Sequentialization is as fun as bungee jumping

Rémi Di Guardia* Olivier Laurent* Lorenzo Tortora de Falco† Lionel Vaux Auclair‡

TLLA 2023

Abstract

We propose a new proof of sequentialization for the proof nets of unit-free multiplicative linear logic with mix. It is based on the search of a splitting $⅋$ by means of a simple new lemma about proof structures: the bungee jumping lemma.

1 Introduction

Proof nets are a major contribution from linear logic [Gir87]. They represent proofs, no longer as trees, but as more general graphs, identifying proofs up to rule commutations. This yields more canonical objects, on which results such as cut elimination become easier to prove. A key theorem to prove on proof nets is that they indeed correspond to proof trees of sequent calculus. The more difficult part is building a tree from a net, a process called sequentialization.

There are many proofs of this result in the literature, but it is still considered as a not so easy theorem. We give an elementary proof of this result in the unit-free multiplicative fragment of linear logic, considering the Danos–Regnier correctness criterion [DR89]. The new proof applies directly in the presence of the *mix* rules, and the *mix*-free case can be easily deduced.

2 Definitions

2.1 Multiplicative Linear Logic with Mix

We focus on unit-free multiplicative linear logic whose formulas are given by: $A ::= X \mid X^\perp \mid A \otimes A \mid A ⅋ A$. The **dual** $(\_)^\perp$ is extended to an involution on all formulas by De Morgan duality: $(X^\perp)^\perp = X$, $(A\otimes B)^\perp = A^\perp ⅋ B^\perp$ and $(A ⅋ B)^\perp = A^\perp \otimes B^\perp$.

We consider the deduction system MLL^{mix}_{hyp} given by *open* proofs in cut-free multiplicative linear logic with *mix* rules:

$$\frac{}{\vdash A^\perp, A}\,(ax) \qquad \frac{\vdash A,\Gamma \quad \vdash B,\Delta}{\vdash A\otimes B,\Gamma,\Delta}\,(\otimes) \qquad \frac{\vdash A,B,\Gamma}{\vdash A ⅋ B,\Gamma}\,(⅋) \qquad \frac{}{\vdash}\,(mix_0) \qquad \frac{\vdash \Gamma \quad \vdash \Delta}{\vdash \Gamma,\Delta}\,(mix_2)$$

This means we allow **open hypotheses** $\vdash A$ (with A a *single* formula) in proofs. An equivalent way of presenting the same objects is by extending the system with a rule $\frac{}{\vdash A}\,(hyp)$. If π is a proof with hypotheses $\vdash A_1, \dots, \vdash A_n$ and conclusion $\vdash B_1,\dots,B_k$, we say that π **is a proof of** $A_1,\dots,A_n \vdash B_1,\dots,B_k$. To be formal, we should be more precise on the way we handle occurrences of formulas (e.g. considering sequents as lists and having an explicit exchange rule) but we keep this implicit, as usual.

If π_1 is a proof of $\Sigma \vdash \Gamma, A$ and π_2 is a proof of $A, \Theta \vdash \Delta$, the **substitution** of π_1 in π_2 is a proof of $\Sigma, \Theta \vdash \Gamma, \Delta$: it is obtained from π_2 by replacing the hypothesis $\vdash A$ with π_1 (this adds Γ to all sequents of π_2 below $\vdash A$).

*Univ Lyon, EnsL, UCBL, CNRS, LIP, Lyon, France, remi.di-guardia@ens-lyon.fr, olivier.laurent@ens-lyon.fr
†Università Roma Tre, Dipartimento di Matematica e Fisica, Rome, Italy, tortora@uniroma3.it
‡Aix Marseille Univ, CNRS, I2M, Marseille, France, lionel.vaux@univ-amu.fr

2.2 Partial Graphs

A (directed multi) **partial graph** is a 4-tuple $(\mathcal{V}, \mathcal{A}, \mathsf{s}, \mathsf{t})$ where $\mathcal{V}$ (**vertices**) and $\mathcal{A}$ (**arcs**) are sets and s (**source**) and t (**target**) are *partial* functions from $\mathcal{A}$ to $\mathcal{V}$. It is **finite** if both $\mathcal{V}$ and $\mathcal{A}$ are finite. An arc is **incident** to a vertex if this vertex is either its source or its target.

Many other notions lift immediately from graphs to partial graphs and moreover from any partial graph, one can recover an **underlying** (directed multi total) graph by restricting $\mathcal{A}$ to elements for which both s and t are defined. Isomorphism of (partial) graphs is denoted $\simeq$.

An **edge** is an arc a together with a **direction** $\{+, -\}$ (we use the notations a^+ and a^-, and a is the **support** of the edge). We extend the notions of source and target to edges by $\mathsf{s}(a^+) = \mathsf{s}(a) = \mathsf{t}(a^-)$ and $\mathsf{t}(a^+) = \mathsf{t}(a) = \mathsf{s}(a^-)$. If ε is a direction, $\overline{\varepsilon}$ is the opposite one. If e is an edge a^ε then its **reverse** $\overline{e}$ is $a^{\overline{\varepsilon}}$. Two edges e_1 and e_2 are **composable** if the target of e_1 is (defined and) equal to the source of e_2.

A **path** p is a pair $(v, \vec{e})$ of a vertex and a finite sequence of composable edges such that $\vec{e}$ is empty (in which case p is an **empty** path) or v is the source of the first edge of $\vec{e}$. The vertex v is the **source** of p and the **target** of p is the target of the last edge in $\vec{e}$, if $\vec{e}$ is not empty, and v otherwise. All sources and targets of elements of $\vec{e}$ must be defined, meaning that we are in fact considering undirected paths (since arcs can be crossed in both directions) in the underlying (total) graph. The source and target of an arc, an edge or a path are their **endpoints**. By considering v followed by the targets of the edges in $\vec{e}$, a path p induces a *non-empty* sequence of vertices. A vertex u **is a vertex of** p if it belongs to this sequence. Since a given vertex may occur more than once in this sequence, we may have to talk about **occurrences of vertices in a path** to distinguish these equal values. An occurrence of a vertex in a path is **internal** if it is neither the source nor the target occurrence. An **arc of a path** is the support of one of its edges. Two paths $p_1 = (v_1, \vec{e}_1)$ and $p_2 = (v_2, \vec{e}_2)$ are **composable** if $\mathsf{t}(p_1) = \mathsf{s}(p_2)$ (both being defined) and, in that case, we define their **concatenation** $p_1 \cdot p_2$ as $(v_1, \vec{e}_1 \cdot \vec{e}_2)$. The **reverse** $\overline{p}$ of a path p is obtained by reversing the order of edges and taking the reverse of each edge. Moreover its source is the target of p (and conversely).

A path γ is a **sub-path** of a given path p if its edges are a contiguous sub-sequence of the edges of p (or γ is an empty path and its source is a vertex of p). Equivalently, γ is a sub-path of p if there exist two paths γ_1 and γ_2 such that $p = \gamma_1 \cdot \gamma \cdot \gamma_2$. In the same spirit, γ is a **prefix** of p if γ_1 is empty, and a **suffix** of p if γ_2 is empty. If v_1 and v_2 are two occurrences of vertices of a path p, there is a unique sub-path γ of p having v_1 and v_2 as endpoints: we denote by $p_{(v_1,v_2)}$ the path γ or $\overline{\gamma}$ with source v_1 and target v_2.

A path is **simple** if its arcs are pairwise distinct and its vertices are pairwise distinct except possibly its endpoints which may be equal. A path is **bouncing** if it contains two consecutive equal arcs. A path is **closed** if it has equal endpoints, otherwise it is **open**. A **cycle** is a non-empty simple closed path.

Lemma 2.1. *A non-bouncing path with pairwise distinct vertices except possibly its endpoints is simple.*

Lemma 2.2 (Concatenation of Simple Paths)**.** *If p_1 and p_2 are two simple open paths and their unique common vertices are the target of p_1 and the source of p_2, and possibly the target of p_2 and the source of p_1, and if the last arc of p_1 is different from the first arc of p_2, then $p_1 \cdot p_2$ is simple.*

2.3 Proof Structures

A **proof structure** is a finite partial graph with labeled vertices and arcs. Arcs are labeled with formulas, and vertices with names of non-*mix* rules: *ax*, $\otimes$ or $⅋$. Vertices are named according to their label: ***ax*-nodes**, **$\otimes$-nodes** and **$⅋$-nodes**. Given a vertex v, arcs with target v are the **premises** of v and arcs with source v are the **conclusions** of v. A vertex is **terminal** if all its conclusions have undefined targets. An arc with undefined source is an **hypothesis** and an arc with undefined target is a **conclusion** of the proof structure. Some additional local constraints are required depending on the label of vertices: each *ax*-node has no premise and two conclusions labeled with dual formulas; each $\otimes$-node has two premises and one conclusion labeled $A \otimes B$ (where A and B are the labels of its premises); each $⅋$-node has two premises and one conclusion labeled $A ⅋ B$ (where A and B are the labels of its premises). To recover the usual notion of proof structure, we should impose an order on the premises of nodes, as well as on the hypotheses and conclusions: we ignore this additional information, since it has no impact on the notion of correctness nor on sequentialization.

Specific notions of paths can be defined on proof structures. A path is **switching** if it does not contain the two premises of any $\parr$-node. A path is **strong** if it is non-empty and it is *not* the case that its source is a $\parr$-node and its first arc is one of its premises. A path is **strong-weak** if it is strong and its target is a $\parr$-node and its last arc is one of its premises. A **bridge** in a path is a pair of two consecutive edges connected through a $\parr$-node κ which are the two premises of κ. In this case, κ is called the bridge **pier**.

Lemma 2.3 (Concatenation of Strong Paths)**.** *The concatenation of two strong (resp. strong bridge-free) paths is a strong (resp. strong bridge-free) path.*

Lemma 2.4 (Strong-Weak Cycles)**.** *A non-empty closed concatenation of simple strong-weak paths contains a cycle as a sub-path.*

Having a bridge is a *local* property of a path while being a switching path is a *global* one. They happen to be related for cycles.

Lemma 2.5 (Local-Global Principle)**.** *A strong bridge-free cycle is a switching cycle.*

Proof. If the two premises of a $\parr$-node are visited, this vertex must be visited twice otherwise it would be a bridge pier. In a cycle, the only vertex visited twice is the unique endpoint and exactly two of its incident arcs are visited. Since the path is strong, if the endpoint is a $\parr$-node, the first arc is not one of its premises. □

Definition 2.6 (Splitting $\parr$-node)**.** *A $\parr$-node is* **splitting** *if there is no cycle containing its conclusion.*

Definition 2.7 (Correctness)**.** *A proof structure is* **DR-correct** *if it does not contain any switching cycle.*

2.4 Desequentialization

We define, by induction on a proof π of $A_1, \ldots, A_n \vdash B_1, \ldots, B_k$, its **desequentialization** $\mathcal{D}(\pi)$ which is a proof structure with hypotheses labeled A_1, ..., A_n and conclusions labeled B_1, ..., B_k.

- If π is reduced to an hypothesis $\vdash A$, then $\mathcal{D}(\pi)$ is the proof structure with no vertex and a single arc (with no source and no target) labeled A.
- If π is reduced to an (ax) rule with conclusion $\vdash A^\perp, A$, then $\mathcal{D}(\pi)$ is the proof structure with one ax-node v and two arcs (with source v and no target) labeled respectively $A^\perp$ and A.
- If the last rule of π is a $(\otimes)$ rule applied to two proofs π_1 and π_2 then $\mathcal{D}(\pi)$ is obtained from the disjoint union of $\mathcal{D}(\pi_1)$ and $\mathcal{D}(\pi_2)$ by adding a new $\otimes$-node v. The conclusions of $\mathcal{D}(\pi_1)$ and $\mathcal{D}(\pi_2)$ with labels A and B corresponding to the principal formulas of the $(\otimes)$ rule now have target v, and we add a new arc, labeled $A \otimes B$, with source v and no target.
- If the last rule of π is a $(\parr)$ rule applied to a proof π_1 then $\mathcal{D}(\pi)$ is obtained from $\mathcal{D}(\pi_1)$ by adding a new $\parr$-node v. The conclusions of $\mathcal{D}(\pi_1)$ with labels A and B corresponding to the principal formulas of the $(\parr)$ rule now have target v, and we add a new arc, labeled $A \parr B$, with source v and no target.
- If π is reduced to a (mix_0) rule, $\mathcal{D}(\pi)$ is the empty proof structure (no vertex, no arc).
- If the last rule of π is a (mix_2) rule applied to two proofs π_1 and π_2 then $\mathcal{D}(\pi)$ is the disjoint union of $\mathcal{D}(\pi_1)$ and $\mathcal{D}(\pi_2)$.

Lemma 2.8 (Desequentialization of a substitution)**.** *If π is the substitution of a proof π_1 for an hypothesis $\vdash A$ in a proof π_2, then $\mathcal{D}(\pi)$ is obtained from the disjoint union of $\mathcal{D}(\pi_1)$ and $\mathcal{D}(\pi_2)$ by identifying the conclusion a of $\mathcal{D}(\pi_1)$ labeled A with the hypothesis a' of $\mathcal{D}(\pi_2)$ labeled A (the source of the obtained arc is $\mathsf{s}(a)$, its target is $\mathsf{t}(a')$ and its label is A).*

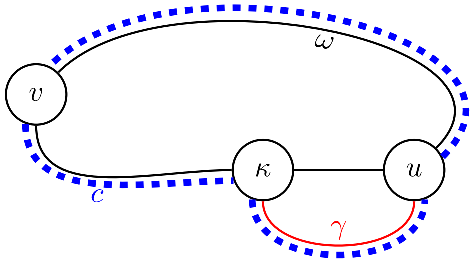

Figure 1: Illustration of the proof of Lemma 3.1

3 Sequentialization

For each vertex v of a proof structure, we denote by $\mathcal{M}_v$ the set of cycles with source v, containing a conclusion of v, and with a minimal number of bridges among all the cycles with source v containing a conclusion of v. The set $\mathcal{M}_v$ is empty if and only if there is no cycle containing a conclusion of v. If $\mathcal{M}_v$ is not empty, it contains a cycle starting with a conclusion of v.

Lemma 3.1 (Bungee Jumping)**.** *Let v be a vertex and ω be a cycle in $\mathcal{M}_v$. If there exists a simple strong bridge-free path γ with source κ the pier of a bridge of ω and with target u a vertex of ω then there exists a switching cycle.*

Proof. By taking a prefix of γ if necessary, we can assume that it does not share any vertex with ω, other than its endpoints κ and u. We can assume $\kappa \neq u$, for otherwise γ is a switching cycle (Lemma 2.5). We use the notation v_1 for the occurrence of v at the source of ω and v_2 for its occurrence at the target of ω. By symmetry (considering the reverse of ω if necessary), we can assume that u is in $\omega_{(\kappa,v_2)}$ and if $u = v$ then ω starts with a conclusion of v.

Consider the closed path $c = \omega_{(v_1,\kappa)} \cdot \gamma \cdot \omega_{(u,v_2)}$ (see Fig. 1): c is non-empty since $\kappa \neq u$, and simple by Lemma 2.2, hence it is a cycle. Moreover, it contains a conclusion of v: either $u = v$ and $\omega_{(v_1,\kappa)}$ starts with a conclusion of v, or both $\omega_{(v_1,\kappa)}$ and $\omega_{(u,v_2)}$ are non-empty, and one contains a conclusion of v. Since $\omega \in \mathcal{M}_v$, c must have at least as many bridges as ω. As it cannot have a bridge at κ nor in γ (γ is strong and bridge-free), we know that it contains a bridge at u and that $\omega_{(\kappa,u)}$ is bridge-free. We then deduce that $\gamma \cdot \omega_{(u,\kappa)}$ is a strong bridge-free cycle: γ is strong and bridge-free, $\omega_{(u,\kappa)}$ is bridge-free and there is no bridge at u (u is a $\parr$-node and at most two of its incident arcs are premises). It is a switching cycle by Lemma 2.5. ☐

We use the notation $v \prec \kappa$ when v is a $\parr$-node and κ is the pier of the *first* bridge of an element of $\mathcal{M}_v$ starting with a conclusion of v. By definition, $\prec$ is irreflexive since κ is internal in a simple path (a cycle) with source v.

Lemma 3.2. *In a DR-correct proof structure, the transitive closure of $\prec$ is a strict partial order relation.*

Proof. Assume we have a cyclic dependency $v_1 \prec v_2 \prec \cdots \prec v_{n+1} = v_1$ ($n \geq 2$). Each $v_i \prec v_{i+1}$ provides a cycle $\omega_i \in \mathcal{M}_{v_i}$ with source v_i and starting with its conclusion, and whose prefix $\gamma_i = \omega_{i(v_i,v_{i+1})}$ is a simple strong-weak bridge-free path. Let γ be the concatenation of the γ_i's (which is bridge-free by Lemma 2.3), it contains a cycle σ by Lemma 2.4. Note that σ cannot be contained in a single γ_i: since γ_i is simple and $\prec$ is irreflexive, no sub-path of γ_i is closed. Hence σ contains an internal occurrence of some v_k ($2 \leq k \leq n$). We focus on the smallest such k: $\omega_{k-1} \in \mathcal{M}_{v_{k-1}}$ and the suffix of σ starting from v_k (which ends in γ_{k-1} thus in ω_{k-1}) satisfy the hypotheses of Lemma 3.1, which contradicts DR-correctness. ☐

Lemma 3.3 (Splitting $\parr$)**.** *A DR-correct proof structure is $\parr$-free or contains a splitting $\parr$-node.*

Proof. If a DR-correct proof structure contains a $\parr$-node, then its set of $\parr$-nodes is finite and non-empty, thus it contains a maximal element with respect to $\prec$ (Lemma 3.2). This $\parr$-node v is splitting. Otherwise $\mathcal{M}_v$ is not empty, let $\omega \in \mathcal{M}_v$ starting with a conclusion of v. Either ω is bridge-free and we contradict DR-correctness (Lemma 2.5), or it contains a bridge and v is not maximal for $\prec$. ☐

Theorem 3.4 (Sequentialization)**.** *Given a DR-correct proof structure ρ, there exists a proof π in MLL^{mix}_{hyp} such that $\rho \simeq \mathcal{D}(\pi)$.*

Proof. We use an induction on the number of vertices and arcs of ρ. If there is at least one $\parr$-node, there is a splitting $\parr$-node v by Lemma 3.3. By removing v (with premises labeled A and B and conclusion labeled $A \parr B$) from ρ (this means that we remove the vertex v but the set of arcs is not modified: the arcs which had v as source or target do not have a source or a target anymore), we obtain two disjoint proof structures: ρ_1 with A and B as labels of some of its conclusions and ρ_2 with a premise labeled $A \parr B$. By induction hypothesis, one gets a proof π_1 such that $\mathcal{D}(\pi_1) \simeq \rho_1$ and a proof π_2 such that $\mathcal{D}(\pi_2) \simeq \rho_2$. We add a $(\parr)$ rule to π_1 and substitute the obtained proof in π_2, obtaining a proof π satisfying $\mathcal{D}(\pi) \simeq \rho$ (Lemma 2.8).

If ρ is $\parr$-free, by DR-correctness, it is cycle-free as any cycle would be a switching cycle. If ρ contains a $\otimes$-node, since the only vertices with premises are $\otimes$-nodes, ρ contains a terminal $\otimes$-node v. By removing v (with premises labeled A and B and conclusion labeled $A \otimes B$) from ρ together with its conclusion, we obtain two disjoint proof structures ρ_1 with A as label of one of its conclusions and ρ_2 with B as label of one of its conclusions. By induction hypothesis, one gets a proof π_1 such that $\mathcal{D}(\pi_1) \simeq \rho_1$ and a proof π_2 such that $\mathcal{D}(\pi_2) \simeq \rho_2$. We add a $(\otimes)$ rule to them and we obtain a proof π satisfying $\mathcal{D}(\pi) \simeq \rho$.

If ρ is $\parr$-free and $\otimes$-free, all vertices are terminal and must be ax-nodes. If ρ contains an ax-node v with conclusions labeled $A^\perp$ and A, by removing v together with its two conclusions, we obtain a proof-structure ρ_1 and, by induction hypothesis, a proof π_1 such that $\mathcal{D}(\pi_1) \simeq \rho_1$. We build: $\pi = \dfrac{\dfrac{}{\vdash A^\perp, A}\,(ax) \qquad \overset{\pi_1}{\vdash \Gamma}}{\vdash A^\perp, A, \Gamma}\,(mix_2)$ which satisfies $\mathcal{D}(\pi) \simeq \rho$.

If ρ is vertex-free, all its arcs have undefined source and target. If ρ contains an arc a labeled A, by removing a, we obtain a proof-structure ρ_1 and, by induction hypothesis, a proof π_1 such that $\mathcal{D}(\pi_1) \simeq \rho_1$. We build: $\pi = \dfrac{\vdash A \qquad \overset{\pi_1}{\vdash \Gamma}}{\vdash A, \Gamma}\,(mix_2)$ which satisfies $\mathcal{D}(\pi) \simeq \rho$. If ρ is empty, $\rho \simeq \mathcal{D}\left(\dfrac{}{\vdash}\,(mix_0)\right)$. □

4 Variations

Now that we have sequentialization/desequentialization for full MLL^{mix}_{hyp}, we can consider some restrictions to specific sub-systems, and characterize sub-systems of the sequent calculus by means of properties of their image in proof structures. A proof π is closed (*i.e.* with no hypothesis) if and only if $\mathcal{D}(\pi)$ is hypothesis-free (*i.e.* the source function s is total).

Given some DR-correct proof structure ρ, the **DR-connectivity degree** $d(\rho)$ is the number of connected components of any of the graphs obtained by removing a premise of each $\parr$-node (note that $d(\rho)$ does not depend on the choice of removed premises). Given a proof π, one can check that $d(\mathcal{D}(\pi)) = 1 + \#mix_2 - \#mix_0$ (where $\#mix_i$ is the number of (mix_i) rules). Conversely, depending on $d(\rho)$, we can transform the proofs π such that $\rho \simeq \mathcal{D}(\pi)$ to obey some constraints on mix-rules, without changing their image by $\mathcal{D}$. Indeed, there is a natural transformation of proofs

$$\dfrac{\vdash \Gamma \qquad \dfrac{}{\vdash}\,(mix_0)}{\vdash \Gamma}\,(mix_2) \quad \rightsquigarrow \quad \vdash \Gamma \qquad\qquad \dfrac{\dfrac{}{\vdash}\,(mix_0) \qquad \vdash \Gamma}{\vdash \Gamma}\,(mix_2) \quad \rightsquigarrow \quad \vdash \Gamma$$

which we call **mix-Rétoré reduction**. It is a confluent and strongly normalizing rewriting system on proofs. Moreover, if π_2 is obtained from π_1 by a sequence of mix-Rétoré reductions then $\mathcal{D}(\pi_1) \simeq \mathcal{D}(\pi_2)$. If π is a mix-Rétoré normal form, either it is reduced to an application of the (mix_0) rule, or it does not contain the (mix_0) rule and it proves a non-empty sequent. In this case, π contains (mix_0) if and only if $\mathcal{D}(\pi)$ is empty, and π contains (mix_2) if and only if $d(\mathcal{D}(\pi)) > 1$. Combined with Theorem 3.4, we obtain:

Theorem 4.1 (Connected sequentialization)**.** *Given a* connected *DR-correct proof structure ρ (*i.e. $d(\rho) = 1$*), there exists a mix-free proof π such that $\rho \simeq \mathcal{D}(\pi)$.*

Our approach can also be extended to richer systems. As usual in the theory of proof nets, dealing with cuts is easy once we know how to deal with $(\otimes)$ rules. Dealing with additive connectives in the spirit of [HG05] requires more work but can be done with a generalization of our approach relying on Lemma 3.1. In the spirit of recent results relating sequentialization of proof nets with results of graph theory [Ngu20], a colored version of the bungee jumping lemma provides a new proof of Yeo's theorem [Yeo97].

References

[DR89] Vincent Danos and Laurent Regnier. "The structure of multiplicatives". In: *Archive for Mathematical Logic* 28 (1989), pp. 181–203. DOI: 10.1007/BF01622878.

[Gir87] Jean-Yves Girard. "Linear logic". In: *Theoretical Computer Science* 50 (1987), pp. 1–102. DOI: 10.1016/0304-3975(87)90045-4.

[HG05] Dominic Hughes and Rob van Glabbeek. "Proof Nets for Unit-free Multiplicative-Additive Linear Logic". In: *ACM Transactions on Computational Logic* 6.4 (2005), pp. 784–842. DOI: 10.1145/1094622.1094629.

[Ngu20] Lê Thành Dũng Nguyễn. "Unique perfect matchings, forbidden transitions and proof nets for linear logic with Mix". In: *Logical Methods in Computer Science* 16.1 (Feb. 2020). DOI: 10.23638/LMCS-16(1:27)2020.

[Yeo97] Anders Yeo. "A Note on Alternating Cycles in Edge-Coloured Graphs". In: *Journal of Combinatorial Theory, Series B* 69.2 (1997), pp. 222–225. DOI: 10.1006/jctb.1997.1728.