semiring stands for some shade of truth. These values are then propagated from the atomic facts to arbitrary queries in the language, which permits one to answer questions such as the minimal cost of a query evaluation, the confidence one can have that the result is true, the number of different ways in which the result can be computed, or the clearance level that is required for obtaining the output, under the assumption that some facts are labeled as confidential, secret, top secret, etc. We refer to [Green and Tannen 2017] for a recent account and many references on the semiring framework for database provenance.

Scenarios to which the semiring provenance approach has been successfully applied include unions of conjunctive queries, positive relational algebra, nested relations, Datalog, XQuery, SQL-aggregates and several others, and it has been implemented in software systems such as Orchestra and Propolis.

MSC2010: 03B70, 03C13, 05C57, 68Q19, 91A43.
Keywords: finite model theory, semiring provenance, games.

For details, see, e.g., [Amsterdamer et al. 2011b; Deutch et al. 2014; Foster et al. 2008; Green 2011; Green et al. 2007; Tannen 2013]. A main limitation of this approach is that is has been largely confined to *positive* query languages. Attempts to add operations that capture *difference of relations* have led to interesting and algebraically challenging, but divergent approaches [Amsterdamer et al. 2011a; Geerts and Poggi 2010; Geerts et al. 2016; Green et al. 2009]. In particular there has been no systematic approach in database theory for tracking *negative information*, and no convincing provenance analysis for languages with full negation.

Here, we would like to develop a new approach for a semiring provenance analysis for model-checking problems of logics with negation, in particular first-order logic and fixed-point logic. This approach is based on several ideas:

• Provenance analysis of logics is intimately connected to provenance analysis of games. In the same way as formula evaluation or model checking can be formulated in game-theoretic terms, also the propagation of provenance values from atomic facts to arbitrary formulae can be viewed as a process on the associated games. Also the typical *results* of a provenance analysis of database queries or logical formulae, concerning for instance confidence scores, costs, required clearance level, or number of "proof trees" have natural game-theoretic interpretations. In fact, provenance analysis of games is of independent interest, and provenance values of positions in a game provide detailed information about the number and properties of the strategies of the players, far beyond the question whether or not a player has a winning strategy from a given position.

• We deal with negation by transformation to negation normal form. This is the common approach for the design of model-checking games and game-based evaluation algorithms. But while this is mainly a matter of convenience (to avoid role switches between players during a play), provenance semantics imposes even stronger reasons for transformations to negation normal form. Indeed, beyond Boolean semantics, negation is not a compositional logical operation: the provenance value of $\neg\varphi$ is not necessarily determined by the provenance value of φ.

• On the algebraic side, we introduce new provenance semirings of polynomials and formal power series, which take negation into account. They are obtained by taking quotients of traditional provenance semirings by congruences generated by products of positive and negative provenance tokens; they are called semirings of dual-indeterminate polynomials or dual-indeterminate power series.

Preliminary accounts of our approach, confined to first-order logic and without the connection to games, but discussing potential applications to issues such as model updates, and reverse provenance analysis (e.g., confidence maximization), have been given in [Tannen 2017; Grädel and Tannen 2017]. Here we put also the provenance analysis of games into focus; in fact we develop our approach here from the perspectives of games. We shall first discuss the case of finite acyclic games which are sufficient for the provenance analysis of first-order logic and its fragments. Most of the central issues of our approach, in particular the view of provenance values in terms of valuations of strategies and plays, appear already in this simple scenario. We shall then discuss reachability games on graphs that admit cycles. These are the games that are relevant for the provenance analysis of logics with least (but without greatest) fixed points. For these it will be necessary to restrict from arbitrary commutative semirings to ω-continuous ones. Such an analysis has previously been carried out for Datalog, but to deal with (atomic) negation we have to combine this with the idea of taking quotients by the duality on indeterminates, which will lead us to semirings of dual-indeterminate power series. Finally we shall outline a provenance approach for safety games and greatest fixed points. Our central algebraic tools here are absorptive semirings, in particular the semiring $\mathbb{S}^\infty[X]$ of generalized absorptive polynomials, admitting also infinite exponents.

This paper is intended to lay foundations for our general approach to a provenance analysis of logic and games, which should take us far beyond the specific cases studied here. The application of the acyclic case to modal and guarded logics has been analyzed in [Dannert and Grädel 2020]. In [Xu et al. 2018] our approach has been applied to database repairs; it has been shown how our treatment of negation, or absent information, can be used to explain and repair missing query answers and the failure of integrity constraints in databases. Further, the potential of the provenance methods developed here for applications in knowledge representation and description logics has been discussed in [Dannert and Grädel 2019]. Work in progress includes the provenance analysis of temporal and dynamic logics in the setting of absorptive semirings, the study of logics of dependence and independence from the point of view of provenance, and the algorithmic analysis of computing provenance values in various settings.

2. Commutative semirings

Definition 1. A *commutative semiring* is an algebraic structure $(K, +, \cdot, 0, 1)$, with $0 \neq 1$, such that $(K, +, 0)$ and $(K, \cdot, 1)$ are commutative monoids, $\cdot$ distributes over $+$, and $0 \cdot a = a \cdot 0 = 0$. A semiring is $+$*-positive* if $a + b = 0$ implies $a = 0$ and $b = 0$. This excludes rings. A semiring is *root-integral* if $a \cdot a = 0$ implies $a = 0$. All semirings considered in this paper are commutative, +-positive, and root-integral. Further, a commutative semiring is *positive* if it is +-positive and has no divisors of 0 (i.e., $a \cdot b = 0$ implies $a = 0$ and $b = 0$). The standard semirings considered in provenance analysis are in fact positive, but for an appropriate treatment of negation we shall introduce later in this paper semirings (of dual-indeterminate polynomials or power series) that have divisors of 0.

Notice that a semiring K is positive if, and only if, the unique function $h : K \to \{0, 1\}$ with $h^{-1}(0) = \{0\}$ is a homomorphism from K into the Boolean semiring $\mathbb{B} = (\{0, 1\}, \vee, \wedge, 0, 1)$. A semiring K is (+)-idempotent if $a + a = a$ for all $a \in K$ and $(+, \cdot)$-idempotent if, in addition, $a \cdot a = a$ for all a. Further, K is *absorptive* if $a + ab = a$ for all $a, b \in K$. Obviously, every absorptive semiring is (+)-idempotent.

Elements of a commutative semiring will be used as truth values for logical statements and as values for positions in games. The intuition is that $+$ describes the *alternative use* of information, as in disjunctions or existential quantifications, or for different possible choices of a player in a game, whereas $\cdot$ stands for the *joint use* of information, as in conjunctions or universal quantifications, or for choices in a game that are controlled by the opponent of the given player. Further, 0 is the value of false statements or losing positions, whereas any element $a \neq 0$ of a semiring K stands for a "nuanced" interpretation of true or as a value of a nonlosing position.

Application semirings. We briefly discuss some specific semirings that provide interesting information about a logical statement or a position in a game:

- The *Boolean semiring* $\mathbb{B} = (\{0, 1\}, \vee, \wedge, 0, 1)$ is the standard habitat of logical truth.
- $\mathbb{N} = (\mathbb{N}, +, \cdot, 0, 1)$ is used here for counting winning strategies in games. It also plays an important role for *bag semantics* in databases.

• $\mathbb{T} = (\mathbb{R}_+^\infty, \min, +, \infty, 0)$ is called the *tropical* semiring. It has many applications in several areas of computer science. It is used here for measuring the cost of strategies.

• The *Viterbi* semiring $\mathbb{V} = ([0, 1], \max, \cdot, 0, 1)$ is isomorphic to $\mathbb{T}$ via $x \mapsto e^{-x}$ and $y \mapsto -\ln y$. We will think of the elements of $\mathbb{V}$ as *confidence scores* and use it to describe the confidence that a player can win from a given position or the confidence assigned to a logical statement.

• The *min-max* semiring on a totally ordered set $(A, \leq)$ with least element a and greatest element b is the semiring $(A, \max, \min, a, b)$.

Provenance semirings. Beyond the traditional application semirings, there are some important provenance semirings of polynomials that are used for a general provenance analysis. These semirings have algebraic *universality* properties (they are freely generated) for various classes of semirings. This allows us to compute provenance values once in a general such semiring and then to specialize it via homomorphisms (i.e., evaluation of the polynomials) to specific application semirings as needed.

• For any set X, the semiring $\mathbb{N}[X] = (\mathbb{N}[X], +, \cdot, 0, 1)$ consists of the multivariate polynomials in indeterminates from X and with coefficients from $\mathbb{N}$. This is the commutative semiring freely generated by the set X.

• By dropping coefficients from $\mathbb{N}[X]$, we get the semiring $\mathbb{B}[X]$ whose elements are just finite sets of distinct monomials. It is the free (+)-idempotent semiring over X.

• By dropping also exponents, we get the semiring $\mathbb{W}[X]$ of finite sums of monomials that are linear in each argument. It is sometimes called the why-semiring.

• The free absorptive semiring $\mathbb{S}[X]$ over X consists of 0, 1 and all antichains of monomials with respect to the componentwise order on their exponents. It is the quotient of $\mathbb{N}[X]$ by the congruence induced by $p \sim q$ for monomials p, q with $p = qr$.

• Finally $\mathsf{PosBool}(X) = (\mathsf{PosBool}(X), \vee, \wedge, \bot, \top)$ is the semiring whose elements are classes of equivalent positive (monotone) Boolean expressions with variables from X (its elements are in bijection with the positive Boolean expressions in irredundant disjunctive normal form). This is the distributive lattice freely generated by the set X.

3. Games

We consider two-player turn-based games on graphs. Such a game is defined by the game graph on which it is played, and by the objectives of the players.

Definition 2. A *game graph* is a structure $\mathcal{G} = (V, V_0, V_1, T, E)$, where $V = V_0 \cup V_1 \cup T$ is the set of positions, partitioned into the sets V_0, V_1 of the two players and the set T of terminal positions, and where $E \subseteq V \times V$ is the set of moves. We denote the set of immediate successors of a position v by $vE := \{w : (v, w) \in E\}$ and require that $vE = \varnothing$ if, and only if, $v \in T$. A play from an initial position v_0 is a finite or infinite path $v_0 v_1 v_2 \cdots$ through $\mathcal{G}$ where the successor $v_{i+1} \in v_i E$ is chosen by Player 0 if $v_i \in V_0$ and by Player 1 if $v_1 \in V_1$. A play ends when it reaches a terminal node $v_m \in T$.

Definition 3. For every game graph $\mathcal{G} = (V, V_0, V_1, T, E)$, and every initial position $v_0 \in V$, the *tree unraveling* of $\mathcal{G}$ from v_0 is the game tree $\mathcal{T}(\mathcal{G}, v_0)$ consisting of all finite paths from v_0. More precisely,

$\mathcal{T}(\mathcal{G}, v) = (V^\#, V_0^\#, V_1^\#, T^\#, E^\#)$, where $V^\#$ is the set of all finite paths $\pi = v_0 v_1 \cdots v_m$ through $\mathcal{G}$, with $V_\sigma^\# = \{\pi v \in V^\# : v \in V_\sigma\}$, $T^\# = \{\pi t \in V^\# : t \in T\}$, and $E^\# = \{(\pi v, \pi v v') : (v, v') \in E\}$. For most game-theoretic considerations, the games played on $\mathcal{G}$ and its unravelings are equivalent, via the canonical projection $\rho : \mathcal{T}(\mathcal{G}, v_0) \to \mathcal{G}$ that maps every path πv to its end point v.

A strategy for a player in a game is a function that selects moves at points that are controlled by that player. A strategy need not be defined at all positions of a player, but it must be closed in the sense that it defines a move from each position that is reachable by a play that is admitted by the strategy. There are several possibilities to define the notion of a strategy formally. For our purposes it is convenient to identify a strategy with the histories of plays that it admits, i.e., to view it as an appropriate subtree of $\mathcal{T}(\mathcal{G}, v_0)$.

Definition 4. A *strategy* of Player σ (for $\sigma \in \{0, 1\}$) from v_0 in a game $\mathcal{G}$ is a subtree of $\mathcal{T}(\mathcal{G}, v_0)$, of the form $\mathcal{S} = (W, F)$ with $W \subseteq V^\#$ and $F \subseteq (W \times W) \cap E^\#$, satisfying the following conditions:

- W is closed under predecessors: if $\pi v \in W$ then also $\pi \in W$.
- If $\pi v \in W \cap V_\sigma^\#$, then $|(\pi v)F| = 1$.
- If $\pi v \in W \cap V_{1-\sigma}^\#$ then $(\pi v)F = (\pi v)E^\#$.

We write $\mathrm{Strat}_\sigma(v_0)$ for the set of all strategies of Player σ from v_0.

In a strategy $\mathcal{S} = (W, F)$, the set W is the part of $\mathcal{T}(\mathcal{G}, v_0)$ on which the strategy is defined, and F is the set of moves that are admitted by the strategy. A strategy $\mathcal{S} \in \mathrm{Strat}_\sigma(v_0)$ induces the set $\mathrm{Plays}(\mathcal{S})$ of those plays from v_0 whose moves are consistent with $\mathcal{S}$. We call $\mathcal{S}$ well-founded if it does not admit any infinite plays; this is always the case on finite acyclic game graphs, but need not be the case otherwise. The set of possible *outcomes* of a strategy $\mathcal{S}$ is the set of terminal nodes that are reachable by a play that is consistent with $\mathcal{S}$. A strategy can also be viewed as a function $\mathcal{S} : W \cap V_\sigma^\# \to V$ such that $\mathcal{S}(\pi v) \in vE$ defines the node to which Player σ moves from πv.

The simplest objectives of players are reachability and safety objectives.

Definition 5. A *reachability objective* for Player σ is given by a set $T_\sigma \subseteq T$ of winning terminal positions. With such an objective, Player σ wins every play that reaches a position in T_σ. Dually, a *safety objective* for Player σ is given by a set $L_\sigma \subseteq T$ of "losing" positions that the player has to avoid, or equivalently, by its complement $S_\sigma = V \setminus L_\sigma$, the region of safe positions inside of which the player has to keep the play. With such an objective Player σ wins every play, finite or infinite, that never reaches a position in L_σ.

Notice that the difference between reachability and safety objectives is relevant only in cases where infinite plays are possible. Indeed, in a game that admits only finite plays, Player σ wins a play with the reachability objective T_σ if, and only if, she wins that play with the safety objective given by $L_\sigma = T \setminus T_\sigma$, so we can always reformulate reachability by safety and vice versa. However, in a game that admits infinite plays, Player σ wins with a reachability objective T_σ if, and only if, her opponent, Player $1-\sigma$, loses with the safety condition $L_{1-\sigma} = T_\sigma$. Hence winning with a reachability objective corresponds to defeating an opponent who plays with a safety objective. If both players play with reachability objectives, then infinite plays are won by neither player.

4. Provenance for well-founded games

We first study the provenance analysis of games for well-founded games, i.e., games that are played on *finite acyclic game graphs* $\mathcal{G} = (V, V_0, V_1, T, E)$, and hence do not admit infinite plays. We introduce *K-valuations* f_0 and f_1 that associate with every position $v \in V$ provenance values $f_0(v)$ and $f_1(v)$, respectively. The idea is that, for $\sigma \in \{0, 1\}$, the function f_σ describes the value of each position from the point of view of Player σ. Such a valuation is induced by its values on the terminal positions, i.e., by a function $f_\sigma : T \to K$, and by a valuation of the moves, i.e., by a function $h_\sigma : E \to K \setminus \{0\}$. Here, the function $f_\sigma : T \to K$ defines the value, for Player σ, of every *terminal position* where, intuitively, $f_\sigma(t) = 0$ means that position t is losing for Player σ. In the simplest case, we can specify reachability objectives T_σ by setting $f_\sigma(t) = 1$ for $t \in T_\sigma$ and $f_\sigma(t) = 0$ otherwise. The functions $h_\sigma : E \to K \setminus \{0\}$ provide a value (or cost) for Player σ of the moves. In many cases valuations of moves are not relevant; we then just put $h_\sigma(vw) = 1$ for all edges $(v, w) \in E$.

The extension of the basic valuations $f_\sigma : T \to K$ and $h_\sigma : E \to K \setminus \{0\}$ to valuations $f_\sigma : V \to K$ for all positions then relies on the idea that a move from v to w contributes to $f_\sigma(v)$ the value $h_\sigma(vw) \cdot f_\sigma(w)$. These contributions are summed up in the case that v is a position for Player σ (i.e., when she chooses herself the successors), and multiplied in the case that v is a position of the opponent (i.e., when she has to cope with any of the possible successors). This is summarized by the following definition.

Definition 6. Let K be a commutative semiring, let $\mathcal{G} = (V, V_0, V_1, T, E)$ be a finite acyclic game graph, and let $\sigma \in \{0, 1\}$ denote one of the two players. A *K-valuation* of $\mathcal{G}$ for Player σ is a function $f_\sigma : V \to K$. It is defined from basic valuations $f_\sigma : T \to K$ and $h_\sigma : E \to K \setminus \{0\}$ via backwards induction, by

$$f_\sigma(v) := \begin{cases} \sum_{w \in vE} h_\sigma(vw) \cdot f_\sigma(w) & \text{if } v \in V_\sigma, \\ \prod_{w \in vE} h_\sigma(vw) \cdot f_\sigma(w) & \text{if } v \in V_{1-\sigma}. \end{cases}$$

An equivalent characterization of the K-valuation f_σ can be obtained by defining provenance values for plays and strategies.

Definition 7. For a play $x = v_0 v_1 \cdots v_m$ from v_0 to a terminal node v_m, we define its valuation for Player σ as $f_\sigma(x) := h_\sigma(v_0 v_1) \cdots h_\sigma(v_{m-1} v_m) \cdot f_\sigma(v_m)$. Let now $\mathcal{S} = (W, F) \subseteq \mathcal{T}(\mathcal{G}, v_0)$ be a strategy for Player σ from v_0 and $\rho_S : (W, F) \to (V, E)$ be the restriction of the canonical homomorphism $\rho : \mathcal{T}(\mathcal{G}, v_0) \to \mathcal{G}$ to $\mathcal{S}$. For any position $v \in V$ and any move $e \in E$, the values

$$\#_\mathcal{S}(v) := |\rho_\mathcal{S}^{-1}(v)| \quad \text{and} \quad \#_\mathcal{S}(e) := |\rho_\mathcal{S}^{-1}(e)|$$

indicate how often the position v and the move e appear in the strategy $\mathcal{S}$. We then define the provenance value $\mathcal{S} \in \mathrm{Strat}_\sigma(v_0)$ as

$$F(\mathcal{S}) := \prod_{e \in E} h_\sigma(e)^{\#_\mathcal{S}(e)} \cdot \prod_{v \in T} f_\sigma(v)^{\#_\mathcal{S}(v)}.$$

In some important special cases, provenance values of strategies coincides with the product of the provenance values over all plays that they admit.

Lemma 8. *If $h_\sigma(e) = 1$ for all moves $e \in E$, or if the underlying semiring is multiplicatively idempotent (i.e., $a^2 = a$ for all a) we have that $F(\mathcal{S}) = \prod_{x \in \mathrm{Plays}(\mathcal{S})} f_\sigma(x)$ for all $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$.*

However, there are simple games where this is not the case. Consider, for instance, the valuation for Player 0 in a game where only the opponent, Player 1, moves: From position v, Player 1 can proceed to w by a move with value $h_0(vw) = a$, and from w he has the choice of moving to either s or t, both options having value 1 for Player 0. There is only one strategy $\mathcal{S}$ for Player 0 (do nothing), with provenance value a. However, the strategy admits two plays, ending in s and t, respectively, both of which have value a. Thus the product over the provenance value of the plays is a^2.

Theorem 9. *For any commutative semiring K and any finite acyclic game $\mathcal{G}$, let $f_\sigma : V \to K$ be the provenance valuation for Player σ, induced by the valuation $f_\sigma : T \to K$ of the terminal nodes and $h_\sigma : E \to K \setminus \{0\}$ of the moves. Then, for every position v*

$$f_\sigma(v) = \sum_{\mathcal{S} \in \mathrm{Strat}_\sigma(v)} F(\mathcal{S}).$$

Proof. For terminal positions v the claim is trivially true. So suppose that $v \in V_\sigma$. Then any strategy $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$ can be written in the form $\mathcal{S} = v \cdot \mathcal{S}'$ for some successor $w \in vE$ and some strategy $\mathcal{S}' \in \mathrm{Strat}_\sigma(w)$. Clearly, $\#_{\mathcal{S}}(t) = \#_{\mathcal{S}'}(t)$ for every terminal position $t \in T$. For the moves we have $\#_{\mathcal{S}}(e) = \#_{\mathcal{S}'}(e)$ for all $e \neq (v, w)$ but $\#_{\mathcal{S}}(e) = 1$ and $\#_{\mathcal{S}'}(e) = 0$ for $e = (v, w)$. This implies $F(S) = h(vw) \cdot F(\mathcal{S}')$. By induction hypothesis $f_\sigma(w) = \sum_{\mathcal{S}' \in \mathrm{Strat}_\sigma(w)} F(\mathcal{S}')$. Hence

$$f_\sigma(v) = \sum_{w \in vE} h_\sigma(vw) \cdot f_\sigma(w) = \sum_{w \in vE} \sum_{\mathcal{S}' \in \mathrm{Strat}_\sigma(w)} h_\sigma(vw) \cdot F(\mathcal{S}') = \sum_{\mathcal{S} \in \mathrm{Strat}_\sigma(v)} F(\mathcal{S}).$$

Finally, let $v \in V_{1-\sigma}$ with $vE = \{w_1, \dots, w_n\}$. Every strategy $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$ has the form $\mathcal{S} = v(\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_n)$, with $\mathcal{S}_i \in \mathrm{Strat}_\sigma(w_i)$. For the terminal nodes $t \in T$ we have $\#_{\mathcal{S}}(t) = \sum_{i \le n} \#_{\mathcal{S}_i}(t)$; similarly, for all moves e from a different position than v, we have $\#_{\mathcal{S}}(e) = \sum_{i \le n} \#_{\mathcal{S}_i}(e)$, but for the moves $e = (v, w_i)$ we have $\#_{\mathcal{S}}(e) = 1$ and $\#_{\mathcal{S}_i}(e) = 0$ for all i. Thus $F(\mathcal{S}) = \prod_{w_i \in vE} h_\sigma(vw_i) \cdot F(\mathcal{S}_i)$. It follows that

$$\begin{aligned} f_\sigma(v) &= \prod_{w_i \in vE} h_\sigma(vw_i) \cdot f_\sigma(w_i) = \prod_{w_i \in vE} h_\sigma(vw_i) \cdot \sum_{\mathcal{S}_i \in \mathrm{Strat}_\sigma(w_i)} F(\mathcal{S}_i) \\ &= \sum_{v \cdot (\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_n) \in \mathrm{Strat}_\sigma(v)} \prod_{w_i \in vE} h_\sigma(vw_i) \cdot F(\mathcal{S}_i) = \sum_{\mathcal{S} \in \mathrm{Strat}_\sigma(v)} F(\mathcal{S}). \end{aligned}$$ □

From this description, we can derive a number of applications of provenance valuations on games. We first consider the information provided by valuations in the general provenance semirings of polynomials. Let $\mathbb{N}[T]$ be the semiring of polynomials with coefficients in $\mathbb{N}$ over indeterminates $t \in T$, where T is the set of terminal positions in an acyclic game graph $\mathcal{G} = (V, V_0, V_1, T, E)$. Let $f_\sigma : V \to \mathbb{N}[T]$ be the valuation induced by setting $f_\sigma(t) = t$ for $t \in T$ and $h_\sigma(vw) = 1$ for all edges (v, w), so that the value of a play is just its outcome, i.e., the terminal position where it ends.

Clearly, we can write $f_\sigma(v)$ as a sum of monomials $m \cdot t_1^{j_1} \cdots t_k^{j_k}$. This provides a detailed description of the number and properties of the strategies that Player σ has from position v.

Theorem 10. *The valuation $f_\sigma(v) \in \mathbb{N}[T]$ is the sum of those monomials $m \cdot t_1^{j_1} \cdots t_k^{j_k}$ (with $m, j_1, \dots, j_k > 0$) such that Player σ has precisely m strategies $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$ with the property that the set of possible outcomes for $\mathcal{S}$ is precisely $\{t_1, \dots, t_k\}$, and precisely j_i plays that are consistent with $\mathcal{S}$ have the outcome t_i.*

This is an immediate consequence of Theorem 9 and Lemma 8. In many cases, somewhat less-detailed information is sufficient, which can be obtained by valuations in less-informative provenance semirings than $\mathbb{N}[T]$:

- Evaluating $f_\sigma(v)$ in the idempotent semiring $\mathbb{B}[T]$ gives us the sum of monomials $t_1^{j_1} \cdots t_k^{j_k}$ for which Player σ has at least one strategy whose multiset of admitted outcomes consists of $t_1, \ldots, t_k$ with multiplicities $j_1, \ldots, j_k$, respectively.

- If we evaluate $f_\sigma(v)$ in $\mathbb{W}[T]$ we get the sum of monomials $t_1 \cdots t_m$ such that Player σ has a strategy whose set of outcomes is $\{t_1, \ldots, t_m\}$. The information on multiplicities of strategies and outcomes is dropped.

- An interesting case is the evaluation in the absorptive semiring $\mathbb{S}[X]$. For two strategies $\mathcal{S}, \mathcal{S}' \in \mathrm{Strat}_\sigma(v)$, we say that $\mathcal{S}$ *absorbs* $\mathcal{S}'$ if for every terminal position $t \in T$, the strategy $\mathcal{S}$ admits fewer plays with outcome t than $\mathcal{S}'$. We call $\mathcal{S}$ *absorption-dominant* if it is not absorbed by any other strategy. Now, $f_\sigma(v) \in \mathbb{S}[X]$ is the sum of monomials $t_1^{j_1} \cdots t_k^{j_k}$ that describe precisely the (multiset of outcomes of the) absorption-dominant strategies of Player σ from v. See Section 11 below for a more detailed analysis of absorption among strategies.

- Finally, the evaluation of $f_\sigma(v) \in \mathsf{PosBool}[T]$ consists of those monomials $t_1 \cdots t_k$ such that $\{t_1, \ldots, t_k\}$ is a minimal set among the sets of outcomes of strategies $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$.

Fix any reachability objective $W \subseteq T$. In any of these provenance semirings, we can write the polynomial $f_\sigma(v)$ as a sum $f_\sigma(v) = f_\sigma^W(v) + g_\sigma^W(v)$, where $f_\sigma^W(v)$ is the sum of those monomials that only contain indeterminates in W and $g_\sigma^W(v)$ contains the rest.

Theorem 11. *For every subset $W \subseteq T$ and every $v \in V$, Player σ has a strategy to reach W from v if, and only if, $f_\sigma^W(v) \neq 0$ (in any of the provenance semirings given above). Moreover, if we set $f(t) = 1$ for $t \in W$ and $f(t) = 0$ for $t \in T \setminus W$, and evaluate f_σ in the semiring $\mathbb{N}$ of natural numbers, then $f_\sigma(v)$ is the number of distinct winning strategies for Player σ to reach W from v.*

Evaluation in other application semirings gives further interesting information about strategies:

Cost of strategies. Given a game $\mathcal{G}$, we associate with Player 0 *cost functions* $f_0 : T \to \mathbb{R}_+$ and $h : E \to \mathbb{R}_+$ for the terminal positions and the moves. We define the cost of a strategy $\mathcal{S} \in \mathrm{Strat}_0(v)$ as the sum of the costs of all moves and outcomes that it admits, weighted by the number of their occurrences.

Proposition 12. *The* cost of an optimal strategy *from v in $\mathcal{G}$ is given by the valuation $f_0(v)$ in the tropical semiring $\mathbb{T} = (\mathbb{R}_+^\infty, \min, +, \infty, 0)$.*

Proof. Since the product in $\mathbb{T}$ is addition in $\mathbb{R}_+^\infty$, the cost of a strategy $\mathcal{S}$ for Player 0, as defined above, coincides with the valuation $f_0(\mathcal{S})$ in $\mathbb{T}$. The summation in $\mathbb{T}$ is minimization in $\mathbb{R}_+^\infty$, so from Theorem 9 we get that

$$f_0(v) = \min_{\mathcal{S} \in \mathrm{Strat}_0(v)} F(\mathcal{S})$$

describes indeed the minimal cost of a strategy for Player 0 from position v. □

Clearance levels. The access control semiring is $\mathbb{A} = (\{\mathsf{P} < \mathsf{C} < \mathsf{S} < \mathsf{T} < 0\}, \min, \max, 0, \mathsf{P})$, where P is "public", C is "confidential", S is "secret", T is "top secret", and 0 is "so secret that nobody can access it!". Let $f_\sigma : T \to \mathbb{A}$ and $h_\sigma : E \to \mathbb{A} \setminus \{0\}$ define access levels for the terminal positions and the moves for Player σ, in the sense that Player σ can make a move e if, and only if, his personal clearance level is at least $h(e)$ and similarly, he can access a terminal position t if, and only if, his clearance level is at least $f_\sigma(t)$.

Proposition 13. *The valuation $f_\sigma(v) \in \mathbb{A}$ describes the* ***minimal clearance level*** *that Player* 0 *needs to win from position v, i.e., to have a strategy that guarantees reaching a terminal position that is accessible for him.*

The proof is a straightforward induction.

Confidence in games. Suppose that $f_\sigma : T \to [0, 1]$ describes the confidence that Player σ puts into t being a winning position for her. We want to compute *confidence scores* $f_\sigma(v)$ to describe the confidence of Player σ that she can win from v. It is natural to define the confidence score $f_\sigma(v)$ as the *maximum* of the confidence scores of the successors $w \in vE$ in the case that $v \in V_\sigma$. For confidence scores of combinations of events whose choice is taken by an opponent, such as for the possible moves from a position $v \in V_{1-\sigma}$, there are different approaches in the literature. A popular one, with which we work here, takes the *product* of the confidence scores of the events from which the opponent chooses. Adopting this definition, the following proposition is immediate.

Proposition 14. *Confidence scores are computed as semiring valuations $f_\sigma : V \to \mathbb{V}$ in the Viterbi semiring* $\mathbb{V} = ([0, 1], \max, \cdot, 0, 1)$.

Min-max games. Finally note that valuations in a min-max semiring $(A, \max, \min, a, b)$ describe the value of positions in games where Player 0 tries to maximize and Player 1 tries to minimize the outcome of the play.

Separating valuations. The K-valuations f_0, f_1 for the two players in a game $\mathcal{G}$, as defined by Definition 6, are a priori completely independent of each other. This admits the treatment of a wide variety of games, without any restrictions on how the objectives of the two players relate to each other. For instance, in a completely cooperative game, the basic valuations of the terminal positions would be the same for Player 0 and Player 1. However, in many games, the objectives of the two players are antagonistic, and valuations f_0 and f_1 should reflect this. This motivates the following definition.

Definition 15. Let $\mathcal{G}$ be a game graph, with valuations f_0, f_1 for the two players in a semiring K, and let $U \subseteq V$ be a set of positions. We say that:

(1) f_0, f_1 for the two players are *separating* on U if for all $u \in U$, either $f_0(u) = 0$ or $f_1(u) = 0$.

(2) f_0, f_1 are *weakly separating* on U if $f_0(u) f_1(u) = 0$ for all $u \in U$. Notice that in the case where K has no divisors of 0, weakly separating valuations are in fact separating.

(3) f_0 and f_1 are *strongly separating* on U if they are separating, and in addition, $f_0(u) + f_1(u) \neq 0$ for all $u \in U$.

Proposition 16. *If two valuations f_0 and f_1 are* (*weakly*) *separating on the terminal positions of $\mathcal{G}$, then they are* (*weakly*) *separating on all positions of $\mathcal{G}$.*

Proof. Recall that all our semirings are assumed to be +-positive. For $v \in V_\sigma$, we have

$$f_\sigma(v) = \sum_{w \in vE} h(vw) f_\sigma(w) \quad \text{and} \quad f_{1-\sigma}(v) = \prod_{w \in vE} h(vw) f_{1-\sigma}(w).$$

It follows that f_0 and f_1 are separating on v if they are separating on all $w \in vE$. Further,

$$\begin{aligned} f_\sigma(v) f_{1-\sigma}(v) = & \Big(\sum_{w \in vE} h_\sigma(vw) f_\sigma(w)\Big)\Big(\prod_{w \in vE} h_{1-\sigma}(vw) f_{1-\sigma}(w)\Big) = \\ & \sum_{w \in vE} \Big(h_\sigma(vw) f_\sigma(w) \prod_{w' \in vE} h_{1-\sigma}(vw') f_{1-\sigma}(w')\Big) = \\ & \sum_{w \in vE} \Big(h_\sigma(vw) h_{1-\sigma}(vw) f_\sigma(w) f_{1-\sigma}(w) \prod_{w' \in vE \setminus \{w\}} h_{1-\sigma}(vw') f_{1-\sigma}(w')\Big). \end{aligned}$$

This proves that f_0 and f_1 are weakly separating on v if they are so on all $w \in vE$. □

The corresponding implication for strongly separating valuations does not hold for all +-positive semirings, but it holds for positive ones.

Proposition 17. *If two valuations f_0 and f_1 into a positive semiring are strongly separating on the terminal positions of $\mathcal{G}$, then they are so on all positions of $\mathcal{G}$.*

Proof. We prove this by induction. Assume that f_0 and f_1 are strongly separating on all $w \in vE$. Then $f_\sigma(v) + f_{1-\sigma}(v) = 0$ only if $f_\sigma(w) = 0$ for all $w \in vE$ and $f_{1-\sigma}(w) = 0$ for at least one $w \in vE$. But this implies $f_0(w) + f_1(w) = 0$ for some $w \in vE$, which contradicts our assumption. □

Note that for the Boolean semiring $K = \mathbb{B}$, this is just Zermelo's theorem on the determinacy of reachability games on well-founded game graphs: from every position, one of the two players has a winning strategy.

Counting positional winning strategies? A strategy is *positional* if it only depends on the current position, and not on the history of the play, i.e., if $\mathcal{S}(\pi v) = \mathcal{S}(\pi' v)$ for all v and all paths πv, $\pi' v$ that lead to v. A positional strategy can be described by a function $s : V_\sigma \to V$ or by a subgraph $\mathcal{S}$ of $\mathcal{G}$ (rather than of $\mathcal{T}(\mathcal{G}, v_0)$).

Given that in the study of games there is (for instance for algorithmic reasons) a strong interest in positional strategies, it is reasonable to ask whether there exist valuations in different semirings that count just the positional strategies. However, invariance under counting bisimulation shows that this is not possible.

Definition 18. Let $\mathcal{G} = (V, V_0, V_1, T, E)$ and $\mathcal{G}' = (V', V_0', V_1', T', E')$ be two game graphs. A *counting bisimulation* between $\mathcal{G}$ and $\mathcal{G}'$ is a relation $Z \subseteq V \times V$ such that for every pair $(v, v') \in Z$ we have

(1) $v \in V_\sigma$ if, and only if, $v' \in V_\sigma'$ and $v \in T$ if, and only if, $v' \in T'$, and

(2) there is a local bijection $z_{vv'} : vE \to v'E'$ between the immediate successors of v and v' such that $(w, z_{vv'}(w)) \in Z$ for every $w \in vE$.

We write $\mathcal{G}, v \sim \mathcal{G}', v'$ if there is a counting bisimulation Z between $\mathcal{G}$ and $\mathcal{G}'$ such that $(v, v') \in Z$. Notice that for any game graph $\mathcal{G}$, the relation $Z = \{(v, \pi v) : v \in V,\ \pi v \in V^{\#}\}$ is a counting bisimulation between $\mathcal{G}$ and its unraveling $\mathcal{T}(\mathcal{G}, v_0)$.

K-valuations of games are invariant under counting bisimilarity in the following sense. Let $\mathcal{G}$ and $\mathcal{G}'$ be two acyclic game graphs with K-valuations $f_\sigma : T \to K$ and $f'_\sigma : T' \to K$ of the terminal positions and $h : E \to K$ and $h' : E' \to K$ of the moves. We say that a counting bisimulation $Z \subseteq V \times V'$ respects these valuations if $f_\sigma(t) = f'_\sigma(t')$ for all $(t, t') \in Z \cap T \times T'$, and $h_\sigma(vw) = h'_\sigma(v'w')$ whenever $(v, v') \in Z$ and $(w, w') \in Z$.

Proposition 19. *Let Z be a counting bisimulation between $\mathcal{G}$ and $\mathcal{G}'$ that respects the basic valuations of the terminal positions and the moves. Then Z respects the valuations of all positions*; *i.e.*, $f_\sigma(v) = f'_\sigma(v')$ *for all* $(v, v') \in Z$.

Proof. Let $(v, v') \in Z$. If v and v' are terminal positions, then $f_\sigma(v) = f'_\sigma(v')$ by assumption. Otherwise, v and v' are both positions of the same player. If they belong to Player σ, then $f_\sigma(v) = \sum_{w \in vE} f_\sigma(w)$. The local bijection $z_{vv'}$ maps every $w \in vE$ to some $w' \in v'E'$ such that, by induction hypothesis, $f_\sigma(w) = f'_\sigma(w')$. Hence $f'_\sigma(v') = \sum_{w' \in v'E'} f'_\sigma(w') = \sum_{w \in vE} f_\sigma(w) = f_\sigma(v)$. If v and v' belong to Player $(1 - \sigma)$ the reasoning is completely analogous, taking a product rather than a sum. □

In particular K-valuations of acyclic games do not change if we replace a game graph $\mathcal{G}$ by one of its unravelings $\mathcal{T}(\mathcal{G}, v)$. Indeed, every valuation $f_\sigma : T \to K$ on the terminal positions of a game graph $\mathcal{G}$ extends to the same valuation for v on $\mathcal{G}$ as on the tree unraveling $\mathcal{T}(\mathcal{G}, v)$. On the other side, every strategy on a tree-shaped game graph is positional. Thus the number of positional winning strategies is certainly not invariant under unraveling and hence not definable by valuations in a semiring.

5. Provenance for first-order logic via model-checking games and dual-indeterminate polynomials

Given a finite relational vocabulary τ and a finite nonempty universe A, we denote by $\mathrm{Atoms}_A(\tau)$ the set of all atoms $R\bar{a}$ with $R \in \tau$ and $\bar{a} \in A^k$. Further, let $\mathrm{NegAtoms}_A(\tau)$ be the set of all negated atoms $\neg R\bar{a}$, where $R\bar{a} \in \mathrm{Atoms}_A(\tau)$, and consider the set of all τ-literals on A,

$$\mathrm{Lit}_A(\tau) := \mathrm{Atoms}_A(\tau) \cup \mathrm{NegAtoms}_A(\tau) \cup \{a \,\mathrm{op}\, b : a, b \in A\},$$

where op stands for $=$ or $\neq$.

Definition 20. Given any commutative semiring K, a *K-interpretation* (for τ and A) is a function $\pi : \mathrm{Lit}_A(\tau) \to K$ that maps equalities and inequalities to their truth values 0 or 1.

We have defined in [Grädel and Tannen 2017] how a semiring interpretation extends to a full valuation $\pi : \mathrm{FO}(\tau) \to K$ mapping any fully instantiated formula $\psi(\bar{a})$ (or equivalently, any first-order sentence of vocabulary $\tau \cup A$) to a value $\pi[\![\psi]\!]$ by setting

$$\pi[\![\psi \vee \varphi]\!] := \pi[\![\psi]\!] + \pi[\![\varphi]\!], \qquad \pi[\![\psi \wedge \varphi]\!] := \pi[\![\psi]\!] \cdot \pi[\![\varphi]\!],$$
$$\pi[\![\exists x \varphi(x)]\!] := \sum_{a \in A} \pi[\![\varphi(a)]\!], \qquad \pi[\![\forall x \varphi(x)]\!] := \prod_{a \in A} \pi[\![\varphi(a)]\!].$$

Negation is handled via negation normal forms: we set $\pi[\![\neg\varphi]\!] := \pi[\![\mathrm{nnf}(\neg\varphi)]\!]$ where $\mathrm{nnf}(\varphi)$ is the negation normal form of φ.

This is equivalent to the game provenance, as defined above, for the model-checking game associated with the formula ψ and the K-interpretation $\pi : \mathrm{Lit}_A(\tau) \to K$. Notice that classically, model-checking games are defined for a formula (assumed to be given in negation normal form) and a fixed structure $\mathfrak{A}$;

see, e.g., [Apt and Grädel 2011, Chapter 4]. However, the game graph of such a model-checking game depends only on the formula ψ and the *universe* A of the given structure $\mathfrak{A}$. It is only the labeling of the terminal positions of the game, as winning for either the Verifier (Player 0) or the Falsifier (Player 1), that depends on which of the literals in $\mathrm{Lit}_A(\tau)$ are true in $\mathfrak{A}$. Hence the definition of a model-checking game readily generalizes to our more abstract provenance scenario.

Definition 21. Let $\psi(\bar{x}) \in \mathrm{FO}(\tau)$ be a first-order formula in negation normal form with a relational vocabulary τ, and let A be a (finite) universe. The model-checking game $\mathcal{G}(A, \psi)$ has positions $\varphi(\bar{a})$, obtained from a subformula $\varphi(\bar{x})$ of ψ, by instantiating the free variables $\bar{x}$ by a tuple $\bar{a}$ of elements of A. At a disjunction $(\psi \vee \varphi)$, Player 0 (Verifier) moves to either ψ or φ, and at a conjunction, Player 1 (Falsifier) makes an analogous move. At a position $\exists x \varphi(\bar{a}, x)$, The Verifier selects an element b and moves to $\varphi(\bar{a}, b)$, whereas at positions $\forall x \varphi(\bar{a}, x)$ the move to the next position $\varphi(\bar{a}, b)$ is done by the Falsifier. The terminal positions of $\mathcal{G}(A, \psi)$ are the literals in $\mathrm{Lit}_A(\tau)$.

A K-interpretation $\pi : \mathrm{Lit}_A(\tau) \to K$ thus provides a valuation of the set $T \subseteq \mathrm{Lit}_A(\tau)$ of terminal positions of the model-checking game $\mathcal{G}(A, \psi)$ for any sentence $\psi \in \mathrm{FO}(\tau \cup A)$. We view it as a valuation f_0 for Player 0. The associated valuation f_1 for Player 1 is obtained by setting $f_1(\varphi) = \pi[\![\neg\varphi]\!]$ for any literal $\varphi \in \mathrm{Lit}_A(\tau)$. Both valuations then extend to full valuations f_0 and f_1 of all positions of $\mathcal{G}(A, \psi)$, including the position ψ itself. The following result is proved by a straightforward induction on formulae.

Theorem 22. *For all positions φ of $\mathcal{G}(A, \psi)$ we have $f_0(\varphi) = \pi[\![\varphi]\!]$ and $f_1(\varphi) = \pi[\![\neg\varphi]\!]$.*

Although this theorem holds without any restrictions on the semiring K and the K-interpretation π, not all such K-interpretations are really meaningful for logic. Indeed the provenance value of complementary literals $R\bar{a}$ and $\neg R\bar{a}$ have to be related in a reasonable way, and as a consequence also the general provenance semirings of polynomials need to be modified. In the simplest case a K-interpretation defines a unique τ-structure.

Definition 23. A semiring interpretation $\pi : \mathrm{Lit}_A(\tau) \to K$ is *model-defining* if for every atom $\varphi \in \mathrm{Atoms}_A(\tau)$ one of $\pi(\varphi)$ and $\pi(\neg\varphi)$ is 0, and the other is $\neq 0$. It uniquely defines the τ-structure $\mathfrak{A}_\pi$ that has universe A, and in which precisely those literals φ are true for which $\pi(\varphi) \neq 0$.

Notice that if K is not the Boolean semiring, then several different K-interpretations may define the same structure. Further, K-interpretations are interesting, and have a number of applications, also in cases where they do not specify a single model; see [Grädel and Tannen 2017].

Dual-indeterminate polynomials. Let X, $\overline{X}$ be two disjoint sets together with a one-to-one correspondence $X \leftrightarrow \overline{X}$. We denote by $p \in X$ and $\bar{p} \in \overline{X}$ two elements that are in this correspondence. We refer to the elements of $X \cup \overline{X}$ as *provenance tokens* and we shall use "positive" and "negative" tokens p and $\bar{p}$ to annotate atoms $R\bar{a} \in \mathrm{Atoms}_A(\tau)$ and negated atoms $\neg R\bar{a} \in \mathrm{NegAtoms}_A(\tau)$, respectively. By convention, if we annotate $R(\bar{a})$ with p then the "negative" token $\bar{p}$ can only be used to annotate $\neg R(\bar{a})$, and vice versa. We refer to p and $\bar{p}$ as *complementary* tokens.

Definition 24. The semiring $\mathbb{N}[X, \overline{X}]$ is the quotient of the semiring of polynomials $\mathbb{N}[X \cup \overline{X}]$ by the congruence generated by the equalities $p \cdot \bar{p} = 0$ for all $p \in X$. This is the same as quotienting by the ideal generated by the polynomials $p\bar{p}$ for all $p \in X$. Observe that two polynomials $g, g' \in \mathbb{N}[X \cup \overline{X}]$ are congruent if, and only if, they become identical after deleting from each of them the monomials that contain complementary tokens. Hence, the congruence classes in $\mathbb{N}[X, \overline{X}]$ are in one-to-one correspondence with the polynomials in $\mathbb{N}[X \cup \overline{X}]$ such that none of their monomials contain complementary tokens. We shall call these *dual-indeterminate polynomials*.

Note that $\mathbb{N}[X, \overline{X}]$ is +-positive and root-integral, but not positive, since it has divisors of 0. Further, we have the following *universality property*:

Proposition 25. *Every function $f : X \cup \overline{X} \to K$ into any commutative semiring K with the property that $f(p) \cdot f(\bar{p}) = 0$ for all $p \in X$ extends uniquely to a semiring homomorphism $h : \mathbb{N}[X, \overline{X}] \to K$ that coincides with f on $X \cup \overline{X}$.*

Definition 26. A *provenance-tracking* interpretation is a mapping $\pi : \mathrm{Lit}_A(\tau) \to X \cup \overline{X} \cup \{0, 1\}$ such that $\pi(\mathrm{Atoms}_A(\tau)) \subseteq X \cup \{0, 1\}$ and $\pi(\mathrm{NegAtoms}_A(\tau)) \subseteq \overline{X} \cup \{0, 1\}$. Further, π maps equalities and inequalities to their truth values 0 or 1.

The idea is that if π annotates a positive or negative atom with a token, then we wish to track that literal through the model-checking computation. On the other hand annotating with 0 or 1 is done when we do not track the literal, yet we need to recall whether it holds or not in the model. See [Grädel and Tannen 2017] for more details and potential applications of provenance-tracking interpretations.

6. Semirings of dual-indeterminate power series and least fixed-point solutions

It is known that the general properties of commutative semirings are not sufficient to deal with unbounded iterations as they occur in fixed-point logic. Even for Datalog, one of the simplest fixed-point formalisms that omits the complications arising with universal quantification and negation, appropriate semirings have the additional property of being *ω-continuous*. The general ω-continuous provenance semirings are no longer semirings of polynomials, but semirings of formal power series, such as $\mathbb{N}^{\infty}[\![X]\!]$. We combine this here with our approach for dealing with negation by taking quotients with respect to the congruence generated by products $p\bar{p}$ of positive and negative provenance tokens. What we obtain are ω-continuous provenance semirings of dual-indeterminate power series, such as $\mathbb{N}^{\infty}[\![X, \overline{X}]\!]$, as well as idempotent, absorptive, and other variants thereof.

A semiring K is *naturally ordered* if the relation $a \leq b :\Leftrightarrow \exists x(a + x = b)$ is a partial order. Note that this relation is reflexive and transitive in every semiring, but it is not always antisymmetric. An ω-chain is a sequence $(a_i)_{i \in \omega}$ with $a_i \leq a_{i+1}$ for all $i \in \omega$.

Definition 27. A commutative semiring K is *ω-continuous* if it is naturally ordered and satisfies the following additional conditions:

- Every ω-chain $(a_i)_i \in \omega$ has a supremum $\sup_{i \in \omega} a_i$ in K. As a consequence, we have a well-defined infinite summation operator $\sum$ such that for every sequence $(b_i)_{i \in \omega}$
$$\sum_{i \in \omega} b_i := \sup\{a_0 + \cdots + a_n : n \in \omega\}.$$
- For every sequence $(a_i)_{i \in \omega}$ in K, every $c \in K$, and every partition $(I_j)_{j \in J}$ of ω, we have $c \cdot \sum_{i \in \omega} a_i = \sum_{i \in \omega} c \cdot a_i$ and $\sum_{j \in J} \sum_{i \in I_j} a_i = \sum_{i \in \omega} a_i$.

In an ω-continuous semiring we further have the Kleene star operation,

$$a^* := \sum_{i\in\omega} a^i = \sup_{i\in\omega}(1 + a + a^2 + \cdots + a^i).$$

A function $f : K \to K$ is ω-continuous if $\sup_{i\in\omega} f(a_i) = f(\sup_{i\in\omega} a_i)$ for every ω-chain $(a_i)_{i\in\omega}$. A consequence of the definition is that any function defined by a polynomial or a power series is ω-continuous in each argument.

Definition 28. Given a semiring K and a finite set X of indeterminates, we denote by $K[\![X]\!]$ the semiring of formal power series (i.e., possibly infinite sums of monomials) with coefficients in K and indeterminates in X, with addition and multiplication defined in the obvious way. If K is ω-continuous and $|X| = n$, then every formal power series $f \in K[\![X]\!]$ induces a well-defined function $f : K^n \to K$ which is ω-continuous in each argument. Further, if K is ω-continuous, then so is $K[\![X]\!]$ [Kuich 1997].

A *system of power series* with indeterminates $X_1, \ldots, X_n$ is a sequence $G = (g_1 \cdots g_n)$ with $g_i \in K[\![X]\!]$ for each i. It induces a function $G : K^n \to K^n$ that is monotone in each argument. By Kleene's fixed-point theorem G has a least fixed point lfp G which coincides with the supremum of the Kleene approximants G^k, defined by $G^0 = 0,\ G^{k+1} = G(G^k)$; i.e., $\text{lfp}\, G = \sup_{k\in\omega} G^k$. We also refer to lfp G as the *least fixed-point solution* of the equation system

$$X_1 = g_1(X_1, \ldots, x_n), \quad \ldots, \quad X_n = g_n(X_1, \ldots, X_n);$$

in short, $X = G(X)$.

Dual-indeterminate power series. Semirings $K[\![X]\!]$ of power series turn out to be appropriate as general provenance semirings for (not necessarily acyclic) reachability games, without any further structure on the terminal nodes, as well as for purely positive fixed-point formalisms, without negation even on the atomic level. However, as soon as we want to deal with fixed-point logics with (atomic) negation we again need to take quotients with respect to the congruence generated by an appropriate correspondence $X \leftrightarrow \overline{X}$ between positive and negative tokens (with the same conventions as in Definition 24).

Definition 29. The semiring $K[\![X, \overline{X}]\!]$ is the quotient of the semiring of power series $K[\![X \cup \overline{X}]\!]$ by the congruence generated by the equalities $p \cdot \bar{p} = 0$ for all $p \in X$. The congruence classes in $K[\![X, \overline{X}]\!]$ are in one-to-one correspondence with the power series in $K[\![X \cup \overline{X}]\!]$ such that none of their monomials contain complementary tokens. We call these *dual-indeterminate power series*.

Again we have a universality property.

Proposition 30. *Every function $f : X \cup \overline{X} \to K$ into an ω-continuous semiring K with the property that $f(p) \cdot f(\bar{p}) = 0$ for all $p \in X$ extends uniquely to an ω-continuous semiring homomorphism $h : \mathbb{N}[\![X, \overline{X}]\!] \to K$ that coincides with f on $X \cup \overline{X}$.*

7. Provenance for reachability games with cycles

We now extend our provenance approach to games that admit infinite plays. We assume that the game graphs are finite, but no longer acyclic. Given a valuation $f_\sigma : T \to K$ in a semiring K for the terminal nodes of a game graph $\mathcal{G}$, the rules defining valuations for the other nodes have now to be read as an equation system (G_σ) in indeterminates X_v (for $v \in V$):

$$\begin{aligned} X_v &= f_\sigma(v) && \text{for } v \in T, \\ X_v &= \sum_{w \in vE} h_\sigma(vw) \cdot X_w && \text{if } v \in V_\sigma, \\ X_v &= \prod_{w \in vE} h_\sigma(vw) \cdot X_w && \text{if } v \in V_{1-\sigma}. \end{aligned} \tag{G_σ}$$

If we assume that the underlying semiring K is ω-continuous, then such a system (G_σ) always has a least fixed-point solution $\operatorname{lfp} G_\sigma$, which can be computed as the limit of its Kleene approximants $G^n : V \to K$ for $n \in \omega$. These Kleene approximants can be seen as valuations in the unravelings $\mathcal{G}^n$ of the game $\mathcal{G}$ up to n moves, defined as follows.

Recall that, for every game graph $\mathcal{G} = (V, V_0, V_1, T, E)$, and every initial position $v_0 \in V$, we have the *tree unraveling* $\mathcal{T}(\mathcal{G}, v_0) = (V^\#, V_0^\#, V_1^\#, T^\#, E^\#)$ consisting of all finite paths from v_0, with the canonical projection $\rho : \mathcal{T}(\mathcal{G}, v_0) \to \mathcal{G}$ that maps every path πv to its end point v.

Definition 31. Given $\mathcal{G}$ with basic valuations $f_\sigma : T \to K$ and $h_\sigma : E \to K \setminus \{0\}$ of the terminal positions and moves, the *truncation* $\mathcal{G}^n = (V^{(n)}, V_0^{(n)}, V_1^{(n)}, T^{(n)}, E^{(n)})$, for $n > 0$, is the restriction of the union of the trees $\mathcal{T}(\mathcal{G}, v)$ (with $v \in V$) to paths of less than n moves, and $\rho^n : \mathcal{G}^n \to \mathcal{G}$ is the restriction of the canonical homomorphism ρ to $\mathcal{G}^n$. Notice that the truncation induces new terminal nodes:

$$T^{(n)} := \{\pi v \in V^{(n)} : v \in T\} \cup \{\pi v \in V^{(n)} : |\pi| = n - 1,\ v \in V \setminus T\}.$$

In $\mathcal{G}^n$, we define the basic valuation of the moves, $h_\sigma^n : E^{(n)} \to K \setminus \{0\}$, in the obvious way, by $h_\sigma^n(e) := h_\sigma(\rho^n(e))$. For the valuation of the terminal nodes $\pi v \in T^{(n)}$, we put $f_\sigma^n(\pi v) = f_\sigma(v)$ if $v \in T$, and $f_\sigma^n(\pi v) = 0$ otherwise, i.e., if πv is an initial segment of a play in $\mathcal{G}$, with $n - 1$ moves, that has not reached a terminal position in T.

The games $\mathcal{G}^n$ are finite acyclic games, and the basic valuations extend to valuations $f_\sigma^n : V^{(n)} \to K$ for all nodes of $\mathcal{G}^n$. By induction, it readily follows that, for all nodes v of $\mathcal{G}$, the Kleene approximants G^n of (G_σ) coincide with these valuations.

Lemma 32. *For all n and all positions v of G, we have $G^n(v) = f_\sigma^n(v)$.*

We denote the strategy space of Player σ from v in $\mathcal{G}^n$ by $\mathrm{Strat}_\sigma^{(n)}(v)$. Since the games $\mathcal{G}^n$ are acyclic, Theorem 9 applies.

Lemma 33. *For every n, and every position v, we have $f_\sigma^n(v) = \sum_{\mathcal{T} \in \mathrm{Strat}_\sigma^{(n)}(v)} F(\mathcal{T})$.*

Valuations of plays and strategies in games with cycles. To generalize Theorem 9 to reachability games with cycles, we first need to extend the valuations of plays and strategies to such games. As in Section 4 a *finite* play $x = v_0 v_1 \cdots v_m$ in $\mathcal{G}$ from v_0 to a terminal node $v_m \in T$ gets the valuation $f_\sigma(x) = h_\sigma(v_0 v_1) \cdots h_\sigma(v_{m-1} v_m) \cdot f_\sigma(v_m)$. The provenance value of an infinite play is defined to be 0. For a strategy $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$, we put $F(\mathcal{S}) := 0$ if $\mathcal{S}$ admits any infinite play. Hence a strategy $\mathcal{S}$ can have a nonzero provenance value only when it admits just finite plays. By König's lemma, it then admits only a finite number of plays, and putting, as in Section 4,

$$F(\mathcal{S}) := \prod_{e\in E} h_\sigma(e)^{\#_{\mathcal{S}}(e)} \cdot \prod_{v\in T} f_\sigma(v)^{\#_{\mathcal{S}}(v)},$$

we have $F(\mathcal{S})$ is well-defined for such strategies, as the values $\#_{\mathcal{S}}(e)$ and $\#_{\mathcal{S}}(v)$ are finite, for all $e \in E$ and $v \in T$. Although the number of different strategies $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$ may well be infinite, Theorem 9 generalizes to reachability games with cycles, with a proof based on Kleene's fixed-point theorem, and the unravelings of $\mathcal{G}$ to finite acyclic games $\mathcal{G}^n$.

Notice that every strategy $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$ for the original game $\mathcal{G}$ induces, in every game $\mathcal{G}^n$, a strategy $\mathcal{S}^{(n)} \in \mathrm{Strat}_\sigma^{(n)}(v)$ for the game $\mathcal{G}_n$.

Lemma 34. *For every strategy $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$ in $\mathcal{G}$ with $F(\mathcal{S}) \neq 0$ there exists some $n_{\mathcal{S}} < \omega$ such that*

- $\mathcal{S} = \mathcal{S}^{(n)}$ *for all* $n \geq n_{\mathcal{S}}$,
- $F(\mathcal{S}^{(m)}) = 0$ *for all* $m < n_{\mathcal{S}}$.

Proof. This readily follows from the fact that a strategy $\mathcal{S}$ with $F(\mathcal{S}) \neq 0$ admits only a finite number of plays, all of which are finite. Let $n_{\mathcal{S}}$ be the maximal length of these plays. Then, for $n \geq n_{\mathcal{S}}$, all plays in $\mathcal{S}$ are already contained in $\mathcal{S}^{(n)}$. For any $m < n_s$, the induced strategy admits an unfinished play; hence $F(\mathcal{S}^m) = 0$. □

Every strategy $\mathcal{T} \in \mathrm{Strat}_\sigma^{(n)}(v)$ can be obtained as the induced strategy of some $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$ such that $\mathcal{T} = \mathcal{S}^{(n)}$. In general $\mathcal{S}$ is not uniquely determined by $\mathcal{T}$ and n. Nevertheless, we have the following.

Lemma 35. *For every position v of $\mathcal{G}$ and every $n < \omega$, we have that in $\mathcal{G}^n$*

$$\sum_{\mathcal{S}\in\mathrm{Strat}_\sigma(v)} F(\mathcal{S}^{(n)}) = \sum_{\mathcal{T}\in\mathrm{Strat}_\sigma^{(n)}(v)} F(\mathcal{T}).$$

Proof. If we have two strategies $\mathcal{S}_1 \neq \mathcal{S}_2$ in $\mathrm{Strat}_\sigma(v)$ with $\mathcal{T} = \mathcal{S}_1^{(n)} = \mathcal{S}_2^{(n)}$, then $\mathcal{T}$ must contain an unfinished play (otherwise $\mathcal{T} = \mathcal{S}_1 = \mathcal{S}_2$), which implies that $F(\mathcal{T}) = 0$. Thus, although the strategy spaces $\mathrm{Strat}_\sigma(v)$ are in general infinite, whereas $\mathrm{Strat}_\sigma^{(n)}(v)$ is finite for each fixed n, those strategies that provide nonzero values to the sums are in one-to-one correspondence, and the two sums have the same value. □

Putting these observations together, we obtain the desired generalization of Theorem 9.

Theorem 36. *For every game graph $\mathcal{G}$ with basic valuations f_σ and h_σ of the terminal positions and moves in an ω-continuous semiring K, we have that, for every position v,*

$$f_\sigma(v) := (\mathrm{lfp}\, G_\sigma)(v) = \sum_{\mathcal{S}\in\mathrm{Strat}_\sigma(v)} F(\mathcal{S}).$$

In the cases where $h(e) = 1$ for all e, or where K is multiplicatively idempotent, we further have that

$$f_\sigma(v) = \sum_{\mathcal{S}\in\mathrm{Strat}_\sigma(v)} \prod_{x\in\mathrm{Plays}(\mathcal{S})} f_\sigma(x).$$

Proof. By the lemmata above, we have that, for every $n < \omega$,

$$G^n(v) = f_\sigma^n(v) = \sum_{\mathcal{T} \in \mathrm{Strat}_\sigma^{(n)}(v)} F(\mathcal{T}) = \sum_{\mathcal{S} \in \mathrm{Strat}_\sigma(v)} F(\mathcal{S}^{(n)}).$$

Since, for every strategy $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$ we have $F(\mathcal{S}) = F(\mathcal{S}^n)$ for sufficiently large n, the result follows by taking suprema. □

For the case of game valuations $f_\sigma : V \to \mathbb{N}[\![T]\!]$, given by the basic valuations $f_\sigma(t) = t$ for terminal positions $t \in T$ and $h_\sigma(vw) = 1$ for all moves $(v, w) \in E$, we again get precise information about the number of strategies that a player has for a specific outcome. Indeed, $f_\sigma(v)$ is a (possibly) infinite sum of monomials $m \cdot t_1^{j_1} \cdots t_k^{j_k}$.

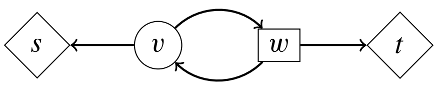
Corollary 37. *Let $f_\sigma : V \to \mathbb{N}[\![T]\!]$ be the valuation of Player σ for the game $\mathcal{G}$ in $\mathbb{N}[\![T]\!]$. For every monomial $m \cdot t_1^{j_1} \cdots t_k^{j_k}$ in $f_\sigma(v)$ (with $m \in \mathbb{N}$ and $j_i > 0$) Player σ has precisely m strategies $\mathcal{S}$ from v with the property that the set of possible outcomes for $\mathcal{S}$ is precisely $\{t_1, \ldots, t_k\}$, and precisely j_i plays that are consistent with $\mathcal{S}$ have the outcome t_i.*

Let $\mathcal{G} = (V, V_0, V_1, T, E)$ be a game with reachability objectives T_0, T_1 for the two players such that $T_0 \cap T_1 = \varnothing$. Let $W_0, W_1 \subseteq V$ be the winning regions for the two players; i.e., W_σ is the set of those positions $v \in V$ such that Player σ has a strategy from v to force the play to T_σ. Note that V is the disjoint union of the W_0, W_1 and U, the set of those positions from which none of the two players has a winning strategy. By Zermelo's theorem both players have strategies to guarantee that each play from U will be at least a draw.

Corollary 38. *Let $f_\sigma : T \to K$ be a valuation of the terminal positions of $\mathcal{G}$ in an ω-continuous semiring, with $f_\sigma(t) \neq 0$ if, and only if, $t \in T_\sigma$. The least fixed-point solution of the equation system F_σ extends this to a valuation $f_\sigma : V \to K$, with $f_\sigma(v) \neq 0$ if, and only if, $v \in W_\sigma$.*

Notice that weakly contradictory valuations f_0 and f_1 on the terminal positions extend to weakly contradictory valuations on all positions. However, even valuations into ω-continuous semirings that are strongly contradictory on the terminal positions are in general only weakly contradictory on the set of all positions, unless $W_0 \cup W_1 = V$, since $f_0(U) = f_1(U) = 0$.

Example 39. We illustrate our findings by the following very simple example of a game where Player 0 moves from v, Player 1 moves from w, and s and t are terminal nodes:



The corresponding equation system for Player 0 has the equations $X_v = s + X_w$ and $X_w = t \cdot X_v$. In $\mathbb{N}^\infty[\![s, t]\!]$ the least fixed-point solution is $f(v) = s \cdot (1 + t + t^2 \cdots)$ and $f(w) = s \cdot (t + t^2 + \cdots)$. If we evaluate it for the reachability objectives $\{s\}$ and $\{t\}$, respectively, we obtain $f(v)[0, t] = f(w)[0, t] = 0$, which illustrates that neither from v nor from w does Player 0 have a strategy to reach t. On the other side, $f(v)[s, 0] = s$ and $f(w)[s, 0] = 0$, which is consistent with the fact that Player 0 has a strategy to reach s from v but not from w.

But the formal power series $f(v)$ and $f(w)$ reveal more information than that. For instance, the fact that $f(v)$ contains, for every n, the monomial $s \cdot t^n$ implies that Player 0 has precisely one strategy S

from v that admits precisely $n+1$ consistent plays, one of which has outcome s and the other n have outcome t; this is the strategy where Player 0 moves from v to w the first n times, and then to s. Notice that Player 0 also has one further strategy, namely the (positional) strategy to move always to w. However, this strategy does not guarantee that the play terminates and therefore has value 0, so it is not visible in the provenance values $f(v)$ and $f(w)$.

8. Provenance analysis for positive LFP

Least fixed-point logic, denoted LFP, extends first-order logic by least and greatest fixed points of definable monotone operators on relations: if $\psi(R, \bar{x})$ is a formula of vocabulary $\tau \cup \{R\}$, in which the relational variable R occurs only positively, and if $\bar{x}$ is a tuple of variables such that the length of $\bar{x}$ matches the arity of R, then $[\text{lfp}\, R\bar{x}\,.\,\psi](\bar{x})$ and $[\text{gfp}\, R\bar{x}\,.\,\psi](\bar{x})$ are also formulae (of vocabulary τ). The semantics of these formulae is that $\bar{x}$ is contained in the least (respectively the greatest) fixed point of the update operator $F_\psi : R \mapsto \{\bar{a} : \psi(R, \bar{a})\}$. Due to the positivity of R in ψ, any such operator F_ψ is monotone and therefore has, by the Knaster–Tarski theorem, a least fixed point $\text{lfp}(F_\psi)$ and a greatest fixed point $\text{gfp}(F_\psi)$. See, e.g., [Grädel et al. 2007] for background on LFP.

Note that in formulae $[\text{lfp}\, R\bar{x}\,.\,\psi](\bar{x})$ one may allow ψ to have other free variables besides $\bar{x}$; these are called parameters of the fixed-point formula. However, at the expense of increasing the arity of the fixed-point predicates and the number of variables, one can always eliminate parameters. For the construction of model-checking games and also for provenance analysis it is convenient to assume that formulae are parameter-free. The duality between least and greatest fixed points implies that for any ψ

$$[\text{gfp}\, R\bar{x}\,.\,\psi](\bar{x}) \equiv \neg[\text{lfp}\, R\bar{x}\,.\,\neg\psi[R/\neg R]](\bar{x}).$$

Using this duality together with De Morgan's laws, every LFP-formula can be brought into *negation normal form*, where negation applies to atoms only.

The fragment of positive least fixed points. We denote by posLFP the fragment of LFP consisting of formulae in negation normal form such that all their fixed-point operators are least fixed points. It is known that, on finite structures (but not in general), posLFP has the same expressive power as full LFP, and thus captures all polynomial-time computable properties of ordered finite structures [Grädel et al. 2007] .

An advantage of dealing with posLFP, rather than full LFP, is that it admits much simpler model-checking games. Indeed the appropriate games for LFP are *parity games*, whereas for posLFP, reachability games are sufficient. This can be exploited to define provenance interpretations for fixed-point formulae, along the lines described in the previous section.

Definition 21 of model-checking games $\mathcal{G}(A, \psi)$ for $\psi \in \text{FO}(\tau)$ extends to formulae $\psi(\bar{x}) \in \text{posLFP}(\tau)$ as follows: for every subformula of ψ of the form $\vartheta := [\text{lfp}\, R\bar{x}\,.\,\varphi(R, \bar{x})](\bar{x})$ we add moves from positions $\vartheta(\bar{a})$ to $\varphi(\bar{a})$, and from positions $R\bar{a}$ to $\varphi(\bar{a})$ for every tuple $\bar{a}$. Since these moves are unique it makes no difference to which of the two players we assign the positions $\vartheta(\bar{a})$ and $R\bar{a}$. The resulting game graphs $\mathcal{G}(A, \psi)$ may contain cycles, but the set T of terminal nodes is again a subset of $\text{Lit}_A(\tau)$.

A K-interpretation $\pi : \text{Lit}_A(\tau) \to K$ into an ω-continuous semiring thus provides a valuation of the terminal positions of the game graph $\mathcal{G}(A, \psi)$ for any $\psi \in \text{posLFP}(\tau)$. By Corollary 38 this extends to a valuation $f_0 : V \to K$ on the set V of all positions $\varphi(\bar{a})$ of $\mathcal{G}(A, \psi)$, including position ψ itself.

Definition 40. For any instantiated subformula φ of a sentence $\psi \in$ posLFP, we define the provenance value $\pi[\![\varphi]\!]$ by its game valuation: $\pi[\![\varphi]\!] := f_0(\varphi)$.

In particular, if π is model-defining, then f_0 provides truth values for all fully instantiated subformula φ of ψ on the structure $\mathfrak{A}_\pi$ that π describes. Indeed $\mathfrak{A}_\pi \models \varphi$ if, and only if, $\pi[\![\varphi]\!] \neq 0$, and in that case the value $\pi[\![\varphi]\!]$ gives us additional information, how and why φ holds in $\mathfrak{A}$, for instance by information on the winning strategies that the Verifier has available for establishing the truth of φ in $\mathfrak{A}_\pi$. However, contrary to the case of first-order logic, in the case where $\mathfrak{A}_\pi \not\models \varphi$, and hence $\pi[\![\varphi]\!] = 0$, we do not get additional information on the reasons why φ is false. The possibility to move to $\neg\varphi$ (or more precisely, its negation normal form) and to do the provenance analysis for that formula does not exist here since $\neg\varphi$ is not a formula of posLFP. In fact, the model-checking game for $\neg\varphi$ is not a reachability game, but a safety game. To deal with safety games and greatest fixed points we shall have to impose additional restrictions on the underlying semirings. We shall discuss this below.

One can define provenance values for posLFP-sentences also directly by a fixed-point interpretation in ω-commutative semirings. The goal is to extend, by induction over the syntax, a K-interpretation $\pi : \mathrm{Lit}_A(\tau) \to K$ to valuations $\pi[\![\psi]\!] \in K$ for all sentences $\psi \in \mathrm{posLFP}(\tau \cup A)$. The rules for first-order operations are defined already, so we just have to consider sentences of form $\psi(\bar{a}) = [\mathrm{lfp}\, R\bar{x}.\varphi(R,\bar{x})](\bar{a})$, with $\varphi \in \mathrm{posLFP}(\tau \cup \{R\})$. If R has arity m, then its K-interpretations of A are functions $g : A^m \to K$. These functions are ordered by $g \le g'$ if, and only if, $g(\bar{a}) \le g'(\bar{a})$ for all $\bar{a} \in A^m$. Given a K-interpretation $\pi : \mathrm{Lit}_A(\tau) \to K$, we denote by $\pi[R \mapsto g]$ the K-interpretation of $\mathrm{Lit}_A(\tau) \cup \mathrm{Atoms}_A(\{R\})$ obtained from π by adding values $g(\bar{c})$ for the atoms $R\bar{c}$. (Notice that R appears only positively in φ, so negated atoms are not needed.)

The formula $\varphi(R,\bar{x})$ now defines, together with π, a monotone update operator F^φ_π on functions $g : A^m \to K$. More precisely, it maps g to

$$F^\varphi_\pi(g) : \bar{a} \mapsto \pi[R \mapsto g][\![\varphi(R,\bar{a})]\!].$$

By Kleene's fixed-point theorem, the operator F^φ_π has a least fixed point $\mathrm{lfp}(F^\varphi_\pi)$ which coincides with the limit of the sequence $(g^n)_{n<\omega}$ with $g^0 := 0$ and $g^{n+1} := F^\varphi_\pi(g^n)$, and which we may define as the provenance value of $[\mathrm{lfp}\, R\bar{x}.\varphi(R,\bar{x})](\bar{a})$. The two definitions coincide.

Proposition 41. *For every formula* $[\mathrm{lfp}\, R\bar{x}.\varphi(R,\bar{x})] \in \mathrm{posLFP}$ *and every* K*-interpretation* $\pi : \mathrm{Lit}_A(\tau) \to K$ *into an* ω*-continuous semiring*, $\pi[\![\, [\mathrm{lfp}\, R\bar{x}.\varphi(R,\bar{x})](\bar{a})]\!] = \mathrm{lfp}(F^\varphi_\pi)(\bar{a})$.

The proof is a rather straightforward adaptation of the correctness proof for model-checking games for LFP; see, e.g., [Grädel et al. 2007, Chapter 3.3].

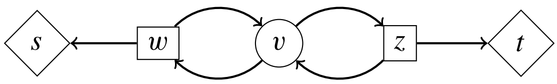
9. Beyond reachability: safety games and greatest fixed points

While the restriction of LFP to its positive fragment comes with no loss of expressive power (on finite structures) and while posLFP is sufficiently powerful to capture a number of interesting and relevant other fixed-point formalisms in computer science, it is nevertheless not really satisfactory. One reason is that the transformation from a fixed-point formula with nonatomic negation into one in posLFP is (contrary to transformations into negation normal form) not a simple syntactic translation. It goes through the stage comparison theorem and can make a formula much longer and more complicated. Further, such transformations are not available for important fixed-point formalism such as the modal μ-calculus, stratified Datalog, transitive closure logics, and even simple temporal languages such as CTL. On the game-theoretic side, reachability games are just the simplest kind of games on graphs, and in many applications players have different and more ambitious goals such as safety, Büchi, parity or Muller objectives. It is thus an important and interesting challenge to lay the foundations of a provenance analysis for full LFP and infinite games with more general objectives, and to apply this approach to the numerous other fixed-point formalisms, in particular in databases and verification.

We defer a detailed treatment of this to forthcoming work. Here we discuss some of the mathematical concepts and challenges that arise in this project, and apply them to the provenance of *safety games*. Recall that the computation of winning positions for safety objectives is a simple, but also in some sense universal, application of greatest fixed points.

The first observation is that we need to impose additional requirements on the semirings that we consider. While ω-continuous semirings are appropriate for a provenance analysis of least fixed points and reachability objectives, they are not always adequate for greatest fixed points. The property of ω-continuity is not sufficient to guarantee the existence of greatest fixed points, and in cases where they exist they do not necessarily provide the information that we are interested in.

Example 42. We consider the game graph



with associated equation system for Player 0 consisting of $X_v = X_w + X_z$, $X_w = f(s) \cdot X_v$, and $X_z = f(t) \cdot X_v$. The least fixed-point solution (in whatever semiring) has values $f(v) = f(w) = f(z) = 0$, which reflects the fact that Player 0 has no strategy to guarantee a finite play. It is not difficult to see that in $\mathbb{N}^\infty[\![s, t]\!]$ this is in fact the unique fixed point, hence in particular the greatest one, which however gives us no information about safety strategies. In $\mathbb{N}^\infty$ instead, under a valuation of the terminal nodes with $f(s) = a \neq 0$ and $f(t) = 0$, we get the greatest fixed point $f(v) = f(w) = \infty$ and $f(z) = 0$. In particular, greatest fixed points do not specialize correctly from $\mathbb{N}^\infty[\![s, t]\!]$ to $\mathbb{N}^\infty$.

We shall see below that we get interesting information on safety strategies by provenance values in the absorptive semiring $\mathbb{S}^\infty[s, t]$.

To make sure that also greatest fixed points of polynomial equation systems exist, we shall require that our semirings are not just ω-continuous, but also *ω-cocontinuous*, i.e., that every descending ω-chain $(a_i)_{i \in \omega}$, with $a_{i+1} \leq a_i$ for all $i \in \omega$, has an infimum $\inf_{i \in \omega} a_i$ in K, which is compatible with the semiring operations in the sense that, for every $c \in K$,

$$c + \inf_{i \in \omega} a_i = \inf_{i \in \omega}(c + a_i) \quad \text{and} \quad c \cdot \inf_{i \in \omega} a_i = \inf_{i \in \omega}(c \cdot a_i).$$

We call such semirings *fully ω-continuous*. Our most important example of such a semiring is $\mathbb{S}^\infty[X]$, the semiring of generalized absorptive polynomials, which we are going to discuss next.

10. Absorptive semirings and generalized absorptive polynomials

Recall that a semiring K is *absorptive* if $a + ab = a$ for all $a, b \in K$, which is equivalent to $1 + a = 1$ for all $a \in K$. Examples include the Viterbi semiring, the tropical semiring, min-max semirings, and further the semiring $\mathbb{S}[X]$ of absorptive polynomials over X. Absorptive semirings are (+)-idempotent and naturally ordered, 1 is the top element, and multiplication decreases elements: $ab \leq b$. In particular, the powers of an element form a descending ω-chain $1 \geq a \geq a^2 \geq \cdots$. If this chain has an infimum then we denote it by a^∞.

In the semiring $\mathbb{S}[X]$, the infima of descending ω-chains $(x^n)_{n<\omega}$ are always 0 and thus not very informative. We therefore complete $\mathbb{S}[X]$ to the semiring $\mathbb{S}^\infty[X]$ by admitting exponents in $\mathbb{N}^\infty$.

Definition 43. Let X be a *finite* set of provenance tokens. A *monomial* over X with exponents from $\mathbb{N}^\infty$ is a function $m : X \to \mathbb{N}^\infty$. Informally, we write m as $x_1^{m(x_1)} \cdots x_n^{m(x_n)}$. Monomial multiplication adds the exponents. Observe also that $x^\infty \cdot x^n = x^\infty$. For any two monomials, m_1, m_2 we say that m_2 *absorbs* m_1 if m_2 has smaller exponents than m_1. Formally, $m_1 \preceq m_2$ if, and only if, $m_1(x) \geq m_2(x)$ for all $x \in X$. Since monomials are functions, this is the pointwise partial order given by the order on $\mathbb{N}^\infty$.

Because $\mathbb{N}^\infty$ is a lattice (with top and bottom) the monomials also inherit a lattice structure. The set of all monomials is, of course, infinite. However, it has some crucial finiteness properties.

Proposition 44. *Every ascending chain and every antichain of monomials is finite.*

Proof. Clearly $(\mathbb{N}^\infty, \leq)$ is a well-order. For any finite set X, the set of monomials $m : X \to \mathbb{N}^\infty$ with the *reverse order* of the absorption order is isomorphic to $(\mathbb{N}^\infty)^k$ with $k = |X|$ and with the componentwise order inherited from $(\mathbb{N}^\infty, \leq)$. This is a well-quasiorder and therefore has no infinite descending chains and no infinite antichains. This implies that in the set of monomials over X with the absorption order, all ascending chains and all antichains are finite. □

Definition 45. We define $\mathbb{S}^\infty[X]$ as the set of antichains of monomials with indeterminates from X and exponents in $\mathbb{N}^\infty$. Writing an antichain as a (formal) sum of its monomials we identify it with a polynomial with coefficients 0 or 1, and call these *generalized absorptive polynomials*. We define polynomial addition and multiplication as usual, except that for coefficients $1 + 1 = 1$, and that we keep only the maximal monomials in the result. The empty antichain corresponds to the 0 polynomial. The 1 polynomial consists of just the monomial in which every indeterminate has exponent 0.

Proposition 46. $(\mathbb{S}^\infty[X], +, \cdot, 0, 1)$ *is an absorptive commutative semiring. Further it is a complete lattice with respect to the natural order, which is fully ω-continuous and moreover completely distributive.*

As a consequence, we can compute not only least fixed point solutions for systems of polynomial equations but also greatest fixed points. In contrast to other semirings with such properties, such as for instance the Viterbi semiring, $\mathbb{S}^\infty[X]$ has one further crucial property. It is *chain-positive* which means that the infimum of every chain of nonzero elements is also nonzero.

As in other semirings of polynomials and power series we can also here take pairs of positive and negative indeterminates, with a correspondence $X \leftrightarrow \overline{X}$ and build the quotient with respect to the congruence generated by the equation $x \cdot \bar{x} = 0$. We thus obtain a new semiring $\mathbb{S}^\infty[X, \overline{X}]$ which provides a natural framework for a provenance analysis for full LFP and other fixed-point calculi. We shall develop this in forthcoming work.

Here we use the semiring $\mathbb{S}^\infty[T]$ to describe a provenance analysis for safety games where T is the set of terminal positions of the given game graph.

11. Absorption among strategies

Definition 47. Let $\mathcal{G} = (V, V_0, V_1, T, E)$ be a finite game graph, and $v \in V$. For two strategies $\mathcal{S}, \mathcal{S}' \in \mathrm{Strat}_\sigma(v)$, we say that $\mathcal{S}$ *absorbs* $\mathcal{S}'$ (in symbols $\mathcal{S} \succeq_a \mathcal{S}'$) if

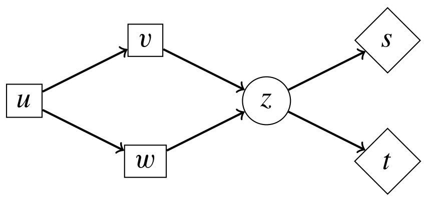
- for all $t \in T$, $\mathcal{S}$ admits at most as many plays with outcome t as $\mathcal{S}'$ does, and
- if $\mathcal{S}$ admits an infinite play, then so does $\mathcal{S}'$.

We call $\mathcal{S}$ *absorption-dominant* if it is maximal with respect to $\succeq_a$.

Absorption-dominant strategies are interesting both for games in general and for logic because they can win "with minimal effort". As a simple example, consider a model-checking game for a formula $\varphi \vee (\varphi \wedge \psi)$. The Verifier can either establish φ or $\varphi \wedge \psi$, but any strategy that establishes the truth of $\varphi \wedge \psi$ will have more plays and more outcomes than one that proves just φ, and will thus be absorbed by it. The absorption-dominant strategies for $\varphi \vee (\varphi \wedge \psi)$ are thus precisely the absorption-dominant strategies for φ.

Notice however that, despite this minimality, absorption-dominant strategies need not be positional, not even in acyclic games.

Example 48. Consider the game



There are four strategies in $\mathrm{Strat}_0(u)$ with provenance values s^2, st, st, and t^2. The positional ones are those with values s^2 and t^2, but all four strategies are absorption-dominant.

However, absorption-dominant strategies are *weakly positional* in the sense that if a node is reached several times during the same play, then, without loss of strategic power, the player can always make the same choice at that node. Absorption among strategies makes sense for both acyclic and cyclic games. In acyclic games, absorption-dominant strategies are described by provenance polynomials in $\mathbb{S}[T]$ (with only finite exponents). But they are even more interesting for the analysis of reachability *and safety* games that admit infinite plays. The fundamental difference between valuations for reachability and safety strategies concerns the valuations of infinite plays. If, as we assume here, reachability and safety goals are defined for terminal nodes, then an infinite play is losing for every reachability objective but winning for every safety objective. As a consequence, the strategies $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$ that enforce all plays to be nonterminating absorb all other strategies in $\mathrm{Strat}_\sigma(v)$ that admit at least one infinite play.

We thus extend the valuations of plays in a game $\mathcal{G}$ (with finite game graph that may contain cycles) to two different valuation functions f_σ^μ and f_σ^ν. For simplicity, we assume trivial valuations on the edges, so for a finite play x ending in t, we just put $f_\sigma^\mu(x) = f_\sigma^\nu(x) = f_\sigma(t)$ but if x is an infinite play, we put $f_\sigma^\mu(x) = 0$ and $f_\sigma^\nu(x) = 1$.

A strategy $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$ may well admit an infinite set of plays. Taking the semiring $\mathbb{S}^\infty[T]$ with the basic valuation $f_\sigma(t) := t$ for the terminal nodes, strategies are described by monomials (or 0), and we put

$$F^\mu(\mathcal{S}) := \prod_{x \in \text{Plays}(\mathcal{S})} f^\mu_\sigma(x) \quad \text{and} \quad F^\nu(\mathcal{S}) := \prod_{x \in \text{Plays}(\mathcal{S})} f^\nu_\sigma(x) = \prod_{t \in T} t^{\#_\mathcal{S}(t)}.$$

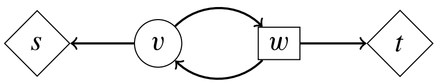
We extend the absorption order $\succeq$ on monomials by $m \succeq 0$ for all m.

Lemma 49. *Let $\mathcal{G}$ be any finite game graph, with valuations of strategies in $\mathbb{S}^\infty[T]$ induced by $f_\sigma(t) = t$ for all $t \in T$. For all strategies $\mathcal{S}, \mathcal{S}' \in \text{Strat}_\sigma(v)$ we have:*

- $0 \neq F^\nu(\mathcal{S}) \succeq F^\mu(\mathcal{S}) \neq 1$.
- *$F^\mu(\mathcal{S}) = 0$ if, and only if, $\mathcal{S}$ admits an infinite play. Otherwise $F^\mu(\mathcal{S}) = F^\nu(\mathcal{S})$.*
- *$F^\nu(\mathcal{S}) = 1$ if, and only if, $\mathcal{S}$ admits only infinite plays.*
- *$\mathcal{S}$ absorbs $\mathcal{S}'$ if, and only if, both $F^\nu(\mathcal{S}) \succeq F^\nu(\mathcal{S}')$ and $F^\mu(\mathcal{S}) \succeq F^\mu(\mathcal{S}')$.*

Proof. Only the last item requires proof. Suppose that $\mathcal{S}$ absorbs $\mathcal{S}'$. If $\mathcal{S}$ admits only finite plays, then $F^\mu(\mathcal{S}) = F^\nu(\mathcal{S}) \succeq F^\nu(\mathcal{S}') \succeq F^\mu(\mathcal{S}')$. If $\mathcal{S}$ admits an infinite play, then so does $\mathcal{S}'$ and $F^\nu(\mathcal{S}) \succeq F^\nu(\mathcal{S}') \succeq F^\mu(\mathcal{S}') = F^\mu(\mathcal{S}) = 0$. In both cases, $F^\nu(\mathcal{S}) \succeq F^\nu(\mathcal{S}')$ and $F^\mu(\mathcal{S}) \succeq F^\mu(\mathcal{S}')$. Conversely, assume that $\mathcal{S}$ does not absorb $\mathcal{S}'$. Then either there is a terminal t such that $\mathcal{S}$ admits more plays with outcome t than $\mathcal{S}'$ does, or $\mathcal{S}$ admits an infinite play, but $\mathcal{S}'$ does not. In the first case, $F^\nu(\mathcal{S}) \not\succeq F^\nu(\mathcal{S}')$ and in the second case $0 = F^\mu(\mathcal{S}) \not\succeq F^\mu(\mathcal{S}') \neq 0$. ☐

Example 50. We return to the game described in Example 39



with equation system G_0 consisting of $X_v = s + X_w$ and $X_w = t \cdot X_v$. In $\mathbb{N}^\infty[\![s,t]\!]$ the least fixed-point solution is $f(v) = s \cdot (1 + t + t^2 \cdots)$ and $f(w) = s \cdot (t + t^2 + \cdots)$. In $\mathbb{S}^\infty[s,t]$ the least fixed-point solution $f^\mu = \text{lfp}\, G_0$ has values $f^\mu(v) = s$ and $f^\mu(w) = st$, which describes the possible outcomes of the unique absorption-dominant strategy that enforces finite plays. The only other absorption-dominant strategy (moving from v to w) has value 0 because it admits an infinite play.

However, the greatest fixed-point solution $f^\nu = \text{gfp}\, G_0$ of this equation system in $\mathbb{S}^\infty[s,t]$ has values $f^\nu(v) = s + t^\infty$ and $f^\nu(w) = st + t^\infty$. Here this second strategy has value t^∞ since it admits infinitely many plays ending in t (and one infinite play with value 1).

Theorem 51. *Let $\mathcal{G} = (V, V_0, V_1, T, E)$ be a game graph and let G_σ be the associated equation system for Player σ. In the semiring $\mathbb{S}^\infty[T]$ this system has least and greatest fixed point solutions* lfp G_σ *and* gfp G_σ *with*

$$(\text{lfp}\, G_\sigma)(v) := \sum_{\mathcal{S} \in \text{Strat}_\sigma(v)} F^\mu(\mathcal{S}) \quad \textit{and} \quad (\text{gfp}\, G_\sigma)(v) := \sum_{\mathcal{S} \in \text{Strat}_\sigma(v)} F^\nu(\mathcal{S}).$$

The values of these sums do not change if we restrict them to the absorption-dominant strategies.

Proof. Since $\mathbb{S}^\infty[T]$ is ω-continuous, the claim for (lfp G_σ) follows from Theorem 36. For the greatest fixed-point solution we use that $\mathbb{S}^\infty[T]$ is also ω-cocontinuous and has the structure of a complete lattice. Thus, (gfp G_σ) is the limit of the descending chain $(G^n)_{n<\omega}$ of approximants starting with $G^0 = 1$, and G^{n+1} is defined by applying the equation system to $G^n : V \to \mathbb{S}^\infty[T]$.

As in the proof of Theorem 36 we argue with the unfoldings $\mathcal{G}^n$ of $\mathcal{G}$ up to $n-1$ moves, but we now put $f^n_\sigma(\pi v) = 1$ for the final node of an "unfinished" play, i.e., with $|\pi| = n-1$ and $v \in V \setminus T$. The valuations f^n_σ extend to all nodes of the (acyclic) game $\mathcal{G}^n$, and again, coincide with the Kleene approximants G^n: for every n and every v we have $G^n(v) = f^n_\sigma(v)$. The different valuation of the terminal nodes in $\mathcal{G}^n$ also has the effect that for any $\mathcal{T} \in \mathrm{Strat}^{(n)}_\sigma(v)$ we have $F(\mathcal{T}) = \prod_{t \in T} t^{\#_\mathcal{T}(t)}$, which is a monomial with only finite exponents. Since $\mathcal{G}^n$ is acyclic

$$f^n_\sigma(v) = \sum_{\mathcal{T} \in \mathrm{Strat}^{(n)}_\sigma(v)} F(\mathcal{T}).$$

Every strategy $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$ for the original game $\mathcal{G}$ induces for each game $\mathcal{G}^n$ a strategy $\mathcal{S}^{(n)}$. Conversely, every strategy $\mathcal{T} \in \mathrm{Strat}^{(n)}_\sigma(v)$ is induced by at least one strategy $\mathcal{S} \in \mathrm{Strat}_\sigma(v)$. Since the semiring $\mathbb{S}^\infty[T]$ is idempotent, we have that, for every $n < \omega$,

$$\sum_{\mathcal{S} \in \mathrm{Strat}_\sigma(v)} F(\mathcal{S}^{(n)}) = \sum_{\mathcal{T} \in \mathrm{Strat}^{(n)}_\sigma(v)} F(\mathcal{T}).$$

As graphs, the sequence $(\mathcal{S}^{(n)})_{n \in \omega}$ of induced strategies is increasing, i.e., $\mathcal{S}^{(1)} \subseteq S^{(2)} \subseteq \cdots$, but their values in $\mathbb{S}^\infty[T]$ are decreasing, i.e., $F(\mathcal{S}^{(1)}) \succeq F(S^{(2)}) \subseteq \cdots$. Further, $F^\nu(\mathcal{S}) = \prod_{t \in T} t^{\#_\mathcal{S}(t)}$ with exponents $\#_\mathcal{S}(t) \in \mathbb{N}^\infty$ and the corresponding exponents in the monomial $F(\mathcal{S}^{(n)})$ tell us how often a terminal position $t \in T$ has been reached by $\mathcal{S}$ after $n-1$ moves. In particular,

$$F^\nu(\mathcal{S}) = \lim_{n \to \infty} F(\mathcal{S}^{(n)}).$$

It is clear that these limits commute with summation over strategies, so we have

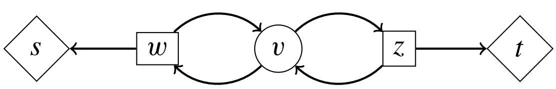
$$\sum_{\mathcal{S} \in \mathrm{Strat}_\sigma(v)} F^\nu(\mathcal{S}) = \lim_{n \to \infty} \sum_{\mathcal{S} \in \mathrm{Strat}_\sigma(v)} F(\mathcal{S}^{(n)}) = \lim_{n \to \infty} \sum_{\mathcal{T} \in \mathrm{Strat}^{(n)}_\sigma(v)} F(\mathcal{T}).$$

Putting everything together we get

$$(\mathrm{gfp}\, G_\sigma)(v) = \lim_{n \to \infty} G^n(v) = \lim_{n \to \infty} f^n_\sigma(v) = \lim_{n \to \infty} \sum_{\mathcal{T} \in \mathrm{Strat}^{(n)}} F(\mathcal{T}) = \sum_{\mathcal{S} \in \mathrm{Strat}_\sigma(v)} F^\nu(\mathcal{S}). \qquad \square$$

These least and greatest fixed points give precise descriptions of the absorption-dominant reachability and safety strategies of the players for each position of the game.

Example 52. We return to the Example 42:



Recall that the associated equation system for Player 0 has the equations $X_v = X_w + X_z$, $X_w = f(s) \cdot X_v$, and $X_z = f(t) \cdot X_v$.

The greatest fixed-point solution in $\mathbb{S}^\infty[s,t]$, computed by iterating from the top element $f = 1$ results in $f^\nu(v) = s^\infty + t^\infty$, $f^\nu(w) = s^\infty + st^\infty$", and $f^\nu(z) = s^\infty t + t^\infty$. Notice that indeed, $f^\nu(v) = f^\nu(w) + f^\nu(z)$ because st^∞ is absorbed by t^∞, and $s^\infty t$ by s^∞. The greatest fixed-point solution indicates that Player 0 has two absorptive strategies (move always to w or move always to z), and gives, for each of the terminal nodes s and t the number of plays ending in that node that the strategy admits. For instance, if the safety objective requires avoiding t, then v and w, the strategy moving to w has infinitely many winning plays ending in s (and one nonterminating play with value 1), but since $f(z)[s, 0] = 0$, Player 0 has no safety strategy from z that avoids t.

12. Outlook

In this paper we have extended the semiring framework for provenance analysis by new elements, so that it can be applied to logics with negation, in particular first-order logic and fixed-point logics, and to an analysis of games that provides detailed information about the number and properties of the strategies of the players.

Our treatment of negation is based on transformations to negation normal form and the use of newly introduced semirings of dual-indeterminate polynomials and dual-indeterminate power series. In particular, ω-continuous semirings $\mathbb{N}^{\infty}[\![X, \overline{X}]\!]$ of dual-indeterminate power series provide an adequate general framework for logics with least fixed points, such as posLFP (and Datalog) and the semiring of absorptive generalized dual-indeterminate polynomials $\mathbb{S}^{\infty}[\![X, \overline{X}]\!]$ permits an adequate treatment of greatest fixed points. We have thus laid foundations for a provenance analysis of general fixed-point logics, and we are currently applying this also to modal, temporal, and dynamic logics.

On the level of games, we have seen that provenance valuations in ω-continuous and absorptive semirings give us very detailed information about strategies for possibly infinite games with reachability and safety objectives. We are currently expanding this to games with more complicated objectives, such as Büchi, co-Büchi or parity games. Since these objectives no longer depend on terminal nodes but on the data occurring in infinite plays, a somewhat different framework has to be used, depending for instance on basic valuations of the edges of the game graph.

References

[Amsterdamer et al. 2011a] Y. Amsterdamer, D. Deutch, and V. Tannen, "On the limitations of provenance for queries with difference", presented at the 3rd Workshop on the Theory and Practice of Provenance, TaPP'11, 2011, available at https://www.usenix.org/legacy/events/tapp11/tech/final_files/Amsterdamer.pdf.

[Amsterdamer et al. 2011b] Y. Amsterdamer, D. Deutch, and V. Tannen, "Provenance for aggregate queries", pp. 153–164 in *PODS '11: proceedings of the 30th ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems* (Athens, Greece), Association for Computing Machinery, New York, 2011.

[Apt and Grädel 2011] K. Apt and E. Grädel (editors), *Lectures in game theory for computer scientists*, Cambridge University Press, 2011. Zbl

[Dannert and Grädel 2019] K. M. Dannert and E. Grädel, "Provenance analysis: a perspective for description logics?", pp. 266–285 in *Description logic, theory combination, and all that*, edited by C. Lutz et al., Lecture Notes in Comput. Sci. **11560**, Springer, 2019. MR

[Dannert and Grädel 2020] K. Dannert and E. Grädel, "Semiring provenance for guarded logics", preprint, 2020, available at https://www.logic.rwth-aachen.de/pub/DannertGraedel.pdf. To appear in *Hajnal Andréka and István Németi on unity of science: from computing to relativity theory through algebraic logic*.

[Deutch et al. 2014] D. Deutch, T. Milo, S. Roy, and V. Tannen, "Circuits for datalog provenance", pp. 201–212 in *Database theory — ICDT 2014*, edited by N. Schweikardt et al., OpenProceedings, Konstanz, 2014. MR

[Foster et al. 2008] J. Foster, T. Green, and V. Tannen, "Annotated XML: queries and provenance", pp. 271–280 in *PODS '08: proceedings of the 27th ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems* (Vancouver, BC, 2008), Association for Computing Machinery, New York, 2008.

[Geerts and Poggi 2010] F. Geerts and A. Poggi, "On database query languages for K-relations", *J. Appl. Log.* **8**:2 (2010), 173–185. MR Zbl

[Geerts et al. 2016] F. Geerts, T. Unger, G. Karvounarakis, I. Fundulaki, and V. Christophides, "Algebraic structures for capturing the provenance of SPARQL queries", *J. ACM* **63**:1 (2016), art. id. 7. MR Zbl

[Grädel and Tannen 2017] E. Grädel and V. Tannen, "Semiring provenance for first-order model checking", preprint, 2017. arXiv

[Grädel et al. 2007] E. Grädel, P. G. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Y. Vardi, Y. Venema, and S. Weinstein, *Finite model theory and its applications*, Springer, 2007. MR

[Green 2011] T. J. Green, "Containment of conjunctive queries on annotated relations", *Theory Comput. Syst.* **49**:2 (2011), 429–459. MR Zbl

[Green and Tannen 2017] T. Green and V. Tannen, "The semiring framework for database provenance", pp. 93–99 in *PODS '17: Proceedings of the 36th ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems* (Chicago, IL), Association for Computing Machinery, New York, 2017.

[Green et al. 2007] T. Green, G. Karvounarakis, and V. Tannen, "Provenance semirings", pp. 31–40 in *PODS '07: Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems* (Beijing, China), Association for Computing Machinery, New York, 2007.

[Green et al. 2009] T. Green, G. Karvounarakis, and V. Tannen, "Reconcilable differences", pp. 212–224 in *ICDT '09: Proceedings of the 12th International Conference on Database Theory*, Association for Computing Machinery, New York, 2009.

[Kuich 1997] W. Kuich, "Semirings and formal power series: their relevance to formal languages and automata", pp. 609–677 in *Handbook of formal languages, Vol. 1*, edited by G. Rozenberg and A. Salomaa, Springer, 1997. MR

[Tannen 2013] V. Tannen, "Provenance propagation in complex queries", pp. 483–493 in *In search of elegance in the theory and practice of computation*, edited by V. Tannen et al., Lecture Notes in Comput. Sci. **8000**, Springer, 2013. MR Zbl

[Tannen 2017] V. Tannen, "Provenance analysis for FOL model checking", *ACM SIGLOG News* **4**:1 (2017), 24–36.

[Xu et al. 2018] J. Xu, W. Zhang, A. Alawini, and V. Tannen, "Provenance analysis for missing answers and integrity repairs", *IEEE Data Eng. Bull.* **41**:1 (2018), 39–50.

Received 2 Aug 2019.

ERICH GRÄDEL:

graedel@logic.rwth-aachen.de
RWTH Aachen University, Aachen, Germany

VAL TANNEN:

val@cis.upenn.edu
University of Pennsylvania, Philadelphia, PA, United States

Moscow Journal of Combinatorics and Number Theory
Vol. 9, No. 3, 2020
https://doi.org/10.2140/moscow.2020.9.229

Generalizations of k-dimensional Weisfeiler–Leman stabilization

Anuj Dawar and Danny Vagnozzi

The family of Weisfeiler–Leman equivalences on graphs is a widely studied approximation of graph isomorphism with many different characterizations. We study these and other approximations of isomorphism defined in terms of refinement operators and Schurian polynomial approximation schemes (SPAS). The general framework of SPAS allows us to study a number of parameters of the refinement operators based on Weisfeiler–Leman refinement, logic with counting, lifts of Weisfeiler–Leman as defined by Evdokimov and Ponomarenko, the invertible map test introduced by Dawar and Holm, and variations of these, as well as to establish relationships between them.

1. Introduction

For convenience, we shall treat graphs as *arc-coloured complete digraphs*; that is to say, as *labelled partitions* of the set of ordered pairs of vertices (hereafter, we refer to the latter as *arcs*). For example, an undirected simple graph can be seen as a partition of its arcs into edges, nonedges, and loops. As such, the *graph isomorphism problem* is that of deciding whether there is a colour-preserving bijection between the sets of vertices of two given graphs. Computationally, this problem is polynomial-time equivalent to finding the orbits of the induced action of the automorphism group of a given graph G on a fixed power of its vertex set V [Mathon 1979]. For short, we refer to the partition of V^k (for any fixed k) obtained by this action as the *orbit partition* of V^k. The graph isomorphism problem (and likewise the problem of determining the orbit partition) is neither known to be solvable in polynomial time nor known to be **NP**-complete. The best known upper bound to their computational time is quasipolynomial. This follows from the well-known result by Babai [2016].

The *classical Weisfeiler–Leman* (WL) algorithm is a well-known method for approximating the orbits of the induced action of the automorphism group of a given graph on the set of pairs of arcs. It can be seen as a generalization of the so-called *naïve colour refinement*. Given a graph G, the WL algorithm produces a *coherent configuration*, which is a partition of the set of arcs of G satisfying certain stability conditions (see Section 2 for the definition). A natural generalization of this algorithm was given by Babai: for each $k \in \mathbb{N}$, the *k-dimensional Weisfeiler–Leman* (WL_k) algorithm outputs a labelled partition of k-tuples of vertices satisfying a similar stability condition and respecting local isomorphism. The running time of the WL_k algorithm on a graph with n vertices is bounded by $n^{O(k)}$. The case $k = 1$ coincides with the naïve colour refinement, and $k = 2$ with the classical Weisfeiler–Leman algorithm.

It follows from a result by Cai, Fürer, and Immerman [Cai et al. 1992] that there is no fixed $k \in \mathbb{N}$ such that for all graphs the k-dimensional Weisfeiler–Leman algorithm outputs the partition of k-tuples of vertices into orbits of the induced action of the automorphism group of the input graph. Indeed, the

MSC2010: primary 05E15, 05E30, 05E99; secondary 03D15.
Keywords: Weisfeiler–Leman, combinatorics, invertible map, graph isomorphism, complexity theory, coherent configuration.

authors show how to construct a graph with $O(k)$ vertices for which WL_k fails to produce the partition into such orbits. Thus, their result implies that a partition induced by this group action can be obtained for all graphs on n vertices only if one chooses k to be $\Omega(n)$. One can informally claim that the strength of the k-dimensional Weisfeiler–Leman algorithm increases with k. More precisely, for unlabelled partitions $\mathcal{P}$ and $\mathcal{Q}$ of some set A we write $\mathcal{P} \preceq_A \mathcal{Q}$, and say $\mathcal{Q}$ is a refinement of $\mathcal{P}$; if, whenever $a, b \in A$ are in the same equivalence class of $\mathcal{Q}$, they are also in the same equivalence class of $\mathcal{P}$. By viewing a labelled partition of A as a function $\gamma : A \to L$ to a set of labels L (which we sometimes refer to as the *colour* set), the unlabelled partition induced by γ is $\{\gamma^{-1}(l) \mid l \in L\}$. We extend the partial order $\preceq_A$ to labelled partitions by writing $\gamma \preceq_A \rho$ to mean that the unlabelled partition induced by ρ refines that induced by γ. Note that this does not require that the codomains of γ and ρ are the same. We omit the subscript A when the set is clear from the context. For a graph Γ, define $\overline{\mathrm{WL}}_1(\Gamma) = \Gamma$; and for $k \geq 2$, set $\overline{\mathrm{WL}}_k(\Gamma)$ to be the labelled partition of the set of arcs induced by the output of the k-dimensional Weisfeiler–Leman algorithm on input Γ. We can now state the following:

$$\overline{\mathrm{WL}}_1(\Gamma) \preceq \overline{\mathrm{WL}}_2(\Gamma) \preceq \cdots \preceq \overline{\mathrm{WL}}_n(\Gamma) = \overline{\mathrm{WL}}_{n+1}(\Gamma) = \cdots = \overline{\mathrm{WL}}_\infty(\Gamma),$$

where n is the number of vertices of Γ and $\overline{\mathrm{WL}}_\infty(\Gamma)$ is the partition into the orbits of the induced action of the automorphism group on arcs. Also,

$$\overline{\mathrm{WL}}_l(\overline{\mathrm{WL}}_k(\Gamma)) = \overline{\mathrm{WL}}_k(\Gamma)$$

for all $l, k \in \mathbb{N}$ with $l \leq k$. This shows that the family of maps from the set of arc-coloured complete digraphs to itself $\{\overline{\mathrm{WL}}_1, \overline{\mathrm{WL}}_2, \dots\}$ forms a *Schurian polynomial approximation scheme* in the following sense, as defined in [Evdokimov and Ponomarenko 1999]:

Definition 1.1 (Schurian polynomial approximation scheme). A family of mappings $\{X_1, X_2, \dots\}$ is said to form a *Schurian polynomial approximation scheme* (SPAS) if for any graph Γ with vertex set V:

(1) $X_k(\Gamma)$ is a graph with vertex set V for all $k \in \mathbb{N}$.

(2) $X_1(\Gamma) \preceq X_2(\Gamma) \preceq \cdots \preceq X_n(\Gamma) = X_{n+1}(\Gamma) = \cdots = X_\infty(\Gamma) = \mathrm{Sch}(\Gamma)$, where $n = |V|$ and $\mathrm{Sch}(\Gamma)$ is the partition of arcs into orbits of the induced action of the automorphism group of Γ.

(3) $X_l(X_m(\Gamma)) = X_m(\Gamma)$ for all $l, m \in \mathbb{N}$ with $l \leq m$.

(4) $X_k(\Gamma)$ is computable in time $n^{O(k)}$.

More informally, one can think of SPAS as a collection of polynomially computable functions indexed by $\mathbb{N}$, each of which takes as input a partition and refines it. Moreover, the higher the index, the closer the output partition is to the orbit partition of the set of arcs.

Definition 1.2 (dominance and equivalence of SPAS). For any two SPAS, $\mathcal{S}_X = \{X_1, X_2, \dots\}$ and $\mathcal{S}_Y = \{Y_1, Y_2, \dots\}$:

(1) we say $\mathcal{S}_X$ *dominates* $\mathcal{S}_Y$, and write $\mathcal{S}_Y \preceq \mathcal{S}_X$, if for each $k \in \mathbb{N}$ there is some $k' \in \mathbb{N}$ such that $Y_k(\Gamma) \preceq X_{k'}(\Gamma)$ for all graphs Γ.

(2) we say $\mathcal{S}_X$ is *equivalent* to $\mathcal{S}_Y$, and write $\mathcal{S}_X \simeq \mathcal{S}_Y$, if $\mathcal{S}_X \preceq \mathcal{S}_Y$ and $\mathcal{S}_Y \preceq \mathcal{S}_X$.

(3) we say $\mathcal{S}_X$ *strictly dominates* $\mathcal{S}_Y$ if $\mathcal{S}_Y \preceq \mathcal{S}_X$, but $\mathcal{S}_X \not\preceq \mathcal{S}_Y$.

In this paper, we deal with the following SPAS: $\mathcal{S}_{\mathbf{WL}}$, $\mathcal{S}_{\mathbf{C}}$, $\mathcal{S}_{\mathbf{C},r}$, $\mathcal{S}_{\mathbf{WL},r}$, and $\mathcal{S}_{\mathbf{IM}(\mathbb{F})}$. Such schemes arise from considering fixed points of *refinement operators*. The formal definition of *refinement operators* is given in Section 4. These are operators that take a partition γ, of V^k, to a refinement of itself. The SPAS $\mathcal{S}_{\mathbf{WL}}$ and $\mathcal{S}_{\mathbf{C}}$ arise from the following well-known concepts: Babai's generalization of the classical Weisfeiler–Leman algorithm for the former and first order logic with counting quantifiers for the latter. In each case, the label of a tuple $\vec{v} \in V^k$ in the refined partition is determined by its label in γ and the partition of V that is induced by considering $\gamma(\vec{u})$ for the tuples $\vec{u}$ obtained by substituting elements of V in $\vec{v}$. For $r \in \mathbb{N}$, $\mathcal{S}_{\mathbf{WL},r}$ and $\mathcal{S}_{\mathbf{C},r}$ are further generalizations of $\mathcal{S}_{\mathbf{WL}}$ and $\mathcal{S}_{\mathbf{C}}$, respectively. In these generalizations, the label associated to each k-tuple $\vec{v}$ is determined by $\gamma(\vec{v})$ and the partition of V^r obtained by considering $\gamma(\vec{u})$ for tuples $\vec{u}$ obtained by substituting r-tuples in $\vec{v}$. Formal definitions of these are given later. Here we note that our first result shows that the parameter r does not strengthen the SPAS $\mathcal{S}_{\mathrm{WL}}$ and $\mathcal{S}_{\mathbb{C}}$.

Theorem 1.3. *For any $r \in \mathbb{N}$, $\mathcal{S}_{\mathbf{WL}} \simeq \mathcal{S}_{\mathbf{C}} \simeq \mathcal{S}_{\mathbf{WL},r} \simeq \mathcal{S}_{\mathbf{C},r}$.*

The reasons for considering the additional parameter r is that it appears to be of interest in another scheme we consider. The scheme $\mathcal{S}_{\mathbf{IM}(\mathbb{F})}$ arises from the *invertible map game* introduced in [Dawar and Holm 2017]. It has been shown to have a close relationship to logics with linear algebraic operators [Dawar et al. 2019] over a field $\mathbb{F}$. The associated refinement operator $\mathrm{IM}_k^{\mathbb{F}}$ maps each k-tuple of vertices $\vec{v}$ and partition γ to a tuple of matrices, and the colour associated to $\vec{v}$ by the refinement of γ is determined by the equivalence class of this tuple of matrices under simultaneous similarity.

For this SPAS we prove the following results:

Theorem 1.4. *For any field $\mathbb{F}$ of characteristic 0, $\mathcal{S}_{\mathbf{IM}(\mathbb{F})} \simeq \mathcal{S}_{\mathbf{WL}}$.*

Theorem 1.5. *For any field $\mathbb{F}$ of positive characteristic, $\mathcal{S}_{\mathbf{IM}(\mathbb{F})}$ strictly dominates $\mathcal{S}_{\mathbf{WL}}$.*

The paper is structured as follows: after a brief overview of the required notions on coherent configurations and algebras, we formally define and discuss the concepts of refinement operators and procedures. We then prove Theorem 1.3 and use a similar method to show the equivalence between $\mathcal{S}_{\mathbf{WL}}$ and $\mathcal{S}_{\mathrm{EP}}$, a SPAS introduced by Evdokimov and Ponomarenko in [1999]. The final two sections contain the proofs of Theorems 1.4 and 1.5 and a short discussion on a variant of the SPAS $\mathcal{S}_{\mathbf{IM}(\mathbb{F})}$, namely $\mathcal{S}_{\mathbf{IMt}(\mathbb{F})}$. This variant is motivated by looking at the difference between the definitions of $\mathcal{S}_{\mathbf{WL}}$ and $\mathcal{S}_{\mathbf{C}}$, and applying a similar variation to the definition of $\mathcal{S}_{\mathbf{IM}(\mathbb{F})}$. The discussion leads to a proof of the following:

Theorem 1.6. *For any field $\mathbb{F}$, $\mathcal{S}_{\mathbf{IM}(\mathbb{F})} \simeq \mathcal{S}_{\mathbf{IMt}(\mathbb{F})}$.*

Throughout the text, all sets are finite. Given two sets V and I, a tuple in V^I is denoted by $\vec{v}$, and its i-th entry by v_i, for each $i \in I$. We use the notation $(v_i)_{i\in I}$ to denote the element of V^I with i-th element equal to v_i. We set $[k] = \{1, 2, \dots, k\} \subset \mathbb{N}$ and $[k]^{(r)} = \{\vec{x} \in [k]^r \mid x_i \neq x_j \ \forall i, j \in [r],\ i \neq j\}$. Recall that a labelled partition of a set A is a function $\gamma : A \to \operatorname{Im}\gamma$. The class of all labelled partitions of A is denoted by $\mathcal{P}(A)$. Recall also that for $\gamma, \rho \in \mathcal{P}(A)$, $\gamma \preceq \rho$ denotes that the unlabelled partition induced by ρ is a refinement of that induced by γ. If $\gamma \preceq \rho$ and $\gamma \succeq \rho$ are both satisfied, we write $\gamma \approx \rho$. Note that $\gamma \approx \rho$ does not imply $\gamma = \rho$, as they may have different codomains. The equivalence class of $a \in A$ with respect to the partition γ is denoted by $[a]_\gamma$. Fix some set V and $k, r \in \mathbb{N}$ with $r \leq k$. For any $\vec{v} \in V^k$, $\vec{u} \in V^r$, and $\vec{\imath} \in [k]^{(r)}$ we define $\vec{v}\langle \vec{\imath}, \vec{u} \rangle \in V^k$ to be the tuple with entries

$$(\vec{v}\langle \vec{\imath}, \vec{u} \rangle)_j = \begin{cases} u_{i_s} & \text{if } j = i_s \text{ for some } s \in [r], \\ v_j & \text{otherwise.} \end{cases}$$

Given two tuples $\vec{v} \in V^r$ and $\vec{w} \in V^s$, their *concatenation* is denoted by $\vec{v} \cdot \vec{w} \in V^{r+s}$. More precisely, $\vec{v} \cdot \vec{w}$ is the tuple with entries

$$(\vec{v} \cdot \vec{w})_i = \begin{cases} v_i & \text{if } i \in [r], \\ w_j & \text{if } i = j + r. \end{cases}$$

For a relation $R \subseteq V^2$, we define the adjacency matrix of R to be the $V \times V$ matrix whose (u, v)-entry is 1 if $(u, v) \in R$ and 0 otherwise. The set of multisets of elements of V is denoted by $\mathrm{Mult}(V)$, and the multiset of entries of a tuple $\vec{v} \in V^I$ is denoted by $\{\{v_i \mid i \in I\}\}$. For all $\gamma \in \mathcal{P}(V^k)$ and natural numbers $r \leq k$, set $\Phi^{\gamma, r} = \mathrm{Im}\, \gamma^{[k]^{(r)}}$.

2. Coherent configurations and coherent algebras

This section introduces notions on coherent configurations and algebras necessary throughout the paper. For a more in-depth account see [Chen and Ponomarenko 2019] or [Cameron 1999]. Our formulation is, in general, different from the more traditional treatment, as we deal with labelled partitions and extend the notion of coherent algebras to arbitrary fields. Also note that rainbows and coherent configurations were originally defined for *unlabelled partitions*. Thus, Definitions 2.1 and 2.2 define, strictly speaking, a *labelled* rainbow and a *labelled* coherent configuration, respectively.

Definition 2.1 (rainbow). A labelled partition ρ of V^2 is said to be a *rainbow* on V if:

(1) There is a set $\mathcal{I} \subseteq \mathrm{Im}\, \rho$ such that

$$\bigcup_{\sigma \in \mathcal{I}} \{\vec{x} \in V^2 \mid \rho(\vec{x}) = \sigma\} = \{(v, v) \in V^2 \mid v \in V\}. \tag{1}$$

(2) For all $(u, v), (u', v') \in V^2$, $\rho(u, v) = \rho(u', v') \iff \rho(v, u) = \rho(v', u')$.

We set $\mathrm{Cel}(\rho) = \{U \subset V \mid \exists \sigma \in \mathrm{Im}\, \rho, \rho(u, u) = \sigma, \forall u \in U\}$ and call its elements the *cells* of ρ.

It was stated in the introduction that, in this paper, graphs are viewed as partitions of the set of their arcs, hence as arc-coloured complete digraphs. For example, an uncoloured loop-free undirected graph can be seen as a complete digraph with its arcs partitioned into three colour classes: edges, nonedges, and loops. Hence, we can always see a graph as a rainbow in the above sense. This view is natural, since our interest is in the partition into orbits of the induced action of the automorphism group on arcs, and this partition is, necessarily, a rainbow. Furthermore, given any group action on V, the partition into the orbits of the induced action on V^2 forms a *coherent configuration* [Cameron 1999][1]:

Definition 2.2 (coherent configuration). A rainbow ρ on V is said to be a *coherent configuration* on V if for each $\sigma, \tau, \kappa \in \mathrm{Im}\, \rho$, there is a constant $p^{\kappa}_{\sigma\tau}$ such that for any $(u, v) \in V^2$ with $\rho(u, v) = \kappa$,

$$\left|\{x \in V \mid \rho(u, x) = \sigma, \rho(x, v) = \tau\}\right| = p^{\kappa}_{\sigma\tau}.$$

[1] Although all group actions give rise to coherent configurations, not all coherent configurations arise from group actions; see [Cameron 1999].

Observe that if X is a union of cells of a coherent configuration, the restriction $\rho|_{X^2}$ is a coherent configuration on X.

The constants $p^{\kappa}_{\sigma\tau}$ are called the *intersection numbers* of ρ and may be interpreted algebraically as follows: For every $\sigma \in \operatorname{Im}\rho$, let A_σ be the adjacency matrix of the relation $\rho^{-1}(\sigma)$. Then for all $\sigma, \tau \in \operatorname{Im}\rho$,

$$A_\sigma A_\tau = \sum_{\kappa \in \operatorname{Im}\rho} p^{\kappa}_{\sigma\tau} A_\kappa.$$

Thus, taking $p^{\kappa}_{\sigma\tau}$ as rational numbers in a field $\mathbb{F}$ of characteristic zero, we see that the $\mathbb{F}$-span of the set $\mathcal{A}_\rho = \{A_\sigma \mid \sigma \in \operatorname{Im}\rho\}$ is an $\mathbb{F}$-algebra. The same is true if we take $\mathbb{F}$ to be a field of characteristic q and consider the constants $p^{\kappa}_{\sigma\tau}$ modulo q. We refer to this algebra as the $\mathbb{F}$-*adjacency algebra of* ρ and denote it by $\mathbb{F}\mathcal{A}_\rho$. Such an algebra is a *coherent algebra* in the following sense:

Definition 2.3 (coherent algebra). A subalgebra of $\operatorname{Mat}_V(\mathbb{F})$ is said to be a *coherent algebra* on V if it is a unital algebra with respect to matrix multiplication and Schur–Hadamard (componentwise) multiplication , and it is closed under transposition.

We indicate the Schur–Hadamard multiplication by $\star$ [2]. At this point, it needs to be pointed out that in most literature, when $\mathbb{F} = \mathbb{C}$, closure under transposition is usually replaced by *closure under Hermitian conjugation* for the definition of a coherent algebra. However, we show in Proposition 2.6, that an algebra satisfies Definition 2.3 if and only if it has a basis of 0-1-matrices satisfying the *coherence conditions* (Definition 2.4). In Section 2.3 of [Chen and Ponomarenko 2019] it is shown that an algebra over $\mathbb{C}$ satisfies Definition 2.3 with closure under transposition replaced by closure under Hermitian conjugation if and only if it has a basis of 0-1-matrices satisfying the coherence conditions. Hence, over $\mathbb{C}$, Definition 2.3 is equivalent to the original one by D. Higman in [1987], but has the advantage that it can be extended to any field.

It is clear from the definition, that for any coherent configuration ρ, the $\mathbb{F}$-adjacency algebra $\mathbb{F}\mathcal{A}_\rho$ is a coherent algebra for any field $\mathbb{F}$. Indeed, the set $\mathcal{A}_\rho$ is the unique basis of 0-1-matrices for $\mathbb{F}\mathcal{A}_\rho$ satisfying the *coherence conditions*:

Definition 2.4. A set of 0-1-matrices $\mathcal{M}$ is said to satisfy the *coherence conditions* if:

(1) $\sum_{A\in\mathcal{M}} A = \mathbb{J}$, where $\mathbb{J}$ is the all 1s matrix.

(2) For some $\mathcal{I} \subseteq \mathcal{M}$, $\sum_{A\in\mathcal{I}} A = \mathbb{I}$, where $\mathbb{I}$ is the identity matrix.

(3) $A^t \in \mathcal{M}$ for all $A \in \mathcal{M}$.

We now show that any coherent algebra over any field has a unique basis of 0-1-matrices satisfying the coherence conditions. We refer to this basis as the *standard basis* of a coherent algebra. The argument that follows is analogous to that used to prove Theorem 2.3.7 in [Chen and Ponomarenko 2019].

Let W be a coherent algebra over V. As explained in Section 2.3 in [Chen and Ponomarenko 2019], one may write

$$\mathbb{J} = \sum_{i\in[r]} E_i, \tag{2}$$

[2] In the literature, the Schur–Hadamard multiplication is often denoted by $\circ$. However, we reserve the latter for function composition.

where $\{E_i \mid i \in [r]\}$ is the full set of primitive idempotents of W with respect to the Schur–Hadamard product. In order to be an idempotent, E_i must be a 0-1 matrix; and hence, the adjacency matrix of some relation $R_i \subseteq V^2$. Since for $i \neq j$, E_i and E_j are orthogonal, for all $u, v \in V$, $(E_i)_{uv} = 1 \implies (E_j)_{uv} = 0$. Thus, from (2), it follows that $\{R_i \mid i \in [r]\}$ forms a partition of V^2.

Lemma 2.5. $\{E_i \mid i \in [r]\}$, *as above, satisfies the coherence conditions.*

Proof. Condition (1) of Definition 2.4 is satisfied because of (2). Because $\mathbb{I} \in W$ is an idempotent, it can be written as a sum of primitive idempotents. Thus, $\{E_i \mid i \in [r]\}$ satisfies condition (2) in Definition 2.4. Finally, E_i^t is also a primitive idempotent, since W is closed under transposition. □

Proposition 2.6. *For any field $\mathbb{F}$, a coherent algebra on V over $\mathbb{F}$ has a unique basis of* 0-1*-matrices satisfying the coherence conditions.*

Proof. The set $B = \{E_i \mid i \in [r]\}$ satisfies the coherence conditions by Lemma 2.5.

Suppose $\mathbb{F}$ is algebraically closed. Then B is a basis for W, since W is commutative with respect to the Schur–Hadamard product and a basis of a semisimple commutative algebra over an algebraically closed field is given by the set of its primitive idempotents.

Suppose $\mathbb{F}$ is not algebraically closed. Since B is a linearly independent set, there is some $B' \subseteq W$ such that $B \cup B'$ is a basis for W. Let $\mathbb{G}$ be the algebraic closure of $\mathbb{F}$, and consider the linear space $\mathbb{G}(B \cup B') \subseteq \mathrm{Mat}_V(\mathbb{G})$. By construction, $\mathbb{G}(B \cup B')$ is closed under transposition, matrix multiplication, and Schur–Hadamard multiplication and is thus a coherent algebra over $\mathbb{G}$. From the above, $\mathbb{G}(B \cup B')$ must then have a basis B'' of 0-1-matrices satisfying the coherence conditions. Since all entries of the elements of B'' are 0 and 1, $B'' \subset \mathbb{F}(B \cup B')$ is a basis for W as well. As B'' is a set of primitive orthogonal idempotents of W, it holds that $B'' \subseteq B$. But B is a linearly independent set, and B'' is a basis for W. Whence $B'' = B = \{E_i \mid i \in [r]\}$.

The uniqueness of B follows from (2), which implies that any basis of 0-1 matrices satisfying the coherence conditions must be the set of primitive idempotents of W. □

We can denote a coherent algebra on V over $\mathbb{F}$ as $\mathbb{F}\mathcal{A}$, where $\mathcal{A}$ is some set of 0-1-matrices satisfying the coherence conditions. It is easily seen that for any $(u, v) \in V^2$ and binary relations S and T on V with adjacency matrices $A_S, A_T \in \mathrm{Mat}_V(\mathbb{F})$, respectively,

$$\left|\{x \in V \mid (u, x) \in S, (x, v) \in T\}\right| = (A_S A_T)_{uv} \tag{3}$$

if $\mathrm{char}(\mathbb{F}) = 0$. Set $\rho_W : V^2 \to [r]$ to be $\rho_W(u, v) = i$ if $(u, v) \in R_i$. Since there are constants $p_{ij}^k \in \mathbb{F}$ such that

$$E_i E_j = \sum_{k \in [r]} p_{ij}^k E_k,$$

it then follows that

$$p_{ij}^k = \left|\{x \in V \mid \rho_W(u, x) = i, \rho_W(x, v) = j\}\right| \tag{4}$$

is the same for all (u, v) such that $\rho_W(u, v) = k$; and hence, ρ_W is a coherent configuration. Otherwise, if $\mathbb{F}$ has characteristic $q > 0$, (3) holds modulo q. In particular if $q > |V|$, ρ_W is a coherent configuration.

Remark 2.7. In the literature, coherent algebras over a field $\mathbb{F}$ of positive characteristic have usually been defined to be the $\mathbb{F}$-span of the adjacency matrices of the relations $\{\rho^{-1}(\sigma) | \sigma \in \mathrm{Im}\,\rho\}$ for some coherent configuration ρ. The latter discussion thus shows that, over fields of positive characteristic, Definition 2.2 defines a potentially larger class of algebras.

The most intuitive morphisms between coherent configurations arise from the algebraic setting. Let $\mathcal{A}, \mathcal{A}' \subseteq \mathrm{Mat}_V(\mathbb{F})$ satisfy the coherence conditions.

Definition 2.8 (isomorphism of coherent algebras). An $\mathbb{F}$-linear bijection $\psi : \mathbb{F}\mathcal{A} \to \mathbb{F}\mathcal{A}'$ is said to be an *isomorphism* of coherent algebras if:

(1) $\psi(\mathbb{I}) = \mathbb{I}$.

(2) $\psi(\mathbb{J}) = \mathbb{J}$.

(3) $\psi(AB) = \psi(A)\psi(B)$ and $\psi(A \star B) = \psi(A) \star \psi(B)$, for all $A, B \in \mathbb{F}\mathcal{A}_\rho$.

That is, ψ preserves the structure of $\mathbb{F}\mathcal{A}$, both as a matrix algebra and as an algebra, with respect to $\star$. As a consequence, the image under ψ of an element of the standard basis of $\mathbb{F}\mathcal{A}$ must be an element of the standard basis of $\mathbb{F}\mathcal{A}'$, since the standard basis of a coherent algebra is the set of its primitive idempotents with respect to $\star$. Conversely, the $\mathbb{F}$-linear extension of any bijection between the standard bases of $\mathbb{F}\mathcal{A}$ and $\mathbb{F}\mathcal{A}'$ is a coherent algebra isomorphism provided it is also a matrix algebra isomorphism.

Let ρ and ρ' be coherent configurations and denote their intersection numbers by $p^{\kappa}_{\sigma\tau}$ and $q^{\kappa'}_{\sigma'\tau'}$, respectively.

Definition 2.9 (algebraic isomorphism). A bijection $\phi : \mathrm{Im}\,\rho \to \mathrm{Im}\,\rho'$ is said to be an *algebraic isomorphism* if for all $\sigma, \tau, \kappa \in \mathrm{Im}\,\rho$,

$$p^{\kappa}_{\sigma\tau} = q^{\phi(\kappa)}_{\phi(\sigma)\phi(\tau)}.$$

Thus, an algebraic isomorphism between coherent configurations induces a bijection between the standard bases of their respective adjacency algebras. Such a bijection linearly extends to a coherent algebra isomorphism.

Crucial to this paper is the fact that when coherent algebras are semisimple (with respect to matrix product), isomorphisms between them assume a very simple form. Indeed, it is an easy consequence of the Skolem–Nöther Theorem that if $\psi : W_1 \to W_2$ is an algebra isomorphism, where W_1 and W_2 are semisimple subalgebras of $\mathrm{Mat}_n(\mathbb{F})$, then there is some $S \in \mathrm{GL}_n(\mathbb{F})$ such that $\psi(A) = SAS^{-1}$ for all $A \in W_1$. The following is a direct consequence of Theorem 4.1.3 in [Zieschang 1996][3]:

Theorem 2.10. *The Jacobson radical of a coherent algebra $\mathbb{F}\mathcal{A}$ is a subspace of the span of the elements of the standard basis whose number of nonzero entries is divisible by* char($\mathbb{F}$).

For a coherent configuration ρ on V, choose $u, v \in V$ such that $\rho(u, v) = \sigma$ and $\rho(u, u) = \tau$. It follows from formula (2.1.5) in [Chen and Ponomarenko 2019] that

$$|\rho^{-1}(\sigma)| = |\rho^{-1}(\tau)|\,\bigl|\{x \in V \mid \rho(u, x) = \sigma\}\bigr|.$$

Since both factors on the right-hand side are no larger than $|V|$, it is clear that the prime factors of the size of any equivalence class of ρ is no larger than $|V|$.

[3]The author actually proves this statement for coherent algebras whose diagonal matrices are multiples of the identity matrix. However, the same argument applies to the more general case; and, in particular, to our more general notion of coherent algebras in the sense of Definition 2.2.

Corollary 2.11. *A coherent algebra on V over $\mathbb{F}$ is semisimple with respect to matrix product if* $\mathrm{char}(\mathbb{F}) = 0$ *or* $\mathrm{char}(\mathbb{F}) > |V|$.

3. Graph-like partitions

In this section, we describe some restrictions to be imposed on the partitions dealt with in the paper. Such restrictions are natural in the sense that they are necessary conditions to be satisfied by a partition of k-tuples into the orbits of an induced action of a group on 1-tuples.

Fix some $k \in \mathbb{N}$, a set V, and let $\gamma \in \mathcal{P}(V^k)$. Define an action of $\mathrm{Sym}(k)$ on V^k by letting, for each $\tau \in \mathrm{Sym}(k)$, $\vec{v}^\tau$ be the element of V^k with i-th entry $v_{\tau^{-1}(i)}$.

Definition 3.1 (invariance). γ is *invariant* if $\gamma(\vec{u}) = \gamma(\vec{v}) \implies \gamma(\vec{u}^\tau) = \gamma(\vec{v}^\tau)$ for all $\vec{u}, \vec{v} \in V^k$ and all $\tau \in \mathrm{Sym}(k)$. [4]

Fix some $r \in [k]$ and $\vec{\imath} \in [k]^r$. For a tuple $\vec{v} \in V^k$, we define its *projection* on $\vec{\imath}$, denoted $\mathrm{pr}_{\vec{\imath}}\vec{v}$, to be the tuple in V^r with j-th entry v_{i_j}. Without ambiguity, we write $\mathrm{pr}_r\vec{v}$ for the tuple $\mathrm{pr}_{(1,\ldots,r)}\vec{v}$. We denote by $\mathrm{pr}_r\gamma$ the partition of V^r given by

$$\mathrm{pr}_r\gamma(\vec{v}) = \gamma(v_1, v_2, \ldots, v_r, v_r, \ldots, v_r),$$

and call it the *r-projection* of γ. Note that if γ is invariant, then for any $r \le k$, $\mathrm{pr}_r\gamma$ is invariant.

Definition 3.2 (r-consistency). γ is said to be *$\vec{\imath}$-consistent* for some $\vec{\imath} \in [k]^r$ if for all $\vec{u}, \vec{v} \in V^k$,

$$\gamma(\vec{u}) = \gamma(\vec{v}) \implies \mathrm{pr}_r\gamma(\mathrm{pr}_{\vec{\imath}}\vec{u}) = \mathrm{pr}_r\gamma(\mathrm{pr}_{\vec{\imath}}\vec{v}).$$

If, in addition, γ is $\vec{\imath}$-consistent for all $\vec{\imath} \in [k]^r$, we say that γ is *r-consistent*.

Observe that if for some $r \le k$, γ is r-consistent, then it is t-consistent for all $t \le r$. One may also verify that if γ is invariant, then it is k-consistent if and only if for all $\vec{u}, \vec{v} \in V^k$,

$$\gamma(\mathrm{pr}_{k-1}\vec{u} \cdot u_{k-1}) = \gamma(\mathrm{pr}_{k-1}\vec{v} \cdot v_{k-1}). \tag{5}$$

Definition 3.3 (graph-like partition). γ is said to be a *graph-like* partition of V^k if it is invariant, r-consistent for all $r \le k$, and for all $\vec{u}, \vec{v} \in V^k$,

$$\gamma(\vec{u}) = \gamma(\vec{v}) \implies (u_i = u_j \implies v_i = v_j,\ \forall\, i, j \in [k]). \tag{6}$$

Note that if γ is graph-like, then $\mathrm{pr}_t\gamma$ is graph-like for all $t \in [k]$. An example of a graph -like partition is that of a coherent configuration, as introduced in Section 2.

Proposition 3.4. *A coherent configuration is a graph-like partition.*

Proof. Let ρ be a coherent configuration on V. Then ρ is a rainbow, and hence satisfies conditions (1) and (2) of Definition 2.1, from which we deduce that it is invariant and satisfies (6). To show that it is 1-consistent, let $u, v, u', v' \in V$ be such that $\rho(u, v) = \rho(u', v')$. For any $\sigma \in \mathrm{Im}\,\rho$,

$$\{x \in V \mid \rho(u, x) = \sigma\} = \bigcup_{\tau \in \mathrm{Im}\,\rho} \{x \in V \mid \rho(u, x) = \sigma, \rho(x, v) = \tau\}.$$

[4] The concept of an invariant partition has already been introduced in Theorem 6.1 in [Evdokimov and Ponomarenko 1999].

The size of the right-hand side of the above equation is independent of the choice of (u, v) from the equivalence class $[(u, v)]_\gamma$. In particular, if $\sigma = \rho(u, u)$, then because $\rho(u, v) = \rho(u', v')$ there is exactly one $x \in V$ such that $\rho(u', x) = \sigma$, namely $x = u'$. Hence, $\rho(u, u) = \rho(u', u')$, and 1-consistency of ρ follows. □

Arguments of this kind appear repeatedly in our proofs of Theorems 1.3, 1.4, and 1.5.

Another graph-like partition which will be useful throughout the paper is that of *atomic types* of k-tuples of vertices of a graph Γ, which we indicate by $\alpha_{k,\Gamma}$. To be precise, we define

$$\alpha_{k,\Gamma} : V^k \to (\operatorname{Im}\Gamma)^{[k]^{(2)}}, \quad \vec{v} \mapsto (\Gamma(v_i, v_j))_{(i,j)\in[k]^{(2)}},$$

where, as the reader may recall, we view graphs as labelled partitions.

Definition 3.5 (Γ-partition). We say that γ is a Γ-partition if $\alpha_{k,\Gamma} \preceq \gamma$ for some graph Γ.

The following result is a useful property of graph-like partitions.

Lemma 3.6. *For any graph-like $\gamma \in \mathcal{P}(V^k)$ and any $\vec{u}, \vec{v} \in V^k$ such that $\gamma(\vec{u}) = \gamma(\vec{v})$,*

$$\gamma(\vec{u}\langle \vec{\imath}, \mathrm{pr}_{\vec{\jmath}}\vec{u}\rangle) = \gamma(\vec{v}\langle \vec{\imath}, \mathrm{pr}_{\vec{\jmath}}\vec{v}\rangle)$$

for all $\vec{\imath} \in [k]^{(r)}$ and $\vec{\jmath} \in [k]^r$.

Proof. Let $\vec{u}, \vec{v}$ be as in the statement. For any $s, t \in [k]$, $\gamma(\vec{u}\langle s, v_t\rangle) = \gamma(\vec{v}\langle s, u_t\rangle)$ holds true by invariance and $(k-1)$-consistency of γ.

Fix some $r < k$ and set $\vec{u}' = \mathrm{pr}_{\vec{\jmath}}\vec{u}$ and $\vec{v}' = \mathrm{pr}_{\vec{\jmath}}\vec{v}$ for some $\vec{\jmath} \in [k]^r$. Suppose that $\gamma(\vec{v}\langle \vec{\imath}, \vec{v}'\rangle) = \gamma(\vec{u}\langle \vec{\imath}, \vec{u}'\rangle)$ holds for all $\vec{\imath} \in [k]^{(r)}$. Because $r < k$, there is some $i' \in [k]$ which is not an entry of $\vec{\imath}$. For some $l \in [k]$, let $x = v_l$ and $y = u_l$. Then,

$$\gamma((\vec{v}\langle \vec{\imath}, \vec{v}'\rangle)\langle i', x\rangle) = \gamma((\vec{u}\langle \vec{\imath}, \vec{u}'\rangle)\langle i', y\rangle)$$

by invariance and $(k-1)$-consistency of γ. From this, one concludes that

$$\gamma(\vec{v}\langle \vec{\imath}\cdot i', \vec{v}'\cdot x\rangle) = \gamma(\vec{u}\langle \vec{\imath}\cdot i', \vec{u}'\cdot y\rangle);$$

and hence, by setting $\vec{q} = \vec{\jmath}\cdot l$,

$$\gamma(\vec{v}\langle \vec{\imath}\cdot i', \mathrm{pr}_{\vec{q}}\vec{v}\rangle) = \gamma(\vec{u}\langle \vec{\imath}\cdot i', \mathrm{pr}_{\vec{q}}\vec{u}\rangle).$$

We deduce the desired statement by induction. □

4. Refinement operators and procedures

As previously stated, all the SPAS in this paper arise from *refinement operators*, which we now define. Fix some $k \in \mathbb{N}$.

Definition 4.1 (refinement operator). A *k-refinement operator R* is a mapping which, for each set V, assigns to each $\gamma \in \mathcal{P}(V^k)$ a partition $R \circ \gamma \in \mathcal{P}(V^k)$ such that $\gamma \preceq R \circ \gamma$.

Since labelled partitions are seen as mappings, the symbol $\circ$ really indicates composition thereof.

Definition 4.2 (fixed point). $\gamma \in \mathcal{P}(V^k)$ is said to be a *fixed point* of a k-refinement operator R if $\gamma \approx R \circ \gamma$. In such cases, we also say that γ is *R-stable*.

Fix some $\gamma \in \mathcal{P}(V^k)$, set $X^0 = \gamma$ and $X^i = R \circ X^{i-1}$. We then have an increasing sequence

$$X^0 \preceq X^1 \preceq \cdots \preceq X^i \preceq \cdots .$$

Because all elements of this sequence are bounded by a labelled partition of V^k with exactly one element per equivalence class, there must be some $s \in \mathbb{N}$ such that for all $i \geq s$, X^i is fixed point of R. For the smallest such s, denote X^s by $[\gamma]^R$.

Definition 4.3 (graph-like operator). A k-refinement operator R is *graph-like* if $R \circ \gamma$ is graph-like for all graph-like $\gamma \in \mathcal{P}(V^k)$ and sets V.

Definition 4.4 (refinement procedure). The family of mappings $\{R_1, R_2, \dots\}$ is said to be a *refinement procedure* if, for each $k \in \mathbb{N}$:

(1) R_k is a graph-like k-refinement operator;

(2) if γ is a graph-like fixed point of R_k then $\mathrm{pr}_{k-1}\gamma$ is a fixed point of R_{k-1}; and

(3) for all sets V and $\gamma \in \mathcal{P}(V^k)$, $[\gamma]^{R_k}$ is computable in time $|V|^{O(k)}$.

For each $k, r \in \mathbb{N}$, the k-refinement operators of interest in this paper are the *Weisfeiler–Leman operators* $\mathrm{WL}_{k,r}$; the *counting logic operators* $\mathbb{C}_{k,r}$; and, for any field $\mathbb{F}$, the *invertible map operators* $\mathrm{IM}_k^{\mathbb{F}}$. For the former two, we are really interested in the case when $r < k$. Thus, when $k \leq r$, for convenience, we let $\mathrm{WL}_{k,r} \circ \gamma = \mathbb{C}_{k,r} \circ \gamma = \gamma$ for all $\gamma \in \mathcal{P}(V^k)$ and sets V. For $r < k$ define

$$\mathrm{WL}_{k,r} \circ \gamma : V^k \to \mathrm{Im}\,\gamma \times \mathrm{Mult}(\Phi^{\gamma,r})^{V^r}, \qquad \vec{v} \mapsto \big(\gamma(\vec{v}), \big\{\!\big\{(\gamma(\vec{v}\langle \vec{\imath}, \vec{u}\rangle)_{\vec{\imath}\in[k]^{(r)}} \mid \vec{u} \in V^r\big\}\!\big\}\big).$$

$$\mathbb{C}_{k,r} \circ \gamma : V^k \to \mathrm{Im}\,\gamma \times (\mathrm{Mult}(\mathrm{Im}\,\gamma^{V^r}))^{[k]^{(r)}}, \quad \vec{v} \mapsto \big(\gamma(\vec{v}), \big(\big\{\!\big\{\gamma(\vec{v}\langle \vec{\imath}, \vec{u}\rangle) \mid \vec{u} \in V^r\big\}\!\big\}\big)_{\vec{\imath}\in[k]^{(r)}}\big).$$

Let $\chi_{\vec{\imath},\sigma}^{\gamma,\vec{v}}$ be the adjacency matrix of the binary relation $\{(x, y) \in V^2 \mid \gamma(\vec{v}\langle\vec{\imath}, (x, y)\rangle) = \sigma\}$. Similarly to above, set $\mathrm{IM}_1^{\mathbb{F}} \circ \gamma = \mathrm{IM}_2^{\mathbb{F}} \circ \gamma = \gamma$; and, for $k > 2$ define

$$\mathrm{IM}_k^{\mathbb{F}} \circ \gamma : V^k \to \mathrm{Im}\,\gamma \times (\mathrm{Mat}_V(\mathbb{F})^{\mathrm{Im}\,\gamma\times[k]^{(2)}}/\sim), \quad \vec{v} \mapsto (\gamma(\vec{v}), ((\chi_{\vec{\imath},\sigma}^{\gamma,\vec{v}})_{\sigma\in\mathrm{Im}\,\gamma})_{\vec{\imath}\in[k]^{(2)}}),$$

where $\sim$ is the equivalence relation on elements of $\mathrm{Mat}_V(\mathbb{F})^{\mathrm{Im}\,\gamma\times[k]^{(2)}}$ under simultaneous similarity. That is, two tuples are equivalent if they lie in the same orbit of $GL_V(\mathbb{F})$ acting on the tuples by conjugation. Although the reader may find the above definitions rather technical, they are not crucial throughout the paper. Indeed, for $k > r$ and $\vec{u}, \vec{v} \in V^k$, the following facts are sufficient:

(1) $\mathrm{WL}_{k,r} \circ \gamma(\vec{u}) = \mathrm{WL}_{k,r} \circ \gamma(\vec{v})$ if and only if $\gamma(\vec{u}) = \gamma(\vec{v})$, and for all $\vec{\phi} \in \Phi^{\gamma,r}$ and $\vec{\imath} \in [k]^{(r)}$

$$\big|\{\vec{x} \in V^r \mid \gamma(\vec{u}\langle\vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}}\}\big| = \big|\{\vec{x} \in V^r \mid \gamma(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}}\}\big|.$$

(2) $\mathbb{C}_{k,r} \circ \gamma(\vec{u}) = \mathbb{C}_{k,r} \circ \gamma(\vec{v})$ if $\gamma(\vec{u}) = \gamma(\vec{v})$, and for all $\sigma \in \mathrm{Im}\,\gamma$ and $\vec{\imath} \in [k]^{(r)}$

$$\big|\{\vec{x} \in V^r \mid \gamma(\vec{u}\langle\vec{\imath}, \vec{x}\rangle) = \sigma\}\big| = \big|\{\vec{x} \in V^r \mid \gamma(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \sigma\}\big|.$$

(3) $\mathrm{IM}_k^{\mathbb{F}} \circ \gamma(\vec{u}) = \mathrm{IM}_k^{\mathbb{F}} \circ \gamma(\vec{v})$ if and only if $\gamma(\vec{u}) = \gamma(\vec{v})$, and for each $\vec{\imath} \in [k]^{(2)}$ there exists some $S \in \mathrm{GL}_V(\mathbb{F})$ such that for all $\sigma \in \mathrm{Im}\,\gamma$,

$$S\chi_{\vec{\imath},\sigma}^{\gamma,\vec{u}}S^{-1} = \chi_{\vec{\imath},\sigma}^{\gamma,\vec{v}}.$$

From this, one may derive the following *stability conditions*:

Proposition 4.5. *For any $\gamma \in \mathcal{P}(V^k)$ and $k > r$:*

(1) *γ is $\mathrm{WL}_{k,r}$-stable if and only if for all $\vec{v} \in V^k$, $\vec{\imath} \in [k]^{(r)}$, and $\vec{\phi} \in \Phi^{\gamma,r}$, the size of the set $\{\vec{x} \in V^r \mid \gamma(\vec{v}\langle \vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}}\}$ is independent of the choice of $\vec{v}$ from the equivalence class $[\vec{v}]_\gamma$.*

(2) *γ is $\mathbb{C}_{k,r}$-stable if and only if for all $\vec{v} \in V^k$, $\vec{\imath} \in [k]^{(r)}$, and $\sigma \in \operatorname{Im}\gamma$, the size of the set $\{\vec{x} \in V^r \mid \gamma(\vec{v}\langle \vec{\imath}, \vec{x}\rangle) = \sigma\}$ is independent of the choice of $\vec{v}$ from the equivalence class $[\vec{v}]_\gamma$.*

(3) *γ is $\mathrm{IM}_k^{\mathbb{F}}$-stable if for all $\vec{u}, \vec{v} \in V^k$ and $\vec{\imath} \in [k]^{(2)}$,*

$$\gamma(\vec{u}) = \gamma(\vec{v}) \Longrightarrow \exists S \in \mathrm{GL}_V(\mathbb{F}),\ S\chi_{\vec{\imath},\sigma}^{\gamma,\vec{u}} S^{-1} = \chi_{\vec{\imath}\,\sigma}^{\gamma,\vec{v}},\ \forall \sigma \in \operatorname{Im}\gamma.$$

Consider the following families of mappings:

(1) For all $r \in \mathbb{N}$, $\mathbf{WL}_r = \{\mathrm{WL}_{1,r}, \mathrm{WL}_{2,r}, \dots\}$.

(2) For all $r \in \mathbb{N}$, $\mathbf{C}_r = \{\mathbb{C}_{1,r}, \mathbb{C}_{2,r}, \dots\}$.

(3) For any field $\mathbb{F}$, $\mathbf{IM}(\mathbb{F}) = \{\mathrm{IM}_1^{\mathbb{F}}, \mathrm{IM}_2^{\mathbb{F}}, \dots\}$.

Proposition 4.6. *The families $\mathbf{WL}_r$, $\mathbf{C}_r$, and $\mathbf{IM}(\mathbb{F})$ are refinement procedures for all $r \in \mathbb{N}$ and fields $\mathbb{F}$.*

For the proof of Proposition 4.6 and that of the next auxiliary lemma, we use the following notations and conventions. Fix a graph-like partition $\gamma \in \mathcal{P}(V^k)$. Let $\bar{\gamma} = \mathrm{pr}_{k-1}\gamma$, and for all $\vec{v} \in V^k$, if $\sigma = \gamma(\vec{v})$, let $\bar{\sigma} = \bar{\gamma}(\mathrm{pr}_{k-1}\vec{v})$. For $\pi \in \mathrm{Sym}(k)$, we let $\sigma^\pi = \gamma(\vec{v}^\pi)$ and define an action on $[k]^{(r)}$ by setting $(\pi(\vec{\imath}))_j = \pi(i_j)$ for all $j \in [r]$. For any $\vec{v} \in V^k$, we denote $\vec{v}' = \mathrm{pr}_{k-1}\vec{v} \cdot v_{k-1}$. Note that $\bar{\sigma}$ and σ^π are well defined, since γ is graph-like.

Lemma 4.7. *Let $\vec{\imath} \in [k]^{(r)}$.*

(1) *If k is an entry of $\vec{\imath}$, then $\gamma(\vec{v}'\langle \vec{\imath}, \vec{x}\rangle) = \gamma(\vec{v}\langle \vec{\imath}, \vec{x}\rangle)$.*

(2) *If $k-1$ is an entry of $\vec{\imath}$, but k is not; then $\gamma(\vec{v}^\pi\langle \vec{\imath}, \vec{x}\rangle) = \gamma(\vec{v}'\langle \vec{\imath}, \vec{x}\rangle)$, where $\pi = (k-1,\ k)$ is a transposition of* $\mathrm{Sym}(k)$.

(3) *If neither k nor $k-1$ are entries of $\vec{\imath}$, then $\gamma(\vec{v}\langle \vec{\imath}, \vec{x}\rangle) = \sigma \Longrightarrow \gamma(\vec{v}'\langle \vec{\imath}, \vec{x}\rangle) = \bar{\sigma}$. In particular,*

$$\{\vec{x} \in V^r \mid \gamma(\vec{v}'\langle \vec{\imath}, \vec{x}\rangle) = \kappa\} = \bigcup_{\substack{\sigma \in \operatorname{Im}\gamma \\ \bar{\sigma} = \kappa}} \{\vec{x} \in V^r \mid \gamma(\vec{v}\langle \vec{\imath}, \vec{x}\rangle) = \sigma\}. \tag{7}$$

Proof. Statements (1) and (2) are trivial to check.

Let $\vec{w} = \vec{v}\langle \vec{\imath}, \vec{x}\rangle$ with $\gamma(\vec{w}) = \sigma$. Then $\vec{w}' = \vec{v}'\langle \vec{\imath}, \vec{x}\rangle$, so $\gamma(\vec{w}') = \bar{\sigma}$ by definition. From this, and the fact that γ is $(k-1)$-consistent, the right-hand side of (7) is a subset of the left-hand side. The reverse inclusion follows from the definition of $\bar{\sigma}$ in terms of σ, and statement (3) follows. □

Proof of Proposition 4.6. We check that each of the families of mappings satisfy the conditions in Definition 4.4. Note that $\gamma \preceq R \circ \gamma$ for any k-refinement operator R. Hence, since γ is graph-like, for any $\vec{u}, \vec{v} \in V^k$, $R \circ \gamma(\vec{u}) = R \circ \gamma(\vec{v})$ implies that $u_i = u_j \Longrightarrow v_i = v_j$ for all $i, j \in [k]$. Thus, to show that $R \circ \gamma$ is graph-like, it suffices to verify that $R \circ \gamma$ is invariant and satisfies (5). That is, for all $\vec{u}, \vec{v} \in V^k$, $R \circ \gamma(\vec{u}) = R \circ \gamma(\vec{v}) \Longrightarrow R \circ \gamma(\vec{u}') = R \circ \gamma(\vec{v}')$.

WL_r is a refinement procedure. We first show that $\mathrm{WL}_{k,r} \circ \gamma$ is invariant and satisfies (5), and is thus a graph-like partition.

Suppose $\mathrm{WL}_{k,r} \circ \gamma(\vec{u}) = \mathrm{WL}_{k,r} \circ \gamma(\vec{v})$. Then $\gamma(\vec{u}) = \gamma(\vec{v})$ by definition, and hence $\gamma(\vec{u}^\tau) = \gamma(\vec{v}^\tau)$ since γ is graph-like and thus invariant. Furthermore, from the invariance of γ, it follows that, for all $\tau \in \mathrm{Sym}(k)$ and $\vec{\phi} \in \Phi^{\gamma,r}$,

$$\{\vec{x} \in V^r \mid \gamma(\vec{v}^\tau\langle\tau(\vec{\imath}), \vec{x}\rangle) = (\phi_{\vec{\imath}})^\tau,\ \forall \vec{\imath} \in [k]^{(r)}\} = \{\vec{x} \in V^r \mid \gamma(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\}.$$

Hence, $\mathrm{WL}_{k,r} \circ \gamma$ is invariant.

For all $\vec{\phi} \in \Phi^{\gamma,r}$, let $\vec{\phi}^\dagger \in \Phi^{\gamma,r}$ be defined as

$$\phi^\dagger_{\vec{\imath}} = \begin{cases} \bar{\phi}_{\vec{\imath}} & \text{if neither } k \text{ nor } k-1 \text{ are entries of } \vec{\imath}, \\ (\phi_{\vec{\imath}})^\pi & \text{if } k-1 \text{ is an entry of } \vec{\imath}, \\ \phi_{\vec{\imath}} & \text{otherwise,} \end{cases}$$

where $\pi = (k-1, k) \in \mathrm{Sym}(k)$. Observe that if $\{\vec{x} \in V^r \mid \gamma(\vec{v}'\langle\vec{\imath}, \vec{x}\rangle) = \psi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\}$ is nonempty for some $\vec{v} \in V^k$ and $\vec{\psi} \in \Phi^{\gamma,r}$, then $\vec{\psi} = \vec{\phi}^\dagger$ for some $\vec{\phi} \in \Phi^{\gamma,r}$. It follows from the definition of $\vec{\phi}^\dagger$ and Lemma 4.7 that

$$\{\vec{x} \in V^r \mid \gamma(\vec{v}'\langle\vec{\imath}, \vec{x}\rangle) = \psi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\} = \bigcup_{\substack{\vec{\phi} \in \Phi^{\gamma,r} \\ \vec{\psi} = \vec{\phi}^\dagger}} \{\vec{x} \in V^r \mid \gamma(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\}. \tag{8}$$

Since $\mathrm{WL}_{k,r} \circ \gamma(\vec{u}) = \mathrm{WL}_{k,r} \circ \gamma(\vec{v})$ for all $\vec{\phi} \in \Phi^{\gamma,r}$,

$$\left|\{\vec{x} \in V^r \mid \gamma(\vec{u}\langle\vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\}\right| = \left|\{\vec{x} \in V^r \mid \gamma(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\}\right|.$$

Thus, because the right-hand side of (8) is a disjoint union, we deduce that

$$\left|\{\vec{x} \in V^r \mid \gamma(\vec{u}'\langle\vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\}\right| = \left|\{\vec{x} \in V^r \mid \gamma(\vec{v}'\langle\vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\}\right|,$$

for all $\vec{\phi} \in \Phi^{\gamma,r}$. This implies $\mathrm{WL}_{k,r} \circ \gamma(\vec{u}') = \mathrm{WL}_{k,r} \circ \gamma(\vec{v}')$; and hence, that $\mathrm{WL}_{k,r} \circ \gamma$ is graph-like.

Suppose now that γ is $\mathrm{WL}_{k,r}$-stable. Observe that for all $\vec{\phi} \in \Phi^{\bar{\gamma},r}$,

$$\{\vec{x} \in V^r \mid \bar{\gamma}(\mathrm{pr}_{k-1}\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k-1]^{(r)}\} = \bigcup_{\substack{\vec{\psi} \in \Phi^{\gamma,r} \\ \bar{\psi}_{\vec{\imath}} = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k-1]^{(r)}}} \{\vec{x} \in V^r \mid \gamma(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \psi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\}. \tag{9}$$

From the $\mathrm{WL}_{k,r}$-stability of γ and the fact that $\gamma(\vec{u}) = \gamma(\vec{v})$, it follows that for all $\vec{\psi} \in \Phi^{\gamma,r}$,

$$|\{\vec{x} \in V^r \mid \gamma(\vec{u}\langle\vec{\imath}, \vec{x}\rangle) = \psi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\}| = |\{\vec{x} \in V^r \mid \gamma(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \psi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\}|.$$

Since the right-hand side of (9) is a disjoint union, it holds that for all $\vec{\phi} \in \Phi^{\bar{\gamma},r}$,

$$\left|\{\vec{x} \in V^r \mid \bar{\gamma}(\mathrm{pr}_{k-1}\vec{u}\langle\vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k-1]^{(r)}\}\right| = \left|\{\vec{x} \in V^r \mid \bar{\gamma}(\mathrm{pr}_{k-1}\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k-1]^{(r)}\}\right|.$$

Thus, $\bar{\gamma}$ is $\mathrm{WL}_{k-1,r}$-stable.

To see that the stable partition $[\gamma]^{\mathrm{WL}_{k,r}}$ can be computed in time $|V|^{O(k)}$, note that the number of iterations of the refinement operator before the fixed point is reached is at most $|V|^k$. At each step, for each k-tuple $\vec{v}$, we need to compute the colour $\mathrm{WL}_{k,r} \circ \gamma(\vec{v})$, which involves checking the colour of $k|V|$ distinct tuples. The total time required at each step is therefore $O(k|V|^{k+1})$. Repeating this for $|V|^k$ steps gives us the required bound.

C_r is a refinement procedure. The proof, in this case, is similar to that of $\mathrm{WL}_{k,r}$.

Suppose $\mathbb{C}_{k,r} \circ \gamma(\vec{u}) = \mathbb{C}_{k,r} \circ \gamma(\vec{v})$. The invariance of γ implies that $\gamma(\vec{u}^\tau) = \gamma(\vec{v}^\tau)$ and that

$$\{\vec{x} \in V^r \mid \gamma(\vec{v}^\tau\langle\tau(\vec{\imath}\,), \vec{x}\rangle) = \sigma^\tau\} = \{\vec{x} \in V^r \mid \gamma(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \sigma\},$$

for all $\tau \in \mathrm{Sym}(k)$ and $\vec{\imath} \in [k]^{(r)}$. Hence,

$$\left|\{\vec{x} \in V^r \mid \gamma(\vec{u}^\tau\langle\tau(\vec{\imath}\,), \vec{x}\rangle) = \sigma^\tau\}\right| = \left|\{\vec{x} \in V^r \mid \gamma(\vec{v}^\tau\langle\tau(\vec{\imath}\,), \vec{x}\rangle) = \sigma^\tau\}\right|.$$

Therefore, $\mathbb{C}_{k,r} \circ \gamma(\vec{u}^\tau) = \mathbb{C}_{k,r} \circ \gamma(\vec{v}^\tau)$; thus showing that $\mathbb{C}_{k,r} \circ \gamma$ is invariant.

As γ is graph-like and $\mathbb{C}_{k,r} \circ \gamma(\vec{u}) = \mathbb{C}_{k,r} \circ \gamma(\vec{v})$, it follows from (1) in Lemma 4.7 that for all $\vec{\imath} \in [k]^{(r)}$ with some entry equal to k,

$$\left|\{\vec{x} \in V^r \mid \gamma(\vec{u}'\langle\vec{\imath}, \vec{x}\rangle) = \sigma\}\right| = \left|\{\vec{x} \in V^r \mid \gamma(\vec{v}'\langle\vec{\imath}, \vec{x}\rangle) = \sigma\}\right|. \tag{10}$$

If $\vec{\imath} \in [k]^{(r)}$ has an entry equal $k-1$ but none equal to k, then (2) in Lemma 4.7 implies that

$$\left|\{\vec{x} \in V^r \mid \gamma(\vec{v}'\langle\vec{\imath}, \vec{x}\rangle) = \sigma\}\right| = \left|\{\vec{x} \in V^r \mid \gamma(\vec{v}^\pi\langle\pi(\vec{\imath}\,), \vec{x}\rangle) = \sigma\}\right|.$$

From the invariance of $\mathbb{C}_{k,r} \circ \gamma$, we deduce that (10) holds also for such values of $\vec{\imath}$. Finally, if no entry of $\vec{\imath}$ is equal to k or $k-1$, γ satisfies (7). As the right-hand side of the latter is a disjoint union and for all $\sigma \in \mathrm{Im}\,\gamma$

$$\left|\{\vec{x} \in V^r \mid \gamma(\vec{u}\langle\vec{\imath}, \vec{x}\rangle) = \sigma\}\right| = \left|\{\vec{x} \in V^r \mid \gamma(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \sigma\}\right|,$$

it follows that (10) holds also when neither k nor $k-1$ are entries of $\vec{\imath}$. Thus, $C_{k,r} \circ \gamma(\vec{u}') = \mathbb{C}_{k,r} \circ \gamma(\vec{v}')$ and $\mathbb{C}_{k,r} \circ \gamma$ is therefore graph-like.

Suppose γ is $\mathbb{C}_{k,r}$-stable. Since γ is graph-like, it holds that

$$\{\vec{x} \in V^r \mid \bar{\gamma}(\mathrm{pr}_{k-1}\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \kappa\} = \bigcup_{\substack{\sigma \in \mathrm{Im}\,\gamma \\ \bar{\sigma} = \kappa}} \{\vec{x} \in V^r \mid \gamma(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \sigma\}, \tag{11}$$

for all $\vec{\imath} \in [k-1]^{(r)}$ and $\kappa \in \mathrm{Im}\,\bar{\gamma}$. Since the right-hand side of the above is a disjoint union and

$$\left|\{\vec{x} \in V^r \mid \gamma(\vec{u}\langle\vec{\imath}, \vec{x}\rangle) = \sigma\}\right| = \left|\{\vec{x} \in V^r \mid \gamma(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \sigma\}\right|,$$

it then follows from (11) that

$$\left|\{\vec{x} \in V^r \mid \bar{\gamma}(\mathrm{pr}_{k-1}\vec{u}\langle\vec{\imath}, \vec{x}\rangle) = \sigma\}\right| = \left|\{\vec{x} \in V^r \mid \bar{\gamma}(\mathrm{pr}_{k-1}\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \sigma\}\right|.$$

Thus, $\bar{\gamma}$ is $\mathbb{C}_{k-1,r}$-stable.

A very similar argument to the case of $\mathrm{WL}_{k,r}$ shows that $[\gamma]^{\mathbb{C}_{k,r}}$ can be computed in time $|V|^{O(k)}$.

$\mathrm{IM}(\mathbb{F})$ is a refinement procedure. We proceed via the same proof strategy as for the above refinement procedures.

Suppose $\mathrm{IM}_k^{\mathbb{F}} \circ \gamma(\vec{u}) = \mathrm{IM}_k^{\mathbb{F}} \circ \gamma(\vec{v})$. Then $\gamma(\vec{u}) = \gamma(\vec{v})$; and hence, $\gamma(\vec{u}^\tau) = \gamma(\vec{v}^\tau)$ for all $\tau \in \mathrm{Sym}(k)$. Note that for any $\sigma \in \mathrm{Im}\,\gamma$, $\chi^{\gamma,\vec{v}}_{\vec{\imath},\sigma} = \chi^{\gamma,\vec{v}^\tau}_{\tau(\vec{\imath}}$ $), \sigma^\tau$. From this one deduces that $\mathrm{IM}_k^{\mathbb{F}} \circ \gamma(\vec{u}^\tau) = \mathrm{IM}_k^{\mathbb{F}} \circ \gamma(\vec{v}^\tau)$; whence, $\mathrm{IM}_k^{\mathbb{F}} \circ \gamma$ is invariant.

From Lemma 4.7, it follows that $\chi^{\gamma,\vec{v}'}_{\vec{\imath},\sigma} = \chi^{\gamma,\vec{v}}_{\vec{\imath},\sigma}$ if k is an entry of $\vec{\imath}$; $\chi^{\gamma,\vec{v}'}_{\vec{\imath},\sigma} = \chi^{\gamma,\vec{v}^{\pi}}_{\pi(\vec{\imath}),\sigma}$ if $k-1$ is an entry of $\vec{\imath}$, but k is not; and

$$\chi^{\gamma,\vec{v}'}_{\vec{\imath},\sigma} = \sum_{\substack{\kappa\in\operatorname{Im}\gamma\\ \bar{\kappa}=\sigma}} \chi^{\gamma,\vec{v}}_{\vec{\imath},\kappa}$$

if neither k nor $k-1$ are entries of $\vec{\imath}$. Since $S\chi^{\gamma,\vec{v}}_{\vec{\imath},\sigma}S^{-1} = \chi^{\gamma,\vec{u}}_{\vec{\imath},\sigma}$ for all $\sigma \in \operatorname{Im}\gamma$, then $S\chi^{\gamma,\vec{v}'}_{\vec{\imath},\sigma}S^{-1} = \chi^{\gamma,\vec{u}'}_{\vec{\imath},\sigma}$ for all $\sigma \in \operatorname{Im}\gamma$. From this, $\mathrm{IM}^{\mathbb{F}}_k \circ \gamma(\vec{u}') = \mathrm{IM}^{\mathbb{F}}_k \circ \gamma(\vec{v}')$ follows; and thus, $\mathrm{IM}^{\mathbb{F}}_k \circ \gamma$ is graph-like.

From the fact that γ is graph-like we deduce that for all $\vec{\imath} \in [k-1]^{(2)}$,

$$\chi^{\bar{\gamma},\mathrm{pr}_{k-1}\vec{v}}_{\vec{\imath},\sigma} = \sum_{\substack{\kappa\in\operatorname{Im}\gamma\\ \bar{\kappa}=\sigma}} \chi^{\gamma,\vec{v}}_{\vec{\imath},\kappa}.$$

Thus, if $S\chi^{\gamma,\vec{v}}_{\vec{\imath},\sigma}S^{-1} = \chi^{\gamma,\vec{u}}_{\vec{\imath},\sigma}$ for all $\sigma \in \operatorname{Im}\gamma$, then $S\chi^{\bar{\gamma},\mathrm{pr}_{k-1}\vec{v}}_{\vec{\imath},\sigma}S^{-1} = \chi^{\bar{\gamma},\mathrm{pr}_{k-1}\vec{u}}_{\vec{\imath},\sigma}$ for all $\sigma \in \operatorname{Im}\gamma$. In particular, if γ is $\mathrm{IM}^{\mathbb{F}}_k$-stable, then $\bar{\gamma}$ is $\mathrm{IM}^{\mathbb{F}}_{k-1}$-stable.

Finally, showing that $[\gamma]^{\mathrm{IM}^{\mathbb{F}}_k}$ can be computed in time $|V|^{O(k)}$ is similar to the previous cases. Again, the number of refinement steps is at most $|V|^k$. However, at each step, and for each pair of tuples, we need to perform a simultaneous similarity test. For this, we rely on the fact that simultaneous similarity is decidable in polynomial time. This follows from the fact that testing simultaneous similarity can be reduced in polynomial time to testing module isomorphism (see [Chistov et al. 1997], for example), and the polynomial-time algorithm for the latter problem over any field is given by Brookbanks and Luks in [2008]. □

For a refinement procedure $\mathbf{R} = \{R_1, R_2, \dots\}$, let $\mathcal{S}_{\mathbf{R}} = \{\bar{R}_1, \bar{R}_2, \dots\}$ be the family of mappings where for all graphs Γ, $\bar{R}_1(\Gamma) = \Gamma$ and $\bar{R}_k(\Gamma) = \mathrm{pr}_2[\alpha_{k,\Gamma}]^{R_k}$ for $k \geq 2$. Then, from the above discussion, $\mathcal{S}_{\mathbf{R}}$ is a SPAS. We define $\mathcal{S}_{\mathbf{WL},r} = \{\overline{\mathrm{WL}}_{1,r}, \overline{\mathrm{WL}}_{2,r}, \dots\}$, $\mathcal{S}_{\mathbf{C},r} = \{\overline{\mathbb{C}}_{1,r}, \overline{\mathbb{C}}_{2,r}, \dots\}$, and $\mathcal{S}_{\mathbf{IM}(\mathbb{F})} = \{\overline{\mathrm{IM}}^{\mathbb{F}}_1, \overline{\mathrm{IM}}^{\mathbb{F}}_2, \dots\}$ to be the SPASs obtained from the refinement procedures $\mathbf{WL}_r$, $\mathbf{C}_r$, and $\mathbf{IM}(\mathbb{F})$, respectively.

5. Proof of Theorem 1.3

Hereafter, when talking of the families of operators $\mathbf{WL}_r$ and $\mathbf{C}_r$, we write $\mathbf{WL} = \{\mathrm{WL}_1, \mathrm{WL}_2, \dots\}$ and $\mathbf{C} = \{\mathbb{C}_1, \mathbb{C}_2, \dots\}$ to denote the families $\mathbf{WL}_1 = \{\mathrm{WL}_{1,1}, \mathrm{WL}_{2,1}, \dots\}$ and $\mathbf{C}_1 = \{\mathbb{C}_{1,1}, \mathbb{C}_{2,1}, \dots\}$, respectively. The distinction between the *procedure* $\mathbf{WL}_k$ and the *operator* WL_k should not cause any confusion, likewise for the procedure $\mathbf{C}_k$ and the operator $\mathbb{C}_k$.

For showing that for two refinement procedures $\mathbf{X} = \{X_1, X_2, \dots\}$ and $\mathbf{Y} = \{Y_1, Y_2, \dots\}$, their respective SPAS satisfy $\mathcal{S}_{\mathbf{X}} \succeq \mathcal{S}_{\mathbf{Y}}$, our general strategy is as follows. For each $k \in \mathbb{N}$, we find some $k' \in \mathbb{N}$ such that for any graph Γ and any Y_k-stable graph-like Γ-partition $\gamma \in \mathcal{P}(V^k)$, there is some $X_{k'}$-stable graph-like Γ-partition $\gamma' \in \mathcal{P}(V^{k'})$ such that $\mathrm{pr}_2\gamma' \succeq \mathrm{pr}_2\gamma$. By taking γ' to be $[\alpha_{k,\Gamma}]^{X_{k'}}$ (which is graph-like by Proposition 4.6), we have that $\bar{X}_k(\Gamma) = \mathrm{pr}_2\gamma' \succeq \mathrm{pr}_2\gamma \succeq \bar{Y}_k(\Gamma)$ for all graphs Γ. The last inequality follows from the fact that $\bar{Y}_k(\Gamma)$ is the 2-projection of a minimal Y_k-stable partition refining $\alpha_{k,\Gamma}$.

For the proofs in this section, the operators $\mathrm{WL}_{k,r}$ and $\mathbb{C}_{k,r}$ are only considered in the case $k > r$, since the statements hold trivially when $k \leq r$.

Lemma 5.1. *For all $k, r \in \mathbb{N}$, any $\mathrm{WL}_{k,r}$-stable partition is $\mathbb{C}_{k,r}$-stable.*

Proof. Let $\gamma \in \mathcal{P}(V^k)$ be $\mathrm{WL}_{k,r}$-stable. The size of $\{\vec{x} \in V^r \mid \gamma(\vec{v}\langle \vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\}$ is then independent of the choice of $\vec{v}$ from the equivalence class $[\vec{v}]_\gamma$ for all $\vec{\phi} \in \Phi^{\gamma,r}$, by Proposition 4.5. For each $\vec{\jmath} \in [k]^{(r)}$ and $\sigma \in \operatorname{Im}\gamma$,

$$\{\vec{x} \in V^r \mid \gamma(\vec{v}\langle \vec{\jmath}, \vec{x}\rangle) = \sigma\} = \bigcup_{\substack{\vec{\phi}\in\Phi^{\gamma,r}\\ \phi_{\vec{\jmath}}=\sigma}} \{\vec{x} \in V^r \mid \gamma(\vec{v}\langle \vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\}.$$

The right-hand side of the above is a disjoint union of sets, whose sizes are independent of the choice of $\vec{v}$, from the equivalence class $[\vec{v}]_\gamma$. Hence, the size of the left-hand side, is also the same for any choice of $\vec{v}$ from the class $[\vec{v}]_\gamma$, implying that γ is $\mathbb{C}_{k,r}$-stable. □

Corollary 5.2. *For any graph* Γ *and* $k, r \in \mathbb{N}$,

$$\overline{\mathbb{C}}_{k,r}(\Gamma) \preceq \overline{\mathrm{WL}}_{k,r}(\Gamma).$$

Lemma 5.3. *For all* $k, r \in \mathbb{N}$, *any* $\mathbb{C}_k$*-stable partition is* $\mathbb{C}_{k,r}$*-stable.*

Proof. Let $\gamma \in \mathcal{P}(V^k)$ be $\mathbb{C}_k$-stable. Then, by definition, it is $\mathbb{C}_{k,1}$ stable.

Suppose γ is $\mathbb{C}_{k,m}$-stable for some $m < k$. Let $\vec{\jmath} \in [k]^{(m)}$. Since $m < k$, there is some $j' \in [k]$ such that $j' \neq j_i$ for all $i \in [m]$. For all $\sigma, \tau \in \operatorname{Im}\gamma$ and $\vec{v} \in V^k$ such that $\gamma(\vec{v}) = \sigma$, define

$$p_{\sigma\tau} = \bigl|\{u \in V \mid \gamma(\vec{v}\langle j', u\rangle) = \tau\}\bigr|$$

and

$$q_{\sigma\tau} = \bigl|\{\vec{w} \in V^m \mid \gamma(\vec{v}\langle \vec{\jmath}, \vec{w}\rangle) = \tau\}\bigr|.$$

Because γ is $\mathbb{C}_k$-stable and, by the induction hypothesis, also $\mathbb{C}_{k,m}$-stable, $p_{\sigma\tau}$ and $q_{\sigma\tau}$ are independent of the choice of $\vec{v}$ from the equivalence class $[\vec{v}]_\gamma$.

Observe that

$$\{\vec{w}\cdot u \in V^{m+1} \mid \gamma(\vec{v}\langle \vec{\jmath}\cdot j', \vec{w}\cdot u\rangle) = \tau\} = \{\vec{w}\cdot u \in V^{m+1} \mid \gamma((\vec{v}\langle \vec{\jmath}, \vec{w}\rangle)\langle j', u\rangle) = \tau\}.$$

From this one deduces that

$$\bigl|\{\vec{w}\cdot u \in V^{m+1} \mid \gamma(\vec{v}\langle \vec{\jmath}\cdot j', \vec{w}\cdot u\rangle) = \tau\}\bigr| = \sum_{\alpha\in\operatorname{Im}\gamma} p_{\sigma\alpha}q_{\alpha\tau}.$$

The right-hand side of the above is independent of the choice of $\vec{v}$ from the equivalence class $[\vec{v}]_\gamma$. Hence, γ is $\mathbb{C}_{k,m+1}$-stable. Thus, by induction, it is $\mathbb{C}_{k,r}$ stable for all r. □

Corollary 5.4. *For any graph* Γ *and* $k, r \in \mathbb{N}$,

$$\overline{\mathbb{C}}_k(\Gamma) \succeq \overline{\mathbb{C}}_{k,r}(\Gamma).$$

Note that Lemmas 5.1 and 5.3 are not restricted to graph-like partitions.

Lemma 5.5. *For all* $k, r \in \mathbb{N}$, *the* k*-projection of a* $\mathbb{C}_{k+r,r}$*-stable graph-like partition is* $\mathrm{WL}_{k,r}$*-stable.*

Proof. Let $\vec{v}, \vec{v}' \in V^k$, and let $\bar{\gamma} = \mathrm{pr}_k\gamma$ for some $\mathbb{C}_{k+r,r}$-stable graph-like partition $\gamma \in \mathcal{P}(V^{k+r})$. Because γ is graph-like, by Lemma 3.6 it follows that for all $\vec{\imath} \in [k+r]^{(r)}$,

$$\gamma(\vec{v}\langle \vec{\imath}, \mathrm{pr}_{\vec{\jmath}}\vec{v}\rangle) = \gamma(\vec{v}'\langle \vec{\imath}, \mathrm{pr}_{\vec{\jmath}}\vec{v}'\rangle),$$

where $\vec{j} = (k+1, k+2, \dots, k+r) \in [k+r]^{(r)}$. In particular, for all $\vec{\imath} \in [k]^{(r)}$,

$$\bar{\gamma}(\mathrm{pr}_k \vec{v}\langle \vec{\imath}, \mathrm{pr}_{\vec{j}} \vec{v}\rangle) = \bar{\gamma}(\mathrm{pr}_k \vec{v}'\langle \vec{\imath}, \mathrm{pr}_{\vec{j}} \vec{v}'\rangle). \tag{12}$$

For all $\vec{v} \in V^{k+r}$, define $\vec{\Delta} \in (\Phi^{\bar{\gamma},r})^{\mathrm{Im}\,\gamma}$ to be

$$(\Delta_{\gamma(\vec{v})})_{\vec{\imath}} = \bar{\gamma}(\mathrm{pr}_k \vec{v}\langle \vec{\imath}, \mathrm{pr}_{\vec{j}} \vec{v}\rangle).$$

One deduces from (12) that $\vec{\Delta}$ is well defined.

For any $\vec{\phi} \in \Phi^{\bar{\gamma},r}$, it follows from the definition of $\vec{\Delta}$ that:

$$\{\vec{u} \in V^r \mid \bar{\gamma}(\mathrm{pr}_k \vec{v}\langle \vec{\imath}, \vec{u}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(r)}\} = \bigcup_{\substack{\sigma \in \mathrm{Im}\,\gamma \\ \Delta_\sigma = \vec{\phi}}} \{\vec{u} \in V^r \mid \gamma(\vec{v}\langle \vec{j}, \vec{u}\rangle) = \sigma\}.$$

$\mathbb{C}_{k,r}$-stability of γ implies that the size of the right-hand side of the above is independent of the choice of $\vec{v}$ from the equivalence class $[\vec{v}]_\gamma$. Since γ is graph-like, the size of the left-hand side is independent of the choice of $\vec{w} \in V^{k+r}$, where $\mathrm{pr}_k \vec{w} \in [\mathrm{pr}_k \vec{v}]_{\bar{\gamma}}$, and the result follows. □

Corollary 5.6. *For any graph* Γ *and* $k, r \in \mathbb{N}$,

$$\overline{\mathbb{C}}_{k+r,r}(\Gamma) \succeq \overline{\mathrm{WL}}_{k,r}(\Gamma).$$

In particular,

$$\overline{\mathbb{C}}_{k+1}(\Gamma) \succeq \overline{\mathrm{WL}}_k(\Gamma).$$

Lemma 5.7. *For all* $k, r \in \mathbb{N}$, *the* $(k-r+1)$*-projection of a* $\mathbb{C}_{k,r}$*-stable graph-like partition is* $\mathbb{C}_{k-r+1}$*-stable.*

Proof. Let $\bar{\gamma} = \mathrm{pr}_{k-r+1}\gamma$, and let $\vec{v} \in V^k$. Fix some $i \in [k-r]$, and let $\vec{j} \in [k]^{(k-r+1)}$ be defined by $\vec{j} = (i, k-r+1, k-r+2, \dots, k-1, k)$. Fix $u \in \vec{V}$, and let $\vec{w} = (v_{k-r+1}, v_{k-r+1}, \dots, v_{k-r+1}) \in V^{k-r}$.

Since γ is graph-like, for any $u' \in V$,

$$\gamma(\vec{v}\langle \vec{j}, u\cdot\vec{w}\rangle) = \gamma(\vec{v}\langle \vec{j}, u'\cdot\vec{w}\rangle) \iff \bar{\gamma}(\mathrm{pr}_k \vec{v}\langle i, u\rangle) = \bar{\gamma}(\mathrm{pr}_k \vec{v}\langle i, u'\rangle);$$

and therefore,

$$\{u' \in V \mid \bar{\gamma}(\mathrm{pr}_k \vec{v}\langle i, u\rangle) = \bar{\gamma}(\mathrm{pr}_k \vec{v}\langle i, u'\rangle)\} = \{u' \in V \mid \gamma(\vec{v}\langle \vec{j}, u'\cdot\vec{w}\rangle) = \gamma(\vec{v}\langle \vec{j}, u\cdot\vec{w}\rangle)\}.$$

The size of the right-hand side of the latter is independent of the choice of $\vec{v}$ from the equivalence class $[\vec{v}]_\gamma$. Hence, as γ is graph-like, the size of the left-hand side is independent of the choice of $\vec{w} \in V^k$, where $\mathrm{pr}_{k-r+1}\vec{w} \in [\mathrm{pr}_{k-r+1}\vec{v}]_{\bar{\gamma}}$. The result follows. □

Corollary 5.8. *For any graph* Γ *and* $k, r \in \mathbb{N}$,

$$\overline{\mathbb{C}}_{k,r}(\Gamma) \succeq \overline{\mathbb{C}}_{k-r+1}(\Gamma).$$

The results proved so far involved showing that the projection of a partition satisfying some stability condition satisfied some other stability condition. In order to prove that $\mathcal{S}_{\mathrm{WL}}$ dominates $\mathcal{S}_{\mathbb{C}}$, we *extend* a WL_k-stable partition to a $\mathbb{C}_{k+1}$-stable one. Let γ be graph-like and be a WL_k-stable partition of V^k. Define $\hat{\gamma} \in \mathcal{P}(V^{k+1})$ as

$$\hat{\gamma}(\vec{v}, w) = (\gamma(\vec{v}), \gamma(\vec{v}\langle 1, w\rangle), \gamma(\vec{v}\langle 2, w\rangle), \ldots, \gamma(\vec{v}\langle k, w\rangle)), \quad \forall \vec{v} \in V^k, w \in V.$$

That is to say, $\hat{\gamma}(\vec{v}, w) = \hat{\gamma}(\vec{v}', w')$ if and only if $\gamma(\vec{v}) = \gamma(\vec{v}')$ and $\gamma(\vec{v}\langle i, w\rangle) = \gamma(\vec{v}'\langle i, w'\rangle)$ for all $i \in [k]$. First, we need to prove that $\hat{\gamma}$ is graph-like.

Lemma 5.9. *Let γ and $\hat{\gamma}$ be as above. If γ is invariant, then $\hat{\gamma}$ is invariant.*

Proof. Because $\mathrm{Sym}(k+1) = \langle \mathrm{Sym}(k), (k, k+1)\rangle$, it is sufficient to show that $\hat{\gamma}(\vec{u} \cdot u') = \hat{\gamma}(\vec{v} \cdot v') \Longrightarrow \hat{\gamma}((\vec{u} \cdot u')^\tau) = \hat{\gamma}((\vec{v} \cdot v')^\tau)$, where $\tau = (k, k+1)$ is a transposition of $\mathrm{Sym}(k+1)$; or, equivalently,

$$\hat{\gamma}((\mathrm{pr}_{k-1}\vec{u}) \cdot u' \cdot u_k) = \hat{\gamma}((\mathrm{pr}_{k-1}\vec{v}) \cdot v' \cdot v_k). \tag{13}$$

From the definition of $\hat{\gamma}$, it holds that

$$\gamma(\vec{u}\langle k, u'\rangle) = \gamma(\vec{v}\langle k, v'\rangle).$$

Hence,

$$\gamma((\mathrm{pr}_{k-1}\vec{u}) \cdot u') = \gamma((\mathrm{pr}_{k-1}\vec{v}) \cdot v'). \tag{14}$$

Also, $\hat{\gamma}(\vec{u} \cdot u') = \hat{\gamma}(\vec{v} \cdot v')$ implies that $\gamma(\vec{u}) = \gamma(\vec{v})$; and hence, since γ is graph-like,

$$\gamma((\mathrm{pr}_{k-1}\vec{u}) \cdot u_k) = \gamma((\mathrm{pr}_{k-1}\vec{v}) \cdot v_k). \tag{15}$$

Since $\gamma(\vec{u}\langle i, u'\rangle) = \gamma(\vec{v}\langle i, v'\rangle)$ for all $i \in [k]$ and γ is invariant; for any $\tau = (i, k) \in \mathrm{Sym}(k)$, $\gamma(\vec{u}\langle i, u'\rangle^\tau) = \gamma(\vec{v}\langle i, v'\rangle^\tau)$ or, equivalently,

$$\gamma((\mathrm{pr}_{k-1}\vec{u}) \cdot u'\langle i, u_k\rangle) = \gamma((\mathrm{pr}_{k-1}\vec{v}) \cdot v'\langle i, v_k\rangle). \tag{16}$$

Combining (14), (15), and (16), we see that (13) follows. $\square$

Consequently, if γ is graph-like, so is $\hat{\gamma}$. Indeed, $\hat{\gamma}$ is k-consistent by construction. Also, since $\hat{\gamma}$ is invariant and γ satisfies (6), then $\hat{\gamma}$ also satisfies (6). Finally, observe that if γ is a Γ-partition for some graph Γ, then so is $\hat{\gamma}$.

Lemma 5.10. *Let γ and $\hat{\gamma}$ be as above. If γ is graph-like and* WL_k *stable, then $\hat{\gamma}$ is $\mathbb{C}_{k+1}$-stable.*

Proof. As observed, if γ is graph-like, then $\hat{\gamma}$ is graph-like by construction. Since γ is WL_k-stable, it follows that for all $\sigma \in \mathrm{Im}\,\hat{\gamma}$ and $\vec{v} \in V^{k+1}$, the size of $\{x \in V \mid \hat{\gamma}(\vec{v}\langle k+1, x\rangle) = \sigma\}$ is independent of the choice of $\vec{v}$ from the equivalence class $[\vec{v}]_{\hat{\gamma}}$. By invariance of $\hat{\gamma}$, one deduces that for all $i \in [k+1]$, the size of $\{x \in V \mid \hat{\gamma}(\vec{v}\langle i, x\rangle) = \sigma\}$ is independent of the choice of $\vec{v}$ in the equivalence class $[\vec{v}]_{\hat{\gamma}}$. Thus, $\hat{\gamma}$ is $\mathbb{C}_{k+1}$-stable. $\square$

Corollary 5.11. *For any graph Γ,*

$$\overline{\mathbb{C}}_{k+1}(\Gamma) \succeq \overline{\mathrm{WL}}_k(\Gamma).$$

In particular, combining this with Corollary 5.6, one deduces that for all graphs Γ and $k \in \mathbb{N}$,

$$\overline{\mathbb{C}}_{k+1}(\Gamma) \approx \overline{\mathrm{WL}}_k(\Gamma). \tag{17}$$

This can be seen as a combinatorial reformulation of Theorem 5.2 in [Cai et al. 1992] proven without referring to the Immerman–Lander pebble game.

Proof of Theorem 1.3. Let Γ be any graph. From (17) it easily follows that $\mathcal{S}_{\mathbf{WL}} \simeq \mathcal{S}_{\mathbf{C}}$.

It follows from Corollaries 5.4 and 5.8 that for all $r, k \in \mathbb{N}$,

$$\overline{\mathbb{C}}_k(\Gamma) \succeq \overline{\mathbb{C}}_{k,r}(\Gamma) \succeq \overline{\mathbb{C}}_{k-r+1}(\Gamma);$$

and hence, for any $r \in \mathbb{N}$,

$$\mathcal{S}_{\mathbf{C}} \simeq \mathcal{S}_{\mathbf{C},r}.$$

From Corollaries 5.6 and 5.2, we deduce that for all $r \in \mathbb{N}$,

$$\mathcal{S}_{\mathbf{WL},r} \simeq \mathcal{S}_{\mathbf{C},r}.$$ □

6. The Evdokimov–Ponomarenko SPAS

In this section, we apply the language and results of this paper to derive the following statement about the SPAS, $\mathcal{S}_{\mathrm{EP}} = \{\mathrm{EP}_1, \mathrm{EP}_2, \dots\}$, introduced by Evdokimov and Ponomarenko in [1999].

Theorem 6.1 (Evdokimov, Ponomarenko, 1999). *For any graph Γ and $k \in \mathbb{N}$,*

$$\overline{\mathrm{WL}}_k(\Gamma) \preceq \mathrm{EP}_k(\Gamma) \preceq \overline{\mathrm{WL}}_{3k}(\Gamma). \tag{18}$$

In particular, $\mathcal{S}_{\mathrm{EP}} \simeq \mathcal{S}_{\mathbf{WL}}$.

We first define the mapping EP_k in terms of the refinement operator WL_2.

For $P \in \mathcal{P}(V^2)$, $k \in \mathbb{N}$, and some symbol Δ, set $P^{(k)} : (V^k)^2 \to \operatorname{Im} P^k \cup \operatorname{Im} P^k \times \{\Delta\}$ to be the following labelled partition of $(V^k)^2$:

$$P^{(k)}(\vec{u}, \vec{v}) = \begin{cases} (P(u_1, v_1), P(u_2, v_2), \dots, P(u_k, v_k), \Delta) & \text{if } \vec{u} = \vec{v} = (u, u, \dots, u) \text{ for some } u \in V, \\ (P(u_1, v_1), P(u_2, v_2), \dots, P(u_k, v_k)) & \text{otherwise,} \end{cases}$$

for all $\vec{u}, \vec{v} \in V^k$. Note that we have denoted elements of $(V^k)^2$ by $(\vec{u}, \vec{v})$ as opposed to $(\vec{u} \cdot \vec{v})$ to emphasize the difference between a set of *pairs* of k-tuples as opposed to a set of $2k$-tuples. Hence, if Γ is a graph, then $[\Gamma^{(k)}]^{\mathrm{WL}_2}$ is a coherent configuration on V^k, which we denote by $\hat{\Gamma}^{(k)}$. Observe that the set

$$I_\Delta = \{(u, u, \dots, u) \mid u \in V\} \subset V^k$$

is a union of cells of $\hat{\Gamma}^{(k)}$. Hence, the restriction $\hat{\Gamma}^{(k)}|_{I_\Delta^2}$ is a coherent configuration on I_Δ. Let $\delta : V^2 \to I_\Delta^2$ be the map

$$(u, v) \mapsto (\vec{u}, \vec{v}), \tag{19}$$

where $\vec{u} = (u, u, \dots, u) \in V^k$ and $\vec{v} = (v, v, \dots, v) \in V^k$. We define $\mathrm{EP}_k(\Gamma) = \hat{\Gamma}^{(k)} \circ \delta$, which is a coherent configuration on V.

Theorem 1.1 in [Evdokimov et al. 1999] shows that the family $\mathcal{S}_{\mathrm{EP}} = \{\mathrm{EP}_1, \mathrm{EP}_2, \dots\}$ forms a SPAS. Furthermore, the authors also prove the following properties for a binary relation R on V. Set $X_R^k = \{(u, v, \dots, v) \in V^k \mid (u, v) \in R\}$.

Proposition 6.2 (Proposition 3.6 in [Evdokimov et al. 1999]). $R = \{(u, v) \in V^2 \mid \mathrm{EP}_k(\Gamma)(u, v) = \sigma\}$ *for some* $\sigma \in \operatorname{Im} \mathrm{EP}_k(\Gamma)$ *if and only if* X_R^k *is a cell of* $\hat{\Gamma}^{(k)}$.

Proposition 6.3 (Proposition 3.6 in [Evdokimov et al. 1999]). *The equivalence classes of* $[\alpha_{k,\Gamma}]^{\mathrm{WL}_k}$ *are unions of cells of* $\hat{\Gamma}^{(k)}$.

From these, one can deduce the left-most relation of (18).

Lemma 6.4. *For any graph* Γ *and* $k \in \mathbb{N}$,

$$\overline{\mathrm{WL}}_k(\Gamma) \preceq \mathrm{EP}_k(\Gamma).$$

Proof. For $k = 1$, the statement is trivial. Let $k \geq 2$, and set $\Lambda = [\alpha_{k,\Gamma}]^{\mathrm{WL}_k}$. By Proposition 6.3 and because Λ is graph-like, for any $u, v \in V$ the equivalence class $[(v, u, \dots, u)]_\Lambda$ is a union of cells of $\hat{\Gamma}^{(k)}$ whose elements are all of the form $(v', u', \dots, u')$. Since $\overline{\mathrm{WL}}_k(\Gamma) = \mathrm{pr}_2\Lambda$, it follows from Proposition 6.2 that its equivalence classes are unions of equivalence classes of $\mathrm{EP}_k(\Gamma)$. Hence $\overline{\mathrm{WL}}_k(\Gamma) \preceq \mathrm{EP}_k(\Gamma)$. □

We now apply Lemmas 5.1 and 5.5 to show the right-most relation of (18). For all $k, p \in \mathbb{N}$, let $\psi_{k,p} : V^{pk} \to (V^k)^p$ be the map

$$\vec{v} \mapsto (\vec{w}_1, \vec{w}_2, \dots, \vec{w}_p),$$

where $\vec{w}_i = (v_{1+(i-1)k}, v_{2+(i-1)k}, \dots, v_{k+(i-1)k})$.

Lemma 6.5. *For any graph* Γ *and* $k \in \mathbb{N}$,

$$\mathrm{EP}_k(\Gamma) \preceq \overline{\mathrm{WL}}_{3k}(\Gamma).$$

Proof. Let $\Lambda = [\alpha_{3k,\Gamma}]^{\mathbb{C}_{3k,k}}$, and set $\hat{\Phi} = \Phi^{\mathrm{pr}_{2k}\Lambda,k}$. By Lemma 5.5, $\mathrm{pr}_{2k}\Lambda$ is a $\mathrm{WL}_{2k,k}$-stable partition of V^{2k}. Hence, for all $\vec{\phi} \in \hat{\Phi}$, the size of

$$\{\vec{x} \in V^k \mid \mathrm{pr}_{2k}\Lambda(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [2k]^{(k)}\}$$

is independent of the choice of $\vec{v}$ from the equivalence class $[\vec{v}]_{\mathrm{pr}_{2k}\Lambda}$. In particular, if $T = \{(1, 2, \dots, k), (k+1, k+2, \dots, 2k)\}$, then for all $\vec{\xi} \in (\mathrm{Im}\,\mathrm{pr}_{2k}\Lambda)^T$,

$$\{\vec{x} \in V^k \mid \mathrm{pr}_{2k}\Lambda(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \xi_{\vec{\imath}},\ \forall \vec{\imath} \in T\} = \bigcup_{\substack{\vec{\phi} \in \hat{\Phi} \\ \phi_{\vec{\imath}} = \xi_{\vec{\imath}},\ \forall \vec{\imath} \in T}} \{\vec{x} \in V^k \mid \mathrm{pr}_{2k}\Lambda(\vec{v}\langle\vec{\imath}, \vec{x}\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [2k]^{(k)}\}.$$

The right-hand side of the above is a disjoint union of sets whose size is independent of the choice of $\vec{v}$ from the equivalence class $[\vec{v}]_{\mathrm{pr}_{2k}\Lambda}$. Hence, the size of the left-hand side is independent of the choice of $\vec{v}$ from the equivalence class $[\vec{v}]_{\mathrm{pr}_{2k}\Lambda}$; and therefore, $(\mathrm{pr}_{2k}\Lambda) \circ \psi_{k,2}^{-1}$ is a WL_2-stable partition of $(V^k)^2$. In particular, it refines $\alpha_{2k,\Gamma} \circ \psi_{k,2}^{-1}$. But $\Gamma^{(k)} \preceq \alpha_{2k,\Gamma} \circ \psi_{k,2}^{-1}$; and thus, since $\hat{\gamma}^{(k)}$ is a minimal WL_2-stable partition of V^{2k} refining $\Gamma^{(k)}$, then $\hat{\Gamma}^{(k)} \preceq (\mathrm{pr}_{2k}\Lambda) \circ \psi_{k,2}^{-1}$. In particular, $\hat{\Gamma}^{(k)}|_{I_\Delta^2} \preceq (\mathrm{pr}_{2k}\Lambda) \circ \psi_{k,2}^{-1}|_{I_\Delta^2}$; and hence, if δ is as defined in (19),

$$\mathrm{EP}_k(\Gamma) \preceq (\mathrm{pr}_{2k}\Lambda) \circ \psi_{k,2}^{-1} \circ \delta.$$

Since Λ is graph-like, Lemma 3.6 implies that for all $u, u', v, v' \in V$,

$$\mathrm{pr}_{2k}\Lambda(u, v, \dots, v) = \mathrm{pr}_{2k}\Lambda(u', v', \dots, v') \implies \mathrm{pr}_{2k}\Lambda(\vec{u} \cdot \vec{v}) = \mathrm{pr}_{2k}\Lambda(\vec{u}' \cdot \vec{v}), \tag{20}$$

where $\vec{u} = (u, u, \ldots, u)$, $\vec{v} = (v, v, \ldots, v)$, $\vec{u}' = (u', u', \ldots, u')$, $\vec{v}' = (v', v', \ldots, v') \in V^k$. It follows that $\mathrm{pr}_2\Lambda = (\mathrm{pr}_{2k}\Lambda) \circ \psi_{k,2}^{-1} \circ \delta$. But $\mathrm{pr}_2\Lambda = \overline{\mathbb{C}}_{3k,k}(\Gamma) \preceq \overline{\mathrm{WL}}_{3k}(\Gamma)$. Thus, by (20),

$$\mathrm{EP}_k(\Gamma) \preceq \overline{\mathrm{WL}}_{3k}(\Gamma). \qquad \square$$

7. Proofs of Theorems 1.4 and 1.5

We proceed using the same strategy as for the proof of Theorem 1.3.

Lemma 7.1. *The k-projection of a graph-like $\mathrm{IM}^{\mathbb{F}}_{k+1}$-stable partition is $\mathbb{C}_k$-stable for any field $\mathbb{F}$ and $k \in \mathbb{N}$.*

Proof. Let $\gamma \in \mathcal{P}(V^{k+2})$ be a graph-like $\mathrm{IM}^{\mathbb{F}}_{k+2}$-stable partition, and set $\bar{\gamma} = \mathrm{pr}_k\gamma$. Suppose $\gamma(\vec{v}) = \gamma(\vec{v}')$. Fix some $j \in [k]$, and let $\vec{\imath} = (j, k+1) \in [k+1]^{(2)}$. As γ is graph-like, there is some $\sigma \in \mathrm{Im}\,\gamma$ such that the matrix $\chi^{\gamma,\vec{v}}_{\sigma,\vec{\imath}}$ is diagonal and nonzero. Since $\gamma(\vec{v}) = \gamma(\vec{v}')$, there is some $S \in \mathrm{GL}_V(\mathbb{F})$ such that $S\chi^{\gamma,\vec{v}}_{\sigma,\vec{\imath}}S^{-1} = \chi^{\gamma,\vec{v}'}_{\sigma,\vec{\imath}}$. So $\chi^{\gamma,\vec{v}}_{\sigma,\vec{\imath}}$ and $\chi^{\gamma,\vec{v}'}_{\sigma,\vec{\imath}}$ have the same rank and hence, the same number of 1s on the diagonal. Thus, for any $w \in V$, the size of the set $\{u \in V \mid \gamma(\vec{v}\langle\vec{\imath}, (u, u)\rangle) = \sigma\}$ is independent of the choice of $\vec{v}$ from the equivalence class $[\vec{v}]_\gamma$. Also,

$$\{u \in V \mid \bar{\gamma}(\mathrm{pr}_k\vec{v}\langle j, u\rangle) = \sigma\} = \{u \in V \mid \gamma(\vec{v}\langle\vec{\imath}, (u, u)\rangle) = \sigma\},$$

and hence, as γ is graph-like, the size of the left-hand side is independent of the choice of $\mathrm{pr}_k\vec{v}$ from the equivalence class $[\mathrm{pr}_k\vec{v}]_{\bar{\gamma}}$. The result follows. $\square$

Corollary 7.2. *For any graph Γ and $k \in \mathbb{N}$,*

$$\overline{\mathrm{IM}}^{\mathbb{F}}_{k+2}(\Gamma) \succeq \overline{\mathrm{WL}}_k(\Gamma).$$

Proof. This follows from Lemma 7.1 and the fact that $\overline{\mathbb{C}}_{k+1}(\Gamma) \approx \overline{\mathrm{WL}}_k(\Gamma)$. $\square$

Lemma 7.3. *The k-projection of a graph-like $\mathbb{C}_{k+1}$-stable partition is $\mathrm{IM}^{\mathbb{F}}_k$-stable for any $\mathbb{F}$ with $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) > |V|$.*

Proof. Let $\gamma \in \mathcal{P}(V^{k+1})$ be a graph-like $\mathbb{C}_{k+1}$-stable partition, and fix some $\vec{\imath} \in [k+1]^{(3)}$ with $i_3 = k+1$. For every $\vec{v} \in V^{k+1}$, define $g_{\vec{v}}$ to be the partition of V^3 given by $g_{\vec{v}}(\vec{x}) = \gamma(\vec{v}\langle\vec{\imath}, \vec{x}\rangle)$ for all $\vec{x} \in V^3$. Since γ is $\mathbb{C}_{k+1}$-stable, it follows that $g_{\vec{v}}$ is $\mathbb{C}_3$ stable for all $\vec{v} \in V^{k+1}$. As γ is graph-like, $\mathrm{pr}_2 g_{\vec{v}}$ is a rainbow, and it is WL_2-stable by Lemma 5.1 and thus, a coherent configuration on V.

Set $\bar{\gamma} = \mathrm{pr}_k\gamma$, and for all $\vec{y} \in V^2$, let $\bar{g}_{\vec{v}}(\vec{y}) = \bar{\gamma}(\mathrm{pr}_k\vec{v}\langle(i_1, i_2), \vec{y}\rangle)$. Then $\bar{g}_{\vec{v}} = \mathrm{pr}_2 g_{\vec{v}}$. Therefore, all nonempty relations of $\bar{g}_{\vec{v}}$ form a coherent configuration whose $\mathbb{F}$-adjacency algebra has standard basis

$$\{\chi^{\bar{\gamma},\mathrm{pr}_k\vec{v}}_{(i_1,i_2),\sigma} \mid \exists \vec{y} \in V^2, \bar{\gamma}(\mathrm{pr}_k\vec{v}\langle(i_1, i_2), \vec{y}\rangle) = \sigma\}.$$

Thus, if $\bar{\gamma}(\mathrm{pr}_k\vec{u}) = \bar{\gamma}(\mathrm{pr}_k\vec{w})$, then $\bar{g}_{\vec{u}}$ and $\bar{g}_{\vec{w}}$ are algebraically isomorphic coherent configurations. More precisely, one can check that $\mathrm{Im}\,\bar{g}_{\vec{u}} = \mathrm{Im}\,\bar{g}_{\vec{v}}$ and that the map

$$\iota : \mathrm{Im}\,\bar{g}_{\vec{u}} \to \mathrm{Im}\,\bar{g}_{\vec{v}}, \quad \sigma \mapsto \sigma$$

is an algebraic isomorphism.

For $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) > |V|$, it follows from Corollary 2.11 that there is some $S \in \mathrm{GL}_V(\mathbb{F})$ such that $S\chi^{\bar{\gamma},\mathrm{pr}_k\vec{u}}_{(i_1,i_2),\sigma}S^{-1} = \chi^{\bar{\gamma},\mathrm{pr}_k\vec{w}}_{(i_1,i_2),\sigma}$ for all $\sigma \in \mathrm{Im}\,\bar{\gamma}$. The result then follows. $\square$

Note that in the above proof, there may be other bijections $\operatorname{Im} \bar{g}_{\vec{u}} \to \operatorname{Im} \bar{g}_{\vec{v}}$ which are algebraic isomorphisms. However, it follows from the definition of the k-refinement operator $\mathrm{IM}_k^{\mathbb{F}}$, that $\vec{u}, \vec{v} \in V^k$ are in the same equivalence class of an $\mathrm{IM}_k^{\mathbb{F}}$-stable partition, only if ι, as above, is an algebraic isomorphism.

Corollary 7.4. *For all $k \in \mathbb{N}$, graph Γ with vertex set V, and field $\mathbb{F}$ such that* $\operatorname{char}(\mathbb{F}) = 0$ *or* $\operatorname{char}(\mathbb{F}) > |V|$,

$$\overline{\mathrm{WL}}_k(\Gamma) \succeq \overline{\mathrm{IM}}_k^{\mathbb{F}}(\Gamma).$$

Proof. This follows from Lemma 7.3 and the fact that $\overline{\mathbb{C}}_{k+1}(\Gamma) \approx \overline{\mathrm{WL}}_k(\Gamma)$. □

The statement of Theorem 1.4 comes from taking $\mathbb{F}$ to be of characteristic 0 in Corollaries 7.2 and 7.4. Theorem 1.5 arises instead from combining Corollary 7.4 with a construction due to Holm [2010]. The construction in the proof of Theorem 7.1 in [Holm 2010] gives, for each $k \in \mathbb{N}$ and prime number p, a graph $\Gamma_{k,p}$ for which $\overline{\mathrm{WL}}_k(\Gamma_{k,p})$ is *strictly coarser* than $\mathrm{Sch}(\Gamma_{k,p})$, but $\overline{\mathrm{IM}}_3^{\mathbb{F}}(\Gamma_{k,p}) = \mathrm{Sch}(\Gamma_{k,p})$ for any field $\mathbb{F}$ with $\operatorname{char}(\mathbb{F}) = p$. Furthermore, from the same result, it also follows that $\overline{\mathrm{IM}}_k^{\mathbb{F}}(\Gamma_{k,p})$ is strictly coarser than $\mathrm{Sch}(\Gamma_{k,p})$ whenever $\operatorname{char}(\mathbb{F}) \neq p$. This shows that the SPAS, $\mathcal{S}_{\mathrm{IM}}(\mathbb{F}_1)$ and $\mathcal{S}_{\mathrm{IM}}(\mathbb{F}_2)$, are incomparable whenever $\operatorname{char}(\mathbb{F}_1) \neq \operatorname{char}(\mathbb{F}_2)$.

8. Yet another refinement operator

There is a subtle difference between the definitions of $\mathrm{WL}_{k,r}$ and $\mathbb{C}_{k,r}$: the colours of $\mathrm{WL}_{k,r} \circ \gamma$ are *multisets of tuples* of colours of γ, whereas the colours of $\mathbb{C}_{k,r} \circ \gamma$ are *tuples of multisets* of colours of γ. We now show that a similar variation in the definition of $\mathrm{IM}_k^{\mathbb{F}}$ gives a refinement procedure whose corresponding SPAS is equivalent to $\mathcal{S}_{\mathbf{IM}(\mathbb{F})}$.

For every $\gamma \in \mathcal{P}(V^k)$, $\vec{v} \in V^k$, and $\vec{\phi} \in \Phi^{\gamma,2}$, let $\chi_{\vec{\phi}}^{\gamma,\vec{v}}$ be the adjacency matrix of the relation $\{\vec{x} \in V^2 \mid \gamma(\vec{v}\langle \vec{\imath}, (u, v)\rangle) = \phi_{\vec{\imath}},\ \forall \vec{\imath} \in [k]^{(2)}\} \subseteq V^2$. Define the mapping $\mathrm{IMt}_k^{\mathbb{F}}$ by setting $\mathrm{IMt}_1^{\mathbb{F}} \circ \gamma = \mathrm{IMt}_2^{\mathbb{F}} \circ \gamma = \gamma$, and for $k > 2$,

$$\mathrm{IMt}_k^{\mathbb{F}} \circ \gamma : V^k \to \operatorname{Im} \gamma \times (\mathrm{Mat}_V(\mathbb{F})^{\Phi^{\gamma,2}} / \sim), \quad \vec{v} \mapsto (\gamma(\vec{v}), (\chi_{\vec{\phi}}^{\gamma,\vec{v}})_{\vec{\phi} \in \Phi^{\gamma,2}}),$$

where the equivalence classes of the relation $\sim$ are the orbits of $GL_V(\mathbb{F})$ acting on the tuples by conjugation.

Similarly to Proposition 4.6, one can show that $\mathrm{IMt}_k^{\mathbb{F}}$ is a graph-like k-refinement operator for all $k \in \mathbb{N}$ and that $\mathrm{IMt}(\mathbb{F}) = \{\mathrm{IMt}_1^{\mathbb{F}}, \mathrm{IMt}_2^{\mathbb{F}}, \dots\}$ is a refinement procedure. Hence, the family $\mathcal{S}_{\mathrm{IMt}(\mathbb{F})} = \{\overline{\mathrm{IMt}}_1^{\mathbb{F}}, \overline{\mathrm{IMt}}_2^{\mathbb{F}}, \dots\}$ is a SPAS for any field $\mathbb{F}$. Also, similarly to Proposition 4.5, one may derive the following stability condition:

Proposition 8.1. *A partition $\gamma \in \mathcal{P}(V^k)$ is $\mathrm{IMt}_k^{\mathbb{F}}$-stable if and only if for all $\vec{u}, \vec{v} \in V^k$,*

$$\gamma(\vec{u}) = \gamma(\vec{v}) \implies \exists S \in \mathrm{GL}_V(\mathbb{F}), \quad \forall \vec{\phi} \in \Phi^{\gamma,2}\ S\chi_{\vec{\phi}}^{\gamma,\vec{u}} S^{-1} = \chi_{\vec{\phi}}^{\gamma,\vec{v}}.$$

In particular, the following result is analogous to Proposition 4.5:

Lemma 8.2. *Any $\mathrm{IMt}_k^{\mathbb{F}}$-stable partition is also $\mathrm{IM}_k^{\mathbb{F}}$-stable.*

Proof. For any $\vec{v} \in V^k, \vec{\imath} \in [k]^{(2)}$, and $\sigma \in \operatorname{Im} \gamma$

$$\chi_{\sigma,\vec{\imath}}^{\gamma,\vec{v}} = \sum_{\substack{\vec{\phi} \in \Phi^{\gamma,2} \\ \phi_{\vec{\imath}} = \sigma}} \chi_{\vec{\phi}}^{\gamma,\vec{v}}.$$

Hence, if $\gamma(\vec{u}) = \gamma(\vec{v})$, there is some $S \in \mathrm{GL}_V(\mathbb{F})$ such that $S\chi_{\vec{\phi}}^{\gamma,\vec{v}}S^{-1} = \chi_{\vec{\phi}}^{\gamma,\vec{u}}$ for all $\vec{\phi} \in \Phi^{\gamma,2}$. Thus,

$$S\chi_{\sigma,\vec{\imath}}^{\gamma,\vec{v}}S^{-1} = \sum_{\substack{\vec{\phi}\in\Phi^{\gamma,2}\\ \phi_{\vec{\imath}}=\sigma}} S\chi_{\vec{\phi}}^{\gamma,\vec{v}}S^{-1} = \sum_{\substack{\vec{\phi}\in\Phi^{\gamma,2}\\ \phi_{\vec{\imath}}=\sigma}} \chi_{\vec{\phi}}^{\gamma,\vec{u}} = \chi_{\sigma,\vec{\imath}}^{\gamma,\vec{u}};$$

from which follows that γ is $\mathrm{IM}_k^{\mathbb{F}}$-stable. □

Corollary 8.3. *For any graph Γ, field $\mathbb{F}$, and $k \in \mathbb{N}$,*

$$\overline{\mathrm{IM}}_k^{\mathbb{F}}(\Gamma) \preceq \overline{\mathrm{IMt}}_k^{\mathbb{F}}(\Gamma).$$

In the following result, the argument, is analogous to that of Lemma 5.5:

Lemma 8.4. *The k-projection of an $\mathrm{IM}_{k+2}^{\mathbb{F}}$-stable partition is $\mathrm{IMt}_k^{\mathbb{F}}$-stable.*

Proof. Let $\vec{u}, \vec{v} \in V^k$ be such that $\gamma(\vec{u}) = \gamma(\vec{v})$, and set $\bar{\gamma} = \mathrm{pr}_k\gamma$. Because γ is graph-like, it follows from Lemma 3.6 that for all $\vec{\imath} \in [k+2]^{(2)}$, $\gamma(\vec{v}\langle\vec{\imath}, \mathrm{pr}_{\vec{\jmath}}\vec{v}\rangle) = \gamma(\vec{u}\langle\vec{\imath}, \mathrm{pr}_{\vec{\jmath}}\vec{u}\rangle)$, where $\vec{\jmath} = (k+1, k+2) \in [k+2]^{(2)}$. In particular, for all $\vec{\imath} \in [k]^{(2)}$,

$$\bar{\gamma}(\mathrm{pr}_k\vec{v}\langle\vec{\imath}, \mathrm{pr}_{\vec{\jmath}}\vec{v}\rangle) = \bar{\gamma}(\mathrm{pr}_k\vec{u}\langle\vec{\imath}, \mathrm{pr}_{\vec{\jmath}}\vec{u}\rangle). \tag{21}$$

Define $\vec{\Delta} \in (\Phi^{\bar{\gamma},2})^{\mathrm{Im}\,\gamma}$ such that

$$(\Delta_{\gamma(\vec{v})})_{\vec{\imath}} = \bar{\gamma}(\mathrm{pr}_k\vec{v}\langle\vec{\imath}, \mathrm{pr}_{\vec{\jmath}}\vec{v}\rangle).$$

One deduces from (21) that $\vec{\Delta}$ is well defined.

For any $\vec{\phi} \in \Phi$,

$$\{\vec{u} \in V^2 \mid \bar{\gamma}(\mathrm{pr}_k\vec{v}\langle\vec{\imath}, \vec{u}\rangle) = \phi_{\vec{\imath}},\ \forall\vec{\imath} \in [k]^{(2)}\} = \bigcup_{\substack{\sigma\in\mathrm{Im}\,\gamma\\ \Delta_\sigma=\vec{\phi}}} \{\vec{u} \in V^2 \mid \gamma(\vec{v}\langle\vec{\jmath}, \vec{u}\rangle) = \sigma\}.$$

Hence, for all $\vec{\phi} \in \Phi$ it holds that

$$\chi_{\vec{\phi}}^{\gamma,\vec{v}} = \sum_{\substack{\sigma\in\mathrm{Im}\,\gamma\\ \Delta_\sigma=\vec{\phi}}} \chi_{\sigma,\vec{\jmath}}^{\gamma,\vec{v}}.$$

Because γ is $\mathrm{IM}_{k+2}^{\mathbb{F}}$-stable and $\gamma(\vec{u}) = \gamma(\vec{v})$, for each $\vec{\imath} \in [k]^{(2)}$ there is some $S \in \mathrm{GL}_V(\mathbb{F})$ such that $S\chi_{\sigma,\vec{\imath}}^{\gamma,\vec{v}}S^{-1} = \chi_{\sigma,\vec{\imath}}^{\gamma,\vec{u}}$ for all $\sigma \in \mathrm{Im}\,\gamma$. Thus, for all $\vec{\phi} \in (\mathrm{Im}\,\bar{\gamma})^{[k]^{(2)}}$,

$$S\chi_{\vec{\phi}}^{\gamma,\vec{v}}S^{-1} = \sum_{\substack{\sigma\in\mathrm{Im}\,\gamma\\ \Delta_\sigma=\vec{\phi}}} S\chi_{\sigma,\vec{\jmath}}^{\gamma,\vec{v}}S^{-1} = \sum_{\substack{\sigma\in\mathrm{Im}\,\gamma\\ \Delta_\sigma=\vec{\phi}}} \chi_{\sigma,\vec{\jmath}}^{\vec{u}} = \chi_{\vec{\phi}}^{\gamma,\vec{u}};$$

whence, $\bar{\gamma}(\vec{u}) = \bar{\gamma}(\vec{v})$. Therefore, $\bar{\gamma}$ is $\mathrm{IMt}_k^{\mathbb{F}}$-stable. □

Corollary 8.5. *For any graph Γ, field $\mathbb{F}$, and $k \in \mathbb{N}$,*

$$\overline{\mathrm{IMt}}_k^{\mathbb{F}}(\Gamma) \preceq \overline{\mathrm{IM}}_{k+2}^{\mathbb{F}}(\Gamma).$$

Theorem 1.6 follows from Corollaries 8.5 and 8.3.

9. Conclusions

The Weisfeiler–Leman algorithm is much studied in the context of graph isomorphism. It is really a family of algorithms, graded by a dimension parameter. A large number of other families of algorithms have been shown to give essentially the same graded approximations of isomorphism. The Schurian polynomial approximation schemes of Evdokimov et al. provide a general framework for comparing these families of algorithms. The invertible map operators of Dawar and Holm provide another such family of algorithms (or, more formally in the language of this paper, *refinement procedure*), but one that has greater distinguishing power than the Weisfeiler–Leman family. In the same way that $\mathbf{WL}_r$ and $\mathbf{C}_r$ were obtained from $\mathbf{WL}$ and $\mathbf{C}$, one can generalize $\mathbf{IM}(\mathbb{F})$ as follows: for every $k, r \in \mathbb{N}$, define the k-refinement operator $\mathrm{IM}^{\mathbb{F}}_{k,r}$ by setting $\mathrm{IM}^{\mathbb{F}}_{k,r} \circ \gamma = \gamma$ when $k \leq 2r$. When $k > 2r$ define:

$$\mathrm{IM}^{\mathbb{F}}_{k,r} \circ \gamma : V^k \times \to \mathrm{Im}\, \gamma \times (\mathrm{Mat}_{V^r}(\mathbb{F})^{\mathrm{Im}\, \gamma \times [k]^{(2r)}} / \sim), \qquad \vec{v} \mapsto (\gamma(\vec{v}), ((\chi^{\gamma, \vec{v}}_{\vec{\imath}, \sigma})_{\sigma \in \mathrm{Im}\, \gamma})_{\vec{\imath} \in [k]^{(2r)}}),$$

where $\chi^{\gamma,\vec{v}}_{\vec{\imath},\sigma}$ is the adjacency matrix of the relation $\{(\vec{x}, \vec{y}) \mid \gamma(\vec{v}\langle \vec{\imath}, \vec{x} \cdot \vec{y}\rangle) = \sigma\} \subseteq (V^r)^2$ and $\sim$ is the relation whose equivalence classes are the orbits of $GL_{V^r}(\mathbb{F})$ acting on the tuples by conjugation. One can show that $\mathbf{IM}_r(\mathbb{F}) = \{\mathrm{IM}^{\mathbb{F}}_{1,r}, \mathrm{IM}^{\mathbb{F}}_{2,r}, \ldots\}$ is a refinement procedure for all $r \in \mathbb{N}$. One can thus derive from it a SPAS, $\mathcal{S}_{\mathbf{IM}(\mathbb{F}),r}$, in the same manner as described in Section 4. While we were able to show that the refinement procedures $\mathbf{WL}_r$ and $\mathbf{C}_r$ do not yield SPASs more powerful than that yielded by $\mathbf{WL}$, the exact relation between $\mathcal{S}_{\mathbf{IM}(\mathbb{F})}$ and $\mathcal{S}_{\mathbf{IM}(\mathbb{F}),r}$ is still unclear and an interesting open question.

References

[Babai 2016] L. Babai, "Graph isomorphism in quasipolynomial time", pp. 684–697 in *STOC'16: Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing* (Cambridge, 2016), edited by D. Wichs and Y. Mansour, ACM, New York, 2016. MR Zbl

[Brooksbank and Luks 2008] P. A. Brooksbank and E. M. Luks, "Testing isomorphism of modules", *J. Algebra* **320**:11 (2008), 4020–4029. MR Zbl

[Cai et al. 1992] J.-Y. Cai, M. Fürer, and N. Immerman, "An optimal lower bound on the number of variables for graph identification", *Combinatorica* **12**:4 (1992), 389–410. MR Zbl

[Cameron 1999] P. J. Cameron, *Permutation groups*, Lond. Math. Soc. Student Texts **45**, Cambridge Univ. Press, 1999. MR Zbl

[Chen and Ponomarenko 2019] G. Chen and I. Ponomarenko, *Coherent configurations*, Central China Normal Univ. Press, Wuhan, 2019.

[Chistov et al. 1997] A. Chistov, G. Ivanyos, and M. Karpinski, "Polynomial time algorithms for modules over finite dimensional algebras", pp. 68–74 in *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation* (Kihei, HI, 1997), edited by W. W. Küchlin, ACM, New York, 1997. MR Zbl

[Dawar and Holm 2017] A. Dawar and B. Holm, "Pebble games with algebraic rules", *Fund. Inform.* **150**:3-4 (2017), 281–316. MR Zbl

[Dawar et al. 2019] A. Dawar, E. Grädel, and W. Pakusa, "Approximations of isomorphism and logics with linear-algebraic operators", art. id. 112 in *46th International Colloquium on Automata, Languages, and Programming* (Patras, Greece, 2019), edited by C. Baier et al., LIPIcs. Leibniz Int. Proc. Inform. **132**, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, Germany, 2019. MR

[Evdokimov and Ponomarenko 1999] S. Evdokimov and I. Ponomarenko, "On highly closed cellular algebras and highly closed isomorphisms", *Electron. J. Combin.* **6** (1999), art. id. 18. MR Zbl

[Evdokimov et al. 1999] S. Evdokimov, M. Karpinski, and I. Ponomarenko, "On a new high-dimensional Weisfeiler–Lehman algorithm", *J. Algebraic Combin.* **10**:1 (1999), 29–45. MR Zbl

[Higman 1987] D. G. Higman, "Coherent algebras", *Linear Algebra Appl.* **93** (1987), 209–239. MR Zbl

[Holm 2010] B. Holm, *Descriptive complexity of linear algebra*, Ph.D. thesis, University of Cambridge, 2010, Available at https://tinyurl.com/holmbphd.

[Mathon 1979] R. Mathon, "A note on the graph isomorphism counting problem", *Inform. Process. Lett.* **8**:3 (1979), 131–136. MR Zbl

[Zieschang 1996] P.-H. Zieschang, *An algebraic approach to association schemes*, Lecture Notes in Math. **1628**, Springer, 1996. MR Zbl

Received 3 Aug 2019. Revised 14 May 2020.

ANUJ DAWAR:

anuj.dawar@cl.cam.ac.uk
Department of Computer Science and Technology, University of Cambridge, Cambridge, United Kingdom

DANNY VAGNOZZI:

danny.vagnozzi@gmail.com
Department of Computer Science and Technology, University of Cambridge, Cambridge, United Kingdom

msp

Moscow Journal of Combinatorics and Number Theory
Vol. 9, No. 3, 2020
https://doi.org/10.2140/moscow.2020.9.253

On defining linear orders by automata

Bruno Courcelle, Irène Durand and Michael Raskin

Motivated by enumeration problems, we define linear orders $\leq_Z$ on Cartesian products $Z := X_1 \times X_2 \times \cdots \times X_n$ and on subsets of $X_1 \times X_2$ where each component set X_i is $[0, p]$ or $\mathbb{N}$, ordered in the natural way. We require that $(Z, \leq_Z)$ be isomorphic to $(\mathbb{N}, \leq)$ if it is infinite. We want linear orderings of Z such that, in two consecutive tuples z and z', at most two components differ, and they differ by at most 1.

We are interested in algorithms that determine the next tuple in Z by using local information, where "local" is meant with respect to certain graphs associated with Z. We want these algorithms to work as well for finite and infinite components X_i. We will formalise them by *deterministic graph-walking automata* and compare their enumeration powers according to the finiteness of their sets of states and the kinds of moves they can perform.

Introduction

This article is motivated by the construction of enumeration algorithms[1] [Durand 2012]. An *enumerator* E_A of a set A is an algorithm that lists its elements. For an example, we may wish to list the minimal dominating sets of a given graph. Each element is determined from the previous one and the current state of the algorithm. The set A may be countable, for example the set of prime numbers. These algorithms can allow repetitions or, on the contrary, they can be designed to eliminate them.

New enumerators can be built from existing ones. For example, an enumerator E_A for a set $A = B \cup C$ can be built from enumerators E_B and E_C for B and C respectively. For the cases where B is finite, E_A can start by enumerating B and afterwards, C. Or, it can output alternatively an element of B and one of C. This is appropriate if B is infinite or if it is extremely large and its cardinality is not known. A library of basic enumerators and operations that build enumerators by combining existing ones has been defined and implemented by I. Durand [2012].

Our initial motivation was to enrich this library by constructions for the Cartesian product $A = B \times C$. Considering $B \times C$ as a matrix, one can enumerate its elements row by row if C is finite, or column by column if B is finite, or in a diagonal way, as in Cantor's enumeration of $\mathbb{N} \times \mathbb{N}$. The latter enumeration can be formalised by the polynomial $P(x, y) = x + (x + y)(x + y + 1)/2$ that defines the rank of a pair (x, y). We do not use it for two reasons. First we want a unique algorithm that works for B and C, either finite or infinite, but P does not compute consecutive ranks for the pairs in $\{0, 1, \ldots, p\} \times \mathbb{N}$. Furthermore, we do not want to use numbers that index the sets B and C. We want to build E_A from enumerators E_B and E_C that produce the next element or report the end of the enumeration. Actually, we will use

MSC2010: 06A05, 05C38, 68R10, 68P10.

Keywords: enumeration algorithm, diagonal enumeration, graph-walking automaton, linear order.

[1] Enumeration is taken in the sense of "listing" and not in that of "counting", as in enumerative combinatorics.

enumerators E_B and E_C that can also produce the previous element, which is easily implementable with a stack.

An enumerator E_A can be seen as an automaton that produces a sequence of elements of A, hopefully exhausting it. It produces a linear order if it does not allow repetitions, and the order type is that of $\mathbb{N}$ if A is infinite.

This view is adequate for the case of $A = B \times C$ where B and C are enumerated without repetitions by E_B and E_C, and hence are linearly ordered. Then $B \times C$ can be considered as graph, shaped as a rectangular, possibly infinite, grid. An enumerator E_A can be formalised as a deterministic graph-walking automaton that traverses the grid and builds a path spanning it that represents the intended enumeration. A *graph-walking automaton* has a "head" that can move in the graph (the grid) from a vertex to a neighbouring one. The decision where to move is taken from the current state and the knowledge of the neighbouring vertices. It is convenient to use north, west, south, southeast etc. as *directions* for describing the moves. For example, the current position is on the eastern border of a finite grid if there is no east directed edge from it.

We propose different constructions of graph-walking automata, and hence of enumerators for Cartesian products $Z := X_1 \times X_2 \times \cdots \times X_n$ and for certain *affine* subsets Z of $X_1 \times X_2$ where each component set X_i is linearly ordered and has order type ω, that of $\mathbb{N}$, if it is infinite. We require that $(Z, \leq_Z)$ be isomorphic to $(\mathbb{N}, \leq)$ if it is infinite. Our orders are inspired by Cantor's diagonal enumeration of $\mathbb{N} \times \mathbb{N}$ establishing a bijection of this set with $\mathbb{N}$.

Each ordered set X_i will be taken equal to $[0, p]$ or $\mathbb{N}$, and ordered in the natural way. We want linear orderings of Z such that, in two consecutive tuples $z = (z_1, \ldots, z_n)$ and $z' = (z'_1, \ldots, z'_n)$, we have $|z_i - z'_i| \leq 1$ for each i. The reason is that we want each step to call the component enumerators for just one forward or backward step. (Cantors's enumeration does not satisfy this condition). Furthermore, we define their *distance* $d(z, z')$ as the number of indices i such that $z_i \neq z'_i$. We have a *dk-ordering* if this distance is always at most k. We will only consider d1- and d2-orderings in order to minimise the number of calls to the component enumerators made at each step.

These requirements can be expressed in terms of graphs $G_1(Z)$ and $G_2(Z)$ that we describe informally for $Z = X_1 \times X_2$. The graph $G_1(Z)$ is a planar rectangular grid with horizontal and vertical edges, and $G_2(Z)$ is $G_1(Z)$ augmented with diagonal edges in each square. A d1-ordering (resp. a d2-ordering) of Z is a Hamiltonian path in $G_1(Z)$ (resp. in $G_2(Z)$) starting at $(0, 0, \ldots, 0)$.

We are interested in algorithms that determine the tuple in Z following a tuple z by using local information, where *local* is meant with respect to $G_1(Z)$ or $G_2(Z)$, and that work as well for finite and infinite components X_i. We will formalise them by means of *deterministic graph-walking automata*, whose runs on a given graph define *walks* (a walk is like a path, but vertices can be visited several times). We will actually construct automata that only define paths, but the general definition cannot guarantee that an automaton defines a path rather than a walk. These automata traverse graphs equipped with an edge labelling where adjacent edges have different labels that we call *directions*. The set of directions is finite. At a vertex reached by a walk, the automaton determines the direction of the next edge to be traversed from the *state* (it may have infinitely many states) and some knowledge of a finite neighbourhood, for example the set of directions of the incident edges, but larger neighbourhoods may be useful, as we will see. After the traversal via the next edge, the state may be changed, according to the used transition rule. The directions for $G_1(Z)$ and $G_2(Z)$ will be among north, west, south, southeast etc. We will not develop a general theory of graph-walking automata (see [Engelfriet and Hoogeboom 2007] for a study in relation with logic, or [Fraigniaud et al. 2005]), but we will define automata well-adapted to the graphs $G_1(Z)$ and $G_2(Z)$.

Some of our main theorems are informally stated as follows.

Theorem 1. *There is no finite or infinite automaton that defines a d*1*-ordering in each set* $X_1 \times X_2$ *where each* X_i *is* $\mathbb{N}$ *or* $[0, p]$ *for some* p *by looking at distance* 1 *of the current vertex. There is a finite one, which looks at distance* 2.

The *height* of a tuple of integers is the sum of values of its components. A *level* is the set of tuples of the same height. We want to build *d2-ℓ-orderings* where levels are traversed consecutively, by increasing order of height. We say that these orderings *respect levels*. No d1-ordering can respect levels, except in very particular cases.

Theorem 2. *For each* n, *there is an automaton with* 2^{n-1} *states that defines a d2-ordering respecting levels, on any set* $Z = X_1 \times X_2 \times \cdots \times X_n$ *such that each* X_i *is* $\mathbb{N}$ *or* $[0, p]$ *for some* p.

The corresponding construction of E_Z from enumerators $E_{X_1}, \ldots, E_{X_2}$ has been implemented in the system TRAG [Durand 2012] (see also the Appendix).

We also characterise the *affine* subsets of $\mathbb{N} \times \mathbb{N}$ having d2-ℓ-orderings defined by finite automata.

Section 1 defines graph-walking automata. Section 2 describes d2-ℓ-orderings of sets $X \times Y$ where X and Y are $\mathbb{N}$ or parts of it, and the corresponding automata. Section 3 compares various d1-orderings of $X \times Y$ as in Section 2. Section 4 studies d2-ℓ-orderings of affine subsets of $\mathbb{N} \times \mathbb{N}$. Section 5 defines d2-$\ell$-orderings of Cartesian products $X_1 \times X_2 \times \cdots \times X_n$ for $n > 2$. Section 6 is a conclusion and the Appendix presents the implementation.

1. Graph-walking automata

Definition 1.1 (directions in graphs). Let $\mathcal{D}$ be a finite set called the set of *directions*. A $\mathcal{D}$-graph is a triple $G = (V, E, \mathrm{dir})$, where V and E are the vertex and edge sets of an undirected graph without loops and parallel edges, and dir is a partial mapping $V \times V \to \mathcal{D}$ such that, for all $x, y, z \in V$, $\mathrm{dir}(x, y)$ is defined if and only if x and y are adjacent, and $\mathrm{dir}(x, y) = \mathrm{dir}(x, z)$ implies $y = z$. We denote by x^d the vertex y such that $\mathrm{dir}(x, y) = d$. The *directions around* a vertex x are those d such that x^d is defined. We denote this set by $\mathcal{D}_G(x)$. It describes the *neighbourhood* of x in G.

Definition 1.2 (graph-walking automata in $\mathcal{D}$-graphs). (a) A *$\mathcal{D}$-graph-walking automaton* (or simply, a *$\mathcal{D}$-automaton*) is a tuple $\mathcal{A} = (Q, \mathcal{T}, q_{\mathrm{init}})$, where Q is the finite or countable set of *states*, $q_{\mathrm{init}} \in Q$ and $\mathcal{T}$ is the set of *transitions*: they are of the form $(q, \delta) \to (d, q')$ or $(q, \delta) \to \mathtt{End}$, where $q, q' \in Q$, $\delta \subseteq \mathcal{D}$ and $d \in \delta$. This means that the direction d is chosen by the transition in the set δ of possible ones. From the final state End, no transition is possible. An automaton is *deterministic*: each pair (q, δ) determines a single transition. If Q is infinite, we assume that it is effectively given, and that (d, q') (or End) such that $(q, \delta) \to (d, q')$ (or $(q, \delta) \to \mathtt{End}$) is computable.

(b) The walk $\pi_{\mathcal{A}}(G, a)$ in G, defined by $\mathcal{A}$ and that starts from $a \in V_G$, is

$$a = b_0 \to b_1 \to b_2 \to \cdots \to b_n \to \cdots$$

defined with the help of the sequence of states

$$q_{\text{init}} = q_0 \to q_1 \to q_2 \to \cdots \to q_n \to \cdots$$

such that, for each $n \geq 0$, $(q_n, \mathcal{D}_G(b_n)) \to (d, q_{n+1})$ and $b_{n+1} := (b_n)^d$. Informally, the state q_n at b_n and the neighbourhood $\mathcal{D}_G(b_n)$ of b_n determine in a unique way a direction d such that $(b_n)^d$ is defined and will be the next visited vertex b_{n+1}; the state q_n is updated to q_{n+1}.

We will construct deterministic $\mathcal{D}$-automata so that they define paths rather than walks.

Definition 1.3 (directions and automata in 2-dimensional grids). (a) Let $Z \subseteq \mathbb{N} \times \mathbb{N}$. We let $G_1(Z)$ be the graph with vertex set Z and edges between (x, y) and $(x, y+1)$ and between (x, y) and $(x+1, y)$. The associated directions are

$$\mathrm{dir}((x, y), (x, y+1)) := \mathtt{N},$$
$$\mathrm{dir}((x, y+1), (x, y)) := \mathtt{S},$$
$$\mathrm{dir}((x, y), (x+1, y)) := \mathtt{E},$$
$$\mathrm{dir}((x+1, y), (x, y)) := \mathtt{W}.$$

The set of directions is $\mathcal{D}_1 := \{\mathtt{N},\mathtt{S},\mathtt{E},\mathtt{W}\}$.

We let $G_2(Z)$ be augmented with "diagonal" edges between $(x, y+1)$ and $(x+1, y)$ and between (x, y) and $(x+1, y+1)$. The associated directions are as above, together with $\mathrm{dir}((x+1, y), (x, y+1)) := \mathtt{NW}$ and similarly for the three other diagonal directions. The set of directions is $\mathcal{D}_2 := \{\mathtt{N},\mathtt{S},\mathtt{E},\mathtt{W},\mathtt{NW},\mathtt{SW},\mathtt{NE},\mathtt{SE}\}$.

(b) We will construct $\mathcal{D}_1$- and $\mathcal{D}_2$-automata that order linearly certain sets Z.

In the next section, we will extend the definitions of $G_1(Z)$ and $G_2(Z)$ to subsets Z of $X_1 \times X_2 \times \cdots \times X_n$ without extending the notion of direction.

2. Definitions and first results for Cartesian products

We will order linearly sets $Z := X_1 \times X_2 \times \cdots \times X_n$ and subsets of $X_1 \times X_2$, where each component set X_i is linearly ordered with order type ω and that of $\mathbb{N}$ in the case it is infinite. We require that $(Z, \leq_Z)$ be isomorphic to $(\mathbb{N}, \leq)$ if it is infinite. Each ordered set X_i will be taken equal to $[0, p]$ or $\mathbb{N}$, and ordered in the natural way.

We want linear orderings of Z such that, in two consecutive tuples $z = (z_1, \ldots, z_n)$ and $z' = (z'_1, \ldots, z'_n)$, we have $|z_i - z'_i| \leq 1$ for each i.

Definition 2.1 (distances, heights and levels). (a) The *distance* $d(z, z')$ of $z = (z_1, \ldots, z_n)$ and $z' = (z'_1, \ldots, z'_n)$ is the number of indices i such that $z_i \neq z'_i$.

(b) In a *dk-ordering*, the distance between any two consecutive tuples is at most k. We will only consider d1- and d2-orderings.

(c) If $Z \subseteq X_1 \times X_2 \times \cdots \times X_n$, we define two graphs:

- $G_1(Z)$ has vertex set Z and an edge between $z = (z_1, \ldots, z_n)$ and $z' = (z'_1, \ldots, z'_n)$ if and only if $|z_i - z'_i| \leq 1$ for each i and $d(z, z') = 1$.
- $G_2(Z)$ is similar with an edge between z and z' if and only if $|z_i - z'_i| \leq 1$ for each i and $d(z, z')$ is 1 or 2.

Hence, a di-ordering of $Z \subseteq X_1 \times X_2 \times \cdots \times X_n$, where i is 1 or 2, is a Hamiltonian path in $G_i(Z)$ starting at (0, 0, …, 0).

(d) The *height* of a tuple of integers is the sum of the values of its components. The *level* k of $Z \subseteq X_1 \times X_2 \times \cdots \times X_n$ is the set of its tuples of height k.

(e) A d2-ℓ-ordering of Z is a d2-ordering such that the levels are traversed consecutively by increasing order of height.

We now present a diagonal enumeration[2] of $\mathbb{N} \times \mathbb{N}$, its extension to certain subsets of the form $X \times Y$ and the corresponding $\mathcal{D}_2$-graph-walking automata.

Definition 2.2 (the diagonal d2-ℓ-ordering $\leq_\Delta$ of $\mathbb{N}\times\mathbb{N}$). We define the *type* $\tau(i, j)$ of a pair $(i, j) \in \mathbb{N}\times\mathbb{N}$ as the following pair, also in $\mathbb{N} \times \mathbb{N}$:

$$\tau(i, j) := \text{IF } i + j \text{ is even THEN } (i + j, i) \text{ ELSE} (i + j, j).$$

Note that (i, j) can be recovered from $\tau(i, j)$:

$$\tau^{-1}(m, n) = \text{IF } m \text{ is even THEN } (n, m - n) \text{ ELSE } (m - n, n).$$

The pairs $(i, j) \in \mathbb{N} \times \mathbb{N}$ are ordered by increasing lexicographic order of their types $\tau(i, j)$. That is,

$$(i, j) \leq_\Delta (i', j') \quad \text{if and only if} \quad \tau(i, j) \leq_{\text{lex}} \tau(i', j').$$

We obtain a d2-ℓ-ordering of $\mathbb{N} \times \mathbb{N}$. The corresponding ordered set is denoted by $\mathbb{N}\Delta\mathbb{N}$. Its level k is the interval of pairs (i, j) such that $i + j = k$. It begins with /00/10,01/02,11,20/30,21,12,03/04, …, where we separate levels with a slash. Odd levels are traversed in reverse lexicographic order.

The pair $\texttt{next}(i, j)$ that follows (i, j) in this order is obtained by the clauses below where we use "$\wedge$" as logical "and":

$$\begin{aligned} \texttt{next}(i, j) = {} & \text{IF } i + j \text{ is even } \wedge\ j > 0 \text{ THEN} (i + 1, j - 1), \text{ ELSE} \\ & \text{IF } i + j \text{ is even } \wedge\ j = 0 \text{ THEN } (i + 1, j), \text{ ELSE} \\ & \text{IF } i + j \text{ is odd } \wedge\ i > 0 \text{ THEN } (i - 1, j + 1), \text{ ELSE} \\ & \text{IF } i + j \text{ is odd } \wedge\ i = 0 \text{ THEN } (i, j + 1), \text{ END} \end{aligned}$$

In terms of automata the property "$i + j$ is even" is handled as a *state* that we call `Down`, and similarly, "$i + j$ is odd" is a state called `Up`. In Figure 1, the vertices 02, 11, 20 of height 2 are ordered "downwards". The Boolean values of the tests "$i = 0$" and "$j = 0$" describe the four possible positions of (i, j) with respect to the borders of $G_2(\mathbb{N} \times \mathbb{N})$ represented in the plane. The condition "$i = 0$" characterises the western border and "$j = 0$" characterises the southern border. There are no northern and eastern borders.

Borders can also be detected by looking at the directions around the current position in the grid, that is a vertex of $G_2(\mathbb{N} \times \mathbb{N})$. For example, the southern border is characterised by the neighbourhoods {`N`, `E`, `NE`} and {`N`, `E`, `W`, `NW`, `NE`}. The second clause can be formalised by automaton transitions expressing the following:

(`Down`, "on the southern border") → ("move to the east", `Up`),

[2]It is not Cantor's enumeration (see Remark 2.6)

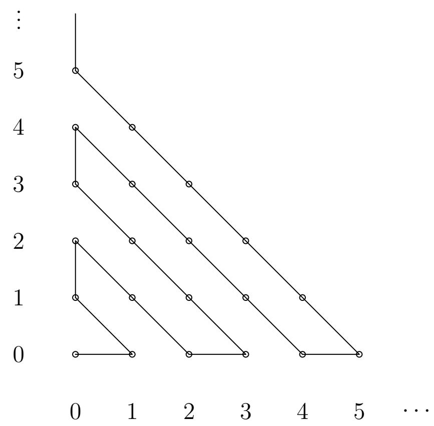


Figure 1. A d2-ℓ-ordering of $\mathbb{N} \times \mathbb{N}$.

that is, in terms of neighbourhoods (see Section 1)

$$(\texttt{Down}, \{\texttt{N}, \texttt{E}, \texttt{NE}\}) \to (\texttt{E}, \texttt{Up}), \quad (\texttt{Down}, \{\texttt{N}, \texttt{E}, \texttt{W}, \texttt{NW}, \texttt{NE}\}) \to (\texttt{E}, \texttt{Up}).$$

In Figure 1, these two transitions define respectively the edges $(0, 0) \to (1, 0)$ and $(2n, 0) \to (2n+1, 0)$ belonging to the path that orders $\mathbb{N} \times \mathbb{N}$. The edges directed to NW and SE, defined by the first and the third clauses, do not change the level $i + j$ of a pair (i, j) because they "go" from it to $(i-1, j+1)$ or $(i+1, j-1)$. The state defined from the arithmetic parity of the level is not changed.

The last clause yields the edge $(0, 3) \to (0, 4)$ derived from the transition expressing the following:

$$(\texttt{Up}, \text{"on the western border"}) \to (\text{"move to the north"}, \texttt{Down}).$$

Definition 2.3 (a d2-ℓ-ordering of $X \times Y \subseteq \mathbb{N} \times \mathbb{N}$). We modify the algorithm of Definition 2.2 so that it defines a d2-ℓ-ordering of $X \times Y$ when X and/or Y is finite. The order is defined from types $\tau(i, j)$ as above. The corresponding Hamiltonian path in $G_2(Z^{(3,6)})$, where $Z^{(3,6)} := [0, 3] \times [0, 6]$, is illustrated in Figure 2. If X is finite, its maximum is denoted by $\max(X)$.

The information about neighbourhood also uses the Boolean tests $i = \max_X$ and $j = \max_Y$ that are always false if X or, respectively Y, is infinite.

In the first clause, the pair $\texttt{next}(i, j)$ is undefined because (i, j) is the last element, and its "value" is the message "none" indicating the end of the enumeration. The clauses are

$$\begin{aligned}
\texttt{next}(i, j) = {} & \text{IF } i = \max_X \ \wedge\ j = \max_Y \ \text{ THEN } \texttt{none} \ \text{ ELSE} \\
& \text{IF } i + j \text{ is even } \ \wedge\ j \neq 0 \ \wedge\ i \neq \max_X \ \text{ THEN } (i+1, j-1) \ \text{ ELSE} \\
& \text{IF } i + j \text{ is even } \ \wedge\ i = \max_X \ \text{ THEN } (i, j+1) \ \text{ ELSE} \\
& \text{IF } i + j \text{ is even } \ \wedge\ j = 0 \ \wedge\ i \neq \max_X \ \text{ THEN } (i+1, j) \ \text{ ELSE} \\
& \text{IF } i + j \text{ is odd } \ \wedge\ i \neq 0 \ \wedge\ j \neq \max_Y \ \text{ THEN } (i-1, j+1) \ \text{ ELSE} \\
& \text{IF } i + j \text{ is odd } \ \wedge\ j = \max_Y \ \text{ THEN } (i+1, j) \ \text{ ELSE} \\
& \text{IF } i + j \text{ is odd } \ \wedge\ i = 0 \ \wedge\ j \neq \max_Y \ \text{ THEN } (i, j+1) \ \text{ END.}
\end{aligned}$$

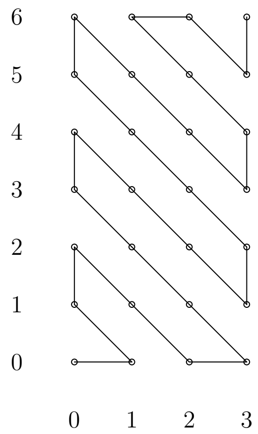


Figure 2. The diagonal enumeration of $Z^{(3,6)}$.

There are nine clauses. The conditions relative to i and j can be replaced by conditions on neighbourhoods, as in Definition 2.2, where "on the southern border" is expressed by "the neighbourhood is {N, E, NE} or {N, E, W, NW, NE}".

Here is an alternative formalization by the automaton, which we will use again. There are nine possible types of position of a vertex (i, j) on the grid if X and Y are not singletons:[3]

- origin (numbered 0),
- on the southern border (1),
- on the western border (2),
- in the middle (3),
- at the southeastern corner (4),
- at the northwestern corner (5),
- on the northern border (6),
- on the eastern border (7) and
- at the northeastern corner (8).

See Figure 3. Each type can be determined by a combination of Boolean conditions such as "$j = 0$" and "$i \neq \max_X$" or of possible neighbourhoods. Position 3 is characterised by the maximum neighbourhood, that is, $\{\mathtt{N}, \mathtt{E}, \mathtt{W}, \mathtt{S}, \mathtt{NW}, \mathtt{NE}, \mathtt{SW}, \mathtt{SE}\} = \mathcal{D}_2$.

In Table 1, and for readability, we define an automaton $\mathcal{B}$ by using the digits 0 to 8 to indicate the types of positions instead of combinations of Boolean conditions or neighbourhoods. Position "2, 3, 5, 6" means of type 2 or 3 or 5 or 6. The initial state is Down. The final state is End. Figure 4 shows the same information as Table 1.

The complete description of $\mathcal{B}$ by a table should include the special cases where X and/or Y is singleton, but we want to keep the table readable.

[3] If X is singleton and Y is not, then 0 and 4 coincide, and so do 2 and 7 and 5 and 8.

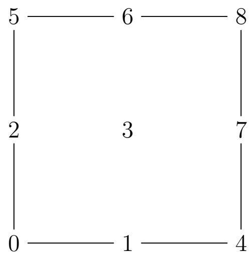


Figure 3. The different types of borders used by $\mathcal{B}$.

Proposition 2.4. *The automaton $\mathcal{B}$ defines a d2-ℓ-ordering.*

Remarks 2.5. (1) $\mathcal{B}$ is actually a $\mathcal{D}_{2\ell}$-automaton, where we define $\mathcal{D}_{2\ell} := \{N, E, W, S, NW, SE\}$. Position 3 is correctly determined by the neighbourhood $\{$N, E, W, S, NW, SE$\}$ and no transitions use the directions NW and SE.

(2) The description in Table 1 is appropriate if X and Y are not singletons. However, the definition of `next` works well in all cases. If X or Y is singleton, there is a unique d2-ℓ-ordering. Otherwise, there

state	position	action	next state
Down	0,1 2,3,5,6 4,7	E SE N	Up Down Up
Up	1,3,4,7 2 5,6	NW N E	Up Down Up
Up or Down	8		End

Table 1. Automaton $\mathcal{B}$.

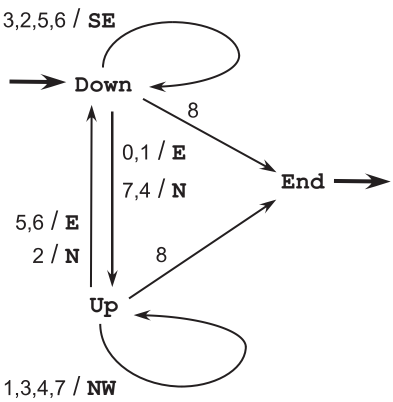


Figure 4. The automaton $\mathcal{B}$ of Proposition 2.4.

are exactly two, one starting by $(0,0) \to (1,0)$ (as in Figures 1 and 2) and the other by $(0,0) \to (0,1)$. The latter one is obtained by taking Up as the initial state and adding to Table 1 the transition from the origin (defined by $i = 0 \wedge j = 0$) to the state Down with a move north. The obtained path starts with /00/01,10/20,11,02/03, ... We will denote by $\mathcal{B}^{\#}$ this modified automaton. In automaton $\mathcal{B}$, the state Up corresponds to the odd levels and Down to the even ones. For $\mathcal{B}^{\#}$, Down corresponds to the odd levels and Up to the even ones.

(3) The automaton $\mathcal{B}$ (as defined by next, see Definition 2.3) also works in the special case where $Y = \{0\}$. All positions satisfy $j = 0 \wedge j = \max_Y$. The transitions used are defined by

$$\text{IF } i + j \text{ is even } \wedge\ j = 0\ \wedge\ i \neq \max_X \ \ \text{THEN } (i + 1, j)$$

and

$$\text{IF } i + j \text{ is odd } \wedge\ i \neq \max_X\ \wedge\ j = \max_Y \ \ \text{THEN } (i + 1, j).$$

Its works also in the special case where $X = \{0\}$. All positions satisfy $i = 0 \wedge i = \max_X$. The transitions used are defined by

$$\text{IF } i + j \text{ is even } \wedge\ i = \max_X \ \ \text{THEN } (i, j + 1)$$

and

$$\text{IF } i + j \text{ is odd } \wedge\ i = 0\ \wedge\ j \neq \max_Y \ \ \text{THEN } (i, j + 1).$$ □

By using $\mathcal{D}_{2\ell}$-automata, we have formalised the construction of Hamiltonian paths in the graphs $G_2(X \times Y)$, which represent d2-ℓ-orderings of $X \times Y$. We will use $\mathcal{D}_1$-automata similarly in graphs $G_1(X \times Y)$ so as to define d1-orderings. In Section 4, we will define automata that define Hamiltonian paths in $G_2(X_1 \times \cdots \times X_n)$.

In a concrete implementation, we use an oracle (a program) that determines the membership in Z of any pair $b = (i, j)$ and the set $\mathcal{D}_G(b)$, especially when Z is defined by affine conditions, such as $i \leq 3j + 5 \wedge j \leq -10i + 30$. See Section 4C.

Remark 2.6. Cantor's bijections $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ are defined by the polynomials

$$P(x, y) = x + (x + y)(x + y + 1)/2 \quad \text{and} \quad P(x, y) = y + (x + y)(x + y + 1)/2.$$

Fueter and Polya proved that no other quadratic polynomial defines such a bijection. References are in [Wikipedia 2015]. The enumeration by the second polynomial starts with 00/10/01/20/..., and hence does not satisfy the condition that the first components in consecutive pairs must differ by at most 1. The corresponding sequence is not a path in $G_2(\mathbb{N} \times \mathbb{N})$.

3. D1-orderings on sets $X_1 \times X_2 \times \cdots \times X_n$

Proposition 3.1. *From a d1-ordering of a finite set $Y \subseteq X_1 \times X_2 \times \cdots \times X_n$, one can define a d1-ordering of $Z := \mathbb{N} \times Y$.*

Proof. D1-orderings are Hamiltonian paths in the graphs $G_1(Y)$ and $G_1(Z)$. Let $P_{a,b}$ from $a = (0, 0, \ldots, 0)$ to some vertex b be a Hamiltonian path in $G_1(Y)$. The opposite path is $P_{b,a}$ from b to a. For each $i \in \mathbb{N}$, let $i \odot P_{a,b}$ be the path $(i, a) \to (i, c_1) \to (i, c_2) \to \cdots \to (i, b)$, where $P_{a,b}$ is $a \to c_1 \to c_2 \to \cdots \to b$. Then, one gets in Z the infinite Hamiltonian path $0 \odot P_{a,b} \to 1 \odot P_{b,a} \to 2 \odot P_{a,b} \to 3 \odot P_{b,a} \to \cdots$ starting from $(0, \ldots, 0) = (0, a) \in Z$. (The arrow $\to$ represents the concatenation of paths). □

Remark 3.2. This construction is related to that of Gray codes; see [Wikipedia 2011]. The 3-ary Gray code with 3 digits is the sequence of 3-tuples in $\{0, 1, 2\} \times (\{0, 1, 2\} \times \{0, 1, 2\})$ that reads

$$000, 001, 002, 012, 011, 010, 020, 021, 022,$$
$$122, 121, 120, 110, 111, 112, 102, 101, 100,$$
$$200, 201, 202, 212, 211, 210, 220, 221, 222.$$

It is thus of the form $0 \odot P \to 1 \odot P' \to 2 \odot P$, where P is

$$00 \to 01 \to 02 \to 12 \to 11 \to 10 \to 20 \to 21 \to 22,$$

and P' is the opposite path. □

Proposition 3.1 does not apply to $Z := \mathbb{N} \times \mathbb{N}$, and an ordering "row by row" is obviously not adequate as its order type will be $\omega + \omega + \cdots = \omega \cdot \omega \neq \omega$. This is a motivation for using the diagonal d2-ℓ-ordering of Definition 2.2. However, d1-orderings can also be defined.

Proposition 3.3. (1) *There is a* d1*-ordering on* $\mathbb{N} \times \mathbb{N}$ *definable by an infinite* $\mathcal{D}_1$*-automaton.*

(2) *Each set* $Z = X_1 \times X_2 \times \cdots \times X_p$*, where each* X_i *is finite or infinite, has a* d1*-ordering.*

Proof. (1) See Figures 5 and 6. Theorem 3.5 will establish a more general result for $G_1(X \times Y)$ where X and/or Y may be finite.

(2) We first consider $\mathbb{N}^p$. We use an induction on p. For $p = 2$, the result holds by Assertion (1). Assume we have a d1-ordering "$\leq_p$" of $\mathbb{N}^p$. Since $(\mathbb{N}^p, \leq_p)$ is isomorphic to $(\mathbb{N}, \leq)$, we have by (1) an ordering of $\mathbb{N}^{p+1} = \mathbb{N} \times (\mathbb{N}^p)$. In this order, a step from a vertex to the next one either modifies the first component (in $\mathbb{N}$) or the second one (in $\mathbb{N}^p$). In the latter case, only one component of $\mathbb{N}^p$ is modified, as $\leq_p$ is a d1-ordering. In both cases, this step modifies a single component of $\mathbb{N}^{p+1}$. Hence, we have a d1-ordering.

If Z is finite, Proposition 3.1 gives the answer. Otherwise, one can permute the components and write $Z = \mathbb{N} \times \cdots \times \mathbb{N} \times X_q \times \cdots \times X_p$, with $X_q, \ldots, X_p$ finite. Thus Z is isomorphic to $\mathbb{N}^q \times (X_q \times \cdots \times X_p)$ and hence to $\mathbb{N} \times Y$ with Y finite, and Proposition 3.1 gives the answer.

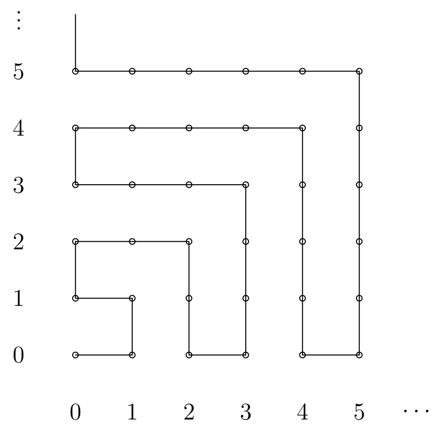


Figure 5. A d1-ordering of $\mathbb{N} \times \mathbb{N}$.

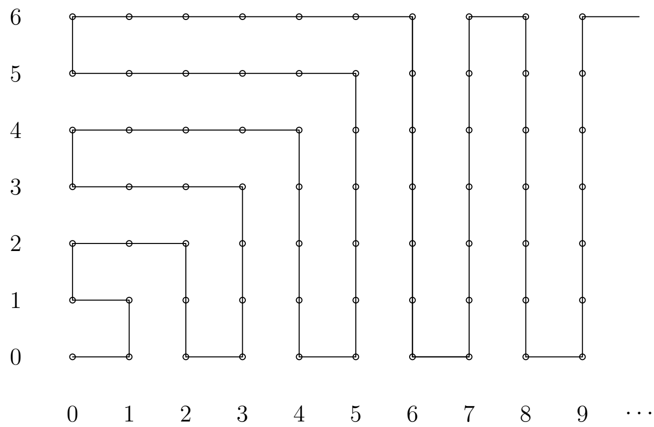


Figure 6. A d1-ordering for [0, 2n]×$\mathbb{N}$.

The computation of the vertex following any z in Z is computable as all definitions and proofs are effective. Hence, there exists a $\mathcal{D}_1$-automaton, with infinitely many states.[4] □

We now examine whether automata can define d1-orderings. Figure 6 shows a d1-ordering of $X \times Y$ where Y is finite of odd cardinality and X is finite or infinite, that is, defined by an infinite $\mathcal{D}_1$-automaton. Whether X and/or Y is finite need not be known at the beginning, but is determined at some point of the computation. This automaton is easy to define with states including counters.[5]

More generally, there are $\mathcal{D}_1$-automata that construct d1-orderings of $X \times Y$ by using some information about X and/or Y. This information can be:

(1) X is finite.

(2) Y is finite.

(3) X is either infinite or finite of odd cardinality.

(4) Y is either infinite or finite of odd cardinality.

(5) X is either infinite or finite of even cardinality.

(6) Y is either infinite or finite of even cardinality.

The automata for Cases (1) and (2) are finite. In Cases (1), (3), (5), Y may be of any type. In the others, X may be of any type. In Cases (3) and (5), the automaton need not know whether X is infinite or not, and similarly for Y in Cases (4) and (6). Without such information, no deterministic automaton can work correctly, as we prove now.

Theorem 3.4. *There is no (finite or infinite) $\mathcal{D}_1$-automaton that constructs a d1-ordering of $X \times Y$ for arbitrary (linearly ordered) sets X and Y.*

Proof. To get a contradiction, we assume the existence of a $\mathcal{D}_1$-automaton $\mathcal{A} = (Q, \mathcal{T}, q_{\text{init}})$ that finds a Hamiltonian path starting at (0,0) in $G_1(X \times Y)$ for any linearly ordered sets X, Y, either $\mathbb{N}$ or $[0, p]$.

[4] As in fly-automata, see [Courcelle and Durand 2016], we allow countable sets of states but transitions must be computable.

[5] For defining the path of Figure 5, one can use a finite $\mathcal{D}_1$-automaton that tests whether the current vertex is on the southwest-northeast diagonal.

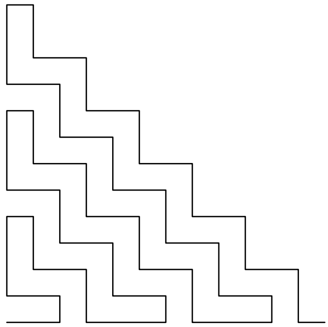


Figure 7. A d1-ordering of $\mathbb{N} \times \mathbb{N}$ that is adaptable to $X \times Y$, where X and/or Y is finite.

This automaton uses only the directions N, E, S, W. The sets X and Y are finite or infinite, which the automaton "does not know": this means that $\mathcal{A}$ works in all cases. The set of states may be infinite, but determinism will yield a contradiction.

The neighbourhood $\mathcal{D}_G(x)$ describes the following possible positions of a vertex x, numbered 0, 1, 2, 3 in Figure 3:

x is the origin: $\mathcal{D}_G(x) = \{\mathtt{N}, \mathtt{E}\}$,
or on the southern border, and not the origin: $\mathcal{D}_G(x) = \{\mathtt{N}, \mathtt{E}, \mathtt{W}\}$,
or on the western border, and not the origin: $\mathcal{D}_G(x) = \{\mathtt{N}, \mathtt{E}, \mathtt{S}\}$,
or in the middle: $\mathcal{D}_G(x) = \{\mathtt{N}, \mathtt{E}, \mathtt{S}, \mathtt{W}\}$.

We first run the automaton on $\mathbb{N} \times \mathbb{N}$. Let P be a Hamiltonian path defined by $\mathcal{A}$ in $G_1(\mathbb{N} \times \mathbb{N})$. It has a subpath $P[a, b]$ from a to b, for some a on the southern border and some b on the western border, such that all intermediate vertices x have neighbourhood $\mathcal{D}_G(x) = \{\mathtt{N}, \mathtt{E}, \mathtt{S}, \mathtt{W}\}$. The initial part $P[(0, 0), a]$ of P is inside the finite portion R of $G_1(\mathbb{N} \times \mathbb{N})$ (drawn on the plane) determined by $P[a, b]$ and the western and southern borders by an obvious planarity argument. We let R contain the vertices of $P[a, b]$ and the initial parts of the borders, from (0, 0) to $(a, 0)$ and from (0, 0) to $(0, b)$.

Let m be the maximal integer such that (m, j) belongs to $P[a, b]$ for some j. Let $c := (m + 1, j)$ for such a j. The vertex c is not in R.

We now consider $\mathcal{A}$ running in $G_1([0, m+1] \times \mathbb{N})$. It follows the path $P[(0, 0), b]$, as it does not distinguish $G_1([0, m + 1] \times \mathbb{N})$ from $G_1(\mathbb{N} \times \mathbb{N})$ when traversing R. The path continues in $G_1([0, m + 1] \times \mathbb{N})$ from b to c outside of R. But after c it must continue southwards, and cannot reach $(m + 1, p)$ for large values of p. Hence we obtained the desired contradiction. □

We now enrich our automata by letting them foresee whether, from a vertex x, they can make two moves east and/or two moves north. That is, we enlarge neighbourhoods. We extend accordingly the definitions of Section 1 and define the set of checkable directions around a vertex as $\mathcal{E} := \{\mathtt{N}, \mathtt{NN}, \mathtt{E}, \mathtt{EE}, \mathtt{S}, \mathtt{W}\}$. If in $\mathcal{D}_G(x)$ we have EE (two consecutive east moves are possible), we must also have E. If E is in $\mathcal{D}_G(x)$ but EE is not, this means that x is at distance 1 from the eastern border. Similar facts hold for NN and N. The corresponding path is in Figure 7.

We first encourage the reader to contemplate Figures 8, 9 and 10. The construction of the path in Figure 8 corresponding to the case $|X| = 7$, $|Y| = 5$ extends to $\mathbb{N} \times \mathbb{N}$; see Step 2 of the proof. Difficulties arise in the cases where X and/or Y have even cardinality. One typical case is shown in Figure 9

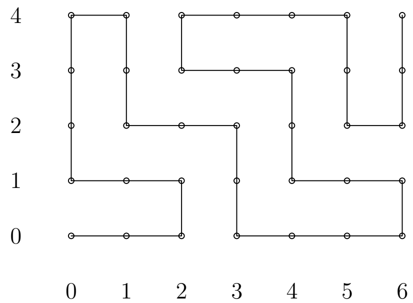


Figure 8. The basic case of $X \times Y$ with sets X, Y of odd cardinalities.

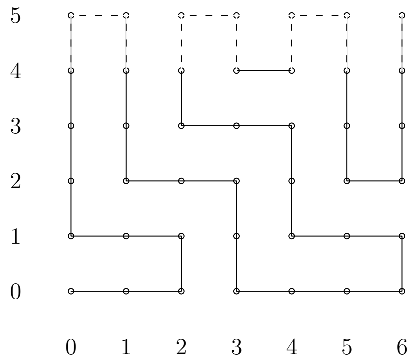


Figure 9. Extended the previous order to accommodate set Y of even cardinality.

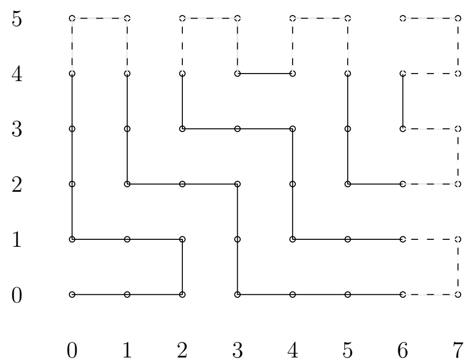


Figure 10. Extended the previous order to accommodate sets X, Y of even cardinality.

corresponding to the case $|X| = 7,\ |Y| = 6$. Another one is for $|X| = 8,\ |Y| = 6$ (Figure 10). Dotted lines indicate modifications from Figure 8.

Theorem 3.5. *There exists a finite $\mathcal{E}$-automaton that constructs a $d1$-ordering of $X \times Y$ for arbitrary* (*linearly ordered*) *sets X and Y.*

Proof. Step 1: The intended automaton will first handle the particular cases where X and/or Y have cardinality 1 or 2. This can be checked from $\mathcal{D}_G((0, 0))$ as $(0, 0)$ is the starting vertex. Hence, Y has

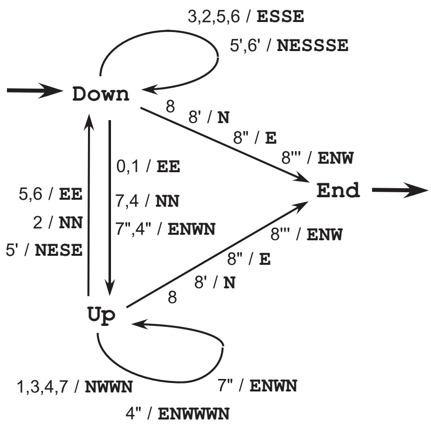


Figure 11. The $\mathcal{E}$-automaton of Theorem 3.5.

state	position	action	new state
Down	0, 1	E;E	Up
	2, 3, 5, 6	E;S;S;E (instead of SE;SE)	Down
	4, 7	N;N	Up
Up	1, 3, 4, 7	N;W;W;N (instead of NW;NW)	Up
	2	N;N	Down
	5, 6	E;E	Up
Up or Down	8		End

Table 2. The automaton $\mathcal{C}$.

cardinality 1 if and only if E is not in $\mathcal{D}_G((0,0))$ and cardinality 2 if and only if EE is not in $\mathcal{D}_G((0,0))$ but E is. We omit details relative to these special cases.

Step 2: We now build an automaton $\mathcal{C}$ intended to work for all sets X, Y that are either infinite or of finite odd cardinality at least 3. It is based on the $\mathcal{D}_{2\ell}$-automaton $\mathcal{B}$ of Proposition 2.4.

From $\mathcal{B}$, we first define a $\mathcal{D}_{2\ell}$-automaton $\mathcal{B}'$ by duplicating actions: N becomes N;N, that is two moves north, SE becomes SE;SE, etc. Because of the assumptions on the cardinalities of X and Y, from each vertex reached by a path with an even number of edges that is defined by $\mathcal{B}'$, if one can make one step east, one can make another one to the east, and similarly for north, southeast and northwest. This automaton need not check the "double" directions EE and NN.

A $\mathcal{D}_1$-automaton $\mathcal{C}$ is defined by Table 2, is obtained from $\mathcal{B}'$ by replacing respectively the "double" moves SE;SE by E;S;S;E and NW;NW by N;W; W;N. This replacement is made explicit in Table 2. The same numbering of types of positions on borders, at corners and in the middle is used, as for describing $\mathcal{B}$ and $\mathcal{B}'$; see Figures 3 and 12. As for $\mathcal{B}'$, one need not check the "double" directions EE and NN.

Claim: The $\mathcal{D}_1$-automaton $\mathcal{C}$ defines a d1-ordering of $X \times Y$ for any sets X, Y as stated.

Proof of claim. First we prove that $\mathcal{C}$ defines a path, i.e., a walk that does not visit the same vertex twice.

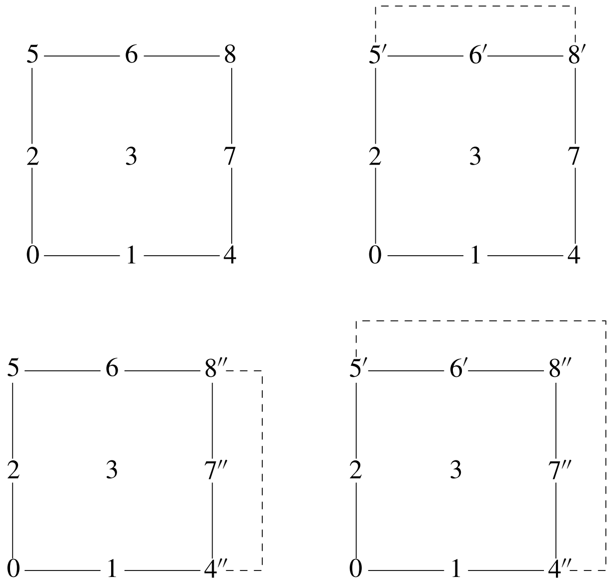


Figure 12. Numbering of types of positions relative to the borders. In particular, close to the northern and eastern borders.

A 2×2 *square* in the grid $G_1(X, Y)$ is a subgraph induced by $[2p, 2p+2] \times [2q, 2q+2]$ for $p, q \geq 0$. Each of them can be coloured black or white, so that two adjacent squares (adjacent by a border, not just a corner) are of different colour. Let $[0, 2] \times [0, 2]$ be white. By $\mathcal{B}'$, it is traversed by the "double" moves NW; NW. Those of the form SE;SE traverse black 2×2 squares.

When defining $\mathcal{C}$, we replace SE;SE by E;S;S;E, so that we go through two more vertices, say x and y, in the middle of the top and bottom borders of that 2×2 square. In the surrounding 2×2 white squares, the replacements are of NW; NW by N;W;W;N, and these replacements involve neither x nor y. A similar observation holds at the borders.

Hence, we have a path. It is easy to check, by a similar argument based on this colouring of the 2×2 squares that it goes through all vertices of $G_1(X, Y)$.

If X and Y are finite, it terminates at the corner numbered 8, i.e., at $(\max(X), \max(Y))$; an example is in Figure 8. □

Step 3: We must handle the three cases where $|X|$ and/or $|Y|$ is even. See Figure 4 showing the four types of borders and corners for finite sets X and Y.

Case 1: $|X|$ is odd or infinite, $|Y|$ is even. The vertices on the row just below the topmost one are in positions of types $5'$, $6'$ and $8'$ (see Figure 12) characterised by the following conditions relative to a vertex x:

$$\begin{aligned} 5':&\ \texttt{N,EE,S} \in \mathcal{D}_G(x), \quad \texttt{W,NN} \notin \mathcal{D}_G(x),\\ 6':&\ \texttt{N,EE,W,S} \in \mathcal{D}_G(x), \quad \texttt{NN} \notin \mathcal{D}_G(x),\\ 8':&\ \texttt{N,EE,W,S} \in \mathcal{D}_G(x), \quad \texttt{E,NN} \notin \mathcal{D}_G(x). \end{aligned}$$

If X is infinite, positions of types 4, 7, $8'$ do not occur.

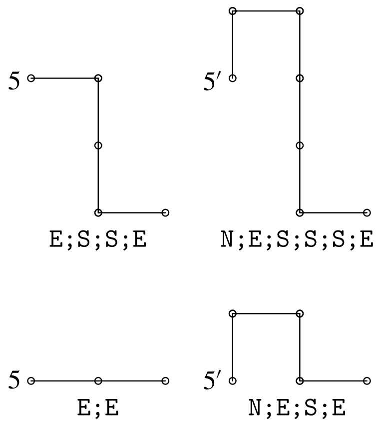


Figure 13. Some detours for vertices close to the northern border.

In order to reach the vertices on the top row, we make small detours defined as follows. When the current state is `Down`:

- Action `E;S;S;E` from vertices of type 5 or 6 is replaced by `N;E;S;S;S;E`, from vertices of type $5'$ or $6'$ (see the top part of Figure 13).
- From vertex $8'$, action is `N`, terminating the path.

When the current state is `Up`:

- Action `E;E` from vertices of type 5 is replaced by `N;E;S;E` from vertices of type $5'$.
- Action `E;E` from vertices of type 6 is replaced by `N;E;S;S;S;E` from vertices of type $6'$.
- From vertex $8'$, action is `N`, terminating the path.

Case 2: $|X|$ is even, $|Y|$ is odd or infinite. Vertices on the column just to the right of the last one are of types $4''$, $7''$ and $8''$, characterised by the following conditions:

$$\begin{aligned} 4'':\ & \mathtt{N,E,W} \in \mathcal{D}_G(x), && \mathtt{S,EE} \notin \mathcal{D}_G(x),\\ 7'':\ & \mathtt{NN,E,S,W} \in \mathcal{D}_G(x), && \mathtt{EE} \notin \mathcal{D}_G(x),\\ 8'':\ & \mathtt{E,S,W} \in \mathcal{D}_G(x), && \mathtt{N,EE} \notin \mathcal{D}_G(x). \end{aligned}$$

If Y is infinite, positions of types 5, 6, $8''$ do not occur.
When the current state is Down:

- Action `N;N` from vertices of type 4 is replaced by: `E;N;W;W;W;N` from vertices of type $4''$.
- Action `N;W;W;N` from vertices of type 4 is replaced by: `E;N;W;W;W;N` from vertices of type $4''$. (See Figure 14).
- Action `N;N` from vertices of type 7 is replaced by: `E;N;W;N`, from vertices of type $7''$.
- From vertex $8''$, action is `E`, terminating the path.

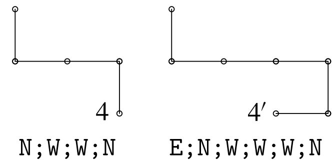


Figure 14. The detour at southeastern corner.

state	position	action	next state
Down	0, 1	E;E	Up
	2, 3, 5, 6	E;S;S;E	Down
	4, 7	N;N	Up
	5′, 6′	N;E;S;S;S;E	Down
	4″, 7″	E;N;W;N	Up
	8′	N	End
	8″	E	
Up	1, 3, 4, 7	N;W;W;N	Up
	2	N;N	Down
	5, 6	E;E	Down
	4″	E;N;W;W;W;N	Up
	5′	N;E;S;E	Down
	6′	N;E;S;S;S;E	Down
	7″	E;N;W;N	Up
Up or Down	8		End
	8′	N	End
	8″	E	End
	8‴	E;N;W	End

Table 3. The $\mathcal{E}$-automaton of Theorem 3.5.

Case 3: $|X|$ and $|Y|$ are even. This case combines Cases 1 and 2. The relevant types of positions replacing 4, 5, 6, 7, 8 from the basic case are $4''$, $5'$, $6'$, $7''$ characterised as above and

$$8'': \quad \mathtt{N},\mathtt{E},\mathtt{S},\mathtt{W} \in \mathcal{D}_G(x), \qquad \mathtt{NN},\mathtt{EE} \notin \mathcal{D}_G(x),$$

From a vertex of type $8''$, the action is E;N;W, terminating the path.

These definitions are collected in Table 3. □

Remark 3.6. It is clear that $\mathbb{Z}$ has no d1-ordering, and that, curiously, $\mathbb{Z} \times \mathbb{Z}$ has one, that is a kind of spiral around the origin. So has $\mathbb{N} \times \mathbb{Z}$ and thus $\mathbb{Z}^p$ for $p > 2$.

4. Diagonal orderings of subsets of $\mathbb{N} \times \mathbb{N}$

Figure 15 shows that the automaton $\mathcal{B}$ of Section 2 can order proper subsets of $\mathbb{N} \times \mathbb{N}$ that are not Cartesian products. In this section, we develop this observation.

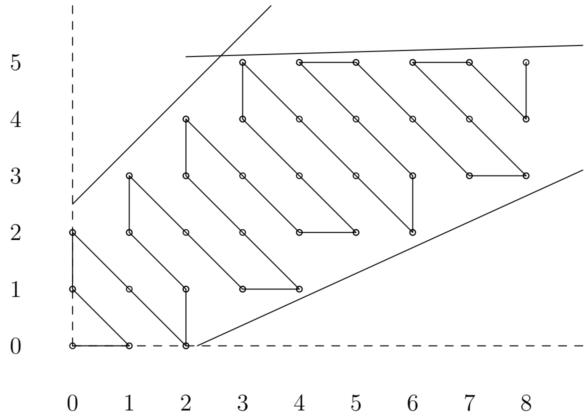


Figure 15. A proper subset of a Cartesian product ordered by automaton $\mathcal{B}$ of Section 2.

4A. ***D2-ℓ-orderings of subsets of*** $\mathbb{N}\times\mathbb{N}$**.** We ask the following questions.

Question 4.1. Which subsets Z of $\mathbb{N}\times\mathbb{N}$ have a d2-ℓ-ordering?

We will consider $\mathcal{D}_2$-automata, more powerful than $\mathcal{D}_{2\ell}$-automata as they can move northeast in addition to northwest, north, east, southwest and south. As we want them to define d2-ℓ-*paths*, i.e., Hamiltonian paths corresponding to d2-ℓ-orderings, they will make no moves southwest.

Question 4.2. When is a d2-ℓ-ordering definable by a finite or infinite $\mathcal{D}_2$-automaton?

If Z is finite and d2-ℓ-orderable, then such an ordering is definable by a finite $\mathcal{D}_2$-automaton with $|Z|$ states. Hence, this question is only interesting for one infinite set Z or for a class of finite and/or infinite sets.

Definition 4.3 (conditions on a set $Z\subseteq\mathbb{N}\times\mathbb{N}$). (a) We denote by Z_k the level k of Z. For each nonempty level Z_k, $\min(Z_k)$ (resp. $\max(Z_k)$) is its unique vertex of minimal (resp. maximal) second coordinate.

(b) We define for $Z\subseteq\mathbb{N}\times\mathbb{N}$ containing (0, 0) the following conditions:

- (C1) The graph $G_2(Z)$ is connected.
- (C2) Each nonempty level Z_k is connected in $G_2(Z)$, and hence, induces a north-west-southeast diagonal path.
- (C3) Each nonempty level can be labelled by `Up` or `Down`, so that if Z_k and $Z_{k'}$ are two consecutive nonempty levels with $k<k'$, then $\texttt{Last}(Z_k)$ is adjacent to $\texttt{First}(Z_{k'})$ in $G_2(Z)$, where[6]
  - if Z_k is labelled by `Down`, then $\texttt{First}(Z_k) := \max(Z_k)$ and $\texttt{Last}(Z_k) := \min(Z_k)$ and
  - if Z_k is labelled by `Up`, then $\texttt{First}(Z_k) := \min(Z_k)$ and $\texttt{Last}(Z_k) := \max(Z_k)$.

Condition (C1) implies that, if Z_k and $Z_{k'}$ are as in (C3), then k' is $k+1$ or $k+2$. If $G_1(Z)$ is connected, then so is $G_2(Z)$ and hence condition (C1) holds; furthermore, if a level is not empty, all previous levels are not either; that is, we have $k'=k+1$ for k, k' as in (C3).

Example 4.4. The set $Z := \{(0, 2i), (1, 2i+1) \mid i\geq 0\}$ satisfies conditions (C1), (C2) and (C3), with all levels labelled by `Up`. It has no level of odd height.

[6]These definitions are the same for $Z_{k'}$.

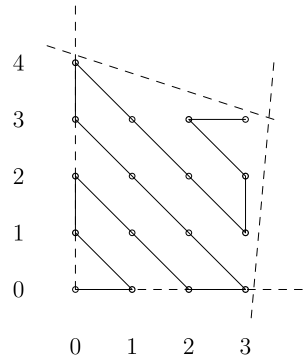


Figure 16. The set W of Example 4.6(1) used in Proposition 4.7.

Proposition 4.5. *A subset of* $\mathbb{N} \times \mathbb{N}$ *containing* $(0, 0)$ *has a d2-*ℓ*-ordering if and only if it satisfies conditions* (C1), (C2) *and* (C3).

Proof. Let Z be a subset of $\mathbb{N} \times \mathbb{N}$ that has a d2-ℓ-ordering with associated path P in $G_2(Z)$. This set satisfies conditions (C1) and (C2). We label by `Up` a nonempty level that is traversed upwards by P, that is from southeast to northwest, and by `Down` in the other case. We label arbitrarily a singleton level $Z_k = \{x\}$ and in either case we have $\texttt{Last}(Z_k) = \texttt{First}(Z_k) = x$. Then condition (C3) holds with this labelling.

Conversely, let Z satisfy (C1) and (C2). From any labelling satisfying (C3), we obtain a path P from $(0, 0)$ with appropriate transitions between levels, which describes a d2-ℓ-ordering of Z. □

Example 4.6. (1) Figure 16 shows an example of a set $W \subseteq \mathbb{N} \times \mathbb{N}$ that satisfies conditions (C1)–(C3). It has a d2-ℓ-path starting with $(0, 0) \to (1, 0)$, shown in this figure. An initial step $(0, 0) \to (0, 1)$ can be extended into a d2-ℓ-path until $(0, 4)$ but not after because $\max(W_4)$ and $\max(W_5)$ are not adjacent. Anticipating the sequel, we observe that W is defined by the conditions $i \leq 3$ and $j \leq -i/3 + 4$.

(2) The related set X of Figure 17 has no d2-ℓ-ordering for a similar reason. It is defined by the conditions $j \leq -i/2 + 4$ and $j \leq -2i + 8$. It satisfies (C1) and (C2).

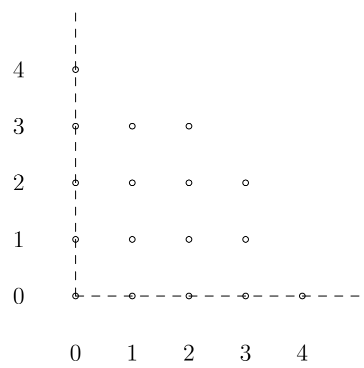


Figure 17. Set X of Example 4.6(2).

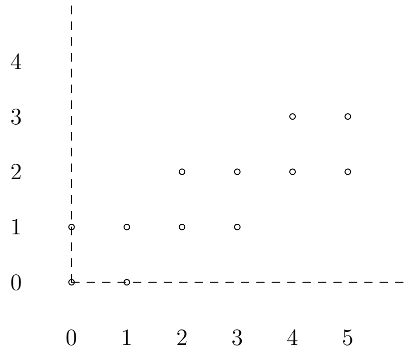


Figure 18. Set Y of Example 4.6(3).

(3) The finite set Y shown in Figure 18 has eight d2-ℓ-orderings. Three of them are:

$$00/10,\ 01/11/21/31,\ 22/32/42/52,\ 43/53, \quad \text{defined by } \mathcal{B},$$
$$00/01,\ 10/11/21/22,\ 31/32/42/43,\ 52/53, \quad \text{defined by } \mathcal{B}^{\#}, \text{ and}$$
$$00/10,\ 01/11/21/22,\ 31/32/42/52,\ 43/53, \quad \text{defined neither by } \mathcal{B} \text{ nor by } \mathcal{B}^{\#}.$$

Its infinite extension Y' defined by $i/2 - 1/2 \leq j \leq i/2 + 1$ has infinitely many d2-ℓ-orderings. □

We continue the study of sets $Z \subseteq \mathbb{N} \times \mathbb{N}$. If $G_1(Z)$ is connected and Z has a d2-ℓ-ordering that is definable by a $\mathcal{D}_2$-automaton, then this ordering is definable by a $\mathcal{D}_{2\ell}$-automaton, actually the same, because no move northeast can be used.

We recall that automata are deterministic and must have computable transitions; see Definition 1.2(a).

Proposition 4.7. (1) *There exists an infinite set of finite sets $Z \subseteq \mathbb{N} \times \mathbb{N}$ that have unique d2-ℓ-orderings, but these orderings are not definable by any finite or infinite $\mathcal{D}_2$-automaton.*

(2) *There exists an infinite set $Z \subseteq \mathbb{N} \times \mathbb{N}$ that has a unique a d2-ℓ-ordering that is not definable by any finite or infinite $\mathcal{D}_2$-automaton.*

Proof. We let $W \subseteq [0, 3] \times [0, 4]$, shown in Figure 16. It has a unique d2-ℓ-path (defined by $\mathcal{B}$) from $s := (0, 0)$ to $t := (3, 3)$. Let $\overline{W} \subseteq [0, 4] \times [0, 3]$ be obtained from W a symmetry with respect to the southwest-northeast diagonal. It has a unique d2-ℓ-path (defined by $\mathcal{B}^{\#}$) also from $s := (0, 0)$ to $t := (3, 3)$.

(1) Let w_n be the word 0^n1. We define $X^{(n)} \subseteq \mathbb{N} \times \mathbb{N}$ by concatenating copies U_i of W or $\overline{W}$ such that U_i is a copy of W if $w_i = 0$ and of $\overline{W}$ otherwise. Two consecutive copies U_i and U_{i+1} are linked by a horizontal edge between t_i and s_{i+1}. See Figure 19 for $X^{(2)}$. Each set $X^{(n)}$ has a unique d2-ℓ-ordering. Assume that a $\mathcal{D}_2$-automaton (equivalently, a $\mathcal{D}_{2\ell}$-automaton) can d2-ℓ-order all the sets $X^{(n)}$. When it reaches a vertex s_i, it cannot "know" whether the next move must be north or east because it cannot know whether U_i is of type W or $\overline{W}$. Infinitely many states would not help.

(2) We now construct X similarly from an infinite word w in $\{0, 1\}^{\omega}$. It has a unique d2-ℓ-ordering. If w is not ultimately periodic, this ordering cannot be defined by a finite $\mathcal{D}_2$-automaton, by an argument similar to that used in (1). It is definable by an infinite one (whose transitions must be computable, see Definition 1.2(a)) if there exists a computable function $f_w\colon \{0, 1\}^* \to \{0, 1\}$ such that $f_w(u)$ defines the

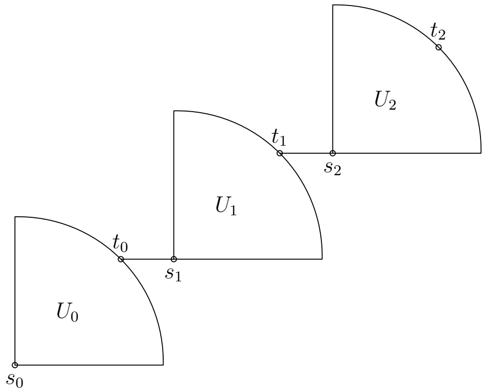


Figure 19. Set $X^{(2)}$ of Proposition 4.7.

letter 0 or 1 that follows u in w in the case where u is a prefix of w (otherwise, it yields 0). As there are uncountably many infinite words and countably many computable functions, there exist uncountably many words w in $\{0, 1\}^\omega$ such that f_w is not computable, and hence uncountably many sets X of the above form with unique d2-ℓ-orderings that are not definable by any $\mathcal{D}_2$–automaton. □

One might consider more powerful automata whose transitions from a vertex x are determined from the state and the neighbourhood of x consisting of vertices at distance at most some p. A similar proof can be done with sets similar to W, of diameter larger than p.

4B. *A $\mathcal{D}_2$-automaton extending $\mathcal{B}$.* We define a $\mathcal{D}_2$-automaton $\mathcal{F}$ that extends $\mathcal{B}$ and is intended to d2-ℓ-order sets Z such that $G_2(Z)$ is connected but $G_1(Z)$ is not. In Table 4, in the "possible directions" column, "…" means "any". (The list of cases read from top to bottom can be implemented by IF THEN ELSE expressions). The initial state is `Down` and the final one is `End`.

Example 4.8. (1) Let Z be defined by $2i/3 \le j \le 3i/2$. Its first levels are $\{(0, 0)\}$, $\varnothing$, $\{(1, 1)\}$, $\varnothing$, $\{(2, 2)\}$, $\{(2, 3), (3, 2)\}$. A d2-ℓ-ordering can be defined by $\mathcal{F}$ that makes northeast moves $(0, 0) \to (1, 1) \to (2, 2)$ and then continues with the transition rules of $\mathcal{B}$.

state	possible directions	action	next state
Down	SE, …	SE	Down
	¬SE, E, …	E	Up
	¬SE, ¬E, N, …	N	Up
	¬SE, ¬E, ¬ N, NE, …	NE	Up
	¬SE, ¬E, ¬ N, ¬ NE, …		End
Up	NW, …	NW	Up
	¬NW, N, …	N	Down
	¬NW, ¬ N, E, …	E	Down
	¬NW, ¬ N, ¬E, NE, …	NE	Down
	¬NW, ¬ N, ¬E, ¬ NE, …		End

Table 4. The $\mathcal{D}_2$-automaton $\mathcal{F}$.

(2) Let Z be defined by $i/2 \leq j \leq (i+1)/2$. Its first levels are $\{(0,0)\}$, $\varnothing$, $\{(1,1)\}$, $\{(2,1)\}$, $\varnothing$, $\{(3,2)\}$. All levels are singleton. It satisfies conditions (C1), (C2) and (C3). □

Our next aim is to characterise the sets $Z \subseteq \mathbb{N} \times \mathbb{N}$ that are d2-ℓ-ordered by the $\mathcal{D}_2$-automaton $\mathcal{F}$ and the $\mathcal{D}_{2\ell}$-automaton $\mathcal{B}$ of Section 4B.

In the following definition, we use the notation of Definition 4.3.

Definition 4.9 (other conditions on sets $Z \subseteq \mathbb{N} \times \mathbb{N}$). We consider the following variant of condition (C3):

(C4) Each nonempty level is labelled by `Down` or `Up`, in such a way that:
- (C4.0) Z_0 is labelled by `Down`.
- (C4.1) If $Z_{k'}$ follows Z_k labelled by `Down` (resp. `Up`) then, it is labelled by `Up` (resp. `Down`).
- (C4.2) If $Z_{k'}$ follows Z_k, then $\texttt{Last}(Z_k)$ is adjacent to $\texttt{First}(Z_{k'})$ in $G_2(Z)$.

If $G_1(Z)$ is connected, we have $k' = k+1$ in conditions (C4.1) and (C4.2).

Example 4.4 satisfies conditions (C1)–(C3) but not condition (C4). By condition (C4), the labelling of nonempty levels is defined in a unique way, because we want to characterise the existence of deterministic automata in the next theorem. Conditions (C1), (C2) and (C4) imply condition (C3).

Theorem 4.10. *Let $Z \subseteq \mathbb{N} \times \mathbb{N}$. It is d2-$\ell$-ordered by automaton $\mathcal{F}$ if and only if it satisfies conditions (C1), (C2) and (C4). It is d2-ℓ-ordered by automaton $\mathcal{B}$ if and only if $G_1(Z)$ is connected (which implies (C1)) and Z satisfies conditions (C2) and (C4).*

Proof. Let $Z \subseteq \mathbb{N} \times \mathbb{N}$ satisfy conditions (C2) and (C4).

If $G_1(Z)$ is connected, then, the automata $\mathcal{F}$ and $\mathcal{B}$ order Z by traversing the levels in the order $Z_0, Z_1, \ldots$. They are in state `Down` on even levels and in state `Up` on the others.

If $G_2(Z)$ is connected but $G_1(Z)$ is not (some levels may be empty), the automaton $\mathcal{F}$ traverses the nonempty levels in increasing order.

Conversely, consider a Hamiltonian d2-ℓ-path defined by $\mathcal{B}$. As its moves that increase the height of a vertex are north and east only, $G_1(Z)$ is connected. This path is a sequence of intervals, all elements of which have same height. This proves condition (C2). The transitions between two levels are by moves north or east. These transitions prove condition (C4).

The proof is similar for a path defined by $\mathcal{F}$. □

4C. *Sets that satisfy conditions* (C1)–(C4). We consider sets defined by conjunctions of arithmetical conditions, and hence that are intersections of finitely many half-planes.

Definition 4.11 (affine subsets of $\mathbb{N} \times \mathbb{N}$). We call *affine*[7] a subset Z of $\mathbb{N} \times \mathbb{N}$ defined by the conjunction of finitely many conditions of the following forms, intended to specify that $(i, j) \in Z$:

(i) $i \leq a$,

(ii) $j \leq bi + c$,

(iii) $j \geq di - e$,

where $a, b, c, d, e \in \mathbb{Q}$, $a, c, d, e \geq 0$.

That $a, c, e \geq 0$ ensures that $(0,0)$ is in Z. We restrict coefficients to rational numbers in order to be able to design algorithms for deciding properties of a given affine set Z that are listed below. Each level Z_k can be enumerated in a straightforward brute force manner.

[7] A more general definition of an affine set could be obtained by omitting the nonnegativity conditions on a, c, d and e.

Questions 4.12. (1) Is a given affine set Z finite?

(2) Is $G_2(Z)$ connected? Is $G_1(Z)$ connected?

(3) Are conditions (C1)–(C3) satisfied?

(4) If they are, does there exist an automaton[8] that defines a d2-ℓ-ordering?

The following examples show a variety of cases.

Example 4.13. We let Z be defined by the following conditions:

(1) $i/2 - 1/3 \leq j \leq i/2$. Then $Z = \{(2n, n) \mid n \in \mathbb{N}\}$ and $G_2(Z)$ is infinite without edges.

(2) $(i-1)/2 \leq j \leq i/2$. Then $Z = \{(2n, n), (2n+1, n) \mid n \in \mathbb{N}\}$ and $G_2(Z)$ is connected but $G_1(Z)$ is not.

(3) $i \leq j \leq i$. Then $G_2(Z)$ is an infinite diagonal southwest-northeast path and $G_1(Z)$ has no edge.

(4) $(4i-1)/10 \leq j \leq i/2$. Then $G_2(Z - \{(0,0)\})$ is connected but $G_2(Z)$ is not as Z_1, Z_2 and Z_3 are empty.

The sets Z of cases (1) and (4) are not d2-ℓ-ordered by any automaton. Those of cases (2) and (3) are by $\mathcal{F}$ but not by $\mathcal{B}$. □

Definition 4.14 (convexity properties). We define for a subset Z of $\mathbb{N} \times \mathbb{N}$ the following convexity properties:

(*horizontal convexity*) If (i, j) and $(i+k, j) \in Z$, then $(i+k', j) \in Z$ for $0 < k' < k$.

(*vertical convexity*) If (i, j) and $(i, j+k) \in Z$, then $(i, j+k') \in Z$ for $0 < k' < k$.

These properties are preserved by intersection. They are satisfied by each set defined by a single inequality, and hence by every affine set. Furthermore, a similar argument shows than an affine set satisfies the following:

(*diagonal convexity*) If $(i, j+k)$ and $(i+k, j) \in Z$, then $(i+k', j+k-k') \in Z$ for $0 < k' < k$, which is equivalent to condition (C2).

We will also use the following two notions:

(*knight convexity*)[9] (H) If $(i, j+1)$ and $(i+2, j) \in Z$, then $(i+1, j) \in Z$.
(V) If $(i, j+2)$ and $(i+1, j) \in Z$, then $(i, j+1) \in Z$.

An affine set Z defined by a inequality of type (i) or (iii) satisfies condition (H); if it is defined by an inequality of type (i) or (ii) with $b \geq 0$, it satisfies condition (V). Diagonal and knight convexities are also preserved by intersection.

Theorem 4.15. *One can decide if an affine subset of* $\mathbb{N} \times \mathbb{N}$ *is finite and if it satisfies conditions* (C1), (C3) *and* (C4).

Lemma 4.16. (1) *An affine subset is finite if its description contains, among other inequalities, one of the form* $j \leq bi + c$ *with* $b < 0$ *or two of the following forms:* $i \leq a$ *and* $j \leq bi + c$ *with* $b \geq 0$, *or* $j \leq bi + c$ *and* $j \geq di - e$ *with* $d > b > 0$. *If so, one can enumerate it and check conditions* (C1), (C3) *and* (C4).

[8] One might also wish to order Z by a d2-ordering, that does not necessarily respect levels. We leave this study for future research.

[9] Named by reference to the move of knights in chess.

(2) *An affine subset defined Z by $bi - e \leq j \leq bi + c$ with $b, c, e \geq 0$ is empty or infinite. This can be decided.*

Proof. (1) The listed inequalities imply finiteness. In each case, the set Z can be enumerated level by level. One can then check whether it satisfies conditions (C1), (C3) and (C4).

(2) Let $b = r/q$. A pair (i, j) of nonnegative integers is in Z if and only if $(i + q, j + r)$ is. The result follows by elementary arithmetic. □

In the description of an affine set, we can eliminate an inequality $i \leq a'$ if we already have $i \leq a$ with $a < a'$ in the description, and similarly for the inequalities of types (ii) and (iii) (of Definition 4.11). After these eliminations, we obtain a *nonredundant description* (although possibly not minimal).

Example 4.17. The following affine sets are infinite:

(1) The set Z defined by $j \geq i/2,\ j \geq i - 2$ and $i \leq 7$ (conditions of type (i) and (iii)).

(2) The set Z' defined by $i/2 \leq j \leq i$ (conditions of type (ii) and (iii)). The level Z'_1 is empty but $G_1(Z' - Z'_0)$ is connected.

(3) The set Z'' defined by $i = j$ (hence by conditions of type (ii) and (iii)). The graph $G_1(Z'')$ has no edge but $G_2(Z'')$ is connected. □

If $Z \subseteq \mathbb{N} \times \mathbb{N}$, and $p < q$, with $p, q \in \mathbb{N} \cup \{\infty\}$, we denote by $Z_{[p,q[}$ the union of the levels Z_k for $p \leq k < q$.

Lemma 4.18. *Let Z be an affine subset of $\mathbb{N} \times \mathbb{N}$. Let Z_p be a nonempty level. The following equivalences hold*:

(1) *$G_1(Z)$ is connected if and only if $G_1(Z_{[0,p+1[})$ and $G_1(Z_{[p,\infty[})$ are so.*

(2) *Z satisfies (C1) and (C3) if and only if $Z_{[0,p+1[}$ and $Z_{[p,\infty[}$ do so and the label* `Up` *or* `Down` *of Z_p is the same in condition (C3) for $Z_{[0,p+1[}$ and for $Z_{[p,\infty[}$.*

(3) *If Z contains $(0, 0)$, then it satisfies (C1) and (C4) if and only if $Z_{[0,p+1[}$ satisfies (C1) and (C4), and $Z_{[p,\infty[}$ satisfies (C1), and its nonempty levels have a labelling that satisfies (C4.1) and (C4.2), such that the label of Z_p is the same for $Z_{[0,p+1[}$ and $Z_{[p,\infty[}$.*

Proof. (1) This is clear since $Z_{[0,p+1[} \cap Z_{[p,\infty[} = Z_p \neq \emptyset$. The same holds for $G_2(Z)$, i.e, for condition (C1).

(2) This is clear from (1) and the definition of (C3).

(3) This is clear from (1) and the definitions. In the labelling of the levels of $Z_{[p,\infty[}$, instead of condition (C4.0), we require that Z_p, the first nonempty level of $Z_{[p,\infty[}$, have same label for $Z_{[0,p+1[}$ and $Z_{[p,\infty[}$. □

Proof of Theorem 4.15. Let Z be an affine subset of $\mathbb{N} \times \mathbb{N}$ defined by a nonredundant description. We have the following five cases, where each one excludes the previous ones. They cover all descriptions of affine sets.

Case 1: We are in the cases covered by Lemma 4.16(1); hence Z is finite and this lemma yields the result.

Case 2: The description of Z consists of inequalities of types (i) and (iii). Then Z is infinite, $G_1(Z)$ is connected and Z satisfies knight convexity of type (H). All levels are nonempty, and one can label them according to (C4.0) and (C4.1). We check condition (C4.2). Let Z_k be labelled by `Down`. Then

$\mathtt{Last}(Z_k) = \min(Z_k) = (i, j)$ is adjacent to $(i, j+1)$ of height $k+1$. By condition (H) we cannot have $(i+2, j-1) \in Z_{k+1}$. Hence $\mathtt{First}(Z_{k+1}) = \min(Z_{k+1})$ is $(i, j+1)$ or $(i+1, j)$ and thus is adjacent to $\mathtt{Last}(Z_k)$. If Z_k is labelled by $\mathtt{Up}$, then $\mathtt{Last}(Z_k) = \max(Z_k) = (0, j)$ and $\mathtt{First}(Z_{k+1}) = \max(Z_{k+1}) = (0, j+1)$ is adjacent to it. The set Z of Example 4.17(1) is of this type.

Case 3: The description of Z consists of inequalities of types (ii) with $b \geq 0$. Then Z is infinite, $G_1(Z)$ is connected and Z satisfies knight convexity of type (V). The proof is similar to that of Case 2.

For the next cases, we define $I(Z)$ as the set of coordinates $(x, y) \in \mathbb{Q} \times \mathbb{Q}$ of the intersection points in the plane of the lines associated with the defining inequalities, including the inequalities $i \geq 0$ and $j \geq 0$. We let $i(Z)$ be the smallest integer i such that $i \geq x + y$ for all (x, y) in $I(Z)$.

Case 4: Z is defined by inequalities of type (ii), of the form $j \leq bi + c$, and of type (iii), of the form $j \geq di - e$ with $b > d \geq 0$ for any such two inequalities. Let $\bar{b}$ be the minimal such b and $\bar{d}$ be the maximal such d, Let $\bar{c}$ and $\bar{e}$ be the corresponding constant coefficients.

We first consider W defined by the inequalities $j \leq \bar{b}i + \bar{c}$ and $j \geq \bar{d}i - \bar{e}$ and we let $r, q \in \mathbb{N}$ be such that $\bar{b} > r/q > \bar{d}$. Let i, j be such that $i \geq (\bar{d}q - \bar{e} - \bar{c})/(\bar{b} - \bar{d})$ and $\bar{d}i + \bar{d}q - \bar{e} \leq j \leq \bar{b}i + \bar{c}$. We have in W the vertices (i, j), $(i+q, j)$ and $(i+q, j+r)$, as one checks easily. Hence, by horizontal and vertical convexities, we have a path in $G_1(W)$ from (i, j) to $(i+q, j+r)$. This path can translated into a path from $(i+nq, j+nr)$ to $(i+(n+1)q, j+(n+1)r)$ for each $n > 0$. These translated paths concatenate into an infinite path in $G_1(W)$ starting from (i, j). By horizontal and vertical convexity again, we obtain that $G_1(W)$ is connected.

We let $p := \max\{i + j, i(Z)\}$. It is clear that $Z \subseteq W$, Z_p is not empty and $G_1(Z_{[p,\infty[})$ is connected. It satisfies conditions (C4.1) and (C4.2), where Z_p can be labelled $\mathtt{Up}$ or $\mathtt{Down}$. For proving (C4.2), we use knight convexities, as in Cases 2 and 3. As $Z_{[0,p+1[}$ is finite, we can enumerate it and use Lemma 4.18 to decide if Z satisfies conditions (C1), (C3) and (C4).

Case 5: The set Z is defined by inequalities, among which are $j \leq bi + c$ and $j \geq bi - e$ with $b = r/q \geq 0$. Let W be defined by these two inequalities (we may have $b = e = 0$ and, necessarily, $c \geq 1$). By Lemma 4.16(2), we can identify the case where W, whence Z, is empty. Assume now W is infinite. Other possible defining inequalities are of the forms $j \leq b'i + c$ and/or $j \geq di - e$ with $b' > b > d$.

Hence Z contains the vertex (nq, nr) for each integer n such that $nr + nq \geq i(Z)$. We let $p = m(r+q)$ be the minimal integer larger than or equal to $i(Z)$ (with $m \in \mathbb{N}$). It is clear that $Z \subseteq W$, Z_p is not empty and $Z_{[p,\infty[} = W_{[p,\infty[}$. However, $G_2(Z_{[p,\infty[})$ need not be connected (see Example 4.13(1)).

The finite set $Z_{[p,p+r+q+1[}$ is isomorphic to $Z_{[p+r+q,p+2(r+q)+1[}$ by a translation of vector (r, q). The intersection of these two sets is Z_{p+r+q}. Hence, $Z_{[p,\infty[}$ is the union of the pairwise isomorphic sets $Z_{[p+n(r+q),p+(n+1)(r+q)+1[}$.

Then we have that $G_1(Z_{[p,\infty[})$ (resp. $G_2(Z_{[p,\infty[})$) is connected if and only if $G_1(Z_{[p,p+r+q+1[})$ (resp. $G_2(Z_{[p,p+r+q+1[})$ is.

Thus $Z_{[p,\infty[}$ satisfies (C3) (resp. (C4)) if and only if $Z_{[p,p+r+q+1[}$ does, which is decidable. As $Z_{[0,p+1[}$ is finite, we can decide it satisfies conditions (C1), (C3) and (C4), and we obtain the final result by means of Lemma 4.18. □

Open question: Is there an efficient algorithm for the decision problem of Theorem 4.15?

5. Dimension > 2

We consider next the definition of d2-ℓ-orderings of sets $X_1 \times X_2 \times \cdots \times X_p$, where $X_1, \ldots, X_p$ are finite or infinite linearly ordered sets, equivalently, $[0, m]$ or $\mathbb{N}$. We will prove that a unique automaton[10] with 2^{p-1} states can define a d2-ℓ-ordering of any such a set, without knowing whether the components X_i are finite or infinite. We will use an induction on p for which we require more facts about orderings of $X_q \times \cdots \times X_p$.

5A. *Levels.* We generalise the notion of level from Definition 2.1. We define it abstractly in a linearly ordered set. The notion of height will arise from that of level.

Definition 5.1 (levelled linear order). (a) A *levelled linear order* (*llo*) is a linear order Z defined as a finite or infinite concatenation of finite nonempty intervals $Z_0, Z_1, \ldots, Z_n, \ldots$ such that $Z_0 < Z_1 < \cdots < Z_n < \cdots$ Each interval is called a *level*. If $m \in Z_j$, then $ht(m) := j$ is the *height* of m.

We define $\mathrm{Lev}(Z)$ as the linearly ordered set $\mathbb{N}$ if Z is infinite (all levels are nonempty), and $[0, p]$ if Z_p is the maximal nonempty level.

(b) The *product* of a linear order $X \subseteq \mathbb{N}$ and an llo Z is the llo on the set $U := X \times Z$ defined as in Definition 2.2, with a notion of type, which depends here on the levels of Z. The Z-*type* of a pair $(i, m) \in X \times Z$ is the triple of integers

$$\sigma(i, m) := \text{IF } i + ht(m) \text{ is even THEN } (i + ht(m), i, m),$$
$$\text{ELSE } (i + ht(m), ht(m), m).$$

A pair (i, m) can be determined from its type $\sigma(i, m)$.

The order $\leq_U$ on pairs (i, m) is increasing lexicographically on the Z-types $\sigma(i, m)$. The level U_k is the interval consisting of the pairs (i, m) such that $i + ht(m) = k$. It is important that each level of Z be finite in the case where Z is infinite.

Intuitively, U is obtained by substituting in $X \times \mathrm{Lev}(Z)$, ordered as in Definition 2.3, the interval $i \odot Z_j$ to (i, j), where $i \odot (s_0, \ldots, s_q)$ denotes the linear order $((x, s_0), (x, s_1), (x, s_2), \ldots, (x, s_q))$.

Example 5.2. (1) An example of an llo is $Z^{(3,6)} := [0, 3] \times [0, 6]$ of Definition 2.3, which we can describe as

$$/00/10, 01/02, 11, 20/30, 21, 12, 03/\ldots/26, 35/36/$$

by separating levels. See Figure 2. Every linear order from Definition 2.3 is an llo, as a notion of level is defined.

(2) From a linear order X and an integer p, we can define an llo where each level has p elements, except the last one, which may have less. For $X := \mathbb{N}$ and $p = 3$, we get the llo $/0, 1, 2/3, 4, 5/6, 7, 8/\ldots$.

(3) From a levelled linear order X and an integer q, we can define an llo by restricting X to its first q levels.

(4) The llo on $[0, 3] \times [0, 3]$ is

$$/00/10, 01/02, 11, 20/30, 21, 12/13, 22, 31/32, 23/33/.$$

[10]It is a graph-walking automaton traversing the graph $G_2(Z)$ but its description will not use directions.

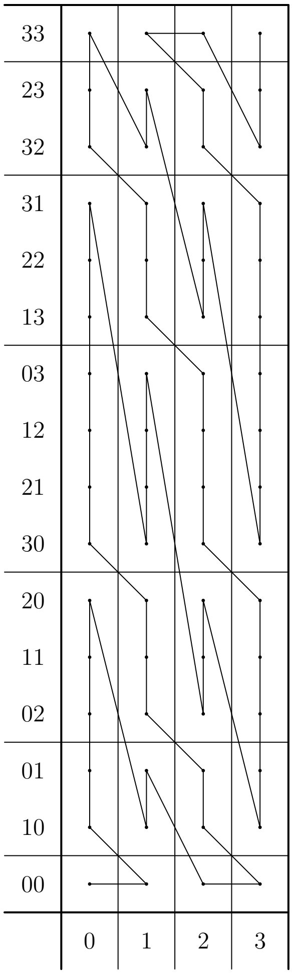


Figure 20. The d2-ℓ-ordering of Example 5.2(5).

(5) The llo on $[0, 3] \times ([0, 3] \times [0, 3])$ is (see Figure 20)

$$/000/100, 010, 001/002, 011, 020, 110, 101, 200/300, 210, 201,$$
$$102, 111, 120, 030, 021, 012, 003/013, 022, 031, 130, \dots,$$
$$301/302, 311, 320, 230, \dots, 023/\dots/233, 332, 323/333/.$$

For conciseness, we write, e.g., 121 instead of (1, 21), a notation that we use below.

Observation 5.3 (concrete descriptions of the levels of $X \times Z$). (1) If X and Z are infinite, the levels U_i of the llo on $U := X \times Z$ of Definition 5.1 are as follows (where $\bullet$ denotes concatenation of sequences):

- If i is even, then $U_i = (0 \odot Z_i) \bullet (1 \odot Z_{i-1}) \bullet \cdots \bullet (i \odot Z_0)$.
- If i is odd, then $U_i = (i \odot Z_0) \bullet (i - 1 \odot Z_1) \bullet \cdots \bullet (0 \odot Z_i)$.

We can determine as follows the pair (y, z') that follows (x, z) in U, where $z \in Z_j$ and thus $(x, z) \in U_{x+j}$. There are three cases with subcases:

(a) z is not last in Z_j (so that (x, z) is not last in U_{x+j}). Then $y = x$ and z' follows z in Z_j.

(b) (x, z) is not last in U_{x+j} but z is last in Z_j. There are two subcases:

(b0) If $x + j$ is even, we must have $j > 0$; otherwise z is last in Z_0 and (x, z) is last in $U_{x+j} = U_x$. Hence, we have $y = x + 1$ and z' is the first element in Z_{j-1} (see the definition of $\mathtt{back}_Z$ below).

(b1) If $x + j$ is odd, we must have $x > 0$; otherwise $(x, z) = (0, z)$ is last in $U_{x+j} = U_j$. Hence, we have $y = x - 1$ and z' is the first element in Z_{j+1}.

(c) If (x, z) is last in U_{x+j} (hence, z is last in Z_j), we have two subcases:

(c0) If $x + j$ is even, then $j = 0$, z is last in Z_0, $y = x + 1$ and z' is the first element in Z_0 (possibly equal to z).

(c1) If $x + j$ is odd, then $x = 0$, z is last in Z_j, $y = 0$ and z' is the first element in Z_{j+1}.

Here, the height of (y, z') is one more than that of (x, z). In the previous two cases, it is the same. We can visualise case (c) as follows:

If i is even, then

$$U_i \bullet U_{i+1} = (0 \odot Z_i) \bullet \cdots \bullet (i \odot Z_0) \bullet ((i+1) \odot Z_0) \bullet \cdots \bullet (0 \odot Z_{i+1}),$$

and if i is odd, then

$$U_i \bullet U_{i+1} = (i \odot Z_0) \bullet \cdots \bullet (0 \odot Z_i) \bullet (0 \odot Z_{i+1}) \bullet \cdots \bullet ((i+1) \odot Z_0).$$

In case (b0) the transition from the last element of Z_j to the first one in Z_{j-1} is called a *back step* in Z. In Case (c0) the transition from z, last in Z_0, to z', first in Z_0, is also a back step inside the level Z_0 of Z, in the case where Z_0 is not singleton. However, Z_0 is singleton if $Z = X_q \times \cdots \times X_p$.

(2) If X and/or Z is finite, this description must be modified. We define m_X in $\mathbb{N} \cup \{\infty\}$ as the least upper bound of X, and, similarly, M_Z as the least upper-bound of $\mathrm{Lev}(Z)$. We let $k \leq m_X + M_Z$ ($U_{m_X+M_Z}$ is the last nonempty level). To describe U_k, we define

$$\alpha := \max\{0, k - M_Z\} = k - \min\{k, M_Z\},$$
$$\beta := \max\{0, k - m_X\} = k - \min\{k, m_X\}.$$

We have $\alpha \leq k - \beta \leq k$ because $k - M_Z \leq m_X$.

If k is even, we have

$$U_k = (\alpha \odot Z_{k-\alpha}) \bullet ((\alpha+1) \odot Z_{k-\alpha-1}) \bullet \cdots \bullet ((k-\beta) \odot Z_\beta).$$

If k is odd, we have

$$U_k = ((k-\beta) \odot Z_\beta) \bullet ((k-\beta+1) \odot Z_{\beta-1}) \bullet \cdots \bullet (\alpha \odot Z_{k-\alpha}).$$

Regarding the determination of the next pair, cases (a) and (b) described above are applicable. For case (c) there are several subcases depending on whether x is maximal and Z_j is the maximal nonempty level of Z.

(c′0) If k is even, then

$$U_k \bullet U_{k+1} = (\alpha \odot Z_{k-\alpha}) \bullet \cdots \bullet ((k-\beta) \odot Z_\beta) \bullet ((k+1-\beta') \odot Z_{\beta'}) \bullet \cdots \bullet (\alpha' \odot Z_{k+1-\alpha}),$$

where $\beta' := k+1-\min\{k+1, m_X\}$ and $\alpha' = k+1-\min\{k+1, M_Z\}$.

We let (x, z) be the last element in $(k-\beta) \odot Z_\beta$, followed by (x', z') in $(k+1-\beta') \odot Z_{\beta'}$. There are three subcases to consider. Examples are from Figure 20:

(i) If $k < m_X$, then $\beta = \beta' = 0,\ x = k,\ x' = k+1$, and z' is the first element in $Z_0 = Z_\beta$. For example, $(x, z) = (2, 00),\ (x', z') = (3, 00)$.

(ii) If $k = m_X$, then $\beta = 0, \beta' = 1,\ x' = x = m_X$, and z' is the first element in Z_1; hence it follows that z is in Z. There is no example because $m_X = 3$.

(iii) If $k > m_X$, then $\beta = k - m_X,\ \beta' = \beta + 1,\ x' = x = m_X$, and z' is the first element in Z_β; hence it follows that z is in Z. For example, $(x, z) = (3, 01),\ (x', z') = (3, 02)$.

In case (i) we have a transition $z \to z'$ inside Z_0.

(c′1) Similarly, if k is odd, then

$$U_k \bullet U_{k+1} = (k-\beta \odot Z_\beta) \bullet \cdots \bullet (\alpha \odot Z_{k-\alpha}) \bullet (\alpha' \odot Z_{k+1-\alpha'}) \bullet \cdots \bullet (k-\beta' \odot Z_{\beta'}).$$

We let (x, z) be the last element in $\alpha \odot Z_{k-\alpha}$, followed by (x', z') in $\alpha' \odot Z_{k+1-\alpha'}$. There are again three subcases:

(iv) If $k < M_Z$, then $\alpha = \alpha' = 0,\ x = x' = 0$, and z' is the first element in Z_{k+1}. Hence it follows that z is in Z. For example, $(x, z) = (0, 23),\ (x', z') = (0, 33)$.

(v) If $k = M_Z$, then $\alpha = 0 = x,\ \alpha' = 1 = x'$, and z' is the first element in $Z_k = Z_{M_Z}$. There is no example because $M_Z = 6$.

(vi) If $k > M_Z$, then $\alpha = x = k - M_Z,\ \alpha' = 1 = x'$, and z' is the first element in Z_{M_Z}. For example, $(x', z') = (1, 23),\ (x', z') = (0, 33)$.

In cases (v), (vi), we have a transition $z \to z'$ inside Z_{M_Z}.

Definition 5.4 (back steps from last elements in their levels). (a) If Z is an llo, we define $\mathtt{back}_Z$ as the set of pairs $(\max(Z_k), \min(Z_{k-1}))$ such that $k > 0$ and Z_k is not empty. If Z_k and Z_{k-1} are singleton, then $\min(Z_{k-1})$ is just the element preceding $\max(Z_k)$ in Z.

(b) The *Last in level* test $\mathtt{Lil}_Z$ applied to $z \in Z$ means that z is the last element in its level.

Example 5.5 (back steps in Cartesian products). Let $Z := [0, 3] \times [0, 3],\ U := [0, 3] \times Z$ and consider in Figure 20 the level

$$U_4 = \big((0, 13), (0, 22), (0, 31), (1, 30), (1, 21), (1, 12), (1, 03), (2, 02), (2, 11), (2, 20), (3, 10), (3, 01)\big).$$

We write ij for a pair (i, j) of Z and (k, ij) for a pair in U corresponding to a triple $(k, (i, j))$ in $[0, 3] \times ([0, 3] \times [0, 3])$. Level U_4 has a transition $(0, 31) \to (1, 30)$ based on a back step in Z from 31 to 30 that decreases height in Z. In this llo, some other $\mathtt{back}_Z$ transitions are used for ordering U: $20 \to 10$ (for $(0, 20) \to (1, 10)$), $33 \to 32$ (for $(2, 33) \to (3, 32)$), $23 \to 13$ (for $(1, 23) \to (2, 13)$) and $33 \to 32$ (for $(2, 33) \to (3, 32)$).

Construction 5.6. We define an automaton $\mathcal{A}_U$ intended to define the llo on $U := X \times Z$ (see Definition 5.1), where X and Z are llo's that satisfy the following conditions:

(1) All levels of X are singleton, and so is the minimal level Z_0 of Z and its maximal one if it is finite.

(2) We are given an automaton $\mathcal{A}_Z$ for Z that defines back steps, and more precisely, such that, based on it, we have routines for the following tests and actions:

$$\texttt{first}_Z, \quad \texttt{last}_Z, \quad \texttt{Lil}_Z, \quad \texttt{next}_Z, \quad \texttt{back}_Z.$$

For X, $\texttt{Lil}_X(x)$ is always true, and $\texttt{back}_X$ is nothing but $\texttt{prev}_X$. We will use the routines $\texttt{first}_X$, $\texttt{last}_X$, $\texttt{next}_X$ and $\texttt{prev}_X$.

Describing automata with directions N, E, SE etc. is no more convenient. We will use Boolean conditions on X and Z instead. If X is an interval of integers, then

$\texttt{first}_X(x)$ is implemented by the test $(x = 0)$?,

$\texttt{last}_X(x)$ is implemented by the test $(x = m_X)$?,

$\texttt{next}_X$ by $x := x + 1$, and

$\texttt{prev}_X$ by $x := x - 1$.

We use these notations for uniformity with those for Z that cannot be easily expressed from integers. (However, see Proposition 5.8 below).

The minimal level U_0 and the maximal one (if U is finite) are singleton. This will allow us to use recursively this construction. For the same reason, we will build an automata $\mathcal{A}_U$ that defines the same five tests and actions as for Z.

Definition of $\mathcal{A}_U$. Its states are pairs (Up, s) and (Down, s), where s is a state of $\mathcal{A}_Z$.

The initial state is (Down, $\texttt{Init}_Z$) where $\texttt{Init}_Z$ is the initial state of $\mathcal{A}_Z$.

We define

$$\texttt{first}_U := \texttt{first}_X \wedge \texttt{first}_Z \quad \text{and} \quad \texttt{last}_U := \texttt{last}_X \wedge \texttt{last}_Z.$$

and $\texttt{Lil}_U$ is defined by Table 5. In Tables 5–8, "state" indicates the first component of the state. The second component is used in the computations of $\texttt{first}_Z$, $\texttt{last}_Z$, $\texttt{Lil}_Z$, $\texttt{next}_Z$ and $\texttt{back}_Z$. It replaces the directions and border conditions used in the automata of Sections 2, 3 and 4. The examples concern $U := [0, 3] \times Z$, where $Z := [0, 3] \times [0, 3]$, and the d2-$\ell$-ordering shown in Figure 20. The actions $\texttt{next}_U$ and $\texttt{back}_U$ are described in Tables 6–8. In Tables 6 and 7, "property" indicates if $\texttt{Lil}_U$ holds before the transition is done. The indicated cases (a), (b), (i), (ii) etc. refer to Observation 5.3.

The number of states of $\mathcal{A}_U$ is twice that of the automaton for Z. If all levels of Z are singleton, then $\mathcal{A}_Z$ has only one state. □

5B. *Application to Cartesian products.*

Theorem 5.7. *There is an automaton with 2^{n-1}states that defines a d2-ℓ-ordering of $U = X_1 \times X_2 \times \cdots \times X_n$, where $X_1, X_2, \ldots, X_n$ are finite or infinite linearly ordered sets.*

Proof. We use Construction 5.6 recursively by writing $U = X_1 \times Z = X_1 \times (X_2 \times (\cdots \times X_n))$. To have a d2-ordering, we must check that the distance between consecutive elements is at most 2.

We let $P(Z)$ for an llo Z (intended to be $X_i \times (\cdots \times X_n)$ for some i) be the conjunction of the following properties:

state	subcondition	examples
Up	$\texttt{first}_X \wedge \texttt{Lil}_Z$	(0, 03)
Up	$\texttt{last}_Z$	(1, 33)
Down	$\texttt{first}_Z$	(0, 00),(2, 00)
Down	$\texttt{last}_X \wedge \texttt{Lil}_Z$	(3, 03), (3, 01)
Up or Down	$\texttt{last}_X \wedge \texttt{last}_Z$	(3, 33)

Table 5. Cases where $\texttt{Lil}_U$ is true.

conditions	subcondition	property	action	new state	examples; cases
$\neg\texttt{Lil}_Z$		$\neg\texttt{Lil}_U$	$\texttt{next}_Z$	Down	(1, 21)→(1, 12)(a)
$\texttt{Lil}_Z$	$\neg\texttt{last}_X\wedge\neg\texttt{first}_Z$	$\neg\texttt{Lil}_U$	$\texttt{next}_X$; $\texttt{back}_Z$	Down	(1, 03)→(2, 02)(b)
$\texttt{Lil}_Z$	$\texttt{last}_X\wedge\neg\texttt{last}_Z$	$\texttt{Lil}_U$	$\texttt{next}_Z$	Up	(3, 03)→(3, 13)(ii)-(iii)
$\texttt{Lil}_Z$	$\texttt{first}_Z\wedge\neg\texttt{last}_X$	$\texttt{Lil}_U$	$\texttt{next}_X$	Up	(2, 00)→(3, 00)(i)
$\texttt{last}_X\wedge\texttt{last}_Z$		$\texttt{Lil}_U$	End		

Table 6. $\texttt{next}_U$ for state=Down.

condition	subconditions	property	action	new state	examples; cases
$\neg\texttt{Lil}_Z$		$\neg\texttt{Lil}_U$	$\texttt{next}_Z$	Up	(0, 21)→(0, 12)(a)
$\texttt{Lil}_Z$	$\neg\texttt{first}_X\wedge\neg\texttt{last}_Z$	$\neg\texttt{Lil}_U$	$\texttt{prev}_X$; $\texttt{next}_Z$	Up	(2, 03)→(1, 13)
$\texttt{Lil}_Z$	$\texttt{first}_X\wedge\neg\texttt{last}_Z$	$\texttt{Lil}_U$	$\texttt{next}_Z$	Down	(0, 23)→(0, 33)
$\texttt{Lil}_Z$	$\neg\texttt{last}_X\wedge\texttt{last}_Z$	$\texttt{Lil}_U$	$\texttt{next}_X$	Down	(1, 33)→(2, 33)
$\texttt{last}_X\wedge\texttt{last}_Z$		$\texttt{Lil}_U$	End		(3, 33)

Table 7. $\texttt{next}_U$ for state=Up.

conditions	subcondition	action	new state	examples
Down	$\neg\texttt{first}_X\wedge\texttt{first}_Z$	$\texttt{prev}_X$	Up	(2, 00)→(1, 00)
Down	$\texttt{last}_X\wedge\texttt{Lil}_Z$	$\texttt{back}_Z$	Up	(3, 03)→(3, 02)
Up	$\texttt{first}_X\wedge\texttt{Lil}_Z$	$\texttt{back}_Z$	Down	(0, 03)→(0, 02)
Up	$\neg\texttt{first}_X\wedge\texttt{last}_Z$	$\texttt{prev}_X$	Down	(1, 33)→(0, 33)

Table 8. $\texttt{back}_U$.

(a) Z_0 is singleton and so is the maximal level if Z is finite. (Think of $X_i \times (\cdots \times X_n)$).

(b) The action $\texttt{back}_Z$ changes a single component.

(c) If $\texttt{Lil}_Z$ holds, then $\texttt{next}_Z$ changes a single component.

(d) If $\texttt{Lil}_Z$ does not hold, then $\texttt{next}_Z$ changes at most two components.

Property $P(Z)$ holds if all levels of Z are singleton, in particular for $Z = X_n \subseteq \mathbb{N}$. Then $\texttt{Lil}_Z$ always holds and $\texttt{back}_Z$ is $\texttt{prev}_Z$.

Next we consider $P(U)$ where $U := X \times Z$ and $P(Z)$ holds.

(a) We have this by the definitions.

(b) This holds by the definition of $\mathtt{back}_U$ in Table 8 and assertion (b) for Z.

(c) Assume that $\mathtt{Lil}_U$ holds. From Table 5, we have $\mathtt{Lil}_Z$ in all cases, because $\mathtt{last}_Z$ implies $\mathtt{Lil}_Z$ and so does $\mathtt{first}_Z$ because Z_0 is singleton (by (a)).

Consider Table 6. The transition whose action is $\mathtt{next}_X$; $\mathtt{back}_Z$ and precondition is $\neg\mathtt{last}_X \wedge \neg\mathtt{first}_Z$ is not compatible with the condition $\mathtt{Lil}_U$ which needs, in state Down, $\mathtt{first}_Z$ or $\mathtt{last}_X$. By $P(Z)$, all transitions change a single component. Similarly, consider Table 7. The transition whose action is $\mathtt{prev}_X$;$\mathtt{next}_Z$ and precondition is $\neg\mathtt{first}_X \wedge \neg\mathtt{last}_Z$ is not compatible with the condition $\mathtt{Lil}_U$, which needs, in state Up, $\mathtt{first}_X$ or $\mathtt{last}_Z$. This proves (c).

(d) This is clear from Tables 6 and 7. □

Condition (a) is necessary for Construction 5.6 to work.

The automaton is the same for sets X_i either infinite or finite with maximal value m_i.

Direct computation of the next element. We take each X_i to be $[0, m_i]$ or $\mathbb{N}$, with known least upper-bound m_i. We wish to compute the n-tuple following a given one, say (3, 0, 2, 4, 0, 0) to take an example, without having to enumerate U until one reaches the given tuple and the one following it.

Proposition 5.8. *Let $U := X_1 \times X_2 \times \cdots \times X_n$ such that the values m_i are known. There exists an algorithm that, for input $\boldsymbol{x} = (x_1, \ldots, x_n)$ such that $x_i \leq m_i$ for all i, determines in time $O(n)$ the n-tuple that follows $\boldsymbol{x}$ in $\leq_U$ without enumerating U.*

Proof. We will compute $\mathtt{next}_U((x_1, \ldots, x_n))$ by means of at most n auxiliary computations of

$$\mathtt{Lil}_{X_i \times \cdots \times X_n}((y_i, \ldots, y_n)),$$
$$\mathtt{next}_{X_i \times \cdots \times X_n}((y_i, \ldots, y_n)),$$
$$\mathtt{back}_{X_i \times \cdots \times X_n}((y_i, \ldots, y_n)))$$

for $1 \leq i \leq n$ and appropriate tuples $(y_i, \ldots, y_n)$.

To simplify notation, we will use $\mathtt{Lil}_i(y_i, \ldots, y_n)$ for $\mathtt{Lil}_{X_i \times \cdots \times X_n}((y_i, \ldots, y_n))$, and similarly for next and back. We fix $U := X_1 \times X_2 \times \cdots \times X_n$ such that the values m_i are known.

If $(y_i, \ldots, y_n) \in X_i \times \cdots \times X_n$, $1 \leq i \leq n$, we define

$$\Lambda_i(y_i, \ldots, y_n) := \big(\mathtt{Lil}_i(y_i, \ldots, y_n), \mathtt{next}_i(y_i, \ldots, y_n), \mathtt{back}_i(y_i, \ldots, y_n)\big),$$

where $\mathtt{Lil}_i(y_i, \ldots, y_n) \in \{\text{true}, \text{false}\}$, $\mathtt{next}_i(y_i, \ldots, y_n)$ is $\perp$ (undefined) if $(y_i, \ldots, y_n) = (m_i, \ldots, m_n)$ and is in $X_i \times \cdots \times X_n$ otherwise, $\mathtt{back}_i(y_i, \ldots, y_n)$ is $\perp$ if $\mathtt{Lil}_i(y_i, \ldots, y_n) = \text{false}$ or $(y_i, \ldots, y_n) = (0, \ldots, 0)$, and it is in $X_i \times \cdots \times X_n$ otherwise.

We compute $\Lambda_1(x_1, \ldots, x_n)$ by recursion, by means of $\Lambda_2(\cdots), \ldots, \Lambda_n(\cdots)$ for appropriate arguments.

For computing $\Lambda_i(y_i, \ldots, y_n)$, we use Tables 5, 6, 7 and 8. The state is Up if $y_i + \cdots + y_n$ is odd and Down if it is even.

If $i = n$, then $\Lambda_n(y_n) := (\text{true}, \ y_n + 1, \perp)$ (or $(\text{true}, \perp, \perp)$ if $y_n = m_n$).

We now examine how to compute $\Lambda_i(y_i, \dots, y_n)$ if $i < n$.

For computing $\mathtt{Lil}_i(y_i, \dots, y_n)$ (see Table 5), we use

$(y_i = 0)?$	for $\mathtt{first}_X$,	
$(y_i = m_i)?$	for $\mathtt{last}$ $\mathtt{Lil}_{i+1}(y_{i+1}, \dots, y_n)$	for $\mathtt{Lil}_Z$ and
$((y_{i+1}, \dots, y_n) = (m_{i+1}, \dots, m_n))?$	for $\mathtt{last}_Z$.	

For computing $\mathtt{next}_i(y_i, \dots, y_n)$ and $\mathtt{back}_i(y_i, \dots, y_n)$ (see Tables 6, 7 and 8), we use

$((y_{i+1}, \dots, y_n) = (0, \dots, 0))?$	for $\mathtt{first}_Z$,
$\mathtt{next}_{i+1}(y_{i+1}, \dots, y_n)$	for $\mathtt{next}_Z$,
$\mathtt{back}_{i+1}(y_{i+1}, \dots, y_n)$	for $\mathtt{back}_Z$

and the same definitions as above for $\mathtt{first}_X$, $\mathtt{last}_X$ and $\mathtt{last}_Z$. □

Example 5.9. Here are some particular cases and examples:

(1) If $\boldsymbol{x} = (m_1, \dots, m_n) \in \mathbb{N}^n$, there is no next element because $\boldsymbol{x}$ is last in U.

(2) If $\boldsymbol{x} = (2p, 0, \dots, 0)$ and $0 \le 2p < m_1$, or $\boldsymbol{x} = (0, 0, \dots, 0, 2p+1)$ and $0 < 2p+1 < m_n$, then, $\boldsymbol{x}$ is last in its level and the following element $\boldsymbol{x}'$ is, respectively, $(2p+1, 0, \dots)$ or $(0, 0, \dots, 0, 2p+2)$.

(3) Let $\boldsymbol{x} = 302400 \in \mathbb{N}^6,\ m_1 > 3,\ m_4 = 2,\ m_5 = 4,\ m_2, m_3, m_6 > 0$,

For $\boldsymbol{x} = 302400$, the state is $\mathtt{Up}$, $\mathtt{Lil}_6(\boldsymbol{x}) =$ false (see Table 5: 3 is not first in X_1 and 02400 is not last in $X_2 \times \cdots \times X_6$) and so $\mathtt{next}_6(\boldsymbol{x}) = 3 \bullet \mathtt{next}_5(02400)$.

The state is now $\mathtt{Down}$; then $\mathtt{Lil}_5(02400) =$ false (0 is not last in X_2, 2400 is not first in $X_3 \times \cdots \times X_6$) and so $\mathtt{next}_6(\boldsymbol{x}) = 3 \bullet 0 \bullet \mathtt{next}_4(2400)$.

The state is $\mathtt{Down}$; then $\mathtt{Lil}_4(2400) = \mathtt{Lil}_3(400) =$ true and

$$\mathtt{next}_4(2400) = 2 \bullet \mathtt{next}_3(400) = \cdots = 2401.$$

Hence $\mathtt{next}_6(\boldsymbol{x}) = 302401$. Note that we did not need to compute $\mathtt{back}$ in this case.

(4) Let now $\boldsymbol{x} = 4323 \in \mathbb{N}^4,\ m_4 = 5,\ m_1 = m_2 = m_3 = 1$.

Then $\mathtt{Lil}_4(4323) =$ false, $\mathtt{Lil}_3(323) =$ true, and $\mathtt{next}_4(\boldsymbol{x}) = (4+1) \bullet \mathtt{back}_3(323) = 5313$. □

Remark 5.10. Computation is accelerated if we note the following facts, stated as in Proposition 5.8.

Claim 1: $\mathtt{Lil}_i(y_i, \dots, y_n)$ implies $\mathtt{Lil}_{i+1}(y_{i+1}, \dots, y_n)$.

Claim 2: $\neg\mathtt{Lil}_{i+1}(y_{i+1}, \dots, y_n)$ implies $\mathtt{next}_i(y_i, \dots, y_n) = y_i \bullet \mathtt{next}_{i+1}(y_{i+1}, \dots, y_n)$.

Open questions 5.11. With the hypothesis of Proposition 5.8, design an algorithm to determine the rank $\mathrm{rk}(\boldsymbol{x})$ of tuple $\boldsymbol{x}$ in the ordering $\le_U$, and conversely, to determine the tuple of given rank i.

These tasks are easy in the case of Definition 2.2, i.e., when $n = 2$ and $m_1 = m_2 = \infty$, because then

$$\begin{aligned}\mathrm{rk}(i, j) = \text{IF } i + j \text{ is even THEN } & (i+j+1)(i+j+2)/2 - j \\ \text{ELSE } & (i+j+1)(i+j+2)/2 - i.\end{aligned}$$

Conversely, given $\mathrm{rk}(i, j) = n$, one determines $i + j$ as the least integer m such that $(m+1)(m+2)/2 \ge n$, from which one obtains i and j depending on its parity.

6. Conclusion

We presented a few open questions in the previous sections; here are some more.

(1) Which "simple" automata can define d1-orderings of $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$, and more generally, of Cartesian products $\mathbb{N} \times \mathbb{N} \times \cdots \times \mathbb{N}$?

(2) Which automata can define d2-orderings, which may not respect levels?

(3) What about sets defined by Boolean combinations of linear inequalities? They may not be convex.

(4) Does there exist a finite automaton that can d2-ℓ-order an affine subset $Z \subseteq \mathbb{N} \times \mathbb{N} \times \cdots \times \mathbb{N}$ defined by conditions of the form $a_1 i_1 + \cdots + a_n i_n \leq b$, with $b \geq 0$ (ensuring that Z contains $(0, 0, \ldots, 0)$). Already for $n = 2$, we have a counterexample in Figure 17. We would need to generalise conditions (C1)–(C4) to larger dimensions.

Appendix

The `Enum` package is part of the `TRAG`[11] system which is written in `Common Lisp`. The code can be found at https://idurand@bitbucket.org/idurand/trag.git. The first version of this package was presented in [Durand 2012]. It offered the possibility of creating basic enumerators (inductive, from a list, ...) and combining existing ones by using operations like taking a product, sequencing or filtering. The general product built on the binary product does not give d2-orderings.

Here we give some hints about how we programmed a bidirectional levelled enumerator which enumerates a (possibly infinite) Cartesian product by producing d2-ℓ-orderings.

A.1. *Enumerators and bidirectional enumerators.*

A.1.1. *General enumerators.* In the following, an enumerator E is identified with the enumerated sequence. In the `Enum` package, each enumerator `E` has at least the following operations:

- `next-element-p (E)`: does there exist a next element?
- `next-element (E)`: move to the next element.

For the implementation, we also need:

- `init-enumerator (E)`: put `E` in its initial state.
- `copy-enumerator (E)`: independent copy of `E`.

Examples with a finite enumerator.

```
ENUM> (defparameter *ABC* (make-list-enumerator '(A B C))) => *ABC*
ENUM> (next-element *ABC*) => A
ENUM> (next-element *ABC*) => B
ENUM> (next-element-p *ABC*) => T
ENUM> (next-element *ABC*) => C
ENUM> (next-element-p *ABC*) => NIL
ENUM> (collect-enum *ABC*) => (A B C) ;; only if finite
```

[11] https://trag.labri.fr

Examples with an infinite enumerator.

```
ENUM> (defparameter *naturals*
        (make-inductive-enumerator 0 (lambda (n) (1+ n)))) => *NATURALS*
ENUM> (next-element *naturals*) => 0
ENUM> (next-element *naturals*) => 1
ENUM> (next-element *naturals*) => 2
ENUM> (init-enumerator *naturals*) => #<INDUCTIVE-ENUMERATOR {100B58E013}>
ENUM> (next-element *naturals*) => 0
ENUM> (next-element *naturals*) => 1
ENUM> (next-element-p *naturals*) => T ;; always true
ENUM> (collect-n-enum *naturals* 9) => (0 1 2 3 4 5 6 7 8) ;; the first 9 values
```

A.1.2. *Bidirectionals enumerators.* A *bidirectional enumerator* `B` has in addition a `way` (`+1` to move forward, `-1` to move backwards), an `initial-way` and the following operations to handle them:

- `initial-way (B)`: initial way,
- `change-initial-way (way, B)`: change `initial-way` to `way`,
- `way (B)`: current way,
- `inverse-way (B)`: inverse `way` of `B`,

together with the following operations:

- `way-next-element-p (way B)`: does there exist a next element in this `way`?
- `way-next-element (way B)`: move to the next element in this `way`.
- `latest-element (B)`: latest object enumerated.

The operations `next-element-p` and `next-element` can be written with `way-next-element-p` and `way-next-element`:

```
(defun next-element-p (B) (way-next-element-p (way B) B))
(defun next-element (B) (way-next-element (way B) B))
```

The implementation of a bidirectional enumerator uses two stacks `past-objects` and `future-objects`, the first one containing the already enumerated objects (that are before the latest one) and the second, the ones that are after, and a slot `latest-object` containing the last enumerated object. If the enumerator is moving backwards, the top element of `past-object` will be produced and moved to `latest-object`; otherwise the top element of `future-object` will be produced and moved to `latest-object`.

Creation and initialization of a bidirectional enumerator. Given a nonempty enumerator `E`, enumerating $e_0, e_1, \ldots$, one can obtain its bidirectional version `B-E` with `(make-bidirectional-enumerator E initial-way)`. In `B-E`, one has access to `E`, the *underlying enumerator*, by `(enum B-E)`. At initialization, if `initial-way` is `-1`, we move forward `(enum B-E)` towards the first element in the positive way, so towards the first element of `E`, e_0, in order to go back to this element at the next call of `next-element`. Consequently, the first call `(next-element-p B-E)` will return `T`, the first call `(next-element B-E)` will return the first element of `E` that is e_0; then `(next-element-p B-E)` will return `Nil` as long as its `way` remains `-1`.

Example of creation and use of a bidirectional enumerator.

```
ENUM> (defparameter *B-NATURALS*
        (make-bidirectional-enumerator *naturals*)) => *B-NATURALS*
ENUM> (next-element *B-NATURALS*) => 0
ENUM> (next-element *B-NATURALS*) => 1
ENUM> (next-element *B-NATURALS*) => 2
ENUM> (way *B-NATURALS*) => 1
ENUM> (inverse-way *B-NATURALS*) => -1
ENUM> (way *B-NATURALS*) => -1
ENUM> (next-element *B-NATURALS*) => 1
ENUM> (next-element *B-NATURALS*) => 0
ENUM> (next-element-p *B-NATURALS*) => NIL
ENUM> (inverse-way *B-NATURALS*) => 1
ENUM> (next-element-p *B-NATURALS*) => T
ENUM> (next-element *B-NATURALS*) => 1
ENUM> (next-element *B-NATURALS*) => 2
ENUM> (latest-element *B-NATURALS*) => 2
ENUM> (way-next-element -1 *B-NATURALS*) => 1
```

A.2. *Enumeration of Cartesian products.* Let $E_1, \ldots, E_p$ be nonempty enumerators (finite or not) such that $E_i = (e^i_0, e^i_1, \ldots)$ if it is infinite, $E_i = (e^i_0, e^i_1, \ldots, e^i_{c_i-1})$ where $c_i = |E_i|$ otherwise. There are no repetitions. Let $T_p = E_1 \times E_2 \times \cdots \times E_p$. The necessity of diagonal enumeration of T_p arises in particular when one of the components is infinite.

A.2.1. *Levelled enumerators of Cartesian products.* The *height* of a tuple $t = (e^1_{j_1}, e^2_{j_2}, \ldots, e^p_{j_p}) \in T_p$ is the sum $\Sigma^p_{i=0} j_i$ of the indices of the elements in the sets E_i. We note here by L^i the i-th *level* of T_p which is the finite set of tuples of height i. Then T_p is the partition of its levels. If T_p is infinite, its number of levels is infinite. The definitions of height and level in Definition 2.1 of Section 2 concern the particular case where the enumerated sequences are $\mathbb{N}$ or intervals $[0, p] \subset \mathbb{N}$.

A *levelled enumerator* enumerates $L^0, L^1, \ldots$ in increasing order. We call *major step* a step that moves to the next level and *minor step* a step inside a level. Levelled enumerators have in addition the predicate

`minor-step-p (E),`

which is true if the next step (`next-element`) does not change the level. It is false when we are done with the enumeration of the current level.

A.2.2. *Bidirectionals levelled enumerators.* A *levelled bidirectional enumerator* is a levelled enumerator which in addition, is bidirectional (it has a `way` and an `initial-way`). When going forward (`way = +1`), it enumerates levels in increasing order: $L^0, L^1, \ldots$ When going backwards (`way = -1`), it enumerates them in decreasing order: $L^i, L^{i-1}, \ldots$ while keeping the forward order inside the levels.

A.2.3. *Diagonal product of a bidirectional enumerator with a bidirectional levelled enumerator.* We implement Definition 5.1(b). Let `X` be a bidirectional enumerator and `Y` be a bidirectional levelled enumerator which when going forward enumerates the levels $Y^0, Y^1, \ldots$. We define below `DP(X, Y)`, the *diagonal product* of `X` and `Y`. When `DP(X, Y)` is created, the initial way of `X` is set to `+1` and the initial way of `Y` is set to `-1`. A *minor step on level* is a step that changes the level of `X` in a way and the level of `Y` in the opposite way but not the level of `D`. In addition to the usual operations we have the accessors `enum-x (D)` and `enum-y (D)` to access respectively to `X` and to `Y`. The other operations are written:

```
(defun latest-element (D)
  (cons (latest-element(enum-x D) (latest-element (enum-y D)))))

(defun minor-step-p (D) ;; precondition (next-element-p D)
  (and (next-element-p (enum-y D))
       (or (next-element-p (enum-x D)) (minor-step-p (enum-y D)))))

(defun way-next-element-p (way D)
  (or (way-next-element-p (way D) (enum-x D))
      (way-next-element-p (way(D) (enum-y D)))))

(defun way-next-element (way D)
  (let* ((enum-x (enum-x enum))
         (enum-y (enum-y enum))
         (next-x (next-element-p enum-x))
         (next-y (next-element-p enum-y)))
    (cond
      ((and next-y (minor-step-p enum-y)) ;; lower-level minor step
       (next-element enum-y))
      ((and next-y next-x) ;; minor-step on level
       ;; each one makes a major in its way
       (next-element enum-x) (next-element enum-y))
      ;; major step
      ((not (or next-x next-y))
       (corner-step enum-x enum-y way))
      (t (sliding-step enum-x enum-y way))))
  (latest-element enum))

(defun sliding-step (X Y way)
  ;; precondition: X or Y can move in its way
  (if (next-element-p Y)
      (way-next-element way Y)
      (way-next-element way X))
  (inverse-way X)
  (inverse-way Y))
```

The call `(sliding-step X Y 1)` corresponds to a *jump-up* (move to upper level) and the call `(sliding-step X Y -1)` corresponds to a back step (move to lower level, see Definition 5.4). If neither `X` nor `Y` can move in their current ways and the enumeration is not over, we are in a case called *corner step*, which may happen only when at least one of the enumerators is finite (otherwise there is always a possible *sliding step*). In the case of a corner step, we reverse the way of the enumerator, which goes in the opposite direction of `way` (of the product enumerator) and move it to the next level according to `way`.

If `way = 1`, we move to the upper level. If `way = -1`, we move to the lower level. The other enumerator changes its way (it could not contribute to the level change because it is blocked in the direction `way`).

```
(defun corner-step (X Y way)
  ;; change the way of the enumerator which goes in opposite direction
  ;; to way and move it; the other enumerator changes way
  (when (plusp (* way (way enum-x)))
    ;; put in enum-x the one that goes in direction -way
    (psetf enum-x enum-y enum-y enum-x))
  (inverse-way enum-x) ;; enum-x will move in direction way
  (next-element enum-x) ;; enum-y will move in direction -way
  (inverse-way enum-y))
```

A.3. ***Diagonal enumeration of a Cartesian product.*** Let `Nil` be the bidirectional levelled enumerator corresponding to the empty product enumerating the singleton set containing a single tuple of length 0: `Nil`$= \{()\}$ it has only one level $L^0 = \{()\}$.

We may show that recursive use of `DP` yields the levelled-ℓ-ordering defined in Definition 5.1 and described in Observation 5.3.

Proposition A.1. *Let $E_1, E_2, \ldots, E_p$ be bidirectional enumerators. The enumerator*

$$\mathtt{DP}(E_1, \mathtt{DP}(E_2, \mathtt{DP}(\ldots, \mathtt{DP}(E_p, \mathtt{Nil}))))$$

is a bidirectional levelled enumerator and defines a d2-ℓ-ordering of $E_1 \times E_2 \times \cdots \times E_p$.

In the examples, we will use only integers so that the level of a tuple is the sum of its elements.

```
ENUM> (defparameter *e2* (make-list-enumerator '(0 1))) => *E2*
ENUM> (defparameter *e3* (make-list-enumerator '(0 1 2))) => *E3*
ENUM> (collect-enum *e2*) => (0 1)
ENUM> (collect-enum *e3*) => (0 1 2)
ENUM> (collect-enum (make-product-enumerator (list *e3* *e3*)))
((0 0) (1 0) (0 1) (0 2) (1 1) (2 0) (2 1) (1 2) (2 2))
ENUM> (collect-enum (make-product-enumerator (list *e3* *e3* *e3*)))
((0 0 0) (1 0 0) (0 1 0) (0 0 1) (0 0 2) (0 1 1) (0 2 0) (1 1 0) (1 0 1)
(2 0 0) (2 1 0) (2 0 1) (1 0 2) (1 1 1) (1 2 0) (0 2 1) (0 1 2) (0 2 2)
(1 2 1) (1 1 2) (2 0 2) (2 1 1) (2 2 0) (2 2 1) (2 1 2) (1 2 2) (2 2 2))
ENUM> (collect-n-enum (make-product-enumerator (list *naturals* *e3*)) 30)
((0 0) (1 0) (0 1) (0 2) (1 1) (2 0) (3 0) (2 1) (1 2) (2 2) (3 1) (4 0) (5 0)
(4 1) (3 2) (4 2) (5 1) (6 0) (7 0) (6 1) (5 2) (6 2) (7 1) (8 0) (9 0) (8 1)
(7 2) (8 2) (9 1) (10 0))
ENUM> (collect-n-enum (make-product-enumerator (list *naturals* *e3*)) 20)
((0 0) (1 0) (0 1) (0 2) (1 1) (2 0) (3 0) (2 1) (1 2) (2 2) (3 1) (4 0) (5 0)
(4 1) (3 2) (4 2) (5 1) (6 0) (7 0) (6 1))
ENUM> (collect-n-enum (make-product-enumerator (list *naturals* *e3* *e3*)) 20)
((0 0 0) (1 0 0) (0 1 0) (0 0 1) (0 0 2) (0 1 1) (0 2 0) (1 1 0) (1 0 1)
(2 0 0) (3 0 0) (2 1 0) (2 0 1) (1 0 2) (1 1 1) (1 2 0) (0 2 1) (0 1 2)
(0 2 2) (1 2 1))
```

Acknowledgements

This research was supported by the GraphEn project of Agence Nationale pour la recherche and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, under grant agreement no. 787367 (PaVeS). We thank the referee for drawing our attention to the Fueter–Pólya theorem.

References

[Courcelle and Durand 2016] B. Courcelle and I. Durand, "Computations by fly-automata beyond monadic second-order logic", *Theoret. Comput. Sci.* **619** (2016), 32–67. MR Zbl

[Durand 2012] I. Durand, "Object enumeration", pp. 43–57 in *Proceedings of the 5th European Lisp Symposium* (Zadar, Croatia, 2012), edited by F. Pehar, 2012.

[Engelfriet and Hoogeboom 2007] J. Engelfriet and H. J. Hoogeboom, "Automata with nested pebbles capture first-order logic with transitive closure", *Log. Methods Comput. Sci.* **3**:2 (2007), art. id. 3. MR Zbl

[Fraigniaud et al. 2005] P. Fraigniaud, D. Ilcinkas, G. Peer, A. Pelc, and D. Peleg, "Graph exploration by a finite automaton", *Theoret. Comput. Sci.* **345**:2-3 (2005), 331–344. MR Zbl

[Wikipedia 2011] "Gray code", Wikipedia entry, 2011, available at https://en.wikipedia.org/wiki/Gray_code.

[Wikipedia 2015] "Fueter–Pólya theorem", Wikipedia entry, 2015, available at https://tinyurl.com/fuetpol.

Received 27 Nov 2019. Revised 2 Apr 2020.

BRUNO COURCELLE:

bruno.courcelle@u-bordeaux.fr
LaBRI, University of Bordeaux and CNRS, Talence, France

IRÈNE DURAND:

irene.durand@u-bordeaux.fr
LaBRI, University of Bordeaux and CNRS, Talence, France

MICHAEL RASKIN:

raskin@mccme.ru
Technische Universität München, Garching bei München, Germany

Moscow Journal of Combinatorics and Number Theory
Vol. 9, No. 3, 2020
https://doi.org/10.2140/moscow.2020.9.293

First-order definitions of subgraph isomorphism through the adjacency and order relations

Oleg Grigoryan, Mikhail Makarov and Maksim Zhukovskii

We study first-order definitions of graph properties over the vocabulary consisting of the adjacency and order relations. We compare logical complexities of subgraph isomorphism in terms of the minimum quantifier depth in two settings: with and without the order relation. We prove that, for pattern-trees, it is at least (roughly) two times smaller in the former case. We find the minimum quantifier depths of $<$-sentences defining subgraph isomorphism for all pattern graphs with at most 4 vertices.

1. Introduction

We consider graphs with the vertex sets $[n] := \{1, \dots, n\}$, $n \in \mathbb{N}$. A *graph property* is a set of graphs closed under isomorphism (isomorphism preserves the adjacency relation $\sim$). In this paper, we focus on *subgraph isomorphism properties* defined below.

Given a *pattern-graph* F, let $\mathcal{S}(F)$ denote the property of containing a (not necessarily induced) subgraph isomorphic to F (i.e., $\mathcal{S}(F)$ is the set of all graphs containing a subgraph isomorphic to F). Let $\Sigma = \{\sim, =, R_1, \dots, R_s\}$ be a finite set of relational symbols, where R_i, $i \in [s]$, represents a certain predicate on $\mathbb{N}$ of arity a_i. Below, $<$ denotes the usual linear order on $\mathbb{N}$.

We consider the first-order logic over the vocabulary Σ. For a sentence φ in this logic and a graph property $\mathcal{P}$, we say that φ expresses $\mathcal{P}$ if

$$G \in \mathcal{P} \Leftrightarrow G \models \varphi.$$

Clearly, for an arbitrary graph F on $[\ell]$ with the edge set $E(F)$, $\mathcal{S}(F)$ is expressed by the first-order sentence

$$\exists x_1 \dots \exists x_\ell \quad \Big[\bigwedge_{i \neq j} x_i \neq x_j\Big] \wedge \Big[\bigwedge_{\{i,j\} \in E(F)} x_i \sim x_j\Big]. \tag{1}$$

Thus, $\mathcal{S}(F)$ is definable in the most laconic first-order logic (over the vocabulary $\{\sim, =\}$).

Consider the parameters $D_\Sigma(F)$ and $W_\Sigma(F)$ defined, respectively, as the minimum quantifier depth (maximum number of variables in a sequence of nested quantifiers, see formal definition in [Libkin 2004], Definition 3.8) and the minimum variable width (the number of distinct variables, see the definition just before Proposition 6.6 in [Libkin 2004]) of a first-order sentence in the vocabulary Σ expressing $\mathcal{S}(F)$. For $\Sigma = \{\sim, =\}$, we will omit the indices and write, simply, $D(F)$ and $W(F)$. For $\Sigma = \{\sim, <\}$, we will write $D_<(F)$ and $W_<(F)$. Finally, $D_{\text{Arb}}(F)$ and $W_{\text{Arb}}(F)$ denote the minimum of $D_\Sigma(F)$ and W_Σ(F)

This work is supported by the grant N 18-71-00069 of Russian Science Foundation.
MSC2010: 03C13, 68Q19.
Keywords: first-order logic, subgraph isomorphism, order, logical complexity, quantifier depth.

over all finite Σ respectively. Notice that the relation $W_\Sigma(F) \le D_\Sigma(F)$ follows from the obvious fact that any first-order sentence of the quantifier depth d can be rewritten using at most d variables.

Since the sentence (1) has the quantifier depth ℓ, we have $D(F) \le \ell$. On the other hand, note that K_ℓ, the complete graph on ℓ vertices, contains F as a subgraph, while $K_{\ell-1}$ does not. Since ℓ first-order variables are necessary in order to distinguish between K_ℓ and $K_{\ell-1}$ (indeed, any quantifier-free subformula of a sentence with $\ell-1$ variables has at most $\ell-1$ variables, and so, it has the same truth value on both $K_{\ell-1}$ and K_ℓ), we have $W(F) = D(F) = \ell$. (However, the problem of estimating the values of these parameters becomes nontrivial for sufficiently large connected *input graphs*; see [Verbitsky and Zhukovskii 2019b; Verbitsky and Zhukovskii 2019c]).

A straightforward conversion of a first-order sentence in the vocabulary $\{\sim, =\}$ defining $\mathcal{S}(F)$ into an algorithm for deciding is there a copy of F in an input graph on n vertices leads to the trivial time bound $O(n^{D(F)})$ which can actually be improved to $O(n^{W(F)})$; see [Libkin 2004, Proposition 6.6]. The preceding discussion shows that first-order sentences in the vocabulary $\{\sim, =\}$ defining $\mathcal{S}(F)$ can only be considered as a very weak computational model for the corresponding decision problem. Notice that the general time bound for the mentioned decision problem established by Nešetřil and Poljak [1985] is $O(n^{(\omega/3)\ell+2})$, where ω is the exponent of fast matrix multiplication, whose value lies between 2 and 2.373 [Le Gall 2014].

However, the above time bounds in terms of the quantifier depth and the variable width apply to the vocabulary $\{\sim, <\}$ as well. It is well known [Libkin 2004; Schweikardt 2013] that there are properties of finite structures that can be defined in first-order logics with $<$ but not without. Thus, it is not surprising, that for certain graphs F on ℓ vertices, $\mathcal{S}(F)$ can be defined much more succinctly in the first-order logic over $\{\sim, <\}$. As a simple example, for $F = K_{1,\ell-1}$ (star graph), it is easy to see that $D_<(F) \le \lceil \log_2(\ell-2)\rceil + 3$ and $W_<(F) \le 3$. Indeed, let

$$\varphi_1(x, y, z) = (x < y) \wedge (x \sim z) \wedge (y \sim z),$$
$$\varphi_k(x, y, z) = \exists w \ \varphi_{\lfloor k/2\rfloor}(x, w, z) \wedge \varphi_{\lceil k/2\rceil}(w, y, z), \quad k \ge 2.$$

Then $\mathcal{S}(F)$ is expressed by the sentence $\exists z \exists x \exists y \quad \varphi_{\ell-2}(x, y, z)$ of the quantifier depth $\lceil \log_2(\ell-2)\rceil + 3$. The same property is expressed by the sentence

$$\exists z \quad \Big(\exists x \ [x \sim z] \wedge \Big[\exists y \ (x < y) \wedge (y \sim z) \wedge \Big(\exists x \ [y < x] \wedge [x \sim z] \wedge [\ldots]\Big)\Big]\Big)$$

with 3 variables.

The natural question to ask is what about an arbitrary tree F? Are $D_<(F)$ and $W_<(F)$ significantly less than ℓ? In this paper, we restrict ourselves with estimating the parameter $D_<(F)$ only. In Section 2, we prove that, for an arbitrary tree F, $D_<(F) \le \frac{1}{2}\ell + \lceil \log_2(\ell+2)\rceil - 1$, which is, roughly, a half of $D(F)$. Unfortunately, we can not prove that it is close to optimal at least for paths. Notice that K_ℓ on $[\ell]$ and $K_{\ell-1}$ on $[\ell-1]$ can not be distinguished by a sentence of the quantifier depth smaller than $\log_2(\ell+1)-2$ (it can be shown, for example, in a usual way via the Ehrenfeucht–Fraïssé game). This implies the trivial lower bound $D_<(F) \ge \log_2(\ell-2)$. On the other hand, $D_{\text{Arb}}(F) \le \text{td}(F) + 2$ [Rossman 2016], where $\text{td}(F)$ is the treedepth of F. In particular, $\text{td}(P_\ell) = \lceil \log_2(\ell+1)\rceil$.

The color coding algorithm of Alon, Yuster and Zwick [Alon et al. 1995] guarantees the time bound $O(n^2)$ for the decision problem for $\mathcal{S}(F)$. Therefore, we can not compete with the best known algorithm.

However, we motivate our interest to the question above exclusively by comparison of the expressive powers of first-order logics with and without the $<$ relation.

In Section 4, we find $D_<(F)$ for all F on at most 4 vertices. For F having 3 vertices, we prove that $D_<(F)=3$ if and only if F is connected. Otherwise, $D_<(F)=2$. For F having 4 vertices, the following is true: $D_<(F)=4$ if and only if F contains a C_4; otherwise, $D_<(F)=3$. In particular, $D_<(K_3)=3$, $D_<(K_4)=4$. It is known that $W_{\mathrm{Arb}}(K_\ell)\geq\frac{1}{4}\ell$ [Rossman 2008]. Moreover, the main result of [He 2015] states that $W_{\mathrm{Arb}}(K_\ell)=\ell$ that, obviously, implies $D_<(K_\ell)=\ell$. Unfortunately, there is no journal version of the latter result. By this reason, we give the proofs for K_3 and K_4 in our paper.

2. Pattern trees of arbitrary size

Theorem 1. *Let F be a tree on ℓ vertices. Then $D_<(F)\leq\frac{1}{2}\ell+\lceil\log_2(\ell+2)\rceil-1$.*

Proof. Consider an arbitrary linear order $<$ on the set of vertices of F: $v_1<\cdots<v_\ell$. Below, we construct a sentence with the desired quantifier depth that defines the existence of a subgraph isomorphic to F and an isomorphism between F and its copy that preserves both $\sim$ and $<$ relations. Having this, it remains to consider a conjunction of all such sentences over all possible linear orders on the vertex set of F.

Let $\mathcal{E}=\{v_2,v_4,\dots\}$ be the set of vertices with even positions and $\mathcal{O}=\{v_1,v_3,\dots\}$ be the set of vertices with odd positions.

Remove from F those edges that have both vertices in $\mathcal{E}$. After that, remove from the obtained forest all isolated vertices. Finally, for every vertex of $\mathcal{E}$ that has degree d bigger than 1 in this graph, cut this vertex into d vertices each with its own unique edge. Denote the trees in the final forest by $F_1,\dots,F_r$. Clearly, if $v\in\mathcal{E}$ belongs to F_j, then v is a leaf. For $j\in[r]$, let F_j have the vertex set $\{v_{k_1^j},\dots,v_{k_{m(j)}^j}\}$ and let $v_{s_1^j},\dots,v_{s_{t(j)}^j}$ be the vertices of F_j that are from $\mathcal{E}$. We will call $v_{s_1^j},\dots,v_{s_{t(j)}^j}$ *distinguished leaves* of F_j.

Let

$$\varphi=\exists x_2\exists x_4\dots\exists x_{2\lfloor\ell/2\rfloor}\left[\bigwedge_{j=1}^{\lfloor\ell/2\rfloor-1}(x_{2j}<x_{2j+2})\right]\wedge\left[\bigwedge_{\{v_{2i},v_{2j}\}\in E(F)}(x_{2i}\sim x_{2j})\right]\left[\bigwedge_{j=1}^{r}\varphi_j(x_{s_1^j},\dots,x_{s_{t(j)}^j})\right],$$

where $\varphi_j(x_{s_1^j},\dots,x_{s_{t(j)}^j})$ states that there exists a copy T_j of F_j on the vertex set $\{x_{k_1^j},\dots,x_{k_{m(j)}^j}\}$ such that

$$v_{i^1}\sim v_{i^2}\text{ in }F_j\Leftrightarrow x_{i^1}\sim x_{i^2}\text{ in }T_j\qquad\text{for }v_{i^1},v_{i^2}\in V(F_j),$$
$$v_{i^1}<v_{i^2}\Leftrightarrow x_{i^1}<x_{i^2}\qquad\text{for }(v_{i^1},v_{i^2})\in[V(F_j)\cup\mathcal{E}]^2\setminus\mathcal{E}^2.$$

As in F_j, the vertices $x_{s_1^j},\dots,x_{s_{t(j)}^j}$ are *distinguished leaves* of T_j. It remains to show that $\varphi_j(x_{s_1^j},\dots,x_{s_{t(j)}^j})$ may be efficiently written (with the quantifier depth not more than $\lceil\log_2(\ell+2)\rceil-1$).

For every i, let $R_i(x)=(x_{2i}<x)\wedge(x<x_{2i+2})$ be the predicate that defines the position of x_{2i+1} in the described copy of F.

Let $j\in[r]$. Clearly, the diameter of F_j is at most $\lceil\ell/2\rceil+1$. Let c_j be a *central vertex* of F_j, i.e., that vertex that minimizes the maximum distance to leaves. Let $c_j=v_{2\gamma-1}$ for a certain $\gamma\in\{0,1,\dots,\lceil\ell/2\rceil\}$. We call a subtree of F_j its *c_j-branch*, if this subtree contains c_j itself, one of its children (here, F_j is rooted in c_j) and all the descendants of this child (a *descendant vertex u* of v is a vertex such that v is in the path from u to the root). Let $F_j^1, \dots, F_j^\mu$ be all the c_j-branches of F_j (μ is the number of children of c_j). Clearly, c_j is a leaf of all these trees. Let us distinguish it. For $q \in [\mu]$, let $v_{k_i^{j,q}}$, $i \in [m(j,q)]$, be all the vertices of F_j^q, and $v_{s_i^{j,q}}$, $i \in [t(j,q)]$, be the distinguished vertices of F_j^q (one of them is $v_{2\gamma+1}$, and all the others are those vertices of F_j^q that are distinguished in F_j).

Then,

$$\varphi_j(x_{s_1^j}, \dots, x_{s_{t(j)}^j}) \;=\; \exists x_{2\gamma+1} \quad R_\gamma(x_{2\gamma+1}) \wedge \Big[\bigwedge_{i=1}^{\mu} \varphi_j^i(x_{s_1^{j,q}}, \dots, x_{s_{t(j,q)}^{j,q}})\Big];$$

here $\varphi_j^i(x_{s_1^{j,q}}, \dots, v_{x_{t(j,q)}^{j,q}})$ states that there exists a copy T_j^i of F_j^i defined on the vertex set $\{x_{k_1^{j,q}}, \dots, x_{k_{m(j,q)}^{j,q}}\}$ such that

$$\begin{aligned} v_{i^1} \sim v_{i^2} \text{ in } F_j^i \Leftrightarrow x_{i^1} \sim x_{i^2} \text{ in } T_j^i &\qquad \text{for } v_{i^1}, v_{i^2} \in V(F_j^i), \\ v_{i^1} < v_{i^2} \Leftrightarrow x_{i^1} < x_{i^2}, &\qquad \text{for } (v_{i^1}, v_{i^2}) \in [V(F_j^i) \cup \mathcal{E}]^2 \setminus [\mathcal{E} \cup \{v_{2\gamma+1}\}]^2. \end{aligned}$$

The vertices $x_{s_1^{j,q}}, \dots, x_{s_{t(j,q)}^{j,q}}$ become the *distinguished leaves* of T_j^i.

The maximum diameter of $F_j^1, \dots, F_j^\mu$ is $\lceil D_j/2 \rceil$, where $D_j \le \lceil \ell/2 \rceil + 1$ is the diameter of F_j. By induction, in at most $\lceil \log_2(\ell+2) \rceil - 1$ steps, we will obtain a forest of edges with all the vertices distinguished. Every such edge is expressed by a formula defining the position of the last distinguished vertices in respect to the vertices of $\mathcal{E}$ and saying that its ends are adjacent. □

Remark. Clearly, two variables are enough to express the "odd" part of F. Therefore, $W_<(F) \le \frac{1}{2}\ell + 2$.

3. The Ehrenfeucht–Fraïssé game

The main tool in our proofs is Ehrenfeucht's theorem. It gives necessary and sufficient conditions of the elementary equivalence of finite structures (in our case graphs) in terms of the so-called Ehrenfeucht–Fraïssé game. We are going to apply this theorem to the vocabulary $\{\sim, <\}$ and, thus, recall the rules of the Ehrenfeucht–Fraïssé game only in these special settings in order to avoid heavy notations.

The *k-round Ehrenfeucht–Fraïssé game* (or, simply, the *k-game*) is played on two graphs G and H whose vertex sets are finite subsets of $\mathbb{N}$. There are two players — *Spoiler* and *Duplicator*. In every round, Spoiler chooses a vertex in G or in H, then Duplicator has to choose a vertex in the other graph. After k rounds, $x_1, \dots, x_k$ are chosen in G and $y_1, \dots, y_k$ are chosen in H. Duplicator wins if and only if, for all distinct $i, j \in [k]$,

$$(x_i \sim x_j) \Leftrightarrow (y_i \sim y_j), \quad (x_i < x_j) \Leftrightarrow (y_i < y_j).$$

The following fact is a corollary of Ehrenfeucht theorem (see details in, e.g., Section 2 of [Verbitsky and Zhukovskii 2019a]).

Lemma 2. *$D_<(F)$ equals the minimum k such that, for any pair of graphs H, G such that $F \subset G$ but $F \not\subset H$, Spoiler has a winning strategy in the k-round game on G and H.*

4. Small patterns

4.1. ***3-patterns.*** There are 4 non-isomorphic graphs on 3 vertices: I_3 (empty graph on 3 vertices), $I_1 \sqcup K_2$, P_3 and K_3.

It is easy to see that $D_<(I_3) = D_<(I_1 \sqcup K_2) = 2$ since containing I_3 is defined by the sentence

$$\exists x \quad (\exists y \quad [y < x]) \wedge (\exists y \quad [x < y])$$

and containing $I_1 \sqcup K_2$ is defined by the sentence

$$\left[\exists x \quad (\exists y \quad [y < x]) \wedge (\exists y \quad [x < y])\right] \wedge \left[\exists x \exists y \quad (x \sim y)\right].$$

Let us switch to $F = K_3$. Let G be the disjoint union of two isolated vertices and K_3. More precisely, the vertex set of G is $\{1, 2, 3, 4, 5\}$ and the set of edges contains only $\{2, 3\}, \{3, 4\}, \{2, 4\}$. Let H be obtained from G by deleting the edge $\{2, 4\}$. It is clear that Duplicator wins the 2-round game on G, H and, therefore, $D_<(K_3) = 3$.

To prove that $D_<(P_3) = 3$, let us consider the following G and H. Let G have the vertex set $\{1, 2, 3, 4, 5\}$ and contain only two edges $\{2, 3\}$ and $\{2, 4\}$. Let H have the vertex set $\{1, 2, 3, 4\}$ and contain the only edge $\{2, 3\}$. Since Duplicator wins the 2-round game on these G and H as well, $D < (P_3) = 3$.

Summing up, for a connected pattern graph F on 3 vertices, $D_<(F) = D(F)$ while, for its complement, $D_<(F) < D(F)$. However, the situation changes for graphs on 4 vertices — $D_<(F)$ becomes smaller even for certain connected patterns F.

4.2. *4-patterns without C_4.* Clearly, for every F on 4 vertices, $D_<(F) \geq 3$ since Duplicator wins the 2-round game on K_4 and K_3. In this section, we prove that $D_<(F) = 3$ for every F that does not contain C_4.

Let us first show that, if F does not contain P_4, then $D_<(F) = 3$.

Let us call two graphs $F, \tilde{F}$ on vertex sets $V(F), V(\tilde{F}) \subset \mathbb{N}$ $(\sim, <)$-*isomorphic*, if there exists a bifection $f : V(F) \to V(\tilde{F})$ such that $u \sim v$ in F if and only if $f(u) \sim f(v)$ in $\tilde{F}$, and $u < v$ in F if and only if $f(u) < f(v)$ in $\tilde{F}$. For a pattern graph F on $[\ell]$, denote $D'_<(F)$ the minimum quantifier depth of a first-order sentence defining the property of containing a subgraph $(\sim, <)$-isomorphic to F.

Claim 3. *Let F be a graph on the vertex set $\{1, 2, 3, 4\}$. If either $3 \nsim 1$, or $1 \nsim 4$, or $4 \nsim 2$, then $D'_<(F) \leq 3$.*

Proof. Let F do not contain the edge $\{1, 3\}$. Then the containment of "ordered" F can be defined by a sentence

$$\exists x_2 \exists x_4 \quad (\exists x_1 \quad [x_1 < x_2] \wedge \phi_{1,2,4}) \quad \wedge \quad (\exists x_3 \quad [x_2 < x_3] \wedge [x_3 < x_4] \wedge \phi_{2,3,4}),$$

where subformulas $\phi_{1,2,4}$ and $\phi_{2,3,4}$ have no quantifiers and define adjacencies between vertices x_1, x_2, x_4 and x_2, x_3, x_4 respectively.

In the same way it can be done for two other pairs. □

From the claim, we immediately get the result for all non-connected patterns and the star.

Corollary 4. *If $P_4 \not\subset F$, then $D_<(F) = 3$.*

Now, let us observe that Claim 3 implies that $D_<(F) = 3$ for 4-path and paw graph F as well.

Theorem 5. *If F is either P_4 or the paw graph (i.e., connected 4-vertex graph having a unique triangle), then $D_<(F) = 3$.*

Proof. We start from $F = P_4$. Let $\mathcal{F}$ be the set of all P_4 on $\{1,2,3,4\}$. Let F_0 be the path 3142. For $F \in \mathcal{F}$, let φ_F be a sentence that defines containment of an "ordered" copy of F (i.e., containment of a subgraph $\tilde{F}$ isomorphic to F with the same order of vertices as in F). From Claim 3, it immediately follows, that for $F \neq F_0$, one may choose a sentence with the quantifier depth 3 on the role of φ_F. Let us show that there exists a sentence φ_0 with the quantifier depth 3 which is tautologically equivalent to $\varphi_{F_0} \wedge \left[\bigwedge_{F\in\mathcal{F}:\, F\neq F_0} \neg\varphi_F\right]$. Indeed, since a desired copy of F_0 should not contain an edge between its second and third vertices (otherwise, there is a copy C_4 and, consequently, a "differently ordered" copy of P_4), conditioning on $\bigwedge_{F\in\mathcal{F}:\, F\neq F_0} \neg\varphi_F$ being true, it remains to say

$$\exists x_1 \exists x_2 \quad (x_1 \sim x_2) \wedge \psi_0(x_1, x_2)$$

and

$$\psi_0(x_1,x_2) = \big(\exists x_3\ [x_2 < x_3] \wedge [x_1 \sim x_3] \wedge [x_2 \nsim x_3]\big) \wedge \big(\exists x_4\ [x_2 < x_4] \wedge [x_1 \sim x_4] \wedge [x_2 \sim x_4]\big). \quad (2)$$

Clearly, the sentence $\varphi_0 \vee \left[\bigvee_{F\in\mathcal{F}:\, F\neq F_0} \varphi_F\right]$ defines the P_4-containment and has the quantifier depth 3.

It remains to prove the same but for a paw graph. Let $\mathcal{F}$ be the set of all paw graphs on $\{1,2,3,4\}$. Let $F_0 \in \mathcal{F}$ have edges $\{3,1\}, \{1,4\}, \{4,2\}, \{1,2\}$, F_0' have edges $\{3,1\}, \{1,4\}, \{4,2\}, \{3,4\}$. As above, for $F \in \mathcal{F}$, let φ_F be a sentence that defines containment of an "ordered" copy of F. From Claim 3, for $F \notin \{F_0, F_0'\}$, one may choose a sentence with the quantifier depth 3 on the role of φ_F. It remains to prove that there exist sentences φ_0 and φ_0' with the quantifier depth 3 that are tautologically equivalent to $\varphi_{F_0} \wedge \left[\bigwedge_{F\in\mathcal{F}:\, F\notin\{F_0,F_0'\}} \neg\varphi_F\right]$ and $\varphi_{F_0'} \wedge \left[\bigwedge_{F\in\mathcal{F}:\, F\notin\{F_0,F_0'\}} \neg\varphi_F\right]$ respectively. Let us prove the existence of the first sentence, the proof for the second one is analogous. The desired sentence is

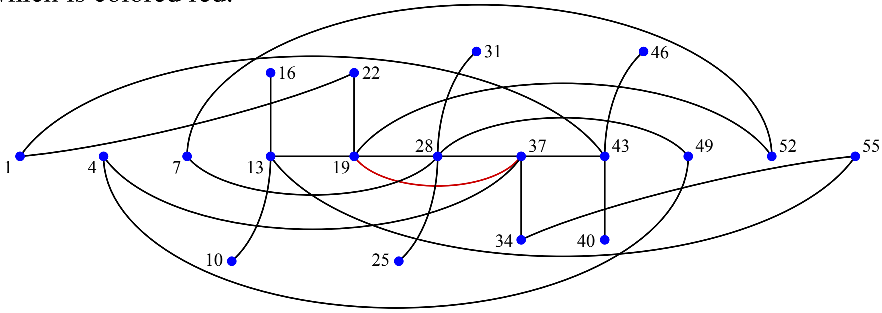
$$\exists x_1 \exists x_2 \quad (x_1 < x_2) \wedge (x_1 \sim x_2) \wedge \psi_0(x_1, x_2),$$

where ψ_0 is defined in (2). Clearly, the sentence $\varphi_0 \vee \varphi_0' \vee \left[\bigvee_{F\in\mathcal{F}:\, F\notin\{F_0,F_0'\}} \varphi_F\right]$ defines the paw-containment and has the quantifier depth 3. □

4.3. *4-patterns with C_4.* In this section, we prove that for all 4-vertex patterns F containing C_4 (i.e., for C_4, the diamond graph $K_4 \setminus e$ and K_4), $D_<(F) = 4$.

Theorem 6. *If $F \supset C_4$ is a 4-vertex graph, then $D_<(F) = 4$.*

Proof. Let us first prove the result for K_4. Consider graphs $G \supset K_4$ and $H \not\supset K_4$ on the vertex set $[55]$. Let V_i be the set of vertices $v \equiv i \mod 3$, $i \in \{0,1,2\}$. All the vertices of V_0 are isolated. All the vertices of V_2 are adjacent to all the vertices of V_1, and there are no adjacencies between the vertices of V_2. The adjacencies between the vertices of V_1 are represented in the figure; G has one more edge than H, $\{19, 37\}$, which is colored red.



By Lemma 2, it is sufficient to show that Duplicator wins the 3-game on G and H. Let us describe the winning strategy. In the first round, Duplicator just copies the Spoiler's move by choosing the same vertex: $x_1 = y_1$.

Assume that, after the Spoiler's choice in the second round, two chosen vertices (in the graph where this choice is made) do not belong to $\{19, 37\}$. Duplicator uses the same strategy as before: $x_2 = y_2$. Clearly, Duplicator wins in the third round by doing the same thing: $x_3 = y_3$. Indeed, if at least one of two vertices of [55] does not belong to $\{19, 37\}$, then the adjacencies between them in G and H are the same.

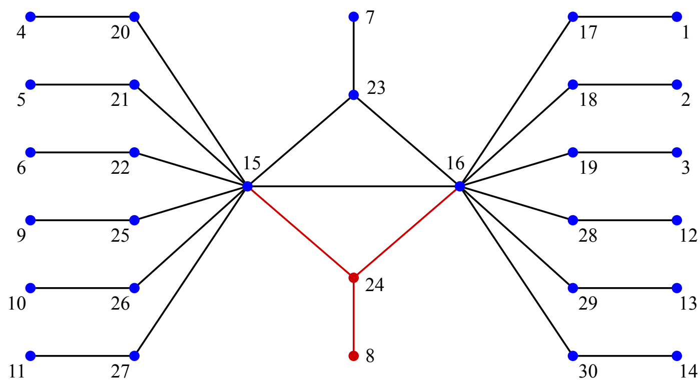
Finally, assume that exactly one of the two vertices chosen by Spoiler belongs to $\{19, 37\}$. Since the graphs are symmetric, we may assume, that this is the vertex 19. Duplicator chooses the same vertex: $x_2 = y_2$.

Clearly, if the vertex chosen by Spoiler in third round does not equal to 37, then Duplicator wins by copying the move of Spoiler. Let Spoiler choose 37. Without loss of generality, $x_2 = 19$.

Assume that Spoiler's move in the third round is in G. If $x_1 \in V_0$, then Duplicator chooses y_3 from V_2 such that, for $j \in \{1, 2\}$, $y_3 > x_j$ if and only if $37 > x_j$. Clearly, there is such a vertex in V_2. If $x_1 \in V_2$, then three situations are possible. If $x_1 < 19$, then Duplicator chooses $y_3 = 28$. If $19 < x_1 < 37$, then $y_3 = 52$. If $x_1 > 37$, then $y_3 = 28$. Assume that $x_1 \in V_1$. If $x_3 \sim x_1$, then Duplicator may find a desired vertex in V_2. If $x_3 \nsim x_1$, then two situations are possible. If $x_1 < 37$, then Duplicator chooses $y_3 = 52$. If $x_1 > 37$, then Duplicator chooses $y_3 = 22$.

It remains to assume that the last Spoiler's move is in H. If $x_1 \in V_0$, then Duplicator chooses y_3 from V_0 such that, for $j \in \{1, 2\}$, $y_3 > x_j$ if and only if $37 > x_j$. If $x_1 \in V_2$, then two situations are possible. If $x_1 < 37$, then Duplicator chooses $y_3 = 43$. If $x_1 > 37$, then $y_3 = 31$. Assume that $x_1 \in V_1$. If $x_3 \nsim x_1$, then Duplicator may find a desired vertex in V_0. If $x_3 \sim x_1$, then four situations are possible. If $x_1 = 4$, then Duplicator chooses 49 If $x_1 = 28$, then $y_3 = 49$. If $x_1 = 34$, then $y_3 = 55$. If $x_1 = 43$, then $y_3 = 40$.

Now, let us switch to C_4 and the diamond graph. Consider the graphs G and H given as follows:



Black edges and blue vertices belong to both graphs, while red vertices (8 and 24) and the edges connected to them belong only to G. Notice that $G \supset K_4 \setminus e \supset C_4$ while $H \not\supset C_4$. By Lemma 2, to finish the proof of Theorem 6, it is sufficient to show that Duplicator wins the 3-game on G and H.

Let us call the vertices that are at most 14 *children*. The vertices adjacent to *children* are called *parents* (if u is a parent of a child v, then $u = v + 16$). Let us call the vertices 23, 24, 7, 8 *central*.

Let us observe the following straightforward properties of G and H.

Claim 7. *If $x_1 = y_1$, $x_2 = y_2$, $x_3 = y_3$, then Duplicator wins.*

Claim 8. *The vertices 23, 24 and 7, 8 in the graph G are indistinguishable with respect to the non-central vertices, i.e., for any non-central vertex u, either both 23 and 24 (or 7 and 8) are adjacent to u, or both are non-adjacent to u, and either both are larger than u, or both are less than u.*

Define $f\colon \mathbb{N} \to \mathbb{N}$ in the following way:

$$f(x) = x, \text{ if } x \notin \{8, 24\}, \quad f(24) = 23, \quad f(8) = 7.$$

From Claim 7 and Claim 8, we get:

Claim 9. *If x, y, z are chosen in graph G, and $f(x), f(y), f(z)$ respectively in graph G and there is not more than one central vertex among x, y, z, then Duplicator wins.*

Since a parent and its child differs by the constant number, and every child is less than any parent, we get the following.

Claim 10. *Assume that (x_1, x_2, x_3), (y_1, y_2, y_3) is a winning configuration for Duplicator such that x_i is a parent if and only if y_i is a parent. If we replace a parent x_i such that its child u_i is not among x_1, x_2, x_3 with u_i, and do the same thing with y_i, then the new configuration is also winning for Duplicator.*

Now, let us change rules of the game in the following way. Spoiler is restricted to choose a parent before its child. If Spoiler chooses a parent, then Duplicator chooses a parent as well. If Spoiler chooses a son (say, x_i) of previously chosen parent (say, x_j), then Duplicator must choose the son of y_j.

According to Claim 10, if Duplicator wins the modified game in 3 rounds, then he wins the original 3-game as well. Below, we describe a winning strategy of Duplicator in the modified game.

In the first round, Spoiler chooses either a vertex x_1 in G, or a vertex y_1 in H. Duplicator's response is $f(x_1)$ (or $f(y_1)$ respectively).

If, after the second move of Spoiler, in the respective graph, both vertices are non-central, then Duplicator chooses $f(x_2)$ (or $f(y_2)$). Then, in the third round, Duplicator chooses the vertex $f(x_3)$ (or $f(y_3)$) and, by Claim 9, wins.

Assume that at least one central vertex is selected. Let in the second round Spoiler choose a vertex v in a graph where a vertex u is already chosen.

(1) If either $v \in \{15, 16\}$, or $u \in \{15, 16\}$, then Duplicator chooses $f(v)$. Without loss of generality, we assume that $v \in \{15, 16\}$. If the last vertex chosen by Spoiler x_3 (y_3) is not central, then Duplicator chooses $f(x_3)$ (resp. $f(y_3)$) and wins by Claim 9. If x_3 (y_3) is the child of $u = x_1$ (resp. $u = y_1$), then Duplicator chooses the child of y_1 (resp. x_1) in the third round and wins. Finally, if Spoiler chooses the second central parent x_3, then Duplicator chooses a neighbor of y_2, which is less than y_1, if $x_3 < x_1$, and a neighbor of y_2, larger than y_1, otherwise.

(2) If v is the child of the vertex chosen in the first round, then Duplicator chooses the respective child in the other graph and then his further winning strategy is obvious by Claim 8.

(3) Finally, assume that both vertices u, v (where u is the vertex chosen in the first round in the graph where Spoiler moves in the second one) are parents, and at least one of them is central. For a parent z, let $z\downarrow$ be the maximum parent less than z (does not exist for 17) and $z\uparrow$ be the minimum parent

larger than z (does not exist for 30). Let us assume that the vertex chosen x_1 be a central parent. Below, without loss of generality, we assume that v is in G. If $x_2 = x_1\uparrow$, then Duplicator chooses $y_2 = y_1\uparrow$. If $x_2 = x_1\downarrow$, then Duplicator chooses $y_2 = y_1\downarrow$. If $x_2 = 17$, then $y_2 = 17$. If $x_2 = 30$, then $y_2 = 30$. Otherwise, Duplicator chooses $y_1\uparrow\uparrow$ (y_1 is less than 30 in this case), if $x_2 > x_1$, and $y_1\downarrow\downarrow$ (y_1 is larger than 17 in this case), if $x_2 < x_1$.

Now, let us assume that u is not central, while v is central. The strategy of Duplicator for the second round is described below. If Spoiler's second move is in G, then the following situations are possible. If $x_1 = 25$, $x_2 = 23$, then $y_2 = 22$. If $x_1 = 25$, $x_2 = 24$, then $y_2 = 23$. If $x_1 = 22$, $x_2 = 23$, then $y_2 = 23$. If $x_1 = 22$, $x_2 = 24$, then $y_2 = 25$. Otherwise, $y_2 = f(x_2)$. If Spoiler's second move is in H, then the following situations are possible. If $y_1 = 25$, $y_2 = 23$, then $x_2 = 24$. If $y_1 = 22$, $y_2 = 23$, then $x_2 = 23$. Otherwise, $x_2 = f(y_2)$.

It remains to prove that, in both cases, Duplicator has a winning move in the last round.

Let us note that, in both graphs, the selected vertices have individual neighbors smaller than any parent, and any neighbor of any parent is less than selected vertices. Any pair of parents, one of which is central, have a common neighbor in both graphs. Also, for any pair of parents $z_1 < z_2$, such that $z_2 \neq z_1\uparrow$, $z_1 \neq 17$, $z_2 \neq 30$, among the vertices non-adjacent to both z_1, z_2 there are vertices of all 3 types: larger than both, smaller than both and between them. There is no common non-neighbor of the first type only if $z_2 = 30$. There is no common non-neighbor of the second type only if $z_1 = 17$. There is no common non-neighbor of the third type only if $z_2 = z_1\uparrow$. The winning strategy of Duplicator follows. □

References

[Alon et al. 1995] N. Alon, R. Yuster, and U. Zwick, "Color-coding", *J. Assoc. Comput. Mach.* **42**:4 (1995), 844–856. MR

[He 2015] Y. He, "k variables are needed to define k-clique in first-order logic", preprint, 2015. arXiv

[Le Gall 2014] F. Le Gall, "Powers of tensors and fast matrix multiplication", pp. 296–303 in *ISSAC 2014—Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ACM, New York, 2014. MR

[Libkin 2004] L. Libkin, *Elements of finite model theory*, Springer, 2004. MR

[Nešetřil and Poljak 1985] J. Nešetřil and S. Poljak, "On the complexity of the subgraph problem", *Comment. Math. Univ. Carolin.* **26**:2 (1985), 415–419. MR

[Rossman 2008] B. Rossman, "On the constant-depth complexity of k-clique", pp. 721–730 in *STOC'08: Proceedings of the 40th Annnual ACM Symposium on Theory of Computing*, ACM, New York, 2008. MR

[Rossman 2016] B. Rossman, "Lower bounds for subgraph isomorphism and consequences in first order logic", Presentation at the Workshop on Symmetry, Logic, Computation at the Simons Institute, Berkeley, November 2016, Available at https://simons.berkeley.edu/talks/benjamin-rossman-11-08-2016.

[Schweikardt 2013] N. Schweikardt, "A short tutorial on order-invariant first-order logic", pp. 112–126 in *Computer science—theory and applications*, Lecture Notes in Comput. Sci. **7913**, Springer, 2013. MR

[Verbitsky and Zhukovskii 2019a] O. Verbitsky and M. Zhukovskii, "On the first-order complexity of induced subgraph isomorphism", *Log. Methods Comput. Sci.* **15**:1 (2019), art. id˙ 25. MR

[Verbitsky and Zhukovskii 2019b] O. Verbitsky and M. Zhukovskii, "Tight bounds on the asymptotic descriptive complexity of subgraph isomorphism", *ACM Trans. Comput. Log.* **20**:2 (2019), art. id. 9. MR

[Verbitsky and Zhukovskii 2019c] O. Verbitsky and M. Zhukovskii, "The descriptive complexity of subgraph isomorphism without numerics", *Theory Comput. Syst.* **63**:4 (2019), 902–921. MR

Received 3 Dec 2019. Revised 6 Feb 2020.

OLEG GRIGORYAN:

olegrg@gmail.com
Higher School of Economics, Moscow, Russia

MIKHAIL MAKAROV:

vbif-98@mail.ru
Moscow Institute of Physics and Technology, Dolgoprudny, Russia

MAKSIM ZHUKOVSKII:

zhukmax@gmail.com
Laboratory of Advanced Combinatorics and Network Applications, Moscow Institute of Physics and Technology, Dolgoprudny, Russia

msp

msp

Moscow Journal of Combinatorics and Number Theory
Vol. 9, No. 3, 2020
https://doi.org/10.2140/moscow.2020.9.303

On limit points of spectra of first-order sentences with quantifier depth 4

Yury Yarovikov

We study the asymptotic behavior of probabilities of first-order properties of sparse binomial random graphs. We consider properties with quantifier depth not more than 4. It is known that the only possible limit points of the spectrum (i.e., the set of all positive α such that $G(n, n^{-\alpha})$ does not obey the zero-one law with respect to the property) of such a property are 1/2 and 3/5. We prove that 1/2 is not a limit point of the spectrum.

1. Introduction

In this paper, we study the asymptotic behavior of probabilities of properties of Erdös-Rényi random graphs $G(N, p)$ that are expressible as sentences in the first-order logic. Here, p is typically a function $p = p(N)$. We consider first-order sentences of quantifier depth no more than k and sparse settings: $p = N^{-\alpha}$, where α is some positive number less than 1.

Recall the definition of the Erdös-Rényi random graph: $G(N, p)$ is a random element of the set of all simple graphs $\Omega_N = \{\mathcal{G} = (\mathcal{V}_N, \mathcal{E})\}$ with the set of vertices $\mathcal{V}_N = \{1, 2, \ldots, N\}$, where N is a positive integer, with the distribution

$$\mathsf{P}_{N,p}(\mathcal{G}) = p^{|E(\mathcal{G})|}(1-p)^{\binom{N}{2}-|E(\mathcal{G})|},$$

where $0 \leq p \leq 1$.

We now define the class of graph first-order properties [Libkin 2004] as follows. A graph first-order property is a property defined by a first-order sentence in the vocabulary $\{=, \sim\}$ consisting of two predicate symbols, "$=$" expressing coincidence of vertices and "$\sim$" expressing adjacency.

Furthermore, we define the quantifier depth of a first-order property as the minimal quantifier depth of a first-order formula [Libkin 2004] expressing that property.

Definition 1. We say that for a function $p = p(N)$, *the zero-one law for first order logic holds* if for each first-order property the probability that the random graph $G(N, p)$ has that property tends to either 0 or 1 as $N \to \infty$.

Theorem 2 [Shelah and Spencer 1988]. *Let $p = N^{-\alpha}$ and α be a positive irrational number. Then the zero-one law for first-order logic holds.*

Note that for $p = N^{-\alpha}$ with α rational and $\alpha \leq 1$, the zero-one law for first-order logic does not hold, as shown in the same paper.

This work is supported by the Russian Foundation for Basic Research (grant 20-31-70025).
MSC2010: 05C80.
Keywords: random graphs, first-order logic, zero-one law.

Definition 3. Let k be a positive integer. For $p = p(N)$, *the zero-one k-law holds* if for each first-order property with quantifier depth no more than k, the probability that the random graph $G(N, p)$ has that property tends to either 0 or 1 as $N \to \infty$.

The obvious corollary of Theorem 2 is that for functions $p = N^{-\alpha}$, the zero-one k-law also holds if α is irrational. For rational α, the situation gets complicated: the zero-one k-law may either hold or not hold, as demonstrated in the following theorem.

Theorem 4 [Zhukovskii 2012]. *Let $p(N) = N^{-\alpha}$ and $\alpha \in \left(0, \frac{1}{k-2}\right)$. Then the zero-one k-law holds. Moreover, if $\alpha = \frac{1}{k-2}$ then the zero-one k-law does not hold.*

Definition 5. *A spectrum of* $k \in \mathbb{N}$ (or, simply, a k-spectrum) is the set of all $\alpha \in (0, 1)$ such that for $p(N) = N^{-\alpha}$ the zero-one k-law does not hold.

In other words, Theorem 4 states that the minimal element of the k-spectrum when $k \geq 3$ equals $\frac{1}{k-2}$.

In [Spencer 2001], it was proven that the spectrum of 14 is infinite. Further, it was proven in [Zhukovskii 2016] that $\frac{1}{2}$ is a limit point of the 5-spectrum. Moreover, it is known that the 3-spectrum is finite [Zhukovskii 2016]. In this paper we will study possible limit points of the 4-spectrum, the existence of which is neither proven nor disproven.

Theorem 6 [Matushkin and Zhukovskii 2018]. *The only limit points of the* 4*-spectrum may be $\frac{1}{2}$ and $\frac{3}{5}$.*

Our main result is stated below.

Theorem 7. *The point $\frac{1}{2}$ is not a limit point of the* 4*-spectrum.*

Thus, the only possible limit point of the 4-spectrum is $\frac{3}{5}$.

In the next section, we recall some known constructions needed to prove Theorem 7. In Sections 3 and 4, we introduce some new tools specific for α close to $\frac{1}{2}$; and finally, we prove the theorem in Section 5.

2. Known definitions and statements

We begin with recalling the Ehrenfeucht game $\mathrm{EHR}(A, B, k)$ on graphs A, B with two players: Spoiler and Duplicator, and a fixed number of rounds k [Ehrenfeucht 1961]. Each round of the game constitutes a move by Spoiler followed by a move by Duplicator: Spoiler selects a vertex from *either* of the two graphs A and B, whereas Duplicator selects a vertex from the *other* graph. Let x_ν denote the vertex selected from A in round ν, and y_ν that from B in round ν, for $1 \leq \nu \leq k$.

Duplicator wins the game if for each $s, t \in \{1, \ldots, k\}$, the following hold:

- $x_s = x_t$ if and only if $y_s = y_t$;
- $x_s \sim x_t$ if and only if $y_s \sim y_t$.

Otherwise, Spoiler wins.

The following theorem holds (see, for example, [Zhukovskii 2012]).

Theorem 8. *For $p = p(N)$ the zero-one k-law holds if and only if*

$$\lim_{N,M\to\infty} \mathsf{P}\big(\textit{Duplicator has a winning strategy in } \mathrm{EHR}(A, B, k)\big) = 1,$$

where A is a random graph distributed as $G(N, p(N))$, B is a random graph distributed as $G(M, p(M))$, and the two are independent of each other.

Definition 9. A graph property L is called *increasing (decreasing)* if for each two graphs G and G', with $V(G) = V(G')$ and $E(G) \subset (\supset) E(G')$, from $G \in L$ it follows that $G' \in L$.

Definition 10. A function $p_0(N)$ is called *the threshold function* for a property L if the following holds: If $p = o(p_0)$ (i.e., $\frac{p(N)}{p_0(N)} \to 0$, $N \to \infty$), then $\mathsf{P}(G(N, p) \in L) \to 0$ as $N \to \infty$; and if $p_0 = o(p)$, then $\mathsf{P}(G(N, p) \in L) \to 1$ as $N \to \infty$ (here, we restrict ourselves with increasing properties only; for decreasing properties, the threshold function is defined in the same way, but the o-conditions should be replaced; see, for example, [Janson et al. 2000]).

Define the *density of a graph* G as $\rho(G) = \frac{e(G)}{v(G)}$. Let us recall the result determining the threshold function for a property to contain an arbitrary given graph G as a subgraph. Define the *maximal density*: $\rho^{\max}(G) = \max\limits_{H \subseteq G} \rho(H)$.

Theorem 11 [Ruciński and Vince 1985]. *The function $p = N^{-1/\rho^{\max}(G)}$ is the threshold function for the property of containing a copy of G.*

Consider graphs $G \supseteq H$, $\tilde{G} \supseteq \tilde{H}$ and positive integers k, l $(k \le l)$ such that

$$V(H) = \{x_1, \ldots, x_k\}, \quad V(G) = \{x_1, \ldots, x_l\},$$
$$V(\tilde{H}) = \{\tilde{x}_1, \ldots, \tilde{x}_k\}, \quad V(\tilde{G}) = \{\tilde{x}_1, \ldots, \tilde{x}_l\}.$$

The graph $\tilde{G}$ is a (G, H)-*extension* of the graph $\tilde{H}$ if

$$(x_i, x_j) \in E(G) \setminus E(H) \Rightarrow (\tilde{x}_i, \tilde{x}_j) \in E(\tilde{G}) \setminus E(\tilde{H}).$$

If the relation

$$(x_i, x_j) \in E(G) \setminus E(H) \Leftrightarrow (\tilde{x}_i, \tilde{x}_j) \in E(\tilde{G}) \setminus E(\tilde{H})$$

holds then $\tilde{G}$ is called a *strict extension* of $\tilde{H}$, and the pairs (G, H) and $(\tilde{G}, \tilde{H})$ are called *isomorphic*.

Note that, throughout the paper, the labeling of vertices of graphs G, H, $\tilde{G}$, and $\tilde{H}$ will follow from the context. Thus, there will be no ambiguity in the definition of isomorphic pairs.

Fix a positive α. For graphs G and H with $H \subset G$, define

$$V(G, H) = V(G) \setminus V(H), \quad E(G, H) = E(G) \setminus E(H),$$
$$v(G, H) = |V(G, H)|, \quad e(G, H) = |E(G, H)|,$$
$$f_\alpha(G, H) = v(G, H) - \alpha e(G, H).$$

If for each graph S such that $H \subset S \subseteq G$, the inequality $f_\alpha(S, H) > 0$ holds, then the pair (G, H) is called α-*safe*. If for each graph S such that $H \subseteq S \subset G$ the inequality $f_\alpha(G, S) < 0$ holds, then the pair (G, H) is called α-*rigid*.

If for each graph S such that $H \subset S \subset G$ the inequality $f_\alpha(S, H) > 0$ holds, but also $f_\alpha(G, H) = 0$; then the pair (G, H) is called α-*neutral*. Note that if (G, H) is α-*neutral*, then for each graph S such that $H \subset S \subset G$, the relation $f_\alpha(G, S) < 0$ holds.

If the pair (G, H) where $H \subset G$ is α-safe, α-rigid or α-neutral, we call G, respectively, an *α-safe, α-rigid, or α-neutral extension* of H.

Let $\tilde{H} \subset \tilde{G} \subset \Gamma$ and $T \subset K$, with $v(T) \le v(\tilde{G})$. The pair $(\tilde{G}, \tilde{H})$ is called (K, T)-*maximal* in Γ, if each subgraph $\tilde{T} \subset \tilde{G}$, with $v(T) = v(\tilde{T})$ and $\tilde{T} \cap \tilde{H} \ne \tilde{T}$, does not have a (K, T)-extension $\tilde{K}$ in $\Gamma \setminus (\tilde{G} \setminus \tilde{T})$ having no edges between $V(\tilde{K}, \tilde{T})$ and $V(\tilde{G}, \tilde{T})$ [Zhukovskii and Raigorodskii 2015].

We will frequently use the following theorem (we write "a.a.s." meaning "asymptotically almost surely"):

Theorem 12 (J. Spencer, S. Shelah, 1988, [Alon and Spencer 1992]). *Let a pair (G, H) with $H \subset G$, be α-safe, and let t be a fixed positive integer. Let $r = v(H)$. Then a.a.s. (as $N \to \infty$) each induced subgraph of $G(N, N^{-\alpha})$ on r vertices has a strict (G, H)-extension that is (K, T)-maximal for each α-rigid pair (K, T) with $v(K, T) \le t$.*

3. New statements and constructions

Hereafter, for a graph G and a subset of its vertices $V \subseteq V(G)$, $G|_V$ denotes the subgraph of G induced on V.

To prove the main theorem, we need to build a finite set $\mathcal{G}$ of "bad" extensions for the following purpose. Let $x_1, \ldots, x_k \in V(A)$ and $y_1, \ldots, y_k \in V(B)$ be the vertices chosen in the game EHR$(A, B, 4)$ after some round k, for $1 \le k \le 4$. Consider two isomorphic extensions K and $\tilde{K}$ of $A|_{\{x_1,\ldots,x_k\}}$ and $B|_{\{y_1,\ldots,y_k\}}$, respectively, such that K and $\tilde{K}$ have a copy in $\mathcal{G}$. Playing as Duplicator, we have to ensure that K and $\tilde{K}$ have similar extensions of a special type; that is, constructed from the (K^*, T^*)-extension, where (K^*, T^*) is a pair of graphs with $v(T^*) = 2$, $e(T^*) = 0$, $v(K^*) = 3$, the only vertex in $V(K^*, T^*)$ being adjacent to each vertex from $V(T^*)$ (a "tick" extension).

To build the aforementioned set of "bad" extensions, we will repeatedly "extend" graphs in the following way. Let Ω be a graph, and let U be its induced subgraph (on an arbitrary subset of its vertices). Let also (A, B), $B \subset A$ be a pair of graphs such that $v(U) = v(B)$. We say that an algorithm *constructs* a strict (A, B)-extension W of U if it performs the following: to Ω it adds $v(A, B)$ vertices (those from the set $V(W) \setminus V(U)$) and $e(A, B)$ edges, so that the pairs (A, B) and (W, U) are isomorphic.

3.1. ***Building the set $\mathcal{G}$.*** Since we are only interested in isomorphism classes of graphs, it will be convenient to assume that all graphs in $\mathcal{G}$ contain a common singleton subgraph H with one vertex: $V(H) = \{z\}$.

Let T be some graph such that $H \subset T$. We call a pair of graphs (K, T), $T \subset K$, *bad* if the following conditions hold:

(1) (K, T) is $\frac{1}{2}$-rigid;

(2) $v(K, T) \le 6$;

(3) $v(T) \ge 2$;

(4) There is at least one edge between $V(K) \setminus V(T)$ and $V(T) \setminus \{z\}$ in the graph K.

Note that we bound $v(K, T)$ by 6, since we are only interested in relatively small extensions while playing the Ehrenfeucht game with 4 steps. The choice of the bound will be clarified in the proof of the main theorem.

Consider a set $\mathcal{G}$ of graphs with the common subgraph H and the following properties:

1. If a pair (G, H) is $\frac{1}{2}$-neutral or $\frac{1}{2}$-rigid and $v(G, H) \le 6$, then there is a graph isomorphic to G, in $\mathcal{G}$ (the image of z is z itself).

2. If $G \in \mathcal{G}$ and a graph $G' \supset G$ is such that the pair (G', G) is bad, then either G' has maximal density at least 2 or there is a graph isomorphic to G' in $\mathcal{G}$ (the image of z is z itself).

3. If $G \in \mathcal{G}$, then the graph obtained by adding some edge into G either has maximum density at least 2 or has an isomorphic graph in $\mathcal{G}$ (the image of z is z itself).

These properties are partially motivated by the fact that maximal density of each graph in $\mathcal{G}$ should be less than 2.

Statement 13. There exists a finite set $\mathcal{G}$ that satisfies 1–3.

Proof. We present an algorithm that constructs such a set $\mathcal{G}$.

(1) We begin with $\mathcal{G}$ that consists of the graphs isomorphic to all (nonempty) $\frac{1}{2}$-neutral extensions G of graph H for which $v(G, H) \leq 6$.

(2) While possible, we choose an arbitrary graph $G \in \mathcal{G}$ and construct $G' \supset G$ such that $\rho^{\max}(G') < 2$ and (G', G) is bad; if $\mathcal{G}$ does not contain a graph isomorphic to G', then G' is added to $\mathcal{G}$.

(3) While possible, we choose an arbitrary graph $G \in \mathcal{G}$ and construct G' such that $V(G) = V(G')$, $E(G) \subset E(G')$, $\rho^{\max}(G') < 2$; if $\mathcal{G}$ does not contain a graph isomorphic to G', then G' is added to $\mathcal{G}$.

We can assume that all the operations from Step 3 are performed after all the operations from Step 2, as the transposition of such operations does not change the resulting graph.

Call $G \in \mathcal{G}$ a 0-*stage graph* if G is added to $\mathcal{G}$ as a result of Step 1 of the algorithm. Call $G \in \mathcal{G}$ an s-*stage graph*, where $s \in \mathbb{N}$, if G is obtained as a result of applying the operation from Step 2 to an $(s-1)$-stage graph. Call a graph $G \in \mathcal{G}$ a *modified s-stage graph* if G is obtained from an s-stage graph by Step 3 of the algorithm.

Let us prove that the algorithm terminates and constructs a finite set. It is sufficient to show that Step 2 of the algorithm terminates, as in Step 3 for each graph in $\mathcal{G}$ we only add a finite number of new graphs.

Consider the value $f_{1/2}(G) = v(G) - \frac{1}{2}e(G)$. Firstly, for each graph G this value is half-integer. Secondly, it is easy to show that for a 0-stage graph G the equality $f_{1/2}(G) = 1$ holds. Finally, if G' is an s-stage graph with $s \in \mathbb{N}$ then it is isomorphic to some $\frac{1}{2}$-rigid (K, T)-extension of some $(s-1)$-stage graph G. So,

$$f_{1/2}(G') = f_{1/2}(G) + f_{1/2}(K, T) \leq f_{1/2}(G) - \tfrac{1}{2}.$$

Thus, if $s \geq 2$ we have

$$f_{1/2}(G') \leq 0,$$

which implies $\rho^{\max}(G') \geq 2$. Consequently, in Step 2, only 1-stage graphs may be obtained. Thus, s-stage graphs with $s > 1$ are not in $\mathcal{G}$. Thus, $\mathcal{G}$ is finite.

If G' is a modified 1-stage graph we have $f_{1/2}(G') \leq \frac{1}{2} - \frac{1}{2} = 0$. Thus, there are also no modified s-stage graphs in $\mathcal{G}$, with $s > 0$.

Let us verify that the set $\mathcal{G}$ that we built satisfies all the three properties. Properties 1 and 3 are evident by construction. The second property may only be disrupted if in Step 3 of the algorithm there appears a graph G that has a $\frac{1}{2}$-rigid extension $G' \notin \mathcal{G}$, with $\rho^{\max}(G') < 2$. Let $\tilde{G} \in \mathcal{G}$ be the graph from which G is obtained in Step 3. Consider $\tilde{G}'$ that is a (G', G)-extension of $\tilde{G}$. Obviously, $E(\tilde{G}') \subset E(G')$. Therefore,

$\rho^{\max}(\tilde{G}') \leq \rho^{\max}(G') < 2$. Hence, $\tilde{G}'$ has been added in Step 2 of the algorithm. Thus, in Step 3 we add G', which leads to a contradiction.

The set $\mathcal{G}$ is successfully constructed. □

Remark 14. From the proof we obtain the equality

$$f_{1/2}(G) = 1 - \frac{s}{2},$$

for each s-stage $G \in \mathcal{G}$.

We now list some additional properties of $\mathcal{G}$ that will help prove the main theorem.

4. For each $G \in \mathcal{G}$ we have $\rho^{\max}(G) < 2$

Proof. Evident by construction. □

5. The degree of each vertex, except possibly for z, in any graph $G \in \mathcal{G}$ is greater than 2, and the degree of z is no less than 1.

Proof. Let us prove that the degree of each vertex except z is greater than 2. It is sufficient to verify the property for graphs of stages 0 and 1. Let some vertex $w \neq z$ from $V(G)$ have a degree no more than 2.

If G is a 0-stage graph then $f_{1/2}(G, G - w) \geq 0$, which contradicts (G, H) being $\frac{1}{2}$-neutral: indeed, it is evident that $G - w \neq H$, for each graph of $\mathcal{G}$ has at least 3 vertices.

If G is a 1-stage graph, let W be a subgraph of G from which G is obtained at Step 2 of the algorithm constructing $\mathcal{G}$. If $w \in V(W)$, a contradiction follows from the fact that W is a 0-stage graph. If $w \in V(G) \setminus V(W)$ the inequality $f_{1/2}(G, G - w) \geq 0$ contradicts the pair (G, W) being $\frac{1}{2}$-rigid.

Let us prove that the degree of z is no less than 1. It is sufficient to verify the property for 0-stage graphs. Assume the contrary. If the degree of z is less than 1 the $\frac{1}{2}$-neutrality of (G, H) implies that

$$2v\big(G|_{V(G)\setminus\{z\}}\big) = e(G, H) = e\big(G|_{V(G)\setminus\{z\}}\big).$$

Therefore, $\rho^{\max}(G) \geq 2$. We come to a contradiction, which proves the statement. □

6. Since there are only 0, 1 and modified 0-stage graphs in $\mathcal{G}$, each one of them has no more than 21 vertices.

7. If a graph $G \in \mathcal{G}$ is not a 0-stage graph, the pair (G, H) is $\frac{1}{2}$-rigid.

Proof. If $G \in \mathcal{G}$ is a modified 0-stage graph then the $\frac{1}{2}$-rigidness follows from the $\frac{1}{2}$-neutralness of the pair (G', H), where G' is a 0-stage graph from which G is obtained by adding edges.

If G is a 1-stage graph then it is a (K, T)-extension of a 0-stage graph G', where (K, T) is a $smash[b]\frac{1}{2}$-rigid pair. Consider an arbitrary graph S such that $H \subseteq S \subset G$. Because of the $\frac{1}{2}$-rigidness of the pair (G, G') and the $\frac{1}{2}$-neutralness of the pair (G', H), we have

$$\begin{aligned} e(G, S) &= e\big(G|_{V(S)\cup V(G')}, S\big) + e\big(G, G|_{V(S)\cup V(G')}\big) \\ &\geq e(G', S \cap G') + \big(G, G|_{V(S)\cup V(G')}\big) \\ &\geq 2\big(v(G') - v(S \cap G') + v(G) - v\big(G, G|_{V(S)\cup V(G')}\big)\big) \\ &= 2(v(G) - v(S)). \end{aligned}$$

Let us show that the last inequality is in fact strict. If it turns into equality then $e(G', S \cap G') = 2(v(G') - v(S \cap G'))$ and $e(G, G|_{V(S) \cup V(G')}) = 2(v(G) - v(G, G|_{V(S) \cup V(G')}))$. The first inequality implies that $S \cap G' = H$. The second one implies that $V(S) \cup V(G') = V(G)$. In other words, $V(S) = V(G) \setminus (V(G') \setminus V(H))$. But, in that case $e(G|_{V(S) \cup V(G')}, S) > e(G', S \cap G')$, because the pair (K, T) contains a bad pair and, therefore, between $V(S) \setminus V(H)$ and $V(G') \setminus V(S)$ there is at least one edge.

Thus, the last inequality is strict, which precisely means that (G, H) is $\frac{1}{2}$-rigid. □

8. Each graph $G \in \mathcal{G}$ is connected and remains connected after removing the vertex z.

Proof. Let us prove that $G - z$ is connected. The remaining part of the statement follows from the fact that the degree of z in G is at least 1.

We now verify the validity of this property for a 0-stage graph G. Assume the contrary: let $G - z$ be disconnected. Then there exist induced subgraphs $T_1, T_2 \subset G$, intersecting only in the vertex z, such that in G there are no edges between $V(T_1) \setminus \{z\}$ and $V(T_2) \setminus \{z\}$, with $v(T_1) > 1$, $v(T_2) > 1$, and $V(T_1) \cup V(T_2) = V(G)$. We then obtain the equality $0 = f_{1/2}(G, H) = f_{1/2}(T_1, H) + f_{1/2}(T_2, H)$. Therefore, there exists $h \in \{1, 2\}$ such that $f_{1/2}(T_h, H) \leq 0$, which contradicts the $\frac{1}{2}$-neutralness of (G, H). For a modified 0-stage graph the validity of the property follows from its validity for 0-stage graphs.

We now verify the validity of this property for a 1-stage graph G'. Let G be a 0-stage graph from which G' is obtained, for which (G', G) is a bad pair. When removing z, all vertices of G remain in a single connected component $S \subset G'$. Put $T_1 = G'|_{V(S) \cup \{z\}}$ and $T_2 = G'|_{V(G') \setminus V(S)}$. Assume the contrary: let T_2 be nonempty. Since the pair (G', G) is bad, the graph G' has got at least one edge between $V(G') \setminus V(G)$ and $V(G) \setminus \{z\}$. Therefore, $T_1 \neq G$ and $V(G) \cup V(T_2) \neq V(G')$. Then, because of the $\frac{1}{2}$-rigidness of the pair (G', G), we have $f_{1/2}(T_1, G) = f_{1/2}(G', G'|_{V(G) \cup V(T_2)}) < 0$. Besides, $f_{1/2}(G', T_1) < 0$. Then we have the inequality

$$\rho(G') = \frac{e(G) + e(T_1, G) + e(G', T_1)}{v(G) + v(T_1, G) + v(G', T_1)} \geq \frac{2(v(G) - 1) + 2v(T_1, G) + 1 + 2v(G', T_1) + 1}{v(G) + v(T_1, G) + v(G', T_1)} = 2,$$

which contradicts the inclusion $G' \in \mathcal{G}$. Thus, for 1-stage graphs the property is also proven. There are no more graphs in $\mathcal{G}$, which completes the proof. □

Remark 15. In order to avoid long notations and explanations, we will frequently write $G \in \mathcal{G}$ to mean that $\mathcal{G}$ contains a subgraph isomorphic to G.

3.2. ***Occurrence of graphs of $\mathcal{G}$ in an arbitrary graph.*** Consider some graph Γ and a vertex $u \in V(\Gamma)$. We call a strict (G, H)-extension R of u *$\mathcal{G}$-maximal* if for each $G' \in \mathcal{G}$ there is no (G', H)-extension of $\Gamma|_u$ in Γ that strictly contains all vertices of R.

We call an induced subgraph $U \subset \Gamma$ *u-bad* if it is a $\mathcal{G}$-maximal (G, H)-extension of $\Gamma|_u$ for some $G \in \mathcal{G}$.

Statement 16. Let graph Γ not contain any subgraphs with maximal density no less than 2 with 41 or less vertices. Then for each vertex $u \in V(\Gamma)$ any two u-bad subgraphs of Γ only intersect in u.

Proof. Assume the contrary. Let $U, W \subseteq \Gamma$ be u-bad subgraphs with $v(U \cap W) > 1$. We can assume that either of U and W is a 0-stage or a 1-stage graph (neither $0'$-stage nor $1'$-stage).

Let U be an s-stage graph. Since $U \in \mathcal{G}$, there exists a sequence of nested graphs U_i, $0 \leq i \leq s + 1$, $U_0 = H$, $U_{s+1} = U$ such that (U_{i+1}, U_i) is bad for all $i \geq 1$, and the pair (U_1, U_0) is $\frac{1}{2}$-neutral or $\frac{1}{2}$-rigid with $v(U_1, U_0) \leq 6$. But then the pair $(\Gamma|_{V(U_1)\cup V(W)}, W)$ is bad due to the $\frac{1}{2}$-neutralness of the pair (U_1, U_0). Hence, either the graph $\Gamma|_{V(U_1)\cup V(W)}$ is in $\mathcal{G}$, or $\rho^{\max}(\Gamma|_{V(U_1)\cup V(W)}) \geq 2$, which contradicts the hypothesis of Statement 16. Hence, $\Gamma|_{V(U_1)\cup V(W)} \in \mathcal{G}$.

Moreover, assume that $\Gamma|_{V(U_r)\cup V(W)}$ has a copy in $\mathcal{G}$ for some r. The pair $(\Gamma_{V(U_{r+1}\cup V(W))}, \Gamma_{V(U_r\cup V(W))})$ is bad. Thus, either $\Gamma|_{V(U_{r+1})\cup V(W)}$ is in $\mathcal{G}$, or $\rho^{\max}(\Gamma|_{V(U_{r+1})\cup V(W)}) \geq 2$, which contradicts the hypothesis of Statement 16. Hence, $\Gamma|_{V(U_{r+1})\cup V(W)}$ is in $\mathcal{G}$.

With $s+1=r$, we obtain that $\Gamma_{V(U\cup W)}$ has a copy in $\mathcal{G}$. Therefore, U is not H-maximal, which contradicts the condition. The statement is proven. □

Enumerate all graphs in $\mathcal{G}$: $\mathcal{G} = \{G_1, \ldots, G_K\}$. Let

$$V(G_i) = \{z = z_1^i, z_2^i, \ldots, z_{v(G_i)}^i\}.$$

Let $\mathcal{U}_i(u)$ be the set of all u-bad (G_i, H)-extensions of the vertex u in Γ.

Let the 0-*neighborhood of the vertex* u be the set

$$U_0(u) = V(\Gamma) \setminus \Big(\bigcup_{i=1}^{K} \bigcup_{U \in \mathcal{U}_i(u)} V(U) \Big).$$

3.3. *(K*, T*)-neighborhood.* Let U be an induced subgraph of Γ. We now define the (K^*, T^*)-*neighborhood* of U. Let

- $W_0 = V(U)$;
- $W_{i+1} = W_i \cup \big\{ v \in V(\Gamma) : |\{w \in W_i : w \sim v\}| \geq 2 \big\}, \quad i \in \mathbb{N}$;
- $W = \bigcup_{i=0}^{\infty} W_i$.

We now define $\Gamma|_W$ to be the (K^*, T^*)-neighborhood of U in Γ.

If for all i the sign "≥" in item 2 is replaced by "=" and $W_{i+1} \setminus W_i$ is an independent set, then the set W is called the *strict* (K^*, T^*)-*neighborhood* of U in Γ.

We now prove an auxiliary statement in order to motivate the definition of (K^*, T^*)-neighborhood.

Statement 17. Let α be a positive rational. Consider graphs G, U, W with $W \subset U \subseteq G$ such that the pair (G, U) is α-neutral and (U, W) is α-safe. Let there also be at least one edge between $V(G) \setminus V(U)$ and $V(U) \setminus V(W)$ in G. Then the pair (G, W) is also α-safe.

Proof. Consider an arbitrary graph G' such that $W \subset G' \subseteq G$. Put $S = G' \cap U$. Due to the fact that (G, U) is α-neutral and (U, W) is α-safe, one can show a chain of inequalities

$$\begin{aligned} e(G', W) = e(S, W) + e(G', S) &\leq \frac{v(S) - v(W)}{\alpha} + e\big(G|_{V(U)\cup V(G')}, U\big) \\ &\leq \frac{v(S) - v(W)}{\alpha} + \frac{v(G') - v(S)}{\alpha} = \frac{v(G') - v(W)}{\alpha}. \end{aligned}$$

Note that $e(S, W) \leq \dfrac{v(S) - v(W)}{\alpha}$ turns into an equality if and only if $S = W$. The inequality

$$e\big(G|_{V(U)\cup V(G')}, U\big) \leq \frac{v(G') - v(S)}{\alpha}$$

turns into an equality if and only if $V(G') \subseteq V(U)$ or $V(U) \cup V(G') = V(G)$.

Assume both inequalities are equalities. Then we have $W = G' \cap U$. If $V(G') \subseteq V(U)$, then $G' = W$, which contradicts the fact that W is a proper subgraph of G'. If, on the other hand, $V(U) \cup V(G') = V(G)$, then $e(G', W) = \frac{v(G')-v(W)}{\alpha} = \frac{v(G)-v(U)}{\alpha} = e(G, U)$ (The first inequality in this chain easily follows from the fact that the previous inequalities turn into equalities). Therefore, there are no edges between the sets $V(G) \setminus V(U)$ and $V(U) \setminus V(W)$ in G, which contradicts the assumption. Hence, the inequality is in fact strict, which completes the proof. □

Corollary 18. *Let α be a positive rational and U be an induced subgraph of G with $\rho^{\max}(U) < \frac{1}{\alpha}$. Let the pair (G, U) be α-neutral. Let there also be at least one edge between the sets $V(U)$ and $V(G) \setminus V(U)$, in G. Then the inequality $\rho^{\max}(G) < \frac{1}{\alpha}$ holds.*

Proof. We obtain the corollary from Statement 17 by putting W to be the empty graph. □

We now formulate some corollaries from Statement 17 that motivate the definition of (K^*, T^*)-neighborhood.

Statement 19. Let Γ be an arbitrary graph. Let U, W be some induced subgraphs of Γ such that $W \subset U$ and (U, W) is $\frac{1}{2}$-safe. Let $G \subset \Gamma$ be the strict (K^*, T^*)-neighborhood of U in Ω, and let also each vertex from the set $V(G) \setminus V(U)$ be adjacent to not more than one vertex from $V(W)$ in Ω. Then the pair (G, W) is also $\frac{1}{2}$-safe.

Proof. Let $U_0 \subset U_1 \subset \ldots \subset U_s$ be a sequence of nested graphs such that $U_0 = U$, $U_s = G$, and for $1 \le i \le s$ the equalities $v(U_i, U_{i-1}) = 1$ hold and $e(U_i, U_{i-1}) = 2$. The pairs (U_i, U_{i-1}) are $\frac{1}{2}$-neutral for all i. Moreover, there are exactly two edges between $V(U_i) \setminus V(U_{i-1})$ and $V(U_{i-1})$ and no more than one edge between $V(U_i) \setminus V(U_{i-1})$ and $V(W)$. Hence, there is at least one edge between the sets $V(U_i) \setminus V(U_{i-1})$ and $V(U_{i-1}) \setminus V(W)$, in U_i.

Let us prove by induction on i that (U_i, W) is $\frac{1}{2}$-safe. The base case when $i = 0$ directly follows from the statement condition. The step case from $i - 1$ to i follows from the fact that (U_{i-1}, W) is $\frac{1}{2}$-safe, (U_i, U_{i-1}) is $\frac{1}{2}$-neutral, and also Statement 17 for $\alpha = \frac{1}{2}$. Put $i = s$, and obtain the desired inequality. □

Corollary 20. *Let Γ be an arbitrary graph. Let U be an induced subgraph of Γ for which the inequality $\rho^{\max}(U) < 2$ holds. If G is the strict (K^*, T^*)-neighborhood of U in Γ, then the inequality $\rho^{\max}(G) < 2$ holds.*

Proof. We obtain the corollary from Statement 19 by putting W to be the empty graph. □

3.4. ***Some auxiliary notations.*** Let $u_1, \ldots, u_m, v_1, \ldots, v_r$ be some vertices in a graph Γ, and put $U = \Gamma|_{\{u_1,\ldots,u_m\}}$. For a tuple $(\alpha_1, \ldots, \alpha_r) \in \{0, 1\}^r$, put

$$\mathcal{N}\big(v_1^{\alpha_1}, \ldots, v_r^{\alpha_r}\big) = \big\{u \in V(\Gamma) \mid \forall k\, (v_k \sim u \leftrightarrow \alpha_k = 1)\big\} \setminus \{v_1, \ldots, v_r\}.$$

Also, in place of v_i^0 we write $\neg v_i$, and in place of v_i^1 we write just v_i. For example,

$$\mathcal{N}(\neg v_1, v_2, \neg v_3) = \big\{u \in V(\Gamma) \mid u \nsim v_1, u \sim v_2, u \nsim v_3, u \notin v_1, v_2, v_3\big\}.$$

Put

$$\delta\big(v_1^{\alpha_1}, \ldots, v_r^{\alpha_r}\big) = \big(\big|\mathcal{N}\big(v_1^{\alpha_1}, \ldots, v_r^{\alpha_r}\big)\big| \ge 1\big).$$

Finally, put

$$\mathcal{N}^U\big(v_1^{\alpha_1},\dots,v_m^{\alpha_m}\big)=\mathcal{N}\big(v_1^{\alpha_1},\dots,v_m^{\alpha_m}\big)\setminus V(U);$$
$$\delta^U\big(v_1^{\alpha_1},\dots,v_r^{\alpha_r}\big)=\big(\big|\mathcal{N}\big(v_1^{\alpha_1},\dots,v_r^{\alpha_r}\big)\setminus V(U)\big|\geq 1\big).$$

Given a set of vertices $V=\{v_1,v_2,\dots,v_r\}$, we write $\neg V$ in place of $\neg v_1,\dots,\neg v_r$.

3.5. ***Definition of the type of a bad subgraph.*** Fix a positive integer m. For a vertex $t\in V(\Gamma)\setminus V(U)$, define $j^U(t)\in\{0,1\}^{m+m2^m}$ as follows: For all k with $1\leq k\leq m$ and for all $S\subseteq V(U)$ put

$$j^U_{u_k}(t)=\delta^U(t,u_k,\neg(V(U)\setminus\{u_k\}));$$

$$j^U_{u_k,S}(t)=\Big(\exists s\in\mathcal{N}^U(t,u_k,\neg(V(U)\setminus\{u_k\})):\\ \big(\bigwedge_{l:\,u_l\in S}\delta^U(s,u_l,\neg t,\neg(V(U)\setminus\{u_l\}))\big)\wedge\big(\bigwedge_{l:\,u_l\in V(U)\setminus S}\neg\delta^U(s,u_l,\neg t,\neg(V(U)\setminus\{u_l\}))\big)\Big).\quad(1)$$

Hereafter, for predicate P we use the notation (P) that takes the value 1 if P is true and 0 otherwise. Finally,

$$j^U(t)=\big(\big(j^U_{u_k}(t)\big)_{k\in\{1,\dots,m\}},\big(j^U_{u_k,S}\big)_{k\in\{1,\dots,m\},S\subseteq V(U)}\big).$$

Note that $j^U_{u_k}(t)=\bigvee_{S\subseteq V(U)}j^U_{u_k,S}(t)$.

Let $u,w\in V(U)$. Define

$$J^U(u,w)=\big\{j^U(t)\mid t\in\mathcal{N}^U(u,w)\big\};$$
$$I^U(u,w)=\big(\big|\mathcal{N}^U(u,w)\big|>1\big).$$

Let a vertex $s\in V(\Gamma)\setminus V(U)$ satisfy $\delta^U(u,w,s)=1$. Define

$$\sigma_1^U(u,w,s)=\delta^U(u,s,\neg w);$$
$$\sigma_2^U(u,w,s)=\delta^U(w,s,\neg u);$$
$$\sigma^U(u,w,s)=\big(\sigma_1^U(u,w,s),\sigma_2^U(u,w,s)\big);$$

$$S^U(u,w)=\begin{cases}\big\{s\in\mathcal{N}^U(\neg u_1,\dots,\neg u_m)\mid\forall t\in\mathcal{N}^U(u,w):\ s\sim t\big\}, & I^U(u,w)=1;\\ \varnothing, & \text{else.}\end{cases}$$

Let

$$\Sigma^U(u,w)=\big\{\sigma^U(u,w,s)\mid s\in S^U(u,w)\big\}.$$

For a u-bad subgraph U isomorphic to G_i, define the i*-type* of U in the following manner (hereafter, u_k is the image of z_k^i under the isomorphism between G_i and U, for $1\leq k\leq m$; the image of $z\in V(G_i)$ is $u=u_1$):

$$T_1^i(U)=(J^U(u_k,u_n))_{k,n};$$
$$T_2^i(U)=(I^U(u_k,u_n))_{k,n};$$
$$T_3^i(U)=(\Sigma^U(u_k,u_n))_{k,n}.$$

The latter three definitions are tuples for which indices of vertices take all pairs of values $1 \le k < n \le m$ in lexicographical order. Let

$$T^i(U) = (T_1^i(U), T_2^i(U), T_3^i(U)).$$

Finally, put

$$\mathcal{J}^i(u) = \{T^i(U) \mid U \in \mathcal{U}_i(u)\}, \quad 1 \le i \le |\mathcal{G}|.$$

3.6. *Construction of some graph pairs associated with the latter definitions.* Now we are almost ready to move on to the lemmas. All we have left is to present a way of constructing graphs that serve as extensions for Duplicator to search when playing the Ehrenfeucht game. Our constructions will be closely related to the value $j^U(t)$.

Fix m vertices $\tilde{u}_1, \ldots, \tilde{u}_m$ and an arbitrary value

$$j = (j_{k,S})_{k \in \{1,\ldots,m\},\, S \subseteq \{\tilde{u}_1,\ldots,\tilde{u}_m\}} \in \{0, 1\}^{m2^m}.$$

We build a pair of graphs (A, B) in the following manner:

a) Mark a vertex $\tilde{t}$.

b) For each nonzero $j_{k,S}$, mark a vertex $\tilde{s}_{k,S}$ and join it with $\tilde{u}_k$ and $\tilde{t}$.

c) For each vertex $\tilde{s}_{k,S}$ and for each l such that $\tilde{u}_l \in S$, mark a vertex and join it with $\tilde{u}_l$ and $\tilde{s}_{k,S}$.

d) The set of all the marked vertices, including $\tilde{u}_1, \ldots, \tilde{u}_m$, and the set of all the drawn edges constitute graph A. The set of vertices $\{\tilde{u}_1, \ldots, \tilde{u}_m, \tilde{t}\}$ and the empty set of edges constitute B.

Definition 21. The constructed pair (A, B) *corresponds* to the tuple $j \in \{0, 1\}^{m2^m}$.

Note that the inequality $v(A, B) \le m(m+1)2^m$ holds, since for each $j_{k,S}$ not more than $m+1$ vertices are added to $V(A, B)$. Also note that the construction of the pair (A, B) comes down to the sequential construction of several (K^*, T^*)-extensions. In other words, the set $V(A, B)$ is a subset of the strict (K^*, T^*)-neighborhood of B in A.

4. Three main lemmas

4.1. *The first lemma.*

Lemma 22. *There exists an $\varepsilon > 0$ such that for each $\alpha \in \left(\frac{1}{2}, \frac{1}{2} + \varepsilon\right)$ a.a.s. (X, Y) satisfies the following condition: for each vertex $x_1 \in V(X)$ there exists a vertex $y_1 \in V(Y)$ such that for all $1 \le i \le |\mathcal{G}|$ the equality*

$$\mathcal{J}^i(x_1) = \mathcal{J}^i(y_1)$$

holds. Here $X \sim G(N, N^{-\alpha})$ and $Y \sim G(M, M^{-\alpha})$ are independent random graphs.

Proof. The informal idea of the proof is as follows. For almost every graph X and an arbitrary vertex $x_1 \in V(X)$, we need to construct a graph Z with a vertex $z_1 \in V(Z)$ such that for all i from 1 to K the equality $\mathcal{J}^i(x_1) = \mathcal{J}^i(z_1)$ holds. We then find a “maximal” copy $\hat{Z}$ of Z (meaning that there are no $\frac{1}{2}$-neutral nor $\frac{1}{2}$-rigid extensions of bounded size of $\hat{Z}$ in Y) in almost every graph Y. Finally, we choose y_1 as the image of z_1 under the isomorphism between Z and $\hat{Z}$.

For each positive integer N let $\mathcal{X}_N \subseteq \Omega_N$ be a set of graphs on N vertices without any subgraphs G with $\rho^{\max}(G) \ge 2$ and $v(G) \le 41$. By Theorem 11, we have $\mathsf{P}(X \in \mathcal{X}_N) \to 1$ for each $\alpha > \frac{1}{2}$. Consider an arbitrary $X \in \mathcal{X}_N$ and a vertex $x_1 \in X$. By Statement 16, any two x_1-bad subgraphs of X only intersect in x_1. We now introduce an algorithm to build a graph $Z = Z(X, x_1)$ with $z_1 \in V(Z)$ such that $\rho^{\max}(Z) < 2$ and the equality $\mathcal{J}^i(x_1) = \mathcal{J}^i(z_1)$ holds for $1 \le i \le |\mathcal{G}|$.

(1) Mark a vertex z_1.

(2) For $1 \le i \le |\mathcal{G}|$, for all types $T \in \mathcal{J}^i(x_1)$:

  (a) Let $T = T^i(U)$ for some $U \in \mathcal{U}_i(x_1)$. Let $x_2, \dots, x_{v(G_i)}$ be the elements of $V(U) \setminus \{x_1\}$.

  (b) Mark $v(G_i) - 1$ vertices $z_2, \dots, z_{v(G_i)}$ and draw edges between them so that the subgraph of Z induced on $z_1, z_2, \dots, z_{v(G_i)}$ is isomorphic to U (with z_k being an image of x_k for all k).

  (c) For all pairs (k, n) such that $1 \le k < n \le v(G_i)$:

    (i) If $S^U(x_k, x_n)$ is nonempty and $|\mathcal{N}^U(x_k, x_n)| = 2$:

      (A) For each vertex $t \in \mathcal{N}^U(x_k, x_n)$ mark a vertex r_t, join it with z_k and z_n, and construct the strict (A, B)-extension of the subgraph induced on $z_1, z_2, \dots, z_{v(G_i)}, r_t$, where (A, B) is the pair of graphs that corresponds to $j^U(t)$ (see Section 3.6).

      (B) For each $\sigma = (\sigma_1, \sigma_2) \in \Sigma^U(x_k, x_n)$ (see Section 3.5) mark a vertex r_σ and join it with the vertices r_t for $t \in \mathcal{N}^U(x_k, x_n)$. If $\sigma_1 = 1$, mark a vertex r_1 and join it with r_σ and z_k. If $\sigma_2 = 1$, mark r_2 and join it with r_σ and z_n.

    (ii) Else:

      (A) For each $j \in J^U(x_k, x_n)$ mark r_j, join it with z_k and z_n, and construct a strict (A, B)-extension of the subgraph induced on $z_1, z_2, \dots, z_{v(G_i)}, r_j$, where (A, B) is the pair of graphs that corresponds to j.

      (B) If $|J^U(x_k, x_n)| = 1$, but $I^U(x_k, x_n) = 1$, repeat Step (A) once again.

(3) Marked vertices and drawn edges constitute the graph Z. □

Statement 23. The inequality $\rho^{\max}(Z) < 2$ holds.

Proof. Let R be an x_1-bad subgraph of X isomorphic to a modified 0-stage graph or a 1-stage graph. By Property 7 of $\mathcal{G}$, the inequality $e(R, R|_{\{x_1\}}) \ge 2v(R, R|_{\{x_1\}}) + 1$ holds. If there exist at least two x_1-bad subgraphs isomorphic to a modified 0-stage graph or a 1-stage graph (denote them by R_1, R_2), then for their union S we have

$$e(S) = e(R_1) + e(R_2) \ge 2v(R_1) + 2v(R_2) - 2 = 2v(S).$$

Thus, $\rho(S) \ge 2$, with $v(S) \le 41$, which contradicts the absence of small dense subgraphs in X.

Therefore, in Step (b) the algorithm constructs no more than one extension of $Z|_{\{z_1\}}$, isomorphic to a modified 0-stage or a 1-stage graph. Denote this subgraph (or the subgraph on a single vertex z_1 if there is no such extension) by U_0. Denote by Z_1 the subgraph induced on z_1 and all the other vertices added in Step (b). We call an induced subgraph $U \subseteq Z$ that contains z_1 *interesting* if $V(U) \setminus z_1$ is equal to the set of vertices built in Step (b) of the algorithm for some $T \in \mathcal{J}^i(x_1)$.

Let us show that

$$\rho^{\max}(Z_1) < 2.$$

Let $U_1, \dots, U_h$ be an enumeration of all interesting subgraphs of Z, but for U_0. For $1 \le k \le h$ denote

$$\Psi_k = Z|_{V(U_0) \cup V(U_1) \cup \ldots \cup V(U_k)}.$$

Put $\Psi_0 = U_0$. Since the algorithm does not introduce an edge between any pair of vertices belonging to two different U_k, for $1 \le k \le h$ the equalities

$$V(\Psi_k) \setminus V(\Psi_{k-1}) = V(U_k) \setminus \{z_1\} \quad \text{and} \quad E(\Psi_k) \setminus E(\Psi_{k-1}) = E(U_k) \setminus E\big(Z|_{\{z_1\}}\big)$$

hold. The pair $(U_k, Z|_{\{z_1\}})$ is $\frac{1}{2}$-neutral for $1 \le k \le h$. Moreover, from Property 5 of $\mathcal{G}$ it follows that for each k, there is at least one edge between z_1 and $V(U_k) \setminus \{z_1\}$. Thus, the pair (Ψ_k, Ψ_{k-1}) is $\frac{1}{2}$-neutral for $1 \le k \le h$, and there is at least one edge between $V(\Psi_{k-1})$ and $V(\Psi_k) \setminus V(\Psi_{k-1})$, in Ψ_k.

Let us prove by induction on k that $\rho^{\max}(\Psi_k) < 2$. The base case when $k = 0$ follows from the fact that the graph $\Psi_0 = U_0$ is isomorphic to a graph from set $\mathcal{G}$. Let us prove the step case from $k-1$ to k. We have $\rho^{\max}(\Psi_{k-1}) < 2$, pair (Ψ_k, Ψ_{k-1}) is $\frac{1}{2}$-neutral, and in Ψ_k there is at least one edge between $V(\Psi_{k-1})$ and $V(\Psi_k) \setminus V(\Psi_{k-1})$. From Corollary 18 with $\alpha = \frac{1}{2}$, we obtain $\rho^{\max}(\Psi_k) < 2$. With $k = h$, we have $\rho^{\max}(Z_1) < 2$.

The construction performed in the algorithm steps, but for Step (b), comes down to construction of strict (K^*, T^*)-extensions. Thus, the graph Z is a strict (K^*, T^*)-neighborhood of its subgraph Z_1. By Corollary 20, we have $\rho^{\max}(Z) < 2$. □

We now study the structure of z_1-bad subgraphs in Z.

Statement 24. A subgraph of Z is z_1-bad if and only if it is interesting.

Proof. The statement boils down to two parts. First, we show that each z_1-bad subgraph in Z is interesting.

Let us show that the vertices of each z_1-bad subgraph are contained in the set $V(U_0) \cup V(U_1) \cup \dots \cup V(U_h)$ (the graphs U_i are defined in the proof of Statement 23). Assume the contrary: let for some i the graph Z contain an induced subgraph $\tilde{G}$ isomorphic to G_i, with the image of $z \in V(G_i)$ under the isomorphism between $\tilde{G}$ and G_i being z_1; moreover, not all vertices of $\tilde{G}$ are marked in Step (b) of the algorithm. Of all such vertices we consider the last added vertex a. This vertex is adjacent to exactly two of the previous vertices. The vertices in $\tilde{G}$ marked later by the algorithm are in interesting subgraphs and, thus, cannot be adjacent to a. Therefore, the degree of a in $\tilde{G}$ is not more than 2, which contradicts Property 5 of $\mathcal{G}$.

Furthermore, let us show that every z_1-bad subgraph of Z has to be fully contained in a single interesting subgraph. Let this not be the case. Then a subgraph U intersects with two interesting subgraphs, in two vertices each (including z_1). But, different interesting subgraphs only intersect in z_1, and edges of Z do not join vertices from different interesting subgraphs. On removing z_1, however, U does not fall into components (by Property 8 of $\mathcal{G}$) and, consequently, cannot contain vertices from different interesting subgraphs, other than z_1. Finally, a z_1-bad subgraph cannot be strictly contained in an interesting subgraph, for this contradicts its $\mathcal{G}$-maximality. Thus, each z_1-bad subgraph in Z is interesting.

Now, consider an interesting subgraph $U \subseteq Z$. Let it not be z_1-bad. Since U is isomorphic to some $G \in \mathcal{G}$, there is a graph $G' \supset U$ that is z_1-bad. Then G' is also interesting (as proven above), which contradicts two interesting subgraphs intersecting only on z_1. The statement is proven. □

Statement 25. For all i the equality $\mathcal{J}^i(x_1) = \mathcal{J}^i(z_1)$ holds. Here x_1 is the vertex from the statement of Lemma 22.

Proof. Consider an arbitrary $T^i(U) \in \mathcal{J}^i(x_1)$ with some x_1-bad U being induced on vertices $x_1, x_2, \dots, x_m$. Let $\tilde{U}$ be a corresponding interesting subgraph of Z on vertices $z_1, z_2, \dots, z_m$. Let us show that $T^i(U) = T^i(\tilde{U})$. It is sufficient to check the equalities for T^i_1, T^i_2, and T^i_3.

Consider arbitrary $1 \le k, n \le m$. Note that if $S^U(x_k, x_n)$ is nonempty then $|\mathcal{N}^U(x_k, x_n)| = 2$. Indeed, first, by the definition of $S^U(x_k, x_n)$ there is $|\mathcal{N}^U(x_k, x_n)| \ge 2$. Moreover, if $|\mathcal{N}^U(x_k, x_n)| \ge 3$, denote by t_1, t_2, t_3 three arbitrary vertices of $\mathcal{N}^U(x_k, x_n)$ and by s an arbitrary vertex of $S^U(x_k, x_n)$. By definition, s is adjacent to each t_ℓ, $1 \le \ell \le 3$. Therefore, $(X|_{V(U)\cup\{t_1,t_2,t_3,s\}}, U)$ is bad, which contradicts the fact that U is x_1-bad.

Consider an arbitrary $t \in \mathcal{N}^U(x_k, x_n)$. Put $j = j^U(t)$. If $S^U(x_k, x_n)$ is nonempty, then $|\mathcal{N}^U(x_k, x_n)| = 2$; and in Step (A) the algorithm marks $r_t \in \mathcal{N}^{\tilde{U}}(z_k, z_n)$. If, on the other hand, $S^U(x_k, x_n)$ is empty, then in Step (A) the algorithm marks $r_j \in \mathcal{N}^{\tilde{U}}(z_k, z_n)$. Denote $\tilde{t} = r_t$ in the first case and $\tilde{t} = r_j$ in the second one.

Let us show that

$$j^U(t) = j^{\tilde{U}}(\tilde{t}).$$

Fix an arbitrary $S \subseteq V(U)$ and k, for some $1 \le k \le m$. Let $\varphi : V(U) \to V(\tilde{U})$ be the isomorphism between the graphs U and $\tilde{U}$ such that $\varphi(x_k) = z_k$ for each k. Put $\tilde{S} = \varphi(S)$. Let us show that $j^U_{k,S}(t) = j^{\tilde{U}}{}_{k,\tilde{S}}(\tilde{t})$. Consider the pair (A, B) that corresponds to $j^U(t)$ (see Section 3.6) and denote by $\tilde{A}$ the subgraph of Z that is the image of A constructed in Step (A) or (A) of the algorithm.

If $j^U_{k,S}(t) = 1$, then in $\tilde{A}$ there exists a vertex $s_{k,\tilde{S}} \in \mathcal{N}^{\tilde{U}}(z_k, \tilde{t})$ for which by construction of (A, B) the following holds:

$$\delta^{\tilde{U}}(s_{k,\tilde{S}}, z_l, \neg\tilde{t}, \neg(V(\tilde{U}) \setminus \{z_1\})) = \begin{cases} 1, & x_l \in S, \\ 0, & \text{else}, \end{cases}$$

which implies that $j^{\tilde{U}}_{k,\tilde{S}}(\tilde{t}) = 1$.

Let $j^U_{k,S}(t) = 0$. Assume that $j^{\tilde{U}}_{k,\tilde{S}}(\tilde{t}) = 1$. Then in $\mathcal{N}^{\tilde{U}}(\tilde{t}, z_k)$ there exists a vertex s such that

$$\delta^{\tilde{U}}(s, z_l, \neg\tilde{t}, \neg(V(\tilde{U}) \setminus \{z_1\})) = \begin{cases} 1, & x_l \in S, \\ 0, & \text{else}. \end{cases}$$

Note that s is not the image of any of the vertices $\tilde{s}_{k,\tilde{S}'} \in V(A)$. Indeed, otherwise we have

$$\delta^{\tilde{U}}\big(s, z_l, \neg\tilde{t}, \neg(V(\tilde{U}) \setminus \{z_1\})\big) = \begin{cases} 1, & x_l \in S' = \{l \mid z_l \in \tilde{S}'\}, \\ 0, & \text{else}, \end{cases}$$

but $S' \ne S$.

But the vertex $s \in \mathcal{N}^{\tilde{U}}(\tilde{t}, z_k)$ may only be marked as the image of some vertex $\tilde{s}_{k,\tilde{S}'} \in V(A)$ since $\tilde{t}$ and z_k have no other common neighbors. We obtain a contradiction. Thus, $j^{\tilde{U}}_{k,\tilde{S}}(\tilde{t}) = 0$, and the equality

$$j^U(t) = j^{\tilde{U}}(\tilde{t})$$

is proven.

We now verify that $T^i_1(U) = T^i_1(\tilde{U})$. The equality boils down to the componentwise equalities $J^U(x_k, x_n) = J^{\tilde{U}}(z_k, z_n)$ for all k, n.

Consider an arbitrary $j \in J^U(x_k, x_n)$ and a vertex $t \in \mathcal{N}^U(x_k, x_n)$ such that $j^U(t) = j$. As proven above, there exists $\tilde{t} \in \mathcal{N}^{\tilde{U}}(z_k, z_n)$ such that $j^U(t) = j^{\tilde{U}}(\tilde{t})$. Thus, the inclusion $J^U(x_k, x_n) \subseteq J^{\tilde{U}}(z_k, z_n)$ holds.

Moving on, an arbitrary vertex $\tilde{t} \in \mathcal{N}^{\tilde{U}}(z_k, z_n)$ may only be marked in Steps (A), (A), or (B) of the algorithm. This means that there exists a vertex $t \in \mathcal{N}^U(x_k, x_n)$ such that $j^U(t) = j^{\tilde{U}}(\tilde{t})$. We have shown the second inclusion, which completes the proof of $T_1^i(U) = T_1^i(\tilde{U})$.

We now verify that $T_2^i(U) = T_2^i(\tilde{U})$. The equality boils down to componentwise equalities $I^U(x_k, x_n) = I^{\tilde{U}}(z_k, z_n)$.

If $I^U(z_k, z_n) = 0$, then $S^U(z_k, z_n) = \varnothing$. Thus, vertices from $\mathcal{N}^{\tilde{U}}(z_k, z_n)$ may only be marked in Step (A) of the algorithm. Therefore, the algorithm marks not more than one vertex in $\mathcal{N}^{\tilde{U}}(z_k, z_n)$, which implies the equality $I^{\tilde{U}}(x_k, x_n) = 0$.

Let $I^U(x_k, x_n) = 1$. Consider the case $S^U(z_k, z_n) \neq \varnothing$. As proven above, $|\mathcal{N}^U(z_k, z_n)| = 2$, and in Step (A) at least two vertices from $\mathcal{N}^{\tilde{U}}(z_k, z_n)$ are marked, which implies the equality $I^U(x_k, x_n) = 1$. Consider the case $S^U(z_k, z_n) = \varnothing$. If $|J^U(x_k, x_n)| > 1$, then in Step (A) the algorithm marks at least two vertices of $\mathcal{N}^{\tilde{U}}(z_k, z_n)$, which implies the equality $I^{\tilde{U}}(x_k, x_n) = 1$. If $|J^U(x_k, x_n)| = 1$, then in Steps (A) and (B) the algorithm marks two vertices of $\mathcal{N}^{\tilde{U}}(z_k, z_n)$, which implies the equality $I^{\tilde{U}}(x_k, x_n) = 1$. The equality $T_2^i(U) = T_2^i(\tilde{U})$ is verified.

Finally, we verify that $T_3^i(U) = T_3^i(\tilde{U})$. The equality boils down to the componentwise equalities $\Sigma^U(x_k, x_n) = \Sigma^{\tilde{U}}(z_k, z_n)$.

If $\Sigma^U(x_k, x_n) = \varnothing$, then either $I^U(x_k, x_n) = 0$ or there exist $t_1, t_2 \in \mathcal{N}^U(x_k, x_n)$ such that $\delta^U(t_1, T_1) = 0$. In the first case, the algorithm marks not more than one vertex $r \in \mathcal{N}^{\tilde{U}}(z_k, z_n)$, which implies the equality $\Sigma^{\tilde{U}}(z_k, z_n) = \varnothing$. In the second case, the algorithm marks at least two vertices $r_1, r_2 \in \mathcal{N}^{\tilde{U}}(z_k, z_n)$, but at the same time the algorithm does not mark any vertices adjacent to r_1 and r_2 but for in $V(\tilde{U})$. Thus, $\Sigma^{\tilde{U}}(z_k, z_n) = \varnothing$.

Examine the case $\Sigma^U(x_k, x_n) \neq \varnothing$. As shown above, $|\mathcal{N}^U(x_k, x_n)| = 2$. Denote by t_1, t_2 the elements of the set $\mathcal{N}^U(x_k, x_n)$. In Step (A), the algorithm marks the vertices $r_{t_1}, r_{t_2} \in \mathcal{N}^{\tilde{U}}(z_k, z_n)$. By construction, the set $\mathcal{N}^{\tilde{U}}(z_k, z_n)$ consists exactly of r_{t_1} and r_{t_2}. Consider an arbitrary $\sigma \in \Sigma^U(x_k, x_n)$ defined by some $s \in S^U(x_k, x_n)$. In Step (A), the algorithm marks $r_\sigma \in \mathcal{N}^{\tilde{U}}(r_{t_1}, r_{t_2})$. Let us show the equality $\sigma^U(x_k, x_n, s) = \sigma^{\tilde{U}}(z_k, z_n, r_\sigma)$. Vertices from sets $\mathcal{N}^{\tilde{U}}(z_h, r_\sigma)$ with $h = k, n$ may only be marked by the algorithm in Step (B). In this step, the vertex that is adjacent to z_h and r_σ is constructed iff $\delta^U(x_h, s) = 1$, which implies the equality $\delta^U(x_h, s) = \delta^{\tilde{U}}(z_h, r_\sigma)$ with $h = k, n$. The equality $\sigma^U(x_k, x_n, s) = \sigma^{\tilde{U}}(z_k, z_n, r_\sigma)$ is verified. Moreover, $r_\sigma \in S^{\tilde{U}}(z_k, z_n)$, and thus the inclusion $\Sigma^U(x_k, x_n) \subseteq \Sigma^{\tilde{U}}(z_k, z_n)$ is verified.

We now verify the reverse inclusion. Consider an arbitrary $\sigma \in \Sigma^{\tilde{U}}(z_k, z_n)$ defined by some $\tilde{s} \in S^{\tilde{U}}(z_k, z_n)$. We have the inclusion $\tilde{s} \in \mathcal{N}^{\tilde{U}}(r_{t_1}, r_{t_2})$. Such a vertex may only be marked by the algorithm in Step (B). Therefore, $\tilde{s} = r_\sigma$ for some $\sigma \in \Sigma^U(x_k, x_n)$. Above we proved the equality $\sigma(\tilde{s}, z_k, z_n) = \sigma$. Thus, we have shown the inclusion $\Sigma^{\tilde{U}}(z_k, z_n) \subseteq \Sigma^U(x_k, x_n)$, which concludes the proof of the equality $T_3^i(U) = T_3^i(\tilde{U})$. Thus, for each i we have

$$\mathcal{J}^i(x_1) \subseteq \mathcal{J}^i(z_1).$$

Now consider an arbitrary z_1-bad $\tilde{U} \subseteq Z$ isomorphic to G_i. By Statement 24 $\tilde{U}$ is interesting; hence, in X there exists an x_1-bad subgraph U for which, by what we have proven above, the equality $T^i(U) = T^i(\tilde{U})$ holds. Therefore,

$$\mathcal{J}^i(z_1) \subseteq \mathcal{J}^i(x_1).$$

Thus, for all $1 \leq i \leq |\mathcal{G}|$ the equalities $\mathcal{J}^i(x_1) = \mathcal{J}^i(z_1)$ hold, and the statement is proven. □

Define $\mathcal{Z}_N = \{Z(X, x_1) \mid X \in \mathcal{X}_N, x_1 \in V(X)\}$.

Statement 26. The number of vertices in an arbitrary graph $Z \in \mathcal{Z}_N$ is bounded above by an absolute constant independent of N.

Proof. The number of iterations of each step of the algorithm is clearly bounded above by an absolute constant. In each step the algorithm marks a bounded number of vertices. Thus, the number of vertices in each $Z \in \mathcal{Z}_N$ is also bounded. $\square$

Define $\mathcal{Z} = \bigcup_{N\in\mathbb{N}} \mathcal{Z}_N$. Put

$$\varepsilon = \frac{1}{\sup\limits_{Z\in\mathcal{Z}} \rho^{\max}(Z)} - \frac{1}{2}.$$

Since $\mathcal{Z}$ is finite and all graphs in $\mathcal{Z}$ have maximal density less than 2, we have $\varepsilon > 0$.

Now, consider the set $\mathcal{Y}_M \subseteq \Omega_M$ of graphs Y such that for each $Z \in \mathcal{Z}$ the following holds: in Y there exists a strict copy of Z that is (K, T)-maximal for each $\frac{1}{2}$-rigid or $\frac{1}{2}$-neutral pair (K, T) with $v(K, T) \le 20$ and, in particular, (K^*, T^*)-maximal. By Theorem 12, for each $\alpha \in \left(\frac{1}{2}, \frac{1}{2}+\varepsilon\right)$ we have $\mathsf{P}(Y \in \mathcal{Y}_M) \to 1$.

Consider arbitrary graphs $Z \in \mathcal{Z}$ and $Y \in \mathcal{Y}_M$. In Y, let us find a copy $\hat{Z}$ of Z that is (K, T)-maximal for each $\frac{1}{2}$-rigid or $\frac{1}{2}$-neutral pair (K, T) with $v(K, T) \le 20$. Let $\psi : V(Z) \to V(\hat{Z})$ be an isomorphism between the graphs. Put $y_1 = \psi(z_1)$.

Statement 27. For all i the equalities $\mathcal{J}^i(z_1) = \mathcal{J}^i(y_1)$ hold.

Proof. We first show that y_1-bad subgraphs of Y are exactly images of z_1-bad subgraphs.

First, we verify that for each z_1-bad subgraph $\tilde{U}$, the subgraph $W = \psi(\tilde{U})$ is y_1-bad. Indeed, otherwise in graph Y there is a subgraph $G' \supset W$ that is y_1-bad. If $G' \subset \hat{Z}$ then $\psi^{-1}(G') \subset Z$ and $\tilde{U} \subset \psi^{-1}(G')$, which contradicts the $\mathcal{G}$-maximality of U. Otherwise, denote by W' the graph $Y|_{V(G')\cap V(\hat{Z})}$. The pair (G', W') is either $\frac{1}{2}$-rigid or $\frac{1}{2}$-neutral by Property 7 of $\mathcal{G}$, with $v(G', W') \le v(G') - 1 \le 20$ (by Property 6 of $\mathcal{G}$). This contradicts (K, T)-maximality of $\hat{Z}$ in Y for each $\frac{1}{2}$-rigid or $\frac{1}{2}$-neutral pair (K, T) with $v(K, T) \le 20$.

Second, we verify that each y_1-bad subgraph W is an image of a z_1-bad subgraph in Z. Assume the contrary. Examine two cases. If $W \not\subseteq \hat{Z}$, then the pair $(W, W|_{V(\hat{Z})\cap V(W)})$ is either $\frac{1}{2}$-rigid or $\frac{1}{2}$-neutral by Property 7 of $\mathcal{G}$, with $v(W, W|_{V(\hat{Z})\cap V(W)}) \le 20$. This contradicts (K, T)-maximality of $\hat{Z}$ in Y for each $\frac{1}{2}$-rigid or $\frac{1}{2}$-neutral pair (K, T) with $v(K, T) \le 20$. If, on the other hand, $W \subseteq \hat{Z}$, then the prototype $\tilde{U}$ of W under ψ is defined. If $\tilde{U}$ is not z_1-bad, then there exists a subgraph G' of Z that contains $\tilde{U}$. Then $\psi(G')$ contains W, which contradicts the fact that W is z_1-bad.

Thus, the set of all y_1-bad subgraphs of Y is equal to the set of images of z_1-bad subgraphs of Z under the isomorphism ψ.

Once again, consider an arbitrary z_1-bad subgraph $\tilde{U}$ of type i and its image W. Let us show that $T^i(\tilde{U}) = T^i(W)$. It is evident from the definition of $\mathcal{J}^i(u)$ that the only vertices that influence the i-type of an y_1-bad subgraph are the ones from the (K^*, T^*)-neighborhood of the subgraph. In other words, upon removing from Y any vertex not in the (K^*, T^*)-neighborhood of $\hat{Z}$, the type $T^i(W)$ does not change. None of the vertices of $V(Y) \setminus V(\hat{Z})$ is in the (K^*, T^*)-neighborhood of subgraph $\hat{Z}$, due to the (K^*, T^*)-maximality of $\hat{Z}$ in Y. Upon removing all vertices of the set $V(Y) \setminus V(\hat{Z})$ from Y, we have the graph $\hat{Z}$ isomorphic to Z. Thus, we have the equality $T^i(\tilde{U}) = T^i(W)$. Similarly, for each y_1-bad subgraph W of Y with type i, there is the equality $T^i(W) = T^i(\psi^{-1}(W))$. Thus, the equalities

$$\mathcal{J}^i(x_1) = \mathcal{J}^i(y_1)$$

are verified for all i.

To sum up, for each $X \in \mathcal{X}_N$ and an arbitrary $x_1 \in V(X)$, we have found a graph $Z = Z(X, x_1)$ and a vertex $z_1 \in V(Z)$ such that $\mathcal{J}^i(x_1) = \mathcal{J}^i(z_1)$ for each i. For every graph $Y \in \mathcal{Y}_M$, we have found a vertex $y_1 \in V(Y)$ such that

$$\mathcal{J}^i(z_1) = \mathcal{J}^i(y_1)$$

for each i. Thus, for each $X \in \mathcal{X}_N$ and a vertex $x_1 \in V(X)$ and for each $Y \in \mathcal{Y}_M$, there exists $y_1 \in V(Y)$ such that

$$\mathcal{J}^i(z_1) = \mathcal{J}^i(y_1)$$

for each i. Finally, as stated at the beginning of the proof, we have $\mathsf{P}(X \in \mathcal{X}_N)$ and $\mathsf{P}(Y \in \mathcal{Y}_M)$ tend to 1, which concludes the proof. □

4.2. ***The second lemma.*** Let, as previously, $u, w \in V(\Gamma)$. Define $W = \Gamma|_{\{u,w\}}$. Let

$$\begin{aligned} j(u, w, t) &= j^W(t); \\ I(u, w) &= I^W(u, w); \\ J(u, w) &= J^W(u, w) = \big\{ j(u, w, t) \mid t \in \mathcal{N}(u, w) \big\}. \end{aligned}$$

Let a vertex $s \in V(\Gamma) \setminus V(W)$ satisfy $\delta^W(u, w, s) = 1$. Define

$$\begin{aligned} \sigma_1(u, w, s) &= \sigma_1^W(u, w, s); \\ \sigma_2(u, w, s) &= \sigma_2^W(u, w, s); \\ \sigma(u, w, s) &= (\sigma_1(u, w, s), \sigma_2(u, w, s)); \\ S(u, w) = S^W(u, w) &= \begin{cases} \big\{ s \in \mathcal{N}(\neg u, \neg w) \mid \forall t \in \mathcal{N}(u, w) : s \sim t \big\}, & I(u, w) = 1; \\ \varnothing, & \text{else}; \end{cases} \\ \Sigma(u, w) &= \big\{ \sigma(u, w, s) \mid s \in S(u, w) \big\}. \end{aligned}$$

Lemma 28. *There exists an $\varepsilon > 0$ such that for each $\alpha \in \left(\frac{1}{2}, \frac{1}{2} + \varepsilon\right)$ a.a.s. (X, Y) satisfies the following condition: For each pair of vertices $x_1, x_2 \in V(X)$ such that $x_2 \sim x_1$ and x_2 is in 0-neighborhood $U^0(x_1)$, and for any vertex $y_1 \in V(Y)$, there exists $y_2 \in V(Y)$ such that*

$$\begin{aligned} y_2 &\in U^0(y_1); \\ y_2 &\sim y_1; \\ J(x_1, x_2) &= J(y_1, y_2); \\ I(x_1, x_2) &= I(y_1, y_2); \\ \Sigma(x_1, x_2) &= \Sigma(y_1, y_2). \end{aligned}$$

Here X, Y are independent random graphs $G(N, N^{-\alpha})$, $G(M, M^{-\alpha})$.

Proof. Unlike the previous proof, we construct a pair of graphs, not a single graph. We then find a strict extension isomorphic to the constructed pair in Y.

For each N, put $\mathcal{X}_N = \Omega_N$. Consider an arbitrary $X \in \mathcal{X}_N$ and two vertices $x_1 \in V(X)$, $x_2 \in U^0(x_1)$ such that $x_1 \sim x_2$.

Let $F = (\{z_1\}, \varnothing)$. We propose the algorithm of constructing a graph $Z = Z(X, x_1, x_2)$, $F \subset Z$ such that the pair (Z, F) is $\frac{1}{2}$-safe and

$$z_2 \in U^0(z_1);$$

$$z_2 \sim z_1;$$

$$J(x_1, x_2) = J(z_1, z_2);$$

$$I(x_1, x_2) = I(z_1, z_2);$$

$$\Sigma(x_1, x_2) = \Sigma(z_1, z_2).$$

(1) Mark vertices z_1, z_2, and join them with an edge.
   (a) If $S(x_1, x_2)$ is nonempty and $|\mathcal{N}(x_1, x_2)| = 2$:
      (i) For each $t \in \mathcal{N}(x_1, x_2)$, mark a vertex r_t, join it with z_1 and z_2, and construct a strict (A, B)-extension of the subgraph induced on z_1, z_2, r_t, where (A, B) is the pair of graphs that corresponds to $j(x_1, x_2, t)$.
      (ii) For each $\sigma \in \Sigma(x_1, x_2)$ mark a vertex r_σ and join it with r_t for all $t \in \mathcal{N}(x_1, x_2)$. For $k = 1, 2$, if $\sigma_k = 1$, mark r_k and join it with r_σ and z_k.
   (b) Else:
      (i) For each $j \in J(x_1, x_2)$, mark r_j, join it with z_1 and z_2, and construct a strict (A, B)-extension of the subgraph induced on z_1, z_2, r_j, where (A, B) is the pair of graphs corresponding to j.
      (ii) If $|J(x_1, x_2)| = 1$, but $I(x_1, x_2) = 1$, repeat Step (i) once again.

(2) Marked vertices and drawn edges constitute the graph Z.

Let us show that if $\Sigma(x_1, x_2)$ is nonempty, then the equality $|\mathcal{N}(x_1, x_2)| = 2$ holds. First, we have

$$|\mathcal{N}(x_1, x_2)| \geq 2,$$

by definition of $S(x_1, x_2)$. Let $|\mathcal{N}(x_1, x_2)| > 2$. Then in the graph X, there exist vertices t_1, t_2, $t_3 \in \mathcal{N}(x_1, x_2)$ and $s \in \mathcal{N}(t_1, t_2, t_3)$ different from x_1 and x_2. Then $X|_{\{x_1, x_2, t_1, t_2, t_3, s\}}$ is a $\frac{1}{2}$-neutral extension of $X|_{\{x_1\}}$, with $v(X_{\{x_1, x_2, t_1, t_2, t_3, s\}}, X|_{\{x_1\}}) = 5$, which contradicts the inclusion $x_2 \in U^0(x_1)$. Thus, $|\mathcal{N}(x_1, x_2)| = 2$.

Statement 29. The constructed pair (Z, F) is $\frac{1}{2}$-safe.

Proof. Construction performed in the algorithm steps other then (1) boils down to sequential construction of strict (K^*, T^*)-extensions of some subgraphs of the graph being constructed. Therefore, Z is a strict (K^*, T^*)-neighborhood of its subgraph $Z|_{\{z_1, z_2\}}$. Moreover, the pair $(Z|_{\{z_1, z_2\}}, F)$ is $\frac{1}{2}$-safe. By Statement 19 we obtain the desired conclusion. □

Statement 30. The following equalities hold:

$$z_2 \in U^0(z_1);$$
$$z_2 \sim z_1;$$
$$J(x_1, x_2) = J(z_1, z_2);$$
$$I(x_1, x_2) = I(z_1, z_2);$$
$$\Sigma(x_1, x_2) = \Sigma(z_1, z_2).$$

Proof. We first prove that $z_2 \in U^0(z_1)$. Assume the contrary, i.e., there exists an induced subgraph $\tilde{U} \subseteq Z$ isomorphic to some $G \in \mathcal{G}$ containing z_1 and z_2, with z_1 being the image of $z \in V(G)$ under the isomorphism between G and $\tilde{U}$. Then $(\tilde{U}, F)$ is either $\frac{1}{2}$-rigid or $\frac{1}{2}$-neutral by Property 7 of $\mathcal{G}$. We obtain the contradiction with the $\frac{1}{2}$-safeness of (Z, F).

The relation $z_2 \sim z_1$ holds by construction of Z.

The other equalities can be verified similar to the corresponding ones in the proof of Statement 25, with $U = X|_{\{x_1,x_2\}}$. □

Define $\mathcal{Z}_N = \{Z(X, x_1, x_2) \mid X \in \mathcal{X}_N;\ x_1, x_2 \in V(X)\}$. As in Lemma 22, the total number of vertices in any $Z \in \mathcal{Z}_N$ is bounded above by an absolute constant.

Put $\mathcal{Z} = \bigcup_{N\in\mathbb{N}} \mathcal{Z}_N$. Let

$$\varepsilon = \inf_{Z\in\mathcal{Z}} \frac{v(Z, F)}{e(Z, F)} - \frac{1}{2}.$$

Obviously, $\varepsilon > 0$.

Let $\mathcal{Y}_M \subseteq \Omega_M$ be a set of graphs on M vertices such that for each $Z \in \mathcal{Z}$ and $y_1 \in V(Y)$ there exists a strict (Z, F)-extension of $Y|_{\{y_1\}}$ in Y that is (K, T)-maximal for each $\frac{1}{2}$-rigid or $\frac{1}{2}$-neutral (K, T) with $v(K, T) \leq 19$. By Theorem 12, for each $\alpha \in \left(\frac{1}{2}, \frac{1}{2} + \varepsilon\right)$ we have $\mathsf{P}(Y \in \mathcal{Y}_M) \to 1$.

Consider an arbitrary $Z \in \mathcal{Z}$, $Y \in \mathcal{Y}_M$ and an arbitrary vertex $y_1 \in V(Y)$. Find a strict (Z, F)-extension $\hat{Z}$ of $Y|_{\{y_1\}}$ in Y such that $\hat{Z}$ is (K, T)-maximal for each $\frac{1}{2}$-rigid or $\frac{1}{2}$-neutral pair (K, T) with $v(K, T) \leq 19$. Let y_2 be an image of z_2 under the isomorphism between Z and $\hat{Z}$.

Statement 31. The following equalities hold:

$$y_2 \in U^0(y_1);$$
$$y_2 \sim y_1;$$
$$J(z_1, z_2) = J(y_1, y_2);$$
$$I(z_1, z_2) = I(y_1, y_2);$$
$$\Sigma(z_1, z_2) = \Sigma(y_1, y_2).$$

Proof. We first verify that $y_2 \in U^0(y_1)$. It is sufficient to show that there exist no induced subgraphs $W \subseteq Y$ isomorphic to some $G \in \mathcal{G}$ such that $y_2 \in V(W)$ and y_1 is the image of z under the isomorphism between G and W. Assume the contrary. Note that $W \not\subseteq \hat{Z}$, because $z_2 \in U^0(z_1)$. Then $(W, W|_{V(W)\cap V(\hat{Z})})$ is $\frac{1}{2}$-rigid by Property 7 of $\mathcal{G}$, with $v(W, W|_{V(W)\cap V(\hat{Z})}) \leq 19$, which contradicts (K, T)-maximality of $\hat{Z}$ in Y for each $\frac{1}{2}$-rigid (K, T) with $v(K, T) \leq 19$.

Moreover, obviously, $y_1 \sim y_2$. The equalities $J(z_1, z_2) = J(y_1, y_2)$, $I(z_1, z_2) = I(y_1, y_2)$, $\Sigma(z_1, z_2) = \Sigma(y_1, y_2)$ can be verified similar to the corresponding ones in Lemma 22. □

Thus, for each $X \in \mathcal{X}_N$ and vertices $x_1, x_2 \in V(X)$ such that $x_1 \sim x_2$, we constructed $Z = Z(X, x_1, x_2)$ and vertices $z_1, z_2 \in V(Z)$ such that the equalities for x_1, x_2, z_1, z_2 from Lemma 28 hold. For each $Y \in \mathcal{Y}_M$ and $y_1 \in V(Y)$ we found $y_2 \in V(Y)$ such that the equalities for x_1, x_2, z_1, z_2 from Lemma 28 hold. Since $\mathsf{P}(X \in \mathcal{X}_N)$ and $\mathsf{P}(Y \in \mathcal{Y}_M)$ tend to 1 (as stated at the beginning of the proof), Lemma 28 is proved. □

4.3. *The third lemma.* We finally introduce our last definitions. Let

$$J'(u, w) = \bigl\{ j(u, w, t) \mid (t \in \mathcal{N}(u, w)) \wedge (\delta(u, w, t) = 0) \bigr\};$$
$$I'(u, w) = (\exists t \in \mathcal{N}(u, w)\ \delta(u, w, \neg t) = 1);$$
$$T(u, w) = \bigl\{ \{t_1, t_2\} \mid t_1, t_2 \in \mathcal{N}(u, w), t_1 \sim t_2 \bigr\};$$
$$\tau(u, w) = \bigl\{ \{ j(u, w, t_1), j(u, w, t_2)\} \mid t_1, t_2 \in \mathcal{N}(u, w), t_1 \sim t_2 \bigr\}.$$

Let us formulate and prove the next lemma.

Lemma 32. *There exists an $\varepsilon > 0$, such that for each $\alpha \in \bigl(\frac{1}{2}, \frac{1}{2} + \varepsilon\bigr)$ a.a.s. (X, Y) satisfies the following condition: For each pair of vertices $x_1, x_2 \in V(X)$ such that $x_2 \not\sim x_1$ and x_2 is in 0-neighborhood $U^0(x_1)$, and for any vertex $y_1 \in V(Y)$, there exists $y_2 \in V(Y)$ such that*

$$y_2 \in U^0(y_1);$$
$$y_2 \not\sim y_1;$$
$$J'(x_1, x_2) = J'(y_1, y_2);$$
$$I'(x_1, x_2) = I'(y_1, y_2);$$
$$\Sigma(x_1, x_2) = \Sigma(y_1, y_2);$$
$$\tau(x_1, x_2) = \tau(y_1, y_2);$$
$$|T(x_1, x_2)| \le 1.$$

Here X, Y are independent random graphs $G(N, N^{-\alpha})$, $G(M, M^{-\alpha})$.

Proof. For each N, put $\mathcal{X}_N = \Omega_N$. Consider an arbitrary $X \in \mathcal{X}_N$ and two vertices $x_1 \in V(X)$, $x_2 \in U^0(x_1)$, $x_1 \not\sim x_2$.

Let us show that $|T(x_1, x_2)| \le 1$. Assume the contrary. Then in X there are at least two pairs of vertices t_1, t_2 and t_3, t_4 that are adjacent to each other as well as to each of x_1 and x_2. If these pairs do not intersect, i.e., $\{t_1, t_2\} \cap \{t_3, t_4\} = \varnothing$, then $X|_{\{x_1,x_2,t_1,t_2,t_3,t_4\}}$ is a $\frac{1}{2}$-neutral extension of $X|_{\{x_1\}}$ with $v(X|_{\{x_1,x_2,t_1,t_2,t_3,t_4\}}, X|_{\{x_1\}}) = 5$. If these pairs do intersect, for instance, $t_1 = t_3$, then $X|_{\{x_1,x_2,t_1,t_2,t_4\}}$ is a $\frac{1}{2}$-neutral extension of $X|_{\{x_1\}}$, with $v(X|_{\{x_1,x_2,t_1,t_2,t_4\}}, X|_{\{x_1\}}) = 4$. Both alternatives contradict $x_2 \in U^0(x_1)$.

Further, let $|T(x_1, x_2)| = 1$ and $S(x_1, x_2)$ be nonempty. Let us prove that $|\mathcal{N}(x_1, x_2)| = 2$. If $|\mathcal{N}(x_1, x_2)| \ge 3$, let t_1, t_2, t_3 be some vertices from $\mathcal{N}(x_1, x_2)$ such that $t_1 \sim t_2$. Since $S(x_1, x_2)$ (see Section 3.5) is nonempty, in the graph X there is a vertex s adjacent to each t_l, with $l = 1, 2, 3$. Then $X|_{\{x_1,x_2,t_1,t_2,t_3,s\}}$ is a $\frac{1}{2}$-neutral extension of $X|_{\{x_1\}}$, with $v(X|_{\{x_1,x_2,t_1,t_2,t_3,s\}}, X|_{\{x_1\}}) = 5$, which contradicts $x_2 \in U^0(x_1)$.

So, put $F = (\{z_1\}, \varnothing)$. We now introduce an algorithm for constructing a graph $Z = Z(X, x_1, x_2)$, $F \subset Z$, such that (Z, F) is $\frac{1}{2}$-safe and

$$z_2 \in U^0(z_1);$$
$$z_2 \not\sim z_1;$$
$$J'(x_1, x_2) = J'(z_1, z_2);$$
$$I'(x_1, x_2) = I'(z_1, z_2);$$
$$\Sigma(x_1, x_2) = \Sigma(z_1, z_2);$$
$$\tau(x_1, x_2) = \tau(z_1, z_2).$$

(1) Mark z_1, z_2.
(2) If $T(x_1, x_2)$ is nonempty:
  (a) Let t_1, t_2 be some adjacent vertices of $\mathcal{N}(x_1, x_2)$.
  (b) Mark r_{t_1} and r_{t_2}, join each with z_1 and z_2 and to each other. With $k = 1, 2$ construct a strict (A, B)-extension of the subgraph induced on z_1, z_2, r_{t_k}, where (A, B) is the pair of graphs that corresponds to $j(x_1, x_2, t_k)$;
  (c) If $S(x_1, x_2)$ is nonempty:
    (i) For each $\sigma \in \Sigma(x_1, x_2)$ mark a vertex r_σ and join it with r_{t_ℓ} for $\ell = 1, 2$. For $k = 1, 2$, if $\sigma_k = 1$, mark r_k and join it with r_σ and z_k.
    (ii) For each $j \in J'(x_1, x_2)$ mark r_j, join it with z_1, z_2, and construct a strict (A, B)-extension of the subgraph induced on z_1, z_2, r_j, where (A, B) is a pair of graphs corresponding to j.
(3) Else:
  (a) If $S(x_1, x_2)$ is nonempty:
    (i) For each $t \in \mathcal{N}(x_1, x_2)$ mark r_t, join it with z_1, z_2, and construct a strict (A, B)-extension of the subgraph induced on z_1, z_2, r_t, where (A, B) is the pair of graphs that corresponds to $j(x_1, x_2, t)$.
    (ii) For each $\sigma \in \Sigma(x_1, x_2)$ mark r_σ and join it with r_t for all $t \in \mathcal{N}(x_1, x_2)$. For $k = 1, 2$, if $\sigma_k = 1$, mark r_k and join it with r_σ and z_k.
  (b) Else:
    (i) For each $j \in J(x_1, x_2)$ mark r_j, join it with z_1, z_2 and construct a strict (A, B)-extension of the subgraph induced on z_1, z_2, r_j, where (A, B) is the pair of graphs corresponding to j.
    (ii) If $|J(x_1, x_2)| = 1$ but $I(x_1, x_2) = 1$, repeat Step (i) once again.
(4) Marked vertices and constructed edges constitute Z.

The rest of the proof of Lemma 32 is similar to the proof of Lemma 28. We conclude the proof of Lemma 32 by proving two statements similar to Statements 29 and 30. □

Statement 33. The pair (Z, F) is $\frac{1}{2}$-safe.

Proof. Let us examine the case $|T(x_1, x_2)| = 0$.

If $S(x_1, x_2) = \varnothing$ or $|\mathcal{N}(x_1, x_2)| = 2$, then construction in all steps of the algorithm but for Step (1) are similar to ones in Lemma 28. Thus, the proof boils down to the proof of the similar statement in Lemma 28.

Let $S(x_1, x_2) \neq \varnothing$ and $|\mathcal{N}(x_1, x_2)| > 2$. If $|\mathcal{N}(x_1, x_2)| \geq 4$, denote by t_1, t_2, t_3, t_4 some vertices of $\mathcal{N}(x_1, x_2)$. Since $S(x_1, x_2)$ is nonempty, in graph X there exists a vertex s joined with each of t_ℓ for $1 \leq \ell \leq 4$, different from x_1 and x_2. Therefore $X|_{\{x_1,x_2,t_1,t_2,t_3,t_4,s\}}$ is a $\frac{1}{2}$-neutral extension of $X|_{\{x_1\}}$, with $v(X_{\{x_1,x_2,t_1,t_2,t_3,t_4,s\}}, X|_{\{x_1\}}) = 6$, which contradicts $x_2 \in U^0(x_1)$. Thus, $|\mathcal{N}(x_1, x_2)| = 3$. Denote by t_1, t_2, t_3 the vertices of $\mathcal{N}(x_1, x_2)$.

Finally, assume $|S(x_1, x_2)| \geq 2$. Let s_1, s_2 be some elements of $S(x_1, x_2)$. The graph $X|_{\{x_1,x_2,t_1,t_2,t_3,s_1,s_2\}}$ is a $\frac{1}{2}$-neutral extension of $X|_{\{x_1\}}$, with $v(X|_{\{x_1,x_2,t_1,t_2,t_3,s_1,s_2\}}, X|_{\{x_1\}}) = 6$, which contradicts $x_2 \in U^0(x_1)$. Thus, $|S(x_1, x_2)| = 1$. By definition of Σ (see Section 3.5), we obtain that $|\Sigma(x_1, x_2)| = 1$.

Denote by σ the single element of $\Sigma(x_1, x_2)$. The algorithm marks the vertices r_{t_l} for $1 \leq l \leq 3$ that are joined with z_1 and z_2 and also a vertex $r_\sigma \in \mathcal{N}(r_{t_1}, r_{t_2}, r_{t_3})$.

Note that $(Z|_{\{z_1,z_2,r_{t_1},r_{t_2},r_{t_3},r_\sigma\}}, F)$ is $\frac{1}{2}$-safe. Moreover, Z is obviously a strict (K^*, T^*)-neighborhood of $Z|_{\{z_1,z_2,r_{t_1},r_{t_2},r_{t_3},r_\sigma\}}$. By Statement 19, we obtain the desired statement for $|T(x_1, x_2)| = 0$.

If $|T(x_1, x_2)| = 1$, then $(Z|_{\{z_1,z_2,r_{t_1},r_{t_2}\}}, F)$ is $\frac{1}{2}$-safe and Z is the strict (K^*, T^*)-neighborhood of $Z|_{\{z_1,z_2,r_{t_1},r_{T_1}\}}$. By Statement 19, the pair (Z, F) is $\frac{1}{2}$-safe, which completes the proof. □

Statement 34. The equalities

$$z_2 \in U^0(z_1);$$

$$z_2 \not\sim z_1;$$

$$J'(x_1, x_2) = J'(z_1, z_2);$$

$$I'(x_1, x_2) = I'(z_1, z_2);$$

$$\Sigma(x_1, x_2) = \Sigma(z_1, z_2);$$

$$\tau(x_1, x_2) = \tau(z_1, z_2)$$

hold.

Proof. We verify that $z_2 \in U^0(z_1)$. Assume the contrary, i.e., let there exist an induced subgraph $\tilde{U} \subseteq Z$ isomorphic to some $G \in \mathcal{G}$ that contains z_1 and z_2, such that z_1 is the image of $z \in V(G)$ under the isomorphism between G and $\tilde{U}$. Then $(\tilde{U}, F)$ is either $\frac{1}{2}$-rigid or $\frac{1}{2}$-neutral due to Property 7 of $\mathcal{G}$. We obtain a contradiction with the $\frac{1}{2}$-safeness of (Z, F).

The relation $z_2 \not\sim z_1$ holds by construction of Z.

Consider an arbitrary $t \in \mathcal{N}^U(x_1, x_2)$. Let $j = j(x_1, x_2, t)$. If $\delta(x_1, x_2, t) = 1$, then $|T(x_1, x_2)| = 1$ and in Step (b) the algorithm marks $r_t \in \mathcal{N}(z_1, z_2)$. If $\delta(x_1, x_2, t) = 0$ and $S(x_1, x_2)$ is nonempty, then in Step (i) the algorithm marks $r_t \in \mathcal{N}(z_1, z_2)$. If $S(x_1, x_2)$ is empty, then in Step (i) the algorithm marks $r_j \in \mathcal{N}(z_1, z_2)$. Let $\tilde{t} = r_t$ in the first or the second case and $\tilde{t} = r_j$ in the third one.

We verify the equality

$$j(x_1, x_2, t) = j(z_1, z_2, \tilde{t}).$$

Let $U = X|_{\{x_1,x_2\}}$. If $\delta(x_1, x_2, t) = 0$, then the proof is similar to the corresponding statement in Lemma 22. Let $\delta(x_1, x_2, t) = 1$. Denote by t' the only vertex of $\mathcal{N}(x_1, x_2, t)$. Note that in the definition of $j^U(t)$, all formulas only contain the existence of vertices adjacent to exactly one vertex of U. Thus, t' does not influence $j^U(t)$ in the sense that upon removing t' from X, the value $j^U(t)$ does not change. With that note, the equality $j(x_1, x_2, t) = j(z_1, z_2, \tilde{t})$ can be proven analogously to the corresponding statement in Lemma 22.

We verify that

$$J'(x_1, x_2) = J'(z_1, z_2).$$

Consider an arbitrary $j \in J'(x_1, x_2)$ given by some $t \in \mathcal{N}(x_1, x_2)$, $\delta(x_1, x_2, t) = 0$. In one of the Steps (i), (i), (ii) a vertex $\tilde{t}$ is marked such that $j(z_1, z_2, \tilde{t}) = j$. Note that the algorithm does not mark any vertices adjacent to $\tilde{t}$, z_1, z_2 at the same time; therefore, $\delta(z_1, z_2, \tilde{t}) = 0$ and $j \in J'(z_1, z_2)$. Thus, the inclusion $J'(x_1, x_2) \subseteq J'(z_1, z_2)$ is shown. Consider now an arbitrary $j \in J'(z_1, z_2)$ given by a vertex $\tilde{t} \in \mathcal{N}(z_1, z_2)$, $\delta(z_1, z_2, \tilde{t}) = 0$. Vertex $\tilde{t}$ may only be marked in one of the aforementioned steps of the algorithm. Thus, there exists a vertex $t \in \mathcal{N}(x_1, x_2)$, $\delta(t, x_1, x_2) = 0$ such that $j(x_1, x_2, t) = j(z_1, z_2, \tilde{t})$. The reverse inclusion is shown, as well as the equality $J'(x_1, x_2) = J'(z_1, z_2)$.

We verify that

$$I'(x_1, x_2) = I'(z_1, z_2).$$

If $T(x_1, x_2)$ is empty, then $T(z_1, z_2)$ is also empty and the equality boils down to $I(x_1, x_2) = I(z_1, z_2)$, which can be proven analogously to the similar one in Lemma 22. Let $T(x_1, x_2)$ be nonempty. If $I'(x_1, x_2) = 1$, then $J'(x_1, x_2)$ is nonempty, and in Step (ii) the algorithm marks the vertex $r_j \in \mathcal{N}(z_1, z_2)$, with $\delta(z_1, z_2, r_j) = 0$, which guarantees $I'(z_1, z_2) = 1$. If, on the other hand, $I'(x_1, x_2) = 0$, then $J'(x_1, x_2)$ is empty and the algorithm does not mark any vertices joined with both z_1 and z_2, but for two adjacent vertices. Thus, $I'(z_1, z_2) = 0$, and the equality $I'(x_1, x_2) = I'(z_1, z_2)$ is shown.

We verify that

$$\Sigma(x_1, x_2) = \Sigma(z_1, z_2).$$

If $S(x_1, x_2)$ is empty, then either $|\mathcal{N}(x_1, x_2)| \leq 1$ and consequently $|\mathcal{N}(z_1, z_2)| \leq 1$ and $S(z_1, z_2)$ is also empty, or $|\mathcal{N}(x_1, x_2)| > 1$; but in this case the algorithm does not mark any vertices adjacent to each vertex of $\mathcal{N}(z_1, z_2)$, but for z_1 and z_2. Thus, $S(z_1, z_2)$ is empty. To sum up, if $S(x_1, x_2)$ is empty, then $S(z_1, z_2)$ is also empty and $\Sigma(x_1, x_2) = \varnothing = \Sigma(z_1, z_2)$.

Let $S(x_1, x_2)$ be nonempty. If $T(x_1, x_2)$ is empty, then the proof of the equality $\Sigma(x_1, x_2) = \Sigma(z_1, z_2)$ is similar to the corresponding equality in Lemma 22. Let $T(x_1, x_2)$ be nonempty. Denote the elements of $\mathcal{N}(x_1, x_2)$ by t_1, t_2. Consider an arbitrary $\sigma \in \Sigma(x_1, x_2)$ given by a vertex $s \in \mathcal{N}^U(t_1, t_2)$. In Step (i), the vertex $r_\sigma \in \mathcal{N}^U(t_1, t_2)$ is marked. Let us show that $\sigma(x_1, x_2, s) = \sigma(z_1, z_2, r_\sigma)$. Fix $k \in \{0, 1\}$. If $\sigma_k(x_1, x_2, s) = 1$, then in Step (i) the algorithm marks a vertex $r_k \in \mathcal{N}(z_k, r_\sigma)$ that is not adjacent to z_{2-k}. Thus, $\sigma_k(z_1, z_2, r_\sigma) = 1$. If $\sigma_k(x_1, x_2, s) = 0$, then each vertex of Z adjacent to z_k and r_σ is a vertex of $\mathcal{N}(z_1, z_2)$. Consequently, $\sigma_k(z_1, z_2, r_\sigma) = 0$. Thus, $\sigma(x_1, x_2, s) = \sigma(z_1, z_2, r_\sigma)$. Finally, a vertex from $S(z_1, z_2)$ may only be marked in Step (i) as r_σ for some $\sigma \in \Sigma(x_1, x_2)$. The equality $\Sigma(x_1, x_2) = \Sigma(z_1, z_2)$ is verified.

Let us show that

$$\tau(x_1, x_2) = \tau(z_1, z_2).$$

If $\tau(x_1, x_2)$ is empty, then the algorithm does not mark any pair of $\mathcal{N}(z_1, z_2)$ adjacent to one another, and $\tau(z_1, z_2)$ is empty. Let $\tau(x_1, x_2)$ be nonempty. Denote by t_1, t_2 the vertices of $\mathcal{N}(x_1, x_2)$ adjacent to one another. The algorithm marks $\tilde{t}_1, \tilde{t}_2 \in \mathcal{N}(z_1, z_2)$ adjacent to one another, and by what we have proved above, we have $j(x_1, x_2, t_k) = j(z_1, z_2, \tilde{t}_k)$ for $k = 1, 2$. There are no other pairs of vertices in Z adjacent to one another and to z_1 and z_2. The equality $\tau(x_1, x_2) = \tau(z_1, z_2)$ is verified. $\square$

5. Proof of the main theorem

Proof of Theorem 7. By Theorem 8, it is sufficient to show that there exists an $\varepsilon > 0$ such that for each $\alpha \in \left(\frac{1}{2}, \frac{1}{2}+\varepsilon\right)$ Duplicator a.a.s. has a winning strategy in the game $\mathrm{EHR}(A, B, 4)$, where $A \sim G(N, N^{-\alpha})$ and $B \sim G(M, M^{-\alpha})$ are independent random graphs.

Let $\varepsilon_1, \varepsilon_2, \varepsilon_3$ be the values obtained from Lemmas 22, 28, 32. Let

$$\varepsilon = \min\{\varepsilon_1, \varepsilon_2, \varepsilon_3\}.$$

Consider an $\alpha \in \left(\frac{1}{2}, \frac{1}{2}+\varepsilon\right)$.

Consider a set $\mathcal{P}$ of all graph pairs (X, Y) satisfying the following conditions:

(1) For each $x_1 \in V(X)$ there exists $y_1 \in V(Y)$ such that $\mathcal{J}^i(x_1) = \mathcal{J}^i(y_1)$ for all i; and, vice versa, for each $y_1 \in V(Y)$ there exists $x_1 \in V(X)$ such that $\mathcal{J}^i(x_1) = \mathcal{J}^i(y_1)$ for all i.

(2) For each pair of vertices $x_1, x_2 \in V(X)$ such that $x_1 \sim x_2$, $x_2 \in U^0(x_1)$ and $y_1 \in V(Y)$, there exists $y_2 \in V(Y)$, $y_2 \sim y_1$ such that the conditions of Lemma 28 hold, and vice versa.

(3) For each pair of vertices $x_1, x_2 \in V(X)$ such that $x_1 \not\sim x_2$, $x_2 \in U^0(x_1)$ and $y_1 \in V(Y)$, there exists $y_2 \in V(Y)$, $y_2 \not\sim y_1$ such that the conditions of Lemma 32 hold, and vice versa.

(4) For each subgraph $S \subseteq X$ with $v(X) \leq 3$ and each $\frac{1}{2}$-safe pair (G, H) such that $v(G, H) \leq 22\cdot 23\cdot 2^{22}$ in X, there exists a strict (G, H)-extension of S that is (K, T)-maximal for each $\frac{1}{2}$-rigid or $\frac{1}{2}$-neutral (K, T) with $v(K, T) \leq 6$, and vice versa.

From Lemmas 22, 28, 32, and Theorem 12 it follows that $\mathsf{P}((A, B) \in \mathcal{P}) \to 1$. Let us prove that in such conditions Duplicator has a winning strategy.

For simplicity, we will often use the following denotation: Let A be an induced subgraph of some graph Γ (for example, $\Gamma = X, Y$) and $a_1, \ldots, a_l$ are some vertices from $V(\Gamma) \setminus V(A)$. By $A + a_1 + \ldots + a_l$ we denote the graph $\Gamma|_{V(A)\cup\{a_1,\ldots,a_l\}}$.

Without loss of generality we may assume that in the first round Spoiler chooses a vertex $x_1 \in V(X)$. Duplicator chooses a vertex $y_1 \in (Y)$ such that

$$\mathcal{J}^i(x_1) = \mathcal{J}^i(y_1)$$

for all i.

From now on during the proof we will only use the equality $\mathcal{J}^i(x_1) = \mathcal{J}^i(y_1)$ and Properties 1–4 of the pair (X, Y). We will not use the explicit construction from Lemma 22. Thus, X and Y are symmetric.

Without loss of generality we may assume that in the second round Spoiler chooses a vertex $x_2 \in V(X)$. We examine several cases.

Case 1. Let x_2 be in a subgraph $U_X \in \mathcal{U}_i(x_1)$ for some $i \geq 1$. Due to the equality $\mathcal{J}^i(x_1) = \mathcal{J}^i(y_1)$, the graph Y contains a y_1-bad subgraph U_Y of type i for which we have

$$T^i(U_X) = T^i(U_Y).$$

Let $\varphi : V(U_X) \to V(U_Y)$ be an isomorphism between U_X and U_Y. Duplicator answers by choosing $y_2 = \varphi(x_2) \in V(Y)$. Note that X and Y are still symmetric.

Without loss of generality we may assume that in the third round Spoiler chooses a vertex of $x_3 \in V(X)$. We now examine several subcases of Case 1.

Subcase 1a. Assume that $x_3 \in V(U_X)$.

Duplicator answers by choosing

$$y_3 = \varphi(x_3) \in V(Y).$$

In round 4, without loss of generality, Spoiler chooses $x_4 \in V(X)$. If $x_4 \in V(U_X)$, then Duplicator chooses $y_4 = \varphi(x_4) \in V(Y)$ and wins due to isomorphism of U_X and U_Y.

Otherwise, if x_4 is joined with not more than one vertex from those chosen before, then the pair $(X|_{\{x_1,x_2,x_3,x_4\}}, X|_{\{x_1,x_2,x_3\}})$ is $\frac{1}{2}$-safe. By Property 4 of the pair (X, Y), in Y there exists a vertex y_4 such that $Y|_{\{y_1,y_2,y_3,y_4\}}$ is a strict $(X|_{\{x_1,x_2,x_3,x_4\}}, X|_{\{x_1,x_2,x_3\}})$-extension of $Y|_{\{y_1,y_2,y_3\}}$. Duplicator chooses y_4 and wins.

If x_4 is adjacent to each of the vertices x_1, x_2, x_3, then $(X|_{\{x_1,x_2,x_3,x_4\}}, X|_{\{x_1,x_2,x_3\}})$ is $\frac{1}{2}$-rigid with $v(X|_{\{x_1,x_2,x_3,x_4\}}, X|_{\{x_1,x_2,x_3\}}) = 1$. Therefore, the subgraph U_X is not x_1-bad in X, which contradicts the definition of U_X.

Let x_4 be adjacent to exactly two of the vertices x_1, x_2, x_3, say x_k and x_n. Then we have $\delta^{U_X}(x_k, x_n) = 1 = \delta^{U_Y}(y_k, y_n)$. By definition, in $V(Y) \setminus V(U_Y)$ there exists a vertex y_4 adjacent to y_k and y_n. Moreover, y_4 is not adjacent to the remaining vertex, same as x_4 cannot be adjacent to each of x_ℓ for $1 \le \ell \le 3$ at the same time. Thus, Duplicator chooses y_4 and wins due to isomorphism of U_X and U_Y and properties of x_4, y_4.

Subcase 1a has been examined.

If $x_3 \notin V(U_X)$, then $|\mathcal{N}(x_3) \cap V(U_X)| \le 2$. Otherwise, the pair $(U_X + x_3, U_X)$ is $\frac{1}{2}$-rigid, with $v(U_X + x_3, U_X) = 1$, which contradicts the fact that U_X is x_1-bad.

Subcase 1b. Assume that $x_3 \notin V(U_X)$ and $|\mathcal{N}(x_3) \cap V(U_X)| = 2$.

Let z_{1X} and z_{2X} be the vertices from U_X adjacent to x_3. Put $z_{1Y} = \varphi(z_{1X})$, $z_{2Y} = \varphi(z_{2X})$. From the equality $J^{U_X}(z_{1X}, z_{2X}) = J^{U_Y}(z_{1Y}, z_{2Y})$, it follows that in Y there exists a vertex $y_3 \in \mathcal{N}^{U_Y}(z_{1Y}, z_{2Y})$ such that

$$j^{U_X}(x_3) = j^{U_Y}(y_3).$$

Duplicator chooses y_3. Note that for all $z \in V(U_X)$, we have $x_3 \sim z$ if and only if $y_3 \sim \varphi(z)$.

In round 4, without loss of generality, Spoiler chooses $x_4 \in V(X)$. If $x_4 \in V(U_X)$, then Duplicator chooses $y_4 = \varphi(x_4)$ and wins.

Otherwise, if x_4 is adjacent to not more than one vertex from x_1, x_2, x_3, then Duplicator wins due to α-safeness of $(X|_{\{x_1,x_2,x_3,x_4\}}, X|_{\{x_1,x_2,x_3\}})$.

If x_4 is adjacent to each of the vertices x_1, x_2, x_3, then the pair $(U_X + x_3 + x_4, U_X)$ is $\frac{1}{2}$-rigid, with $v(U_X + x_3 + x_4, U_X) = 2$, which contradicts the fact that U_X is x_1-bad.

If x_4 is adjacent to exactly two of x_1, x_2, x_3, then two possibilities may occur.

If x_4 is adjacent to x_1, x_2, then $x_4 \in \mathcal{N}^{U_X}(x_1, x_2)$. Let us show that there is a vertex in $V(Y) \setminus V(U_Y)$ adjacent to y_1 and y_2, different from y_3. If $x_3 \in \mathcal{N}^{U_X}(x_1, x_2)$, then $I^{U_X}(x_1, x_2) = 1 = I^{U_Y}(y_1, y_2)$. If, on the other hand, $x_3 \notin \mathcal{N}^{U_X}(x_1, x_2)$, then $y_3 \notin \mathcal{N}^{U_Y}(y_1, y_2)$ and $\delta^{U_X}(x_1, x_2) = 1 = \delta^{U_Y}(y_1, y_2)$. In both cases there exists a vertex $y_4 \in V(Y) \setminus V(U_Y)$ adjacent to y_1 and y_2. Note that y_4 cannot be adjacent to y_3, otherwise U_Y is not y_1-bad in Y. Thus, Duplicator chooses y_4 and wins.

Finally, let x_4 be adjacent to x_3 and some x_k for $k \in \{1, 2\}$, but not adjacent to x_l, $l \in \{1, 2\} \setminus \{k\}$. Consider the equality $j^{U_X}(x_3) = j^{U_Y}(y_3)$. We have

$$j^{U_X}_{x_k}(x_3) = 1 = j^{U_Y}_{y_k}(y_3),$$

which implies the existence of a vertex $y_4 \in \mathcal{N}^{U_Y}(y_k, y_3, \neg y_l)$ in Y. Duplicator chooses $y_4 \in V(Y)$ and wins.

Subcase 1c. Assume that $x_3 \notin V(U_X)$, $|\mathcal{N}(x_3) \cap V(U_X)| = 1$, and in $V(X) \setminus V(U_X)$ there exists a vertex t_X adjacent to x_3 such that $|\mathcal{N}(t_X) \cap V(U_X)| = 2$.

Denote by z_{1X} a vertex of U_X adjacent to x_3, and put $z_{1Y} = \varphi(z_{1X})$.

Let z_{2X} and z_{3X} be the vertices of U_X adjacent to t_X. Put $z_{2Y} = \varphi(z_{2X})$, $z_{3Y} = \varphi(z_{3X})$. Due to $J^{U_X}(z_{2X}, z_{3X}) = J^{U_Y}(z_{2Y}, z_{3Y})$, there exists a vertex $t_Y \in \mathcal{N}^{U_Y}(z_{2Y}, z_{3Y})$ such that the equality $j^{U_X}(t_X) = j^{U_Y}(t_Y)$ holds. Note that for all $z \in V(U_X)$, we have $t_X \sim z$ if and only if $t_Y \sim \varphi(z)$). Define

$$\begin{aligned} S_X &= \bigl\{z \in V(U_X) \mid \delta^U(x_3, z, \neg(V(U_X) \setminus z))\bigr\};\\ S_Y &= \bigl\{\varphi(z) \mid z \in S_X\bigr\}. \end{aligned}$$

From the equality

$$j^{U_X}_{z_{1X}, S_X}(t_X) = j^{U_Y}_{z_{1Y}, S_Y}(t_Y),$$

it follows that in $\mathcal{N}^{U_Y}(t_Y, z_{1Y}, \neg(V(U_Y) \setminus \{z_{1Y}\}))$ there exists a vertex y_3 such that

$$\Bigl(\bigwedge_{z \in S_Y} \delta^{U_Y}\bigl(y_3, z, \neg t_Y, \neg(V(U_Y) \setminus \{z\})\bigr)\Bigr) \wedge \Bigl(\bigwedge_{z \in V(U_Y) \setminus S_Y} \neg\delta^{U_Y}\bigl(y_3, z, \neg t_Y, \neg(V(U_Y) \setminus \{z\})\bigr)\Bigr).$$

Duplicator chooses such a vertex $y_3 \in V(Y)$.

Without loss of generality, in round 4 Spoiler chooses $x_4 \in V(X)$.

If $x_4 \in V(U_X)$, then Duplicator chooses a vertex $\varphi(y_4) \in V(Y)$ and, obviously, wins. Otherwise, if the vertex x_4 is adjacent to each of the previous vertices, then $(U_X + x_3 + x_4 + t_X, U_X)$ is $\frac{1}{2}$-rigid, with $v(U_X + x_3 + x_4 + t_X, U_X) = 3$; and, consequently, U_X is not x_1-bad, which leads to a contradiction.

If y_4 is adjacent to precisely two previously chosen vertices, then two options are possible. If y_4 is adjacent to x_1 and x_2, then Duplicator's winning strategy is analogous to the one in Subcase 1b.

Finally, let x_4 be adjacent to x_3 and some x_k with $k \in \{1, 2\}$, but not adjacent to x_ℓ, $\ell \in \{1, 2\} \setminus \{k\}$. Then, due to the choice of y_3 in Y there is a vertex

$$y_4 \in \mathcal{N}^{U_Y}\bigl(y_3, y_k, \neg t_Y, \neg(V(U_Y) \setminus \{y_k\})\bigr).$$

Duplicator chooses such a vertex y_4 and wins.

If y_4 is adjacent to one or none of the previous vertices, then Duplicator wins due to $\frac{1}{2}$-safeness of the corresponding pair of graphs.

Subcase 1d. Assume that $x_3 \notin V(U_X)$, $|\mathcal{N}(x_3) \cap V(U_X)| = 1$, and in $V(U_X) \setminus V(U)$ there is no vertex adjacent to x_3 for which $|\mathcal{N}(t_X) \cap V(U_X)| = 2$. Denote by z_{1X} a vertex from $V(U_X)$ adjacent to x_3, and put $z_{1Y} = \varphi(z_{1X})$.

Consider the pair of graphs $(\tilde{A}_1, \tilde{B}_1)$, where $V(\tilde{B}_1) = \{\tilde{z}_1, \dots, \tilde{z}_m, \tilde{t}\}$, that corresponds to $j^{U_X}(x_3)$. Let

$$\tilde{B}_1' = \tilde{B}_1|_{\{\tilde{z}_1, \dots, \tilde{z}_v\}} \quad \text{and} \quad \tilde{A}_1' = \tilde{A}_1 + \{\tilde{z}_1, t\},$$

which indicates the graph $\tilde{A}_1$ along with the edge $\{\tilde{z}_1, \tilde{t}\}$. Here, $\tilde{z}_1$ is the image of z_{1X} in $\tilde{B}_1'$.

Statement 35. The pair $(\tilde{A}'_1, \tilde{B}'_1)$ is $\frac{1}{2}$-safe.

Proof. The statement follows from the $\frac{1}{2}$-safeness of $(\tilde{B}_1, \tilde{B}'_1)$, the fact that $\tilde{A}'$ is a strict (K^*, T^*)-neighborhood of $\tilde{B}_1$, and also Statement 19. □

Finally, note that $v(\tilde{A}'_1, \tilde{B}'_1) \le m(m+1)2^m$. By Property 4 of pair (X, Y), in Y there is a strict (K^*, T^*)-maximal $(\tilde{A}'_1, \tilde{B}'_1)$-extension of subgraph U_Y. Thus, Duplicator chooses $y_3 \in V(Y)$ that corresponds to $\tilde{t} \in V(\tilde{A}_1)$. It is easy to verify that

$$j^{U_X}(x_3) = j^{U_Y}(y_3).$$

Indeed, the only vertices that may influence the value $j^{U_Y}(y_3)$ are vertices from (K^*, T^*)-neighborhood of $U_Y + y_3$. But, from (K^*, T^*)-maximality of the strict $(\tilde{A}'_1, \tilde{B}'_1)$-extension, it follows that the (K^*, T^*)-neighborhood coincides with an already found extension.

Without loss of generality assume Spoiler chooses $x_4 \in V(X)$ in round 4.

If $x_4 \in V(U_X)$, then Duplicator chooses a vertex $y_4 = \varphi(x_4) \in V(Y)$ and wins. Otherwise, if x_4 is adjacent to exactly two vertices chosen before, then Duplicator wins due to the equalities

$$j^{U_X}(x_1, x_2, x_3) = j^{U_Y}(y_1, y_2, y_3) \quad \text{and} \quad \delta^{U_X}(x_1, x_2) = \delta^{U_Y}(y_1, y_2)$$

and the fact that in $V(Y) \setminus V(U_Y)$ there is no vertex adjacent to all three chosen before. If Spoiler chooses a vertex that is adjacent to not more than one of the previously chosen, then Duplicator wins due to the $\frac{1}{2}$-safeness of a corresponding extension. Spoiler also cannot choose a vertex that is adjacent to all three of the previous ones and not in U_X.

Subcase 1e. Assume that $x_3 \notin V(U_X)$, x_3 does not have neighbors in U_X and $\delta^{U_X}(x_1, x_2, x_3) = 0$. This case can be verified analogously to Subcase 1d. In Subcase 1e, we need to remove the edge $\{\tilde{z}_1, \tilde{t}\}$ from the pair $(\tilde{A}'_1, \tilde{B}'_1)$.

Subcase 1f. Assume that $x_3 \notin V(U_X)$, x_3 does not have neighbors in U_X and $\delta^{U_X}(x_1, x_2, x_3) = 1$.

Let $x_3 \notin S^{U_X}(x_1, x_2)$. Let t_X be a vertex of $\mathcal{N}^{U_X}(x_1, x_2, x_3)$.

Consider the pair of graphs $(\tilde{A}_2, \tilde{B}_2)$, where $V(\tilde{B}_2) = \{\tilde{z}_1, \dots, \tilde{z}_v, \tilde{t}_X, \tilde{t}\}$, that corresponds to $j^{U_X+t_X}(x_3)$. Put

$$\tilde{B}'_2 = \tilde{B}_2|_{\{\tilde{z}_1,\dots,\tilde{z}_v,\tilde{t}_X\}}, \tilde{A}'_2 = \tilde{A}_2 + \{\tilde{t}_X, \tilde{t}\},$$

where $\tilde{t}_X$ is an image of t_X in B'. As in Statement 35, the pair $(\tilde{A}', \tilde{B}')$ is $\frac{1}{2}$-safe. Finally, note that $v(\tilde{A}'_1, \tilde{B}'_1) \le (m+1)(m+2)2^{m+1}$.

Since $\delta^{U_Y}(y_1, y_2) = 1$, in $\mathcal{N}^{U_Y}(y_1, y_2)$ there is at least one vertex t_Y. By Property 4 of the pair (X, Y), in Y there is a strict (K^*, T^*)-maximal $(\tilde{A}'_2, \tilde{B}'_2)$-extension of $U_Y + t_Y$. Thus, Duplicator chooses a vertex $y_3 \in V(Y)$ that corresponds to $\tilde{t} \in V(\tilde{A}_2)$. It is not hard to verify that

$$j^{U_X+t_X}(x_3) = j^{U_Y+t_Y}(y_3).$$

Moreover, not a single vertex of U_X may be adjacent to y_3, due to strictness of the found $(\tilde{A}'_2, \tilde{B}'_2)$-extension.

Without loss of generality, in round 4 Spoiler chooses a vertex $x_4 \in V(X)$. If $x_4 \in V(U_X)$, then Duplicator chooses $y_4 = \varphi(x_4)$ and wins. If $x_4 \in \mathcal{N}^{U_X}(x_1, x_2, x_3)$, then Duplicator chooses t_Y and wins. If x_4 is adjacent to no more than two of the previous vertices, then two options are possible. If x_4 is adjacent to x_1 and x_2, then $I^{U_X}(x_1, x_2) = 1 = I^{U_Y}(y_1, y_2)$. Spoiler chooses an arbitrary element of

$\mathcal{N}^{U_Y}(y_1, y_2)$, different from t_Y, as y_4. Due to strictness of the found $(\tilde{A}_2', \tilde{B}_2')$-extension, y_4 is not adjacent to y_3, and Duplicator wins. If x_4 is adjacent to x_3 and x_k for some $k \in \{1, 2\}$, then the existence of the winning strategy for Duplicator can be verified as in Subcase 1d.

Let $x_3 \in S^{U_X}(x_1, x_2)$. Due to the equality $\Sigma^{U_X}(x_1, x_2) = \Sigma^{U_Y}(y_1, y_2)$, in Y there is a vertex $y_3 \in S^{U_Y}(y_1, y_2)$ such that

$$\sigma^{U_X}(x_1, x_2, x_3) = \sigma^{U_Y}(y_1, y_2, y_3).$$

Duplicator chooses this vertex y_3.

Without loss of generality, in round 4 Spoiler chooses a vertex $x_4 \in V(X)$. If $x_4 \in V(U_X)$, then Duplicator chooses a vertex $y_4 = \varphi(x_4)$ and wins, since x_4 is not adjacent to x_3 and y_4 is not adjacent to y_3. Otherwise, if x_4 is adjacent to each of the previously chosen, then it is enough to verify that there is a vertex $y_4 \in V(Y)$ adjacent to each vertex chosen before in Y. First, due to inclusion $y_3 \in S^{U_Y}(y_1, y_2)$, the set $\mathcal{N}^{U_Y}(y_1, y_2)$ is nonempty; and, second, each element of this set is adjacent to y_3. Hence, such a vertex y_4 exists.

If x_4 is adjacent to exactly two of the previously chosen vertices, then one of them is x_3, since $x_3 \in S^{U_X}(x_1, x_2)$, which means that x_3 is adjacent to each common neighbor of x_1 and x_2. Let also $x_k \sim x_4$ for some $k \in \{1, 2\}$. Then we have $\sigma_k^{U_X}(x_1, x_2, x_3) = 1 = \sigma_k^{U_Y}(y_1, y_2, y_3)$, which implies the existence of $y_4 \in V(Y)$ adjacent to y_3 and y_k, but not adjacent to x_l, $l \in \{1, 2\} \setminus \{k\}$. Thus, Duplicator chooses x_4 and wins. If x_4 is adjacent to not more than one of the previous vertices, then Duplicator wins due to $\frac{1}{2}$-safeness of the corresponding pair of graphs.

Case 1 has been examined.

Case 2. Let x_2 be an element of the 0-neighborhood of x_1, with $x_2 \sim x_1$. By Property 2 of the pair (X, Y), in Y there is a vertex y_2 adjacent to y_1 for which the conclusion of Lemma 28 holds.

Duplicator chooses this vertex y_2. All properties of the graphs X and Y are symmetric. Let $U_X = X|_{\{x_1,x_2\}}$ and $U_Y = Y|_{\{y_1,y_2\}}$. Further analysis of Case 2 is analogous to that of Case 1.

Case 3. Let x_2 be an element of the 0-neighborhood of x_1, with $x_2 \not\sim x_1$. By definition of (X, Y), in Y there is a vertex y_2 not adjacent to y_1 such that Lemma 32 conditions hold. Duplicator chooses a vertex y_2. Note that X and Y are now symmetric.

Without loss of generality, in round 3 Spoiler chooses a vertex $x_3 \in V(X)$.

Subcase 3a. Assume that x_3 is adjacent to both x_1, x_2, and in X there is a vertex t_X adjacent to each x_k for $1 \le k \le 3$. Since $\tau(x_1, x_2) = \tau(y_1, y_2)$, in Y there are vertices $y_3, t_Y \in \mathcal{N}(y_1, y_2)$ such that $j(x_1, x_2, x_3) = j(y_1, y_2, y_3)$ and $t_Y \sim y_3$. Duplicator chooses the vertex y_3.

Further, without loss of generality, in round 4 Spoiler chooses a vertex $x_4 \in V(X)$. If x_4 is adjacent to each of the previous vertices, then Duplicator chooses $t_Y \in V(Y)$ and wins. If x_4 is adjacent to each of x_1, x_2, then

$$I'(x_1, x_2) = 1 = I'(y_1, y_2).$$

Thus, in $\mathcal{N}(y_1, y_2)$ there is at least one vertex besides y_3 and t_Y, with $y_4 \not\sim y_3$. Therefore Duplicator wins. If x_4 is adjacent to x_3 and some x_k for $1 \le k \le 2$, then Duplicator wins due to the equality $j(x_1, x_2, x_3) = j(y_1, y_2, y_3)$. If, finally, x_4 is adjacent to not more than one of the previous vertices, then Duplicator wins due to the $\frac{1}{2}$-safeness of a corresponding extension.

Subcase 3b. Assume that x_3 is adjacent to both x_1, x_2 and in X there is no vertex adjacent to each of x_k for $1 \le k \le 3$. Since $J'(x_1, x_2) = J'(y_1, y_2)$, in Y there exists a vertex y_3 such that firstly,

$$j(x_1, x_2, x_3) = j(y_1, y_2, y_3);$$

and, second, in Y there is no vertex adjacent to each of y_k for $1 \le k \le 3$. With this note, Subcase 3b is similar to Subcase 1a.

Subcase 3c. Assume that x_3 is adjacent to exactly one of x_1, x_2. This subcase is similar to Subcases 1c and 1d.

Subcase 3d. Assume that x_3 is not adjacent to any of the previous vertices. The item is similar to Subcases 1e and 1f.

Case 3 has been examined, and the theorem is fully proven. □

6. Conclusion

In this paper, we achieved progress in studying limit points of 4-spectrum within the context of the zero-one k-law. Future research on this topic can be aimed at studying the last possible limit point of 4-spectrum, namely $\frac{3}{5}$.

References

[Alon and Spencer 1992] N. Alon and J. H. Spencer, *The probabilistic method*, Wiley, New York, 1992. MR Zbl

[Ehrenfeucht 1961] A. Ehrenfeucht, "An application of games to the completeness problem for formalized theories", *Fund. Math.* **49** (1961), 129–141. MR Zbl

[Janson et al. 2000] S. Janson, T. Łuczak, and A. Rucinski, *Random graphs*, Wiley, New York, 2000. MR Zbl

[Libkin 2004] L. Libkin, *Elements of finite model theory*, Springer, 2004. MR Zbl

[Matushkin and Zhukovskii 2018] A. D. Matushkin and M. E. Zhukovskii, "First order sentences about random graphs: small number of alternations", *Discrete Appl. Math.* **236** (2018), 329–346. MR Zbl

[Ruciński and Vince 1985] A. Ruciński and A. Vince, "Balanced graphs and the problem of subgraphs of a random graph", *Congr. Numer.* **49** (1985), 181–190. MR

[Shelah and Spencer 1988] S. Shelah and J. Spencer, "Zero-one laws for sparse random graphs", *J. Amer. Math. Soc.* **1**:1 (1988), 97–115. MR Zbl

[Spencer 2001] J. Spencer, *The strange logic of random graphs*, Algor. Combinator. **22**, Springer, 2001. MR Zbl

[Zhukovskii 2012] M. Zhukovskii, "Zero-one k-law", *Discrete Math.* **312**:10 (2012), 1670–1688. MR Zbl

[Zhukovskii 2016] M. E. Zhukovskii, "On infinite spectra of first order properties of random graphs", *Mosc. J. Comb. Number Theory* **6**:4 (2016), 73–102. MR Zbl

[Zhukovskii and Raigorodskii 2015] M. E. Zhukovskii and A. M. Raigorodskii, "Random graphs: models and asymptotic characteristics", *Uspekhi Mat. Nauk* **70**:1(421) (2015), 35–88. In Russian; translated in *Russian Math. Surv.* **70**:1 (2015), 33–81. MR Zbl

Received 4 Dec 2019. Revised 4 May 2020.

YURY YAROVIKOV:
yu-rovikov@yandex.ru
Moscow Institute of Physics and Technology, Moscow, Russia

Moscow Journal of Combinatorics and Number Theory
Vol. 9, No. 3, 2020
https://doi.org/10.2140/moscow.2020.9.333

Weakly distinguishing graph polynomials on addable properties

Johann A. Makowsky and Vsevolod Rakita

A graph polynomial P is weakly distinguishing if for almost all finite graphs G there is a finite graph H that is not isomorphic to G with $P(G) = P(H)$. It is weakly distinguishing on a graph property $\mathcal{C}$ if for almost all finite graphs $G \in \mathcal{C}$ there is $H \in \mathcal{C}$ that is not isomorphic to G with $P(G) = P(H)$. We give sufficient conditions on a graph property $\mathcal{C}$ for the characteristic, clique, independence, matching, and domination and ξ polynomials, as well as the Tutte polynomial and its specializations, to be weakly distinguishing on $\mathcal{C}$. One such condition is to be addable and small in the sense of C. McDiarmid, A. Steger and D. Welsh (2005). Another one is to be of genus at most k.

1. Introduction and preliminaries

Unless otherwise stated, we only consider simple (finite, loopless, undirected graphs with no parallel edges) graphs with vertices labeled $1, \dots, n$. For a graph $G = (V(G), E(G))$ and $A \subseteq E(G)$, we denote by $G\langle A\rangle$ the graph $(V(G), A)$, and by $k(A)$ the number of connected components of $G\langle A\rangle$. A graph property is a family of graphs that is closed under isomorphisms. For a graph property $\mathcal{C}$, denote by $\mathcal{C}(n)$ the graphs of order n in $\mathcal{C}$. We only consider properties such that $\mathcal{C}(n)$ is nonempty for all sufficiently large n.

Let P be a graph polynomial. We say that two non isomorphic graphs G and H are P-mates if $P(G) = P(H)$, and that G is P unique if it has no P mates. P is *trivial* if all graphs G, H are P-mates. P is *complete* if all graphs G are P-unique.

In this paper we investigate conditions which imply that almost all graphs in a graph property $\mathcal{C}$ have a P-mate. More formally, we give the following definitions:

V. Rakita was partially supported by a Jacobs Scholarship for excellent students.
MSC2010: primary 05C31; secondary 05C10, 05C30, 05C69, 05C80.
Keywords: graph polynomials, random graphs, Bollobás–Pebody–Riordan conjecture, addable graph classes.

Let P be a graph polynomial, and denote by $\mathcal{G}(n)$ the family of graphs of order n with vertices labeled $1, \dots, n$, and by $U_P(n)$ the set of P unique graphs of order n.

Definition 1.1. P is *weakly distinguishing* if

$$\lim_{n\to\infty} \frac{|U_P(n)|}{|\mathcal{G}(n)|} = 0$$

and P is *almost complete* if

$$\lim_{n\to\infty} \frac{|U_P(n)|}{|\mathcal{G}(n)|} = 1$$

In [Bollobás et al. 2000], Bollobás, Pebody and Riordan conjectured:

Conjecture 1 (BPR conjecture). The chromatic and Tutte polynomials are almost complete.

In [Makowsky and Zhang 2019], the analogous question for r-regular hypergraphs was considered, and for $r \geq 3$ the conjecture was refuted. In [Noy 2003] it was observed, as a remark in the conclusions, that the independence polynomial $\mathrm{In}(G; x)$, discussed in Section 7, is weakly distinguishing on all finite graphs. In [Makowsky and Rakita 2019], it was proven that an infinite number of graph polynomials, among them the independence, clique and harmonious polynomials, are weakly distinguishing.

A natural way to approach the question whether a graph polynomial P is weakly distinguishing, almost complete or otherwise, is to ask, given a graph property $\mathcal{C}$, whether almost all graphs in $\mathcal{C}$ are P unique in $\mathcal{C}$. Let $\mathcal{C}$ be a graph property and denote by $U_{P,\mathcal{C}}(n) = \{G \in \mathcal{C}(n) : G$ has no P-mate in $\mathcal{C}\}$.

Definition 1.2. P is *weakly distinguishing on* $\mathcal{C}$ if

$$\lim_{n\to\infty} \frac{|U_{P,\mathcal{C}}(n)|}{|\mathcal{C}(n)|} = 0,$$

and P is *almost complete on* $\mathcal{C}$ if

$$\lim_{n\to\infty} \frac{|U_{P,\mathcal{C}}(n)|}{|\mathcal{C}(n)|} = 1.$$

In this paper we prove that many well studied graph polynomials are weakly distinguishing on infinitely many graph properties $\mathcal{C}$, listed in Example 3.2. However, all these properties $\mathcal{C}$ are small (in the sense that they are sets of measure 0 in the collection of all graphs), so these results do not imply that any of the above polynomials are weakly distinguishing or almost complete for arbitrary properties $\mathcal{C}$.

The graph polynomials which we show are weakly distinguishing for $\mathcal{C}$ include the characteristic polynomial, the domination polynomial, and the ξ-polynomial, which is a generalization of the both the matching and the Tutte polynomial. These three polynomials are mutually incomparable in distinctive power by Proposition 2.3. Once we know that ξ is weakly distinguishing for a graph property $\mathcal{C}$, this holds also for all the graph polynomials which are substitution instances of ξ in $\mathcal{C}$. This includes the matching polynomials, the independence polynomial, the chromatic polynomial, the Tutte polynomial, the Euler polynomial, and many others; see Section 7.

This paper is organized as follows: In Section 2 we introduce a method to compare distinctive power of graph polynomials and formulate how to use it (Lemma A). In Section 3, we give a graph theoretic background on addable properties, and graphs of genus at most k, and formulate our main tools, Lemma B and Lemma C. In Sections 4 and 5 we prove our results: Theorems 4.4 and 4.7 for the characteristic polynomial, Theorems 5.5 and 5.8 for the domination polynomial. In Section 6 we prove the same for the ξ-polynomial and in Section we apply Lemma A to derive the corresponding results for the matching polynomials, the independence polynomial, the chromatic polynomial, the Tutte polynomial, the Euler polynomial, and many others. In Section 8 we draw conclusions and present open problems.

2. Comparing graph polynomials

In this section we provide a tool (Lemma A) which allows us to show that many graph polynomials are weakly distinguishing.

Definition 2.1. Let $\mathcal{C}$ be a graph property, P and Q be two graph polynomials and G and H two finite graphs.

(i) G and H are *similar* if they have the same number of vertices, edges and connected components.

(ii) $P <^{\mathcal{C}}_{d.p} Q$ or Q *is at least as distinctive as* P *in* $\mathcal{C}$ if for all graphs $G, H \in \mathcal{C}$, $Q(G; \bar{x}) = Q(H; \bar{x})$ implies $P(G; \bar{y}) = P(H; \bar{y})$.

(iii) $P \sim^{\mathcal{C}}_{d.p} Q$ or P *and* Q *are of the same distinctive power in* $\mathcal{C}$ if $P <^{\mathcal{C}}_{d.p} Q$ and $Q <^{\mathcal{C}}_{d.p} P$.

(iv) $P <^{\mathcal{C}}_{s.d.p} Q$ or P *and* Q *are of the same distinctive power in* $\mathcal{C}$ *on similar graphs* if if for all similar graphs $G, H \in \mathcal{C}$, $Q(G; \bar{x}) = Q(H; \bar{x})$ implies $Q(G; \bar{y}) = Q(H; \bar{y})$.

(v) $P \sim^{\mathcal{C}}_{s.d.p} Q$ if $P <^{\mathcal{C}}_{s.d.p} Q$ and $Q <^{\mathcal{C}}_{s.d.p} P$.

(vi) For all graph properties $\mathcal{C}$ $P <^{\mathcal{C}}_{d.p} Q$ implies $P <^{\mathcal{C}}_{s.d.p} Q$.

If $\mathcal{C}$ consists of all finite graphs, we omit it.

The partial preorders $<_{d.p.}$ and $<_{s.d.p.}$ between graph polynomials (or general graph invariants) are studied extensively in [Makowsky et al. 2019]. A complete (trivial) graph polynomial is a maximal (minimal) element with respect to $<_{d.p.}$

Example 2.2. (i) The chromatic polynomial $\chi(G; x)$ and the Tutte polynomial satisfy $\chi(G; x) <_{s.d.p.} T(G; x, y)$ but not $\chi(G; x) <_{d.p.} T(G; x, y)$; see Section 7, because $T(G; x, y)$ does not determine the order of G in the presence of isolated vertices.

(ii) The characteristic polynomial $P_A(G; x)$ and the matching polynomial (defect aka acyclic) $\mu(G; x)$ from Section 7 are $d.p.$-equivalent on forests; see [Godsil and Gutman 1981].

(iii) Let $\bar{G}$ be the (loopless) complement graph of a simple graph G, and $P(G; \bar{x})$ be a (possibly multivariate) graph polynomial. Put $\bar{P}(G; \bar{x}) = P(\bar{G}, \bar{x})$. Then $\bar{P}(G; \bar{x}) \sim_{d.p.} P(G, \bar{x})$.
If we relativize this to a graph property $\mathcal{C}$, $\bar{P}(G; \bar{x}) \sim^{\mathcal{C}}_{d.p.} P(G, \bar{x})$ holds provided $\mathcal{C}$ is closed under complement graphs.

Proposition 2.3. *The following graph polynomials are pairwise incomparable by* $<_{s.d.p}$:

(i) *The chromatic polynomial* $\chi(G; x)$ *and the independence polynomial* $\mathrm{In}(G; x)$ *from Section 7.*

(ii) *The characteristic polynomial* $P_A(G, x)$ *from Sections 4 and the chromatic polynomial* $\chi(G; x)$.

(iii) *The characteristic polynomial* $P_A(G; x)$ *and the independence polynomial* $\mathrm{In}(G; x)$.

(iv) *The characteristic polynomial* $P_A(G, x)$ *and the domination polynomial* $\mathrm{Dom}(G; x)$. *from Sections 4 and 5.*

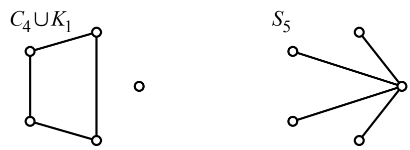


Figure 1. P_A-mates.

Figure 2. A path of length 5.

(v) *The characteristic polynomial and the ξ-polynomial from Section 6.*

(vi) *The domination polynomial and the ξ-polynomial.*

Proof. The first three examples are from [Makowsky et al. 2019].

For (iv)-(vi) we first note that all graphs of order less than 8 and all trees of order less than 10 are ξ-unique; see [Trinks 2012].

(A) Consider the graphs $C_4 \sqcup K_1$ and S_5 in Figure 1. We have

$$P_A(S_5, x) = P_A(C_4 \sqcup K_1, x) = x^2(x+2)(x-2), \mathrm{Dom}(S_5, 1) = 1, \quad \mathrm{Dom}(C_4 \sqcup K_1, 1) = 0,$$

and $\xi(S_5, x) \neq \xi(C_4 \sqcup K_1, x)$, since they are of order 5.

(B) Let P_5 and $\hat{P}_5$ be the graphs of order 5 shown in Figure 2. Then $\mathrm{Dom}(P_5, x) = \mathrm{Dom}(\hat{P}_5, x)$, since every dominating set of $\hat{P}_5$ is also a dominating set of P_5. Further,

$$P_A(P_5, x) = -x(x-1)(x+1)(x^2-3) \neq P_A(\hat{P}_5, x) = -x(x^2-x-3)(x^2+x-1),$$

hence $P_A(P_5, x) \neq P_A(\hat{P}_5, x)$. Thus $\xi(P_5, x) \neq \xi(\hat{P}_5, x)$, since both P_5 and $\hat{P}_5$ are ξ-unique.

(C) The graphs G_1 and G_2 of Figure 3 are of order 10. Therefore $\xi(G_1, x) = \xi(G_2, x)$. We also have

$$\begin{aligned}\mathrm{Dom}(G_1; x) &= x^{10} + 10x^9 + 40x^8 + 82x^7 + 92x^6 + 56x^5 + 16x^4,\\ \mathrm{Dom}(G_2; x) &= x^{10} + 10x^9 + 41x^8 + 86x^7 + 94x^6 + 48x^5 + 9x^4,\\ P_A(G_1; x) &= x^2(x^4 - x^3 - 4x^2 + 2x + 3)(x^4 + x^3 - 4x^2 - 2x + 3),\\ P_A(G_2; x) &= x^2(x-1)(x+1)(x^2-2)(x^4 - 5x^2 + 3),\end{aligned}$$

Hence $\mathrm{Dom}(G_1, x) \neq \mathrm{Dom}(G_2, x)$ and $P_A(G_1, x) \neq P_A(G_2, x)$.

Now (A) and (B) proves (iv), (A) and (C) proves (v), and (B) and (C) proves (vi). □

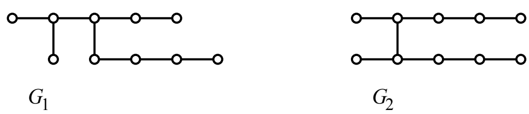


Figure 3. ξ-mates.

In Section 7 we use the following observation:

Lemma A. *Let $P(G;\bar{x})$ and $Q(G;\bar{y})$ two graph polynomials and $\mathcal{C}$ and $\mathcal{D}$ be graph properties with $\mathcal{D} \subseteq \mathcal{C}$. Assume that $P <^{\mathcal{C}}_{s.d.p} Q$ and Q is weakly distinguishing in $\mathcal{C}$. Then also P is weakly distinguishing in $\mathcal{C}$ but not necessarily in $\mathcal{D}$.*

Proof. Clearly, $P <^{\mathcal{C}}_{s.d.p} Q$ implies that $U_P(n) \subseteq U_Q(n)$. Hence

$$\lim_{n\to\infty} \frac{|U_P(n)|}{|\mathcal{G}(n)|} \leq \lim_{n\to\infty} \frac{|U_Q(n)|}{|\mathcal{G}(n)|}. \qquad \square$$

3. Addable properties and graphs of genus *k*

We discuss properties on which the Tutte, domination, matching, ξ, clique and characteristic polynomials are weakly distinguishing. These properties were studied in [McDiarmid, Steger, Welsh 2005; McDiarmid 2008].

Small addable classes.

Definition 3.1. A graph property $\mathcal{A}$ is *decomposable* if it is closed under disjoint union, and for all $G \in \mathcal{A}$ every component of G is in $\mathcal{A}$.

A graph property $\mathcal{A}$ is *bridge addable* if for each graph $G \in \mathcal{A}$ and every two vertices u, v in different components of G the graph obtained by adding an edge between u and v is also in $\mathcal{A}$.

A graph property $\mathcal{A}$ is *addable* if it is decomposable and bridge addable.

Example 3.2. The following properties are easily seen to be addable:

- planar graphs,
- outerplanar graphs,
- series-parallel graphs,
- graphs with tree width at most k for $k \geq 2$,
- k-colorable graphs for $k \geq 2$,
- graphs with no cycles of length greater than k,
- graphs with no K_k minor for $k \geq 2$.

Let $\mathcal{A}$ be a graph property. If $\mathcal{A}$ is minor closed (that is, closed under deletion of vertices and edges, and under contraction of edges), the Graph Minor theorem says that $\mathcal{A}$ is characterized by a finite set of forbidden minors (see [Lovász 2006] for more on graph minors and the graph minor theorem). We can characterize addable minor closed graph properties in terms of their forbidden minors:

Proposition 3.3 [McDiarmid 2009]. *Let $\mathcal{A}$ be a minor closed graph property. Then $\mathcal{A}$ is addable if and only if each excluded minor of $\mathcal{A}$ is 2-connected.*

Note that any nonempty, minor closed addable graph property $\mathcal{A}$ contains all forests, as it contains the graph with a single vertex, and is closed under taking disjoint unions of graphs in $\mathcal{A}$ and under adding edges between different components.

In addition to being addable, we will require the properties we consider to be small:

Definition 3.4. Let $\mathcal{A}$ be a graph property, and denote by $\mathcal{A}_n$ the graphs of order n in $\mathcal{A}$. We say that a graph property $\mathcal{A}$ is small if there exists a constant $a > 0$ such that $|\mathcal{A}_n| \leq a^n n!$ for all sufficiently large n.

The following result is convenient:

Theorem 3.5 [Norine et al. 2006]. *Let $\mathcal{C}$ be a proper minor closed graph property. Then $\mathcal{C}$ is small.*

Let H be a graph on vertex set $\{1, \dots, h\}$ and let G be a graph on the vertex set $\{1, \dots, n\}$ where $n > h$. Let $W \subseteq V(G)$ with $|W| = h$, and let the root r_W denote the least element in W. We say that W is a pendant appearance of H in G if (a) the increasing bijection from $\{1, \dots, h\}$ to W gives an isomorphism between H and and the induced subgraph $G[W]$ of G, and (b) there exists exactly one edge between W and $V(G) - W$, and this edge is incident with the root r_W. Our method for proving graph polynomials are weakly distinguishing will relay on the following theorem:

Theorem 3.6 [McDiarmid, Steger, Welsh 2005]. *Let $\mathcal{C}$ be a non-empty, addable, and small graph property, let $H \in \mathcal{C}$ be a connected graph, and let R_n be a random graph selected uniformly at random from the graphs of order n in $\mathcal{C}$. Denote by $f_H(R_n)$ the number of pendant appearances of H in R_n. Then there are constants $\alpha > 0, n_0 \in \mathbb{N}$ such that for all $n > n_0$,*

$$\mathbb{P}[f_H(R_n) \leq \alpha n] < e^{-\alpha n}$$

For our purposes, we shall only need a weaker corollary of Theorem 3.6:

Corollary 3.7. *Let $\mathcal{A}$ be an addable proper minor closed graph property, H a fixed connected graph in $\mathcal{A}$. Denote by $\mathcal{A}'_n$ the set of all n vertex graphs $G \in \mathcal{A}$ that have at least one pendent appearance of H. Then* $\lim_{n\to\infty} |\mathcal{A}'_n|/|\mathcal{A}_n| = 1$.

All the properties listed in Example 3.2 are addable and minor closed, hence the corollary applies to them. Using this corollary, we get:

Lemma B. *Let P be a graph polynomial, and $\mathcal{A}$ a small addable graph property. If there is a fixed $H \in \mathcal{A}$ such that every graph $G \in \mathcal{A}$ with a pendant appearance of H has a P mate $G' \in \mathcal{A}$, then P is weakly distinguishing on $\mathcal{A}$.*

Proof. By Corollary 3.7, almost all graphs $G \in \mathcal{A}$ have a pendant appearance of H, and hence almost all graphs $G \in \mathcal{A}$ have a P-mate, so P is weakly distinguishing on $\mathcal{A}$. □

Graphs with genus at most k. Let G be a graph. The genus of G, denoted $g(G)$, is the minimal genus of a surface in which G can be embedded. (For more on graphs embedded in surfaces see [Lando and Zvonkin 2004; Mohar and Thomassen 2001], for example.) It is easy to see that $g(G)$ is well defined, as for all G, $g(G) \leq |E(G)|$. For $k \in \mathbb{N}$, denote by $\mathcal{C}_k$ the property of graphs with genus at most k. Note that in general $\mathcal{C}_k$ is not addable - for instance, the genus of K_5, the complete graph with 5 vertices, is 1, but the genus of a disjoint union of two copies of K_5 has genus 2- so we can not apply Theorem 3.6 to $\mathcal{C}_k$. However, the same result does hold for $\mathcal{C}_k$ with a slight modification:

Theorem 3.8 [McDiarmid 2008]. *Let H be a connected graph, $k \in \mathbb{N}$, and let R_n be a random graph selected uniformly at random from the graphs of order n in $\mathcal{C}_k$. Denote by $f_H(R_n)$ the number of pendant appearances of H in R_n. Then there are constants $\alpha > 0, n_0 \in \mathbb{N}$ such that for all $n > n_0$,*

$$\mathbb{P}[f_H(R_n) \leq \alpha n] < e^{-\alpha n}$$

Again, we only need a weaker corollary of this theorem:

Corollary 3.9. *Let H be a fixed connected planar graph, $k \in \mathbb{N}$ and let $\mathcal{A} = \mathcal{C}_k$ and let $\mathcal{A}'_n$ be the set of all n vertex graphs $G \in \mathcal{A}$ that have at least one pendent appearance of H. Then* $\lim_{n\to\infty} |\mathcal{A}'_n|/|\mathcal{A}_n| = 1$.

Lemma C. *Let P be a graph polynomial, and $\mathcal{C}_k$ the property of graphs with genus at most k. If there is a fixed planar graph H such that every graph G with a pendant appearance of H has a P mate $G' \in \mathcal{C}_k$, then P is weakly distinguishing on $\mathcal{C}_k$.*

Proof. By Corollary 3.9, almost all graphs $G \in \mathcal{C}_k$ have a pendant appearance of H, and hence almost all graphs $G \in \mathcal{C}_k$ have a P-mate, so P is weakly distinguishing on $\mathcal{C}_k$. □

4. The characteristic polynomial

Definition 4.1. Let G be a graph, and denote by A_G its adjacency matrix. The characteristic polynomial of G, denoted $P_A(G)$ is defined as the characteristic polynomial of A_G. The set of roots of $P_A(G)$ is referred to as the spectrum of G.

A P_A-unique graph is usually referred to as a graph determined by its spectrum, and if G and H are two non isomorphic graphs such that $P_A(G) = P_A(H)$, G and H are said to be cospectral.

The characteristic polynomial and particularly its roots are widely studied; see [Brouwer and Haemers 2012], for example, for an introduction. A classic result of Schwenk states that almost all trees are not determined by their spectrum (see [Schwenk 1973] for details). Using Corollary 3.7, we can extend this result to all small addable graph properties.

We use the following recurrence relation for the characteristic polynomial:

Lemma 4.2 (see [Clarke 1970]). *Let G_1 and G_2 be two graphs, and let $v_1 \in V(G_1)$,$v_2 \in V(G_2)$. Denote by H the graph obtained from the disjoint union of G_1 and G_2 by adding an edge from v_1 to v_2. Then*

$$P_A(H, x) = P_A(G_1, x)P_A(G_2, x) - P_A(G_1 - v_1, x)P_A(G_2 - v_2, x) \tag{4-1}$$

We now take two graphs considered in [Schwenk 1973], labeled R and S in Figure 4. They are isomorphic, and we can check that

$$P_A(S - v, x) = P_A(R - u, x) = x^8 - 6x^6 + 10x^4 - 4x^2.$$

Lemma 4.3. *Let G_S be a graph with a pendent appearance of S rooted at v. Denote by G_R the graph obtained from G by replacing the pendant appearance of S in G with a pendant appearance of R, rooted at u. Then G_S and G_R are cospectral mates.*

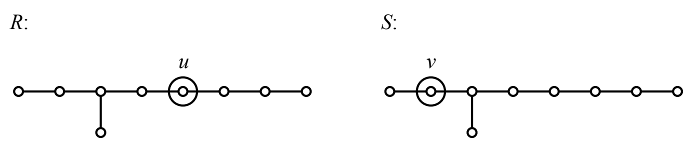


Figure 4. Cospectral graphs.

Proof. Denote by $w \in V(G_S)$ the vertex adjacent to v that is not in the pendant appearance of S, and by H the graph obtained from G_S by deleting all the vertices in the pendant appearance of S. By applying relation (4-1) to G_S with the edge wv, we get

$$P_A(G_S, x) = P_A(S, x)P_A(H, x) - P_A(S - v, x)P_A(H - w, x)$$

By applying relation (4-1) to G_R with the edge wu we get

$$P_A(G_R, x) = P_A(R, x)P_A(H, x) - P_A(R - u, x)P_A(H - w, x)$$

But since $P_A(R, x) = P_A(S, x)$ and $P_A(S - v, x) = P_A(R - u, x)$, we get that $P_A(G_S, x) = P_A(G_R, x)$, so G_S and G_R are cospectral mates. □

As unrooted graphs, S and R are isomorphic. Thus, we have:

Theorem 4.4. *Let $\mathcal{A}$ be a small addable graph property such that $S \in \mathcal{A}$. Then P_A is weakly distinguishing on $\mathcal{A}$.*

Proof. From lemmas B and 4.3 we get that P_A is weakly distinguishing on $\mathcal{A}$. □

Corollary 4.5. *Let $\mathcal{A}$ be a proper minor closed addable graph property. Then P_A is weakly distinguishing on $\mathcal{A}$.*

Proof. Since all forests are in $\mathcal{A}$, and S is a tree, by Theorems 4.4 and 3.5, P_A is weakly distinguishing on $\mathcal{A}$. □

Since all the properties in list 3.2 are addable and minor closed, we have:

Corollary 4.6. *P_A is weakly distinguishing on all the properties listed in Example 3.2.*

Similarly, we get:

Theorem 4.7. *Denote by $\mathcal{C}_k$ the class of graphs of genus at most k. Then P_A is weakly distinguishing on $\mathcal{C}_k$ for all $k \in \mathbb{N}$.*

Proof. Since S is a tree, from lemmas C and 4.3 we get that P_A is weakly distinguishing on $\mathcal{C}_k$. □

Instead of the characteristic polynomial P_A, which is based on the adjacency matrix, one can also look at the analogue graph polynomial P_L which is based on the Laplacian matrix $L_G = D_G - A_G$, where D_G is the diagonal matrix with diagonal elements $d_{v,v} = deg(v)$ for each $v \in V(G)$. Now $P_L(G; x)$ is the characteristic polynomial of L_G. An early survey about the Laplacian polynomial may be found in [Mohar 1991]. The graph polynomials are $d.p.$-equivalent on regular graphs, but $d.p.$-incomparable on all finite graphs; see [Brouwer and Haemers 2012] and [Makowsky et al. 2019]. Our proofs of Theorems 4.4 and 4.7 do not work for $P_L(G; x)$.

Problem 4.1. Find two graphs R and S that can be used two prove analogues of Theorems 4.4 and 4.7.

Figure 5. A path of length 5.

5. The domination polynomial

The *domination polynomial* is the generating function of dominating sets in a given graph G. More formally:

Definition 5.1. Let G be a graph. A set $S \subseteq V(G)$ is called a dominating set if for every $v \in V(G)$, either $v \in S$ or v has a neighbor $u \in S$. Define $\mathrm{Dom}(G; x) = \sum_{S \subseteq V(G)} x^{|S|}$, where the sum is over all dominating sets of G.

The domination polynomial was extensively studied in recent years; see [Alikhani and Peng 2014] for a survey.

Lemma 5.2 [Kotek et al. 2012]. *Let G be a graph. We say a vertex $v \in V(G)$ is a stem if it has a neighbor $u \in V(G)$ with $deg(u) = 1$. Assume G has two stems $u, u' \in G$, $(u, u') \in E(G)$, and denote by G' the graph resulting from G by deleting the edge (u, u'). Then for every set $S \subseteq V(G)$, S is a dominating set of G if and only if S is a dominating set of G'.*

Proof. If S is a dominating set of G', it is clearly also a dominating set of G. On the other hand, let S be a dominating set of G. After deleting (u, u'), the only vertices that are perhaps not dominated by S are u, u'. But u has a vertex of degree 1 as a neighbor, so either it or u are in S, and in either case u is dominated. The same applies to u'. So S is a dominating set of G'. □

Theorem 5.3. *Every graph G that has two distinct stems has a* Dom-*mate.*

Proof. This is a direct consequence of the lemma. □

Corollary 5.4. *Let G be a graph that has a pendant appearance of P_5 rooted at r* (see Figure 5), *and let $\hat{G}$ be the graph obtained from G by replacing the pendant appearance of P_5 with a pendant appearance of $\hat{P}_5$ rooted at r. Then G and $\hat{G}$ are* Dom-*mates.*

Theorem 5.5. *Let $\mathcal{A}$ be a small addable graph property such that $P_5, \hat{P}_5 \in \mathcal{A}$. Then* Dom *is weakly distinguishing on $\mathcal{A}$.*

Proof. From Lemma B and Corollary 5.4 we get that Dom is weakly distinguishing on $\mathcal{A}$. □

Corollary 5.6. *Let $\mathcal{A}$ be a proper minor closed addable graph property, such that $\hat{P}_5 \in \mathcal{A}$. Then* Dom *is weakly distinguishing on $\mathcal{A}$*

Proof. Since all forests are in $\mathcal{A}$, $P_5 \in \mathcal{A}$, and hence from Theorems 5.5 and 3.5 Dom is weakly distinguishing on $\mathcal{A}$. □

As in the previous sections, we have:

Corollary 5.7. Dom *is weakly distinguishing on all the properties listed in Example 3.2.*

Similarly, we get:

Theorem 5.8. *Denote by $\mathcal{C}_k$ the property of graphs of genus less than k. Then* Dom *is weakly distinguishing on $\mathcal{C}_k$ for all $k \in \mathbb{N}$.*

Proof. Since P_5 and $\hat{P}_5$ are planar, from Lemma C and Corollary 5.4 we get that Dom is weakly distinguishing on $\mathcal{C}_k$. □

6. The ξ polynomial

The ξ polynomial, introduced in [Averbouch et al. 2008; 2010], generalizes the matching and the Tutte polynomial. It is defined via a recurrence relation:

Let G be a graph and e an edge of G. We denote by $G - x$ the graph resulting from G by deleting the edge e, by G/e the graph resulting from G by contracting the edge e, and by $G \dagger e$ the graph resulting from G by extracting the edge e, i.e., deleting e together with its adjacent vertices.

Definition 6.1. Let G be a graph. Define the trivariate polynomial $\xi(G; x, y, z)$ using the recursive relation

$$\xi(G; x, y, z) = \xi(G - e; x, y, z) + y\xi(G/e; x, y, z) + z\xi(G \dagger e; x, y, z), \tag{6-1}$$
$$\xi(G \sqcup H; x, y, z) = \xi(G; x, y, z)\xi(H; x, y, z)$$

with the base conditions $\xi(K_1; x, y, z) = x$ and $\xi(\varnothing; x, y, z) = 1$.

An alternative representation of the ξ polynomial was introduced in [Trinks 2012]:

Definition 6.2. Let G be a graph. The covered components polynomial $C(G; x, y, z)$ of G is defined as

$$C(G; x, y, z) = \sum_{A \subseteq E(G)} x^{k(G\langle A\rangle)} y^{|A|} z^{c(G\langle A\rangle)},$$

where $G\langle A\rangle$ is the graph $(V(G), A)$, $k(G\langle A\rangle)$ is the number of connected components of $G\langle A\rangle$ and $c(G\langle A\rangle)$ is the number of covered connected components of $G\langle A\rangle$, that is connected components in $G\langle A\rangle$ with at least one edge.

We sometimes write $C(G)$ for $C(G; x, y, z)$ to simplify long expressions. The two definitions are connected by the following result:

Proposition 6.3 [Trinks 2012]. *For all graphs G,*

$$C(G; x, y, z) = \xi(G; x, y, xyz - xy). \tag{6-2}$$

Corollary 6.4. *Let G and H be graphs. Then $C(G; x, y, z) = C(H; x, y, z)$ if and only if $\xi(G; x, y, z) = \xi(H; x, y, z)$*

We will use the following recurrence relation:

Theorem 6.5 [Trinks 2012]. *Let G_1, G_2 be graphs, and $v_1 \in V(G_1)$, $v_2 \in V(G_2)$. Let H be the graph obtained from G_1 and G_2 by identifying v_1 with v_2. Then*

$$C(G) = \left(\frac{1}{xz} + \frac{2}{x}\right)C(G_1)C(G_2) + \left(-\frac{1}{z} - 1\right)\big(C(G_1)C(G_2 - v_2) + C(G_1 - v_1)C(G_2)\big) + \left(\frac{x}{z} + x\right)C(G_1 - v_1)C(G_2 - v_2). \tag{6-3}$$

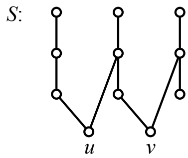


Figure 6. A tree with pseudosimilar vertices.

We next apply Theorem 6.5 to exhibit a pair of C mates.

Lemma 6.6. *Consider the graph S in Figure 6. Let G_v be a graph with a pendent appearance of S rooted at v, and let G_u be the graph obtained from G_v by replacing the pendant appearance of S rooted at v by a pendant appearance of S rooted at u. Then G_v and G_u are C mates.*

Proof. We apply relation (6-3) to G_v, with $G_1 = S$ and G_2 being the graph H_1 obtained from G_v by replacing the pendant appearance of S with a single vertex w. We get

$$C(G_v) = \Big(\frac{1}{xz} + \frac{2}{x}\Big)C(S)C(H_1) + \Big(-\frac{1}{z} - 1\Big)\big(C(S)C(H_1 - w) + C(S - v)C(H_1)\big) + \Big(\frac{x}{z} + x\Big)C(S - v)C(H_1 - w). \quad (6\text{-}4)$$

Similarly, we apply relation (6-3) to G_u, with $G_1 = S$ and G_2 being the graph H_2 obtained from G_u by replacing the pendant appearance of S with a single vertex w. We get

$$C(G_u) = \Big(\frac{1}{xz} + \frac{2}{x}\Big)C(S)C(H_2) + \Big(-\frac{1}{z} - 1\Big)\big(C(S)C(H_2 - w) + C(S - u)C(H)\big) + \Big(\frac{x}{z} + x\Big)C(S - u)C(H_2 - w).$$

Note that $H_1 \cong H_2$, $H_1 - w \cong H_2 - w$, and $S - v \cong S - u$, and hence $C(G_v) = C(G_u)$. □

As a direct consequence of Lemma 6.6 and Corollary 6.4, we have:

Corollary 6.7. *Let G_v be a graph with a pendent appearance of S (depicted in Figure 6) rooted at v, and let G_u be the graph obtained from G_v by replacing the pendant appearance of S rooted at v by a pendant appearance of S rooted at u. Then G_v and G_u are ξ mates.*

From Lemma B and Corollary 6.7 we get:

Theorem 6.8. *Let $\mathcal{A}$ be a small addable graph property such that $S \in \mathcal{A}$. Then ξ is weakly distinguishing on $\mathcal{A}$.*

Corollary 6.9. *Let $\mathcal{A}$ be a proper minor closed addable graph property. Then ξ is weakly distinguishing on $\mathcal{A}$.*

Proof. Since all forests are in $\mathcal{A}$, $S \in \mathcal{A}$, and hence from Theorems 6.8 and 3.5, M is weakly distinguishing on $\mathcal{A}$. □

As in the previous section, we get:

Corollary 6.10. *ξ is weakly distinguishing on all the properties listed in Example 3.2.*

Similarly, we get:

Theorem 6.11. *Denote by $\mathcal{C}_k$ the class of graphs of genus less than k. Then ξ is weakly distinguishing on $\mathcal{C}_k$ for all $k \in \mathbb{N}$.*

Proof. Since S is a tree, from lemmas C and 6.7 we get that ξ is weakly distinguishing on $\mathcal{C}_k$. □

7. ξ-invariants

We investigate consequences of Theorems 6.8 and 6.11 using Lemma A. From Theorem 6.8 we get:

Corollary 7.1. *Let $\mathcal{A}$ be a small addable graph property such that $S \in \mathcal{A}$, and let P be a graph polynomial such that $P <^{\mathcal{C}}_{s.d.p} \xi$. Then P is weakly distinguishing on $\mathcal{A}$.*

Corollary 7.2. *Let $\mathcal{A}$ be a proper minor closed addable graph property, and let P be a graph polynomial such that $P <^{\mathcal{C}}_{s.d.p} \xi$. Then P is weakly distinguishing on $\mathcal{A}$.*

From Theorem 6.11 we get:

Corollary 7.3. *Let $\mathcal{C}_k$ the class of graphs of genus less than k, and let P be a graph polynomial such that $P <^{\mathcal{C}}_{s.d.p} \xi$. Then P is weakly distinguishing on $\mathcal{C}_k$ for all $k \in \mathbb{N}$.*

We next consider results from the literature showing that there are many graph polynomials P where we can apply these results. Among them we find

(i) the generalized chromatic polynomial from [Dohmen et al. 2003], including the chromatic polynomial;

(ii) the matching polynomials;

(iii) the independence polynomial, including the vertex cover polynomial;

(iv) the Tutte polynomial, including the flow polynomial, the reliability polynomial and the Euler polynomial.

The generalized chromatic polynomial. The generalized chromatic polynomial $\mathrm{GC}(G; x, y)$ was introduced in [Dohmen et al. 2003]. and Z. Let Y and Z be two disjoint sets of colors. A generalized coloring of a graph $G = (V(G), E(G))$ is a map $c : V(G) \to (Y \sqcup Z)$ such that for all $(u, v) \in E$, if $c(u) \in Y$ and $c(v) \in Y$, then $c(u) = c(v)$. Y is called the set of proper colors. For two positive integers $x \geq y$, the value of the polynomial $\mathrm{GC}(G; x, y)$ is the number of generalized colorings by x colors, where y of them are proper. The chromatic polynomial $\chi(G; x)$ is obtained for the case $x = y$.

Theorem 7.4 [Averbouch et al. 2010, Proposition 22]. *For all graphs G,*

$$\mathrm{GC}(G; x, y) = \xi(G; x, -1, x - y).$$

Therefore $\chi <^{\mathcal{C}}_{d.p} CG <^{\mathcal{C}}_{d.p} \xi$.

The matching polynomial. Several variants of the matching polynomial are considered in the literature:

Definition 7.5. Let G be a graph with n vertices. A *matching* in G is a spanning subgraph of G in which every connected component is either an isolated vertex or two vertices connected by a single edge. We say a matching is of size k if it has exactly k edges. Denote by $m_k(G)$ the number of k matchings in G.

The *matching acyclic polynomial* (also known as the matching defect polynomial) is defined as

$$\mu(G;x)=\sum_{k=0}^{\lfloor n/2\rfloor}(-1)^k m_k(G)x^{n-2k}$$

The *matching generating polynomial* is defined as

$$g(G;x)=\sum_{k=0}^{\lfloor n/2\rfloor} m_k(G)x^k$$

The *bivariate matching polynomial* of G is defined as

$$M(G;w_1,w_2)=\sum_{k=0}^{\lfloor n/2\rfloor} m_k(G)w_1^{n-2k}w_2^k$$

For an introduction to the bivariate matching polynomial, see [Farrell 1979]. For a recent survey on the acyclic matching polynomial; see [Gutman 2016].

For our purposes, we note the following fact:

Fact 7.1. All of the polynomials in Definition 7.5 are of the same distinctive power on similar graphs:.

$$\mu \sim_{s.d.p} g \sim_{s.d.p} M.$$

In fact, we also have $\mu \sim_{d.p} M$, but for the edgeless graphs E_n of order n we have $g(E_n,x)=g(E_m,x)$ for all $m,n\geq 1$, but $\mu(E_n,x)\neq\mu(E_m,x)$ for $n\neq m$.

Thus we will only consider the bivariate matching polynomial.

Theorem 7.6 [Averbouch et al. 2010, Proposition 20]. *For all graphs G*

$$M(G;x,y)=\xi(G;x,0,y);$$

therefore $M <_{s.d.p} \xi$.

Independence and clique polynomial. The independence polynomial is defined as

$$\mathrm{In}(G;x)=\sum_{A\subseteq V(G)} x^{|A|}$$

where the graph induced graph $G[A]$ is edgeless.

The vertex cover polynomial $V(G;x)$ is defined as

$$\mathrm{VC}(G;x)=\sum_{A\subseteq V(G)} x^{|A|}$$

where A is a vertex cover of G.

The following is taken from (but possibly not originally due to) [Trinks 2012].

Proposition 7.7. *For all graphs G we have*:

(i) $\mathrm{In}(G;x)=\mathrm{GC}(G;x+1,1)$.

(ii) $\mathrm{VC}(G;x)=x^n\,\mathrm{In}\big(G;\frac{1}{x}\big)$.

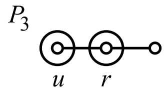


Figure 7. A path of length 3.

Hence, VC $\sim_{s.d.p}$ In $<_{s.d.p}$ GC.

The clique polynomial is defined as

$$\mathrm{Cl}(G; x) = \sum_{A \subseteq V(G)} x^{|A|},$$

where the graph induced graph $G[A]$ is a complete graph.

We note that $\mathrm{In}(G; x) = \mathrm{Cl}(\bar{G})$ for simple graphs. Therefore In $\sim_{d.p}$ Cl by Example 2.2(iii).

Both $\mathrm{In}(G; x)$ and $\mathrm{Cl}(G; x)$ were shown in [Makowsky and Rakita 2019] to be weakly distinguishing on all finite graphs. In the light of the above, $\mathrm{In}(G; x)$ is also weakly distinguishing on small addable graph properties, and on graphs of genus at most k.

This also holds if $\mathcal{C}$ is addable, small, and closed under complements. If $\mathcal{C}$ is addable and small, but not closed under complements we can use the following lemma:

Lemma 7.8. *Let G_r be a graph such that G has a pendant appearance of P_3* (see Figure 7) *rooted at r, and G_u the graph obtained from G_r by replacing the pendant appearance of P_3 by a pendant appearance of P_3 rooted at u. Then G_r and G_u are clique mates.*

Proof. For all $k > 2$, G_r and G_u have the same number of k cliques (since no k clique can include vertices from the pendant appearance), and G and G' have the same number of vertices and edges, and hence the same number of 1 and 2 cliques. So $\mathrm{Cl}(G_r; x) = \mathrm{Cl}(G_u; x)$. □

From Lemmas B and 7.8 we get:

Theorem 7.9. *Let $\mathcal{A}$ be a small addable graph property such that $P_3 \in \mathcal{A}$. Then* Cl *is weakly distinguishing on $\mathcal{A}$.*

The Tutte polynomial. Let $G = (V(G), E(G))$ be a graph. The Tutte polynomial $T(G; x, y)$ is defined by

$$T(G; x, y) = \sum_{A \subseteq E(G)} (x-1)^{k(A)-k(E)} (y-1)^{k(A)+|A|-|V(G)|}.$$

The partition function $Z(G; q, w)$ is defined by

$$Z(G; q, w) = \sum_{A \subseteq E(G)} q^{k(F)} w^{|F|}.$$

They are related by the equation

$$T(G; x, y) = (x-1)^{-k(E)} (y-1)^{-|V(G)|} Z(G; (x-1)(y-1), (y-1)).$$

The chromatic polynomial $\chi(G; x)$ can be obtained from the Tutte polynomial by

$$\chi(G; x) = (-1)^{|V(G)|-k(G)} x^{k(G)} T(G; (1-x), 0)$$

From this we get:

Proposition 7.10. $\chi <_{s.d.p} T \sim_{s.d.p} Z <_{d.p} C \sim_{d.p.} \xi$.

A graph is *Eulerian* if all its vertices have even degree. It does not have to be connected. The Euler polynomial $\mathcal{E}(G; x)$ of a graph is defined by

$$\mathcal{E}(G; x) = \sum_{\substack{A \subseteq E(G) \\ (V,A) \text{ is Eulerian}}} x^{|A|};$$

it is related to the Tutte polynomial as follows [Aigner 2007, Chapter 10, p. 468]:

$$\mathcal{E}(G; x) = (1-x)^{|E(G)-|V(G)|+k(G)} x^{|V(G)|-k(G)} T\Big(G; \frac{1}{x}, \frac{1+x}{1-x}\Big).$$

Hence we get:

Proposition 7.11. $\mathcal{E} <_{s.d.p} T$.

The flow polynomial $\mathrm{Fl}(G, x)$ and the reliability polynomial $R(G; p)$ are related to the Tutte polynomial by

$$\mathrm{Fl}(G; x) = (-1)^{|E(G)-|V(G)|+k(G)} T(G; 0, 1-x)$$
$$R(G; p) = (p)^{|E(G)-|V(G)|+k(G)} (1-p)^{|V(G)|-k(G)} T\Big(G; 1, \frac{1}{p}\Big).$$

Hence we get:

Proposition 7.12. $\mathrm{Fl} <_{s.d.p} T$ *and* $R <_{s.d.p} T$.

8. Conclusion

We have shown that many graph polynomials are weakly distinguishing on all proper minor closed addable graph properties, including many interesting properties such as planar graphs, graphs with tree width at most k, and K_k free graphs (for $4 < k \in \mathbb{N}$). In addition, we proved that the domination, characteristic, and the edge elimination polynomial $\xi(G; x, y, z)$ are weakly distinguishing on the properties of graphs with genus less then k for all $k \in \mathbb{N}$. This also applies to graph polynomials derivable from ξ, such as the generalized chromatic polynomial, Tutte polynomial and its variants, the matching, the independence, and the clique polynomials.

Our results relied on the fact that proper minor closed addable properties are, in a sense, small, and so they do not settle the question whether the domination, characteristic or Tutte polynomials are weakly distinguishing, almost complete, or otherwise on all graphs.

We have shown that for the above graph polynomials P, the sequence $\alpha_P^{\mathcal{C}}(n) = \dfrac{|U_P^{\mathcal{C}}(n)|}{|\mathcal{C}(n)|}$ tends to 0 as n tends to infinity.

The following questions are natural extensions of the work in this paper:

Problem 8.1. What can be said about $\alpha_P^{\mathcal{C}}(n)$ when $\mathcal{C}$ is assumed to be a hereditary or monotone graph property?

Problem 8.2. Can we find a graph polynomial P and a graph property $\mathcal{C}$ such that we can prove that $\alpha_P^{\mathcal{C}}(n) \geq \beta$ for all sufficiently large n for some fixed $\beta \in (0, 1]$?

Problem 8.3. For P one of the above polynomials and $\mathcal{A}$ a proper minor closed addable property, select a random graph G_n uniformly at random in $\mathcal{A}_n$, and denote $[G_n] = \{H \in \mathcal{A}_n : P(G) = P(H)\}$. What can be said about the limit distribution of the random variable $|[G_n]|$?

Acknowledgements

Preliminary versions of our results were presented at the 19th International Conference on Random Structures and Algorithms, 15 to 19 July 2019, Zürich, Switzerland. The results of Sections 6 and 7 were partially obtained during Seminar 19401 in Dagstuhl (Comparative Theory for Graph Polynomials) in October 2019. We want to thank the participants of the Dagstuhl Seminar Delia Garijo, Anna de Mier, Marc Noy, Elena Ravve and Peter Tittmann for valuable discussions and comments. Thanks go especially to Peter Tittmann for his guidance and comments about invariants derivable from ξ-polynomial, which helped us formulating Section 7.

References

[Aigner 2007] M. Aigner, *A course in enumeration*, Graduate Texts in Mathematics **238**, Springer, 2007. MR Zbl

[Alikhani and Peng 2014] S. Alikhani and Y.-h. Peng, "Introduction to domination polynomial of a graph", *Ars Combin.* **114** (2014), 257–266. MR Zbl

[Averbouch et al. 2008] I. Averbouch, B. Godlin, and J. A. Makowsky, "A most general edge elimination polynomial", pp. 31–42 in *Graph-theoretic concepts in computer science*, edited by H. Broersma et al., Lecture Notes in Comput. Sci. **5344**, Springer, 2008. MR Zbl

[Averbouch et al. 2010] I. Averbouch, B. Godlin, and J. A. Makowsky, "An extension of the bivariate chromatic polynomial", *European J. Combin.* **31**:1 (2010), 1–17. MR Zbl

[Bollobás et al. 2000] B. Bollobás, L. Pebody, and O. Riordan, "Contraction-deletion invariants for graphs", *J. Combin. Theory Ser. B* **80**:2 (2000), 320–345. MR Zbl

[Brouwer and Haemers 2012] A. E. Brouwer and W. H. Haemers, *Spectra of graphs*, Springer, 2012. MR Zbl

[Clarke 1970] F. H. Clarke, *The characteristic polynomial of a graph*, master's thesis, McGill University, 1970, Available at https://escholarship.mcgill.ca/concern/theses/gh93h0968?locale=en.

[Dohmen et al. 2003] K. Dohmen, A. Pönitz, and P. Tittmann, "A new two-variable generalization of the chromatic polynomial", *Discrete Math. Theor. Comput. Sci.* **6**:1 (2003), 69–89. MR Zbl

[Farrell 1979] E. J. Farrell, "An introduction to matching polynomials", *J. Combin. Theory Ser. B* **27**:1 (1979), 75–86. MR Zbl

[Godsil and Gutman 1981] C. D. Godsil and I. Gutman, "On the theory of the matching polynomial", *J. Graph Theory* **5**:2 (1981), 137–144. MR Zbl

[Gutman 2016] I. Gutman, "A survey on the matching polynomial", pp. 77–99 in *Graph polynomials*, edited by M. Dehmer et al., CRC Press, Boca Raton, FL, 2016. MR

[Kotek et al. 2012] T. Kotek, J. Preen, F. Simon, P. Tittmann, and M. Trinks, "Recurrence relations and splitting formulas for the domination polynomial", *Electron. J. Combin.* **19**:3 (2012), art. no. 47, 27 pp. MR Zbl

[Lando and Zvonkin 2004] S. K. Lando and A. K. Zvonkin, *Graphs on surfaces and their applications*, Encyclopaedia of Mathematical Sciences **141**, Springer, 2004. MR Zbl

[Lovász 2006] L. Lovász, "Graph minor theory", *Bull. Amer. Math. Soc.* (*N.S.*) **43**:1 (2006), 75–86. MR Zbl

[Makowsky and Rakita 2019] J. A. Makowsky and V. Rakita, "On weakly distinguishing graph polynomials", *Discrete Math. Theor. Comput. Sci.* **21**:1 (2019), art. no. 4, 10 pp. MR Zbl

[Makowsky and Zhang 2019] J. A. Makowsky and R. X. Zhang, "On P-unique hypergraphs", *Australas. J. Combin.* **73** (2019), 456–465. MR Zbl

[Makowsky et al. 2019] J. A. Makowsky, E. V. Ravve, and T. Kotek, "A logician's view of graph polynomials", *Ann. Pure Appl. Logic* **170**:9 (2019), 1030–1069. MR Zbl

[McDiarmid 2008] C. McDiarmid, "Random graphs on surfaces", *J. Combin. Theory Ser. B* **98**:4 (2008), 778–797. MR Zbl

[McDiarmid 2009] C. McDiarmid, "Random graphs from a minor-closed class", *Combin. Probab. Comput.* **18**:4 (2009), 583–599. MR Zbl

[McDiarmid, Steger, Welsh 2005] C. McDiarmid, A. Steger, and D. J. A. Welsh, "Random planar graphs", *J. Combin. Theory Ser. B* **93**:2 (2005), 187–205. MR Zbl

[Mohar 1991] B. Mohar, "The Laplacian spectrum of graphs", pp. 871–898 in *Graph theory, combinatorics, and applications* (Kalamazoo, MI, 1988), vol. 2, edited by Y. Alavi et al., Wiley, New York, 1991. MR Zbl

[Mohar and Thomassen 2001] B. Mohar and C. Thomassen, *Graphs on surfaces*, Johns Hopkins University Press, Baltimore, MD, 2001. MR Zbl

[Norine et al. 2006] S. Norine, P. Seymour, R. Thomas, and P. Wollan, "Proper minor-closed families are small", *J. Combin. Theory Ser. B* **96**:5 (2006), 754–757. MR Zbl

[Noy 2003] M. Noy, "Graphs determined by polynomial invariants", *Theoret. Comput. Sci.* **307**:2 (2003), 365–384. MR Zbl

[Schwenk 1973] A. J. Schwenk, "Almost all trees are cospectral", pp. 275–307 in *New directions in the theory of graphs* (Ann Arbor, MI, 1971), edited by F. Harary, Academic Press, New York, 1973. MR Zbl

[Trinks 2012] M. Trinks, "The covered components polynomial: a new representation of the edge elimination polynomial", *Electron. J. Combin.* **19**:1 (2012), art. no. 50, 31 pp. MR Zbl

Received 5 Dec 2019. Revised 1 Mar 2020.

JOHANN A. MAKOWSKY:

janos@cs.technion.ac.il
Department of Computer Science, Technion - IIT, Haifa, Israel

VSEVOLOD RAKITA:

vsevolod@campus.technion.ac.il
Department of Computer Science, Technion - IIT, Haifa, Israel

Guidelines for Authors

Authors may submit manuscripts in PDF format on-line at the submission page.

Originality. Submission of a manuscript acknowledges that the manuscript is original and and is not, in whole or in part, published or under consideration for publication elsewhere. It is understood also that the manuscript will not be submitted elsewhere while under consideration for publication in this journal.

Language. Articles are usually in English or French, but articles written in other languages are welcome.

Required items. A brief abstract of about 150 words or less must be included. It should be self-contained and not refer to bibliography keys. If the article is not in English, two versions of the abstract must be included, one in the language of the article and one in English. Also required are keywords and a Mathematics Subject Classification for the article, and, for each author, affiliation (if appropriate) and email address.

Format. Authors are encouraged to use LaTeX and the standard amsart class, but submissions in other varieties of TeX, and exceptionally in other formats, are acceptable. Initial uploads should normally be in PDF format; after the refereeing process we will ask you to submit all source material.

References. Bibliographical references should be complete, including article titles and page ranges. All references in the bibliography should be cited in the text. The use of BibTeX is preferred but not required. Tags will be converted to the house format, however, for submission you may use the format of your choice. Links will be provided to all literature with known web locations and authors are encouraged to provide their own links in addition to those supplied in the editorial process.

Figures. Figures must be of publication quality. After acceptance, you will need to submit the original source files in vector graphics format for all diagrams in your manuscript: vector EPS or vector PDF files are the most useful.

Most drawing and graphing packages — Mathematica, Adobe Illustrator, Corel Draw, MATLAB, etc. — allow the user to save files in one of these formats. Make sure that what you are saving is vector graphics and not a bitmap. If you need help, please write to graphics@msp.org with as many details as you can about how your graphics were generated.

Bundle your figure files into a single archive (using zip, tar, rar or other format of your choice) and upload on the link you been provided at acceptance time. Each figure should be captioned and numbered so that it can float. Small figures occupying no more than three lines of vertical space can be kept in the text ("the curve looks like this:"). It is acceptable to submit a manuscript with all figures at the end, if their placement is specified in the text by means of comments such as "Place Figure 1 here". The same considerations apply to tables.

White Space. Forced line breaks or page breaks should not be inserted in the document. There is no point in your trying to optimize line and page breaks in the original manuscript. The manuscript will be reformatted to use the journal's preferred fonts and layout.

Proofs. Page proofs will be made available to authors (or to the designated corresponding author) at a Web site in PDF format. Failure to acknowledge the receipt of proofs or to return corrections within the requested deadline may cause publication to be postponed.

Moscow Journal of Combinatorics and Number Theory

vol. 9 no. 3 2020

Thematic Issue on

Complexity