



HAL
open science

Stellar Resolution: Multiplicatives

Boris Eng, Thomas Seiller

► **To cite this version:**

| Boris Eng, Thomas Seiller. Stellar Resolution: Multiplicatives. 2020. hal-02895111v1

HAL Id: hal-02895111

<https://hal.science/hal-02895111v1>

Preprint submitted on 9 Jul 2020 (v1), last revised 26 Jan 2021 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1 Stellar Resolution: Multiplicatives

2 Boris Eng 

3 LIPN – UMR 7030 Université Sorbonne Paris Nord, France

4 engboris@hotmail.fr

5 Thomas Seiller 

6 CNRS, LIPN – UMR 7030 Université Sorbonne Paris Nord, France

7 <https://www.seiller.org>

8 seiller@lipn.fr

9 Abstract

10 We present a new asynchronous model of computation named *Stellar Resolution* based on first-
11 order unification [30, 48]. This model of computation is obtained as a formalisation of Girard’s
12 transcendental syntax programme, sketched in a series of three articles [29, 27, 28]. As such, it is
13 the first step towards a proper formal treatment of Girard’s proposal to tackle first-order logic in
14 a proofs-as-program approach [28]. After establishing formal definitions and basic properties of
15 stellar resolution, we explain how it generalises traditional models of computation, such as logic
16 programming and combinatorial models such as Wang tilings. We then explain how it can represent
17 multiplicative proof-structures [18], their cut-elimination and the correctness criterion of Danos-
18 Regnier [10]. Further use of realisability techniques lead to dynamic semantics for Multiplicative
19 Linear Logic, following previous Geometry of Interaction models.

20 **2012 ACM Subject Classification** Theory of computation → Interactive computation; Theory of
21 computation → Linear logic; Theory of computation → Logic; Mathematics of computing → Graph
22 theory; Theory of computation → Program semantics; Theory of computation → Constraint and
23 logic programming

24 **Keywords and phrases** Models of Computation, Linear Logic, Semantics, Geometry of Interaction

25 **Digital Object Identifier** 10.4230/LIPIcs...

26 1 Introduction

27 We present a new asynchronous model of computation named *Stellar Resolution* based
28 on first-order unification [30, 48]. This model arises from work in proof theory, and more
29 precisely proof-theoretic semantics related to the Curry-Howard correspondence between
30 proofs and programs [8, 31]. While Curry-Howard traditionally assimilates terms in the
31 λ -calculus with proofs in intuitionistic logic, stellar resolution terms – named constellations
32 – will be assimilated with (generalisations of) proofs in linear logic.

33 Linear Logic was introduced by Girard [18] as a refinement of intuitionistic logic inspired
34 from semantics of λ -calculus [20]. Soon after the introduction of linear logic, the *geometry of*
35 *interaction* programme emerged [22], aiming at defining semantics of proofs and programs
36 accounting for the dynamics of the cut-elimination procedure. This *dynamic semantics*
37 approach, a major inspiration behind game semantics [32, 1], distinguishes itself from
38 denotational semantics in which cut-elimination is represented as equality.

39 However, the geometry of interaction ambition went beyond the dynamic semantics
40 aspects. It also aimed to reconstruct logic – in particular linear logic – from what could be
41 understood as an untyped model of computation by using realisability techniques. In some
42 way, the idea is the same as the reconstruction of simple types from pure λ -calculus. In
43 λ -calculus, one defines an orthogonality relation between terms t and contexts $E(\cdot)$ by letting
44 $t \perp E(\cdot) \Leftrightarrow E(t)$ is strongly normalising. A type A is then defined as a set of contexts T_A –
45 understood as tests –, and a term t is considered typable with the type A when $t \in T_A^\perp$, i.e.



© Boris Eng and Thomas Seiller;

licensed under Creative Commons License CC-BY

Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

46 $\forall E(\cdot) \in T_A, t \perp E(\cdot)$. We refer the reader to, e.g., Riba’s work[47] for more details.

47 In this aspect of reconstructing linear logic, several geometry of interaction models were
 48 defined using operators algebras [21, 19, 25], unification algebras [23], graphs [50, 52] and
 49 graphings [54, 56, 53]. Although all these models did define rich models, that were in
 50 particular used to study computational complexity [5, 2, 3, 55, 57], they failed with respect
 51 to two different aspects. Firstly, the objects used to interpret even the most basic proofs were
 52 most of the time infinite objects and even when they were not, types were defined through an
 53 infinite number of tests. Secondly, the obtained models did interpret soundly the fragments
 54 of linear logic considered, but no completeness results exist¹.

55 Recently, Girard published a series of articles [26, 29, 27, 28] sketching the main lines
 56 of a new kind of model that would have the qualities of geometry of interaction models,
 57 but improve on them at least concerning the first failure mentioned above. Those articles
 58 are interesting and claim great improvements, proposing in particular a Curry-Howard
 59 interpretation of first-order logic [28]. However, these articles are too inexact in form to
 60 serve satisfactorily as the basis of a mathematical theory². The current work is the first step
 61 towards a proper formal account of the model, with underlying motivation the representation
 62 of first-order logic and its possible applications in relation with descriptive complexity results
 63 [35], such as the Immerman-Vardi theorem [59, 34].

64 **Contributions.** We formally describe a model of computation named stellar resolution,
 65 which extends the model of computation vaguely described by Girard. We prove the main
 66 properties of the model. In particular, while Girard claimed the failure of the Church-Rosser
 67 property, we are able to prove it holds for stellar resolution (Theorem 29). We also relate
 68 it to standard models of computation such as Wang tilings [60] and abstract tiles assembly
 69 models [61, 45], which have applications in bio-computing [49, 62]. We then explain how
 70 this model captures the dynamics of cut-elimination for MLL, the multiplicative fragment of
 71 linear logic (Theorem 47), and the correctness criterion for proof-structures – a syntax for
 72 MLL (Theorem 54). Lastly, we explain how realisability techniques similar to those used
 73 in λ -calculus can be used to define types that organise into a denotational semantics for
 74 MLL (a $*$ -autonomous category), and prove soundness and completeness of the model w.r.t.
 75 MLL+MIX, an extension of MLL with the so-called MIX rule (Theorems 72 & 74).

76 2 Stellar Resolution

77 2.1 First-order Terms and Unification

78 Stellar resolution is based on the theory of unification. We here recall basic definitions and
 79 refer the reader to the article of *Lassez et al.* [38] for more details.

80 ► **Definition 1** (Signature). *A signature $\mathbb{S} = (\mathcal{V}, \mathcal{F}, \mathbf{ar})$ consists of an infinite countable set*
 81 *\mathcal{V} of variables, a countable set \mathcal{F} of function symbols whose arities are given by $\mathbf{ar} : \mathcal{F} \rightarrow \mathbf{N}$.*

82 ► **Definition 2** (First-order term). *Let $\mathbb{S} = (\mathcal{V}, \mathcal{F}, \mathbf{ar})$ be a signature. The set of terms $\mathcal{T}_{\mathbb{S}}$ is*
 83 *inductively defined by the grammar $t, u ::= x \mid f(t_1, \dots, t_n)$ with $x \in \mathcal{V}, f \in \mathcal{F}, \mathbf{ar}(f) = n$.*

84 We now fix a signature $\mathbb{S} = (\mathcal{V}, \mathcal{F}, \mathbf{ar})$ until the end of this section.

¹ While this aspect is a failure somehow, it is also a feature as the models are very rich and open other paths of reflexion.

² The formulation is borrowed from Church’s critic of von Mises notion of kollektiv [7].

85 ▶ **Definition 3** (Substitution). *A substitution is a function $\theta : \mathcal{V} \rightarrow \mathcal{T}_{\mathbb{S}}$ extended from variables*
 86 *to terms by $\theta(f(u_1, \dots, u_k)) = f(\theta u_1, \dots, \theta u_k)$.*

87 *A renaming is a substitution α such that $\alpha(x) \in \mathcal{V}$ for all $x \in \mathcal{V}$.*

88 ▶ **Definition 4** (Unification problem). *An equation is an unordered pair $t \doteq u$ of terms in $\mathcal{T}_{\mathbb{S}}$.*
 89 *A unification problem is a set $P = \{t_1 \doteq u_1, \dots, t_n \doteq u_n\}$ of equations.*

90 *A solution for P is a substitution θ such that for all equations $t_i \doteq u_i$ in P , $\theta t_i = \theta u_i$.*

91 ▶ **Definition 5** (Matching). *Two terms t and u are matchable if there exists a renaming α*
 92 *such that $\{\alpha t \doteq u\}$ has a solution.*

93 ▶ **Theorem 6** (Unification algorithm). *The problem of deciding if a solution to a given unific-*
 94 *ation problem P exists is decidable. Moreover, there exists a minimal solution $\text{Solution}(P)$*
 95 *w.r.t the preorder $\theta \preceq \psi \Leftrightarrow \exists \theta'. \psi = \theta' \circ \theta$, unique up to renaming.*

96 Let us note that several algorithms were designed to compute the unique solution when
 97 it exists, such that the Martelli-Montanari unification algorithm [40].

98 2.2 Stars and Constellations

99 The stellar resolution model is based on first-order unification but extends it the notions of
 100 *polarities* and *colours*. Intuitively, polarised terms are prefixed with either a $+$ or $-$ sign
 101 indicating their ability to be composed (with a term of opposite polarity). Colours allow for
 102 the consideration of various types of composition. We define the core objects of our model.

103 ▶ **Definition 7** (Coloured Signature). *A coloured signature is a 4-tuple $\mathbb{S} = (\mathcal{V}, \mathcal{F}, \mathcal{C}, \mathbf{ar})$*
 104 *where $(\mathcal{V}, \mathcal{F}, \mathbf{ar})$ is a signature and $\mathcal{C} \subseteq \mathcal{F}$ is a specified set of colours. The set of terms $\mathcal{T}_{\mathbb{S}}$*
 105 *is defined as the set of terms over the signature $(\mathcal{V}, \mathcal{F}, \mathbf{ar})$.*

106 We will now work with the coloured signature $\mathbb{S} = (\mathcal{V}, \mathcal{F}, \mathcal{C}, \mathbf{ar})$ unless specified otherwise.

107 ▶ **Definition 8** (Ray). *A ray is a term in the grammar $r ::= +c(t_1, \dots, t_n) \mid -c(t_1, \dots, t_n) \mid t$,*
 108 *where $\{t_1, \dots, t_n\} \subseteq \mathcal{T}_{\mathbb{S}}$ and $c \in \mathcal{C}$ with $\mathbf{ar}(c) = n$. The underlying term of a ray is defined by*
 109 $[+c(t_1, \dots, t_n)] = [-c(t_1, \dots, t_n)] = c(t_1, \dots, t_n)$ and $[t] = t$.

110 ▶ **Notation 9**. We will sometimes write $+c.t$ (resp. $-c.t$) instead of $+c(t)$ (resp. $-c(t)$) for
 111 unary colours when it is convenient.

112 ▶ **Definition 10** (Star). *A star σ over a coloured signature \mathbb{S} is a finite and non-empty*
 113 *multiset of rays, i.e. a finite set $|\sigma|$ together with a map $\rho_\sigma : |\sigma| \rightarrow \text{rays}(\mathbb{S})$. The set of*
 114 *variables appearing in σ is written $\mathbf{vars}(\sigma)$. Stars are written as multisets $\sigma = [r_1, \dots, r_n]$.*

115 ▶ **Definition 11** (Substitutions and α -equivalence). *Given a substitution θ , its action extends*
 116 *to rays by letting $\theta(\pm c(t_1, \dots, t_n)) = \pm \theta(c(t_1, \dots, t_n))$ with $\pm \in \{+, -\}$. It also extends to*
 117 *stars: $\theta[r_1, \dots, r_n] = [\theta r_1, \dots, \theta r_n]$.*

118 *We say that two stars σ_1, σ_2 are α -equivalent, written $\sigma_1 \approx_\alpha \sigma_2$, when there exists a*
 119 *renaming α such that $\alpha \sigma_1 = \sigma_2$.*

120 ▶ **Notation 12**. In this work, stars will be considered up to α -equivalence. We therefore
 121 define $\text{Stars}(\mathbb{S})$ as the set of all stars over a coloured signature \mathbb{S} , quotiented by \approx_α .

122 ▶ **Definition 13** (Constellation). *A constellation Σ is a (countable) multiset of stars, i.e.*
 123 *a countable (possibly infinite) set $|\Sigma|$ together with a map $\alpha_\Sigma : |\Sigma| \rightarrow \text{Stars}(\mathbb{S})$. The*
 124 *variables appearing in Σ are considered (sometimes implicitly) bound to their star i.e*
 125 $\bigcap_{e \in |\Sigma|} \mathbf{vars}(\alpha_\Sigma(e)) = \emptyset$. *The set of rays of Σ is defined as $\text{rays}(\Sigma) = \{(s, r) \mid s \in |\Sigma|, r \in$*
 126 $|\alpha_\Sigma(s)|\}$. *A finite constellation will sometimes be written $\Sigma = a_1 \sigma_1 + \dots + a_n \sigma_n$ ($a_i \in \mathbf{N}$).*

XX:4 Stellar Resolution: Multiplicatives

127 ► **Example 14.** Here is an example of two finite and an infinite constellation:

128 ■ $\Sigma_{ex} = [g(x), f(x), +a(f(x))] + [-a(y), +b(y)] + [x, -b(g(x))] + [+b(x), x]$

129 ■ $\Sigma_a = [-a(x), +a(x), -b(x)]$

130 ■ $\Sigma_{\mathbf{N}}$ is defined by $|\Sigma_{\mathbf{N}}| = \mathbf{N}$ and $\alpha_{\Sigma_{\mathbf{N}}}(n) = [-nat.s^n(0), +nat.s^{n+1}(0)]$

131 where $s^0(t) = t$ and $s^{n+1}(t) = s(s^n(t))$, nat is a colour and s and 0 are function symbols.

132 2.3 Diagrams and Reduction

133 Constellations are gatherings of stars destined to be connected together through their rays of
134 opposite polarities. We define the unification graph of a constellation which is a multigraph
135 specifying which stars can be connected together through the unification of their rays.

136 ► **Definition 15 (Unification graph).** *The unification graph $\Sigma[\mathcal{A}]$ of a constellation Σ w.r.t.*
137 *a set of colours $\mathcal{A} \subseteq \mathcal{C}$ is the undirected multigraph $(V^{\Sigma[\mathcal{A}]}, E^{\Sigma[\mathcal{A}]}, s^{\Sigma[\mathcal{A}]})$ where $V^{\Sigma[\mathcal{A}]} = |\Sigma|$,*
138 *$E^{\Sigma[\mathcal{A}]}$ is the set of unordered pairs $\{((s, r), (s', r')) \in \text{rays}(\Sigma) \mid r \text{ and } r' \text{ are matchable}\}$, and*
139 *$s^{\Sigma[\mathcal{A}]}$ is the function defined by $s^{\Sigma[\mathcal{A}]((s, r), (s', r'))} = \{s, s'\}$.*

140 ► **Example 16.** If we take the constellations from example 14, the unification graph $\Sigma_{\mathbf{N}}[nat]$
141 is an infinite linear graph and $\Sigma_{ex}[b, c, d, e]$ is a linear graph $\bullet - \bullet - \bullet - \bullet$ of 4 vertices. The
142 graph $\Sigma_a[a, b]$ is a vertex with a loop.

143 The unification graph is used to define *diagrams*, which are graphs obtained by composing
144 occurrences of stars in a constellation along dual rays. They represent actual connexions
145 between the stars of a constellation, following the connexions allowed by the unification
146 graph.

147 ► **Definition 17 (Diagram).** *Let \mathcal{A} be a set of colours $\mathcal{A} \subseteq \mathcal{C}$. An \mathcal{A} -diagram δ over a*
148 *constellation Σ is a finite connected graph³ D_δ and a graph homomorphism $\delta : D_\delta \rightarrow \Sigma[\mathcal{A}]$*
149 *which is ray injective, i.e. for all $v \in V^{D_\delta}$ the function $\{e \in E^{D_\delta} \mid v \in s^{D_\delta}(e)\} \rightarrow$*
150 *$|\delta(v)|; \{(\delta(v), r), (s', r')\} \mapsto r$ is injective. If D_δ is a tree, we say δ is tree-like.*

151 ► **Notation 18.** Given an \mathcal{A} -diagram δ , we define its set of *paired rays* as the set $\text{paired}(\delta) =$
152 $\{(v, r) \mid \exists e \in E^{D_\delta}, v \in s^{D_\delta}(e), (\delta(v), r) \in \text{rays}(\Sigma)\}$ and its set of *free rays* as $\text{free}(\delta) =$
153 $\{(v, r) \mid v \in V^{D_\delta}, (\delta(v), r) \in \text{rays}(\Sigma), (v, r) \notin \text{paired}(\delta)\}$.

154 ► **Example 19.** If we take the constellations from example 14 together with their unification
155 graph in example 16, any subchain of $\Sigma_{\mathbf{N}}[nat]$ and $\Sigma_{ex}[b, c, d, e]$ can give rise to a diagram.
156 As for, $\Sigma_a[a, b]$, we can make a diagram be connecting as many occurrences of the star
157 $[-a(x), +a(x), -b(x)]$ as we want.

158 ► **Remark 20.** The notion of diagram suggested by Girard [29] coincides with the notion of
159 *tree-like diagram* in this work. However, as we will explain below, the more general notion of
160 diagram considered here allows for the representation of interesting models of computation,
161 such as Wang tilings [60].

162 As mentioned above, we consider the underlying graphs of diagrams to have natural
163 numbers as vertices. This is used here to produce a family of substitutions $(\theta(v, _))_{v \in \mathbf{N}}$ of
164 pairwise disjoint codomains. This is used in the following definition to rename all occurrences
165 of stars in diagrams with pairwise disjoint sets of free variables.

³ The underlying graph is considered up to renaming of the vertices and edges. For practical purpose and without loss of generality, we will consider that vertices are natural numbers.

166 ► **Definition 21** (Underlying unification problem). *The underlying unification problem of an*
 167 *\mathcal{A} -diagram δ over a constellation Σ is defined as*

$$168 \quad \mathcal{P}(\delta) = \{\theta(v, [r]) \doteq \theta(v', [r']) \mid f \in E^{D_\delta}, s^{D_\delta}(f) = \{v, v'\} \wedge \delta(f) = ((\delta(v), r), (\delta(v'), r'))\}.$$

169 In some ways, diagrams generalise trails in a graph. Where paths describe a possible
 170 trajectory for a particle, diagrams describe the possible trajectories of a wave that can
 171 simultaneously spread in several directions when encountering forks. We now introduce the
 172 notion of *saturated diagrams*, which correspond to *maximal paths*.

173 ► **Definition 22** (Saturated Diagram). *We define a preorder \sqsubseteq on \mathcal{A} -diagrams over a con-*
 174 *stellation Σ by: $\delta \sqsubseteq \delta'$ if there exists an isomorphism φ from a graph D of $D_{\delta'}$ to D_δ such*
 175 *that $\delta = \delta' \circ \varphi$.*

176 *A maximal \mathcal{A} -diagram w.r.t. \sqsubseteq is called saturated.*

177 ► **Example 23.** For the constellations from example 14 and their unification graph in
 178 example 16, $\Sigma_{\mathbf{N}}$ has no saturated diagram since it is always possible to extend any star
 179 $[-nat.s^n(0), +nat.s^{n+1}(0)]$ by connecting it to $[-nat.s^{n+1}(0), +nat.s^{n+2}(0)] \in \Sigma_{\mathbf{N}}$. Similarly,
 180 Σ_a can always connect any star with a new copy of itself so no saturated diagram can be
 181 constructed. The star $[x, -b(g(x))] \in \Sigma_{ex}$ can be connected to either $[g(x), f(x), +a(f(x))] +$
 182 $[-a(y), +b(y)]$ or just $[+b(x), x]$ and these two connexions can't be extended. Therefore, they
 183 form saturated diagrams.

184 2.4 Evaluation and Normal form

185 Diagrams can be evaluated by confronting and annihilating the rays connected together
 186 similarly to a chemical reaction between molecules which will have an effect on the remaining
 187 free rays. This reaction defines a unification problem. When a solution (a substitution) to
 188 this problem exists, one can apply it to all the free rays involved in the diagram. Edges in
 189 the diagram then represent equalities and the diagram can be *contracted* into a single star.
 190 This operation is called *actualisation*.

191 ► **Definition 24** (Correct diagram and actualisation). *Consider a constellation Σ and a set*
 192 *of colours $\mathcal{A} \subseteq \mathcal{C}$. An \mathcal{A} -diagram δ is correct if $\mathbf{free}(\delta) \neq \emptyset$ and the associated unification*
 193 *problem $\mathcal{P}(\delta)$ has a solution.*

194 *The actualisation of a correct diagram δ is the star $\Downarrow \delta$ defined as $|\Downarrow \delta| = \mathbf{free}(\delta)$ and*
 195 *$\rho_{\Downarrow \delta} : (e, r) \in \mathbf{free}(\delta) \mapsto \psi(\theta(e, r))$, where $\psi = \text{Solution}(\mathcal{P}(\delta))$ and $\theta(e, _)$ is the renaming*
 196 *used in Definition 21.*

197 ► **Notation 25.** We write $\text{Correct}_{\mathcal{A}}^k(\Sigma)$ (resp. $\text{Correct}_{\mathcal{A}, \text{tree}}^k(\Sigma)$) for the set of correct \mathcal{A} -
 198 diagrams (resp. correct tree-like \mathcal{A} -diagrams) over Σ with k vertices, and $\text{Sat}_{\mathcal{A}}^k(\Sigma)$ (resp.
 199 $\text{Sat}_{\mathcal{A}, \text{tree}}^k(\Sigma)$) the set of saturated \mathcal{A} -diagrams in $\text{Correct}_{\mathcal{A}}^k(\Sigma)$ (resp. in $\text{Correct}_{\mathcal{A}, \text{tree}}^k(\Sigma)$).

200 Note that the approach proposed by Girard used a step-by-step procedure merging two
 201 adjacent stars in a diagram. It is possible to prove that his approach and our lead to the
 202 same result, and more precisely that a sequence of merging coincides with a step-by-step
 203 execution of the Martelli-Montanari unification algorithm [40]. Stating and proving this
 204 refined result is however outside of the scope of this paper.

205 ► **Definition 26** (Normalisation). *The normalisation or execution (resp. tree-like norm-*
 206 *alisation) of a constellation Σ w.r.t. a set of colours $\mathcal{A} \subseteq \mathcal{C}$ is defined by $\text{Ex}_{\mathcal{A}}^*(\Sigma) =$
 207 $\bigcup_{k=0}^{\infty} \Downarrow \text{Sat}_{\mathcal{A}, * }^k(\Sigma)$ ($* \in \{\emptyset, \text{tree}\}$), where $\Downarrow \text{Sat}_{\mathcal{A}, * }^k(\delta)$ denotes the set $\{\Downarrow \delta \mid \delta \in \text{Sat}_{\mathcal{A}, * }^k(\delta)\}$.*

208 ► **Example 27.** The constellation Σ_{ex} is the only constellation from example 14 which has
 209 saturated diagrams. The only correct diagram δ connects $[x, -b(g(x))]$ and $[+b(y), y]$. We
 210 have to solve the problem $\mathcal{P}(\delta) = \{b(g(x)) \doteq b(y)\}$ producing $\theta = \{y \mapsto g(x)\}$ and we finally
 211 obtain $\text{Ex}(\Sigma_{ex}) = \theta[x, y] = [x, g(x)]$ since Σ_{ex} has only two diagrams. One can check that
 212 the other diagram fails.

213 ► **Definition 28 (Strong normalisation).** A constellation Σ is strongly normalising w.r.t. a
 214 set of colours $\mathcal{A} \subseteq \mathcal{C}$ if and only if $\text{Ex}(\Sigma)$ is a finite constellation. When $\mathcal{A} = \mathcal{C}$, we simply
 215 say that Σ is strongly normalising.

216 ► **Theorem 29 (Associativity / Church-Rosser).** Let Σ be a constellation, and \mathcal{A}, \mathcal{B} be two
 217 disjoint sets of colours, i.e. $\mathcal{A} \subseteq \mathcal{C}$, $\mathcal{B} \subseteq \mathcal{C}$, and $\mathcal{A} \cap \mathcal{B} = \emptyset$. Then:

$$218 \quad \text{Ex}_{\mathcal{B}}^*(\text{Ex}_{\mathcal{A}}^*(\Sigma)) = \text{Ex}_{\mathcal{A} \cup \mathcal{B}}^*(\Sigma) = \text{Ex}_{\mathcal{A}}^*(\text{Ex}_{\mathcal{B}}^*(\Sigma)), \quad (* \in \{\emptyset, \text{tree}\})$$

219 **Proof (Sketch).** We establish the first isomorphism, which is enough by symmetry. First
 220 note that the disjointness of the sets of colours implies that the set of edges in $\Sigma[\mathcal{A} \cup \mathcal{B}]$ is
 221 the disjoint union of the sets of edges in $\Sigma[\mathcal{A}]$ and those in $\Sigma[\mathcal{B}]$. As a consequence, any
 222 diagram on $\Sigma[\mathcal{A}]$ (resp. $\Sigma[\mathcal{B}]$) can be thought of as a diagram on $\Sigma[\mathcal{A} \cup \mathcal{B}]$.

223 Let $\delta : D_\delta \rightarrow \text{Ex}_{\mathcal{A}}^*(\Sigma)[\mathcal{B}]$ be a correct saturated \mathcal{B} -diagram of $\text{Ex}_{\mathcal{A}}^*(\Sigma)$. For each $s \in$
 224 $|\text{Ex}_{\mathcal{A}}^*(\Sigma)|$, the star $\text{Ex}_{\mathcal{A}}^*(\Sigma)(s)$, which we will write σ_s , corresponds to a diagram $\delta_s : D_s \rightarrow \Sigma[\mathcal{A}]$.
 225 We can therefore build a diagram $\bar{\delta}$ over $\Sigma[\mathcal{A} \cup \mathcal{B}]$ by *blowing up* δ along the diagrams δ_s .
 226 More precisely, we construct the graph \bar{D}_δ obtained by replacing each vertex s by the graph
 227 D_s ; this is well-defined as each ray in $\delta(s)$ comes from a unique ray from a star $\delta_s(s')$ for some
 228 $s' \in V^{D_s}$, and therefore each edge e in D_δ of source s becomes an edge of source the unique
 229 star $s' \in V^{D_s}$. The morphism δ then extends uniquely to a morphism $\bar{\delta} : \bar{D}_\delta \rightarrow \Sigma[\mathcal{A} \cup \mathcal{B}]$.

230 whose action on the subgraphs D_s coincides with that of δ_s (as a morphism into $\Sigma[\mathcal{A} \cup \mathcal{B}]$).
 231 To check that this mapping from \mathcal{B} -diagrams on $\text{Ex}_{\mathcal{A}}^*(\Sigma)$ to $(\mathcal{A} \cup \mathcal{B})$ -diagrams on Σ is
 232 indeed an isomorphism, one can directly define an inverse mapping. For this purpose, the
 233 essential remark is that given a $(\mathcal{A} \cup \mathcal{B})$ -diagram $\bar{\delta}$ on Σ , one can recover the underlying
 234 \mathcal{B} -diagrams on $\text{Ex}_{\mathcal{A}}^*(\Sigma)$ as the restriction of $\bar{\delta}$ to the connected components of the graph
 235 obtained from \bar{D}_δ by removing the edges mapped to \mathcal{A} -coloured edges in $\Sigma[\mathcal{A} \cup \mathcal{B}]$. The
 236 underlying graph of the corresponding \mathcal{B} -diagram on $\text{Ex}_{\mathcal{A}}^*(\Sigma)$ is then defined from $D_{\bar{\delta}}$ by
 237 contracting each of these connected components to a single vertex. ◀

238 ► **Remark 30.** In Girard's first paper on Transcendental Syntax [29], the constellation
 239 $\Sigma = [+a.x, -a.x, +b.x]$ is mentioned as a counter-example for the Church-Rosser property.
 240 Here, we have $\text{Ex}_{\{a\}}^{\text{tree}}(\text{Ex}_{\{b\}}^{\text{tree}}(\Sigma)) = \text{Ex}_{\{b\}}^{\text{tree}}(\text{Ex}_{\{a\}}^{\text{tree}}(\Sigma)) = \emptyset$ (because no saturated diagram on a
 241 nor on b can be constructed). Our understanding of Girard's failure comes from his limitation
 242 to strongly normalising constellations, so that $\text{Ex}_{\{a\}}^{\text{tree}}(\Sigma)$ was not defined.

243 2.5 Remarkable examples of constellations

244 We now explain how the stellar resolution model generalises several established frameworks
 245 from the literature.

246 2.5.1 (Hyper)graphs

247 As a first example, we detail how graphs (and more generally hypergraphs) can be encoded,
 248 and how the execution coincides with the computation of paths. A consequence of this result
 249 is that models of computation such as *Interaction Graphs* [50, 52] are special cases of our
 250 construction.

251 ► **Definition 31** (Encoding of hypergraphs). *Let $G = (V, E, s, t)$ be a directed hypergraph,*
 252 *i.e. s, t are maps $E \rightarrow \wp(V)$. We suppose given a signature in which there exists distinct*
 253 *unary function symbols to represent the elements of V and E . Then, for each $e \in E$ with*
 254 *$s(e) = \{v_1, \dots, v_n\}$ and $t(e) = \{w_1, \dots, w_m\}$, we define the star*

$$255 \quad \bar{e} = [-v_1(x), -v_2(x), \dots, -v_n(x), +w_1(e(x)), +w_2(e(x)), \dots, +w_m(e(x))].$$

256 *The graph G is then represented by the constellation \bar{G} defined as the multiset $\alpha : E \rightarrow \bar{e}$.*

257 One can check that this encoding has the property that diagrams $\rho : T \rightarrow U(\bar{G})$ in which T
 258 is a *linear graph* (i.e. a graph $\bullet \rightarrow \bullet \rightarrow \dots \rightarrow \bullet$) are in bijection with walks in G ; moreover
 259 the diagram is saturated if and only if the walk is *maximal*.

260 2.5.2 Logic programming

261 The stellar resolution is very close to concepts coming from logic programming since they all
 262 use first-order terms to do computations. The main difference is that logic programming
 263 has practical motivations and enjoy logical interpretations through model theory while our
 264 approach is purely computational and doesn't obey logic. We choose to encode first-order
 265 disjunctive clauses [48].

266 ► **Definition 32** (Encoding of first-order disjunctive clauses). *We have the following encoding*
 267 *for first-order atoms: $P(t_1, \dots, t_n)^\star = +P(t_1, \dots, t_n)$ and $(\neg P(t_1, \dots, t_n))^\star = -P(t_1, \dots, t_n)$*
 268 *where P is a colour. A clause is encoded as $\{A_1, \dots, A_n\}^\star = [A_1^\star, \dots, A_n^\star]$ where A_1, \dots, A_n are*
 269 *first-order atoms. A set of clauses is translated into $\{C_1, \dots, C_n\}^\star = C_1^\star + \dots + C_n^\star$. Clauses*
 270 *are connected through the resolution rule: from $\Gamma \cup \{P(t_1, \dots, t_n)\}$ and $\Delta \cup \{\neg P(u_1, \dots, u_n)\}$,*
 271 *we infer $\theta(\Gamma \cup \Delta)$ where $\theta = \text{Solution}(\{t_i \doteq u_i\}_{1 \leq i \leq n})$.*

272 The actualisation of a diagram corresponds to several applications of the resolution rule.
 273 In particular, a saturated diagram is a full computation (instead of a partial/unfinished
 274 computation). The unification graph we use is reminiscent of dependency graphs [44, 12]
 275 and the normalisation of the computation of answer sets [14, 13]. Although not described
 276 here, it may also be possible to simulate goal-directed inferences (such as SLD [36] or SLO
 277 [39] resolution) by using unpolarised rays.

278 Another close model is the model of flows [23, 4] made of couples of terms $t \leftarrow u$ where t
 279 and u have exactly the same variables.

280 ► **Definition 33** (Encoding of flows). *Let $f = t \leftarrow u$ be a flow. We define its encoding*
 281 *as $f^\star = [+t, -u]$. A wiring $F = f_1 + \dots + f_n$ is a set of flows and it is encoded into*
 282 *$F^\star = f_1^\star + \dots + f_n^\star$.*

283 A wiring will describe a graph where flows represent edges. Matching (definition 5) decides
 284 which vertices the edges connect. The execution of a wiring $\text{Ex}(F)$ computing maximal paths
 285 in corresponding graph corresponds to $\text{Ex}(F^\star)$.

286 2.5.3 Wang tiles and abstract tile assembly models

287 Wang tiles [60] are square tiles with a colour on each side. For instance: $W = \{\begin{smallmatrix} \blacksquare & \blacktriangle \\ \blacktriangle & \blacktriangle \end{smallmatrix}\}$. We
 288 are interested in the possible tilings made by putting copies of the tiles side by side such
 289 that two adjacent sides have the same colour. Our setting naturally generalises the structure
 290 and behaviour of Wang tiles in \mathbf{N}^2 .

291 ► **Definition 34** (Encoding of Wang tiles). *Let $t^i = (c_w^i, c_e^i, c_s^i, c_n^i)$, $i = 1, \dots, k$, be a set of*
 292 *Wang tile characterised by west, east, south and north colours. We encode each Wang tile in*
 293 \mathbb{N}^2 *by the star $(t^i)^\star = [-h(c_w^i(x), x, y), -v(c_s^i(y), x, y), +h(c_e^i(s(x)), x, y), +v(c_n^i(s(y)), x, y)]$*

294 This supposes the signature to contain a unary function symbol for each colour used in the
 295 Wang tiles set considered. We use two additional ternary colours h (for *horizontal axis*) and
 296 v (for *vertical axis*) for the rays. The terms $s^n(x), s^n(y)$ for $n \leq 0$ represent coordinates on
 297 the horizontal and vertical axis encoded as natural numbers, and forbids geometric patterns
 298 that would be impossible in the original Wang tile framework (e.g. a tile connected to itself).
 299 A (possibly cyclic) diagram will correspond to a finite tiling.

300 Wang tilings are a Turing-complete model of computation [6]. As the simulation works
 301 by representing the space-time diagram of the execution, it is enough to restrict to Wang
 302 tiles over \mathbb{N}^2 by considering machines with a single right-infinite tape (i.e. Turing's original
 303 *a-machine* model [58]). As a consequence, we obtain the following theorem.

304 ► **Theorem 35** (Turing-completeness). *Stellar resolution can simulate Turing machines.*

305 Wang tiles are very close to the abstract tile assembly models (aTAM) [61, 45] which can
 306 be used as a model of computation with applications in DNA computing [49]. Our model is
 307 able to represent aTAM, and even generalise it.

308 A tile $t = (g_w, g_e, g_s, g_n)$ in the aTAM is a Wang tile where the sides $g \in \mathcal{G}$ (called
 309 glue types instead of colours) are associated to an integer $\text{str}(g) \geq 0$ called their strength.
 310 An additional model parameter is called the *temperature*, which we write τ . A tile can be
 311 attached to an assembly (i.e. a diagram in the stellar resolution terminology) if adjacent
 312 sides have matching glue type and strength, and the total strength of its connections is
 313 at least⁴ τ . To encode those we use colours h, v of arity 3 as above but decline them in
 314 pairs h_-, h_+ and v_-, v_+ , together with a binary symbol G , and define for all glue type
 315 $g \in \mathcal{G}$ the term $\text{gl}(g)(x) = G(g(x), s^{\text{str}(g)}(0))$. Temperature is dealt with by adding *ambiance*
 316 *stars* that are required to connect between tile encodings. Ambiance stars connect to a tile
 317 translation on several sides and force the sum of forces to be larger than the temperature.
 318 Ambiance stars are defined as $A^{a,b,c,d} = [-v_-(G(x_a, s^a(y_a)), z, w), +v_+(G(x_b, s^b(y_b)), z, w),$
 319 $-h_-(G(x_c, s^c(y_c)), z, w), +h_+(G(x_d, s^d(y_d)), z, w), +v_+(G(x_a, s^a(y_a)), z_1, w_1),$
 320 $-v_-(G(x_b, s^b(y_b)), z_2, w_2), +h_+(G(x_c, s^c(y_c)), z_3, w_3), -h_-(G(x_d, s^d(y_d)), z_4, w_4)]$ for all $0 \leq$
 321 $a, b, c, d \leq \tau$ such that $a + b + c + d = \tau$. We also require the addition of stars $[-v_+], [+v_-],$
 322 $[-h_+], [+v_-]$ to *plug* connexions where no tiles can be added.

323 ► **Definition 36** (Encoding of self-assembling tiles). *Let $t^i = (g_w^i, g_e^i, g_s^i, g_n^i)$, $i \in I$ (countable*
 324 *but possibly infinite) be a set of self-assembling tiles. The tile t^i is encoded as the star $(t^i)^\star =$
 325 $[-h_+(\text{gl}(g_w^i)(x), x, y), -v_+(\text{gl}(g_s^i)(y), x, y), +h_-(\text{gl}(g_e^i)(s(x)), x, y), +v_-(\text{gl}(g_n^i)(s(y)), x, y)]$.*

326 *The aTAM model with tiles $(t^i)_{i \in I}$ and temperature τ is encoded as $\sum_{a+b+c+d=\tau} A^{a,b,c,d} +$
 327 $\sum_{i \in I} (t^i)^\star + [-v_+] + [+v_-] + [-h_+] + [+v_-]$.*

3 Interpreting the computational content of MLL

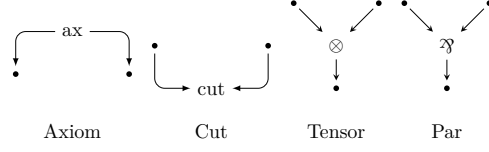
329 In this section, we restrict the definitions to the tree-like normalisation i.e $\text{Ex}(\Sigma)$ will be a
 330 shortcut for $\text{Ex}_{\mathcal{A}}^{\text{tree}}(\Sigma)$ where \mathcal{A} is a set of colour left implicit.

⁴ As a consequence, a tile can connect to an assembly in temperature 2 through two faces of strength 1, but cannot connect through a single of those faces.

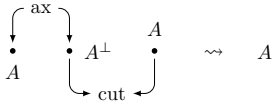
$A, B = X_i \mid X_i^\perp \mid A \otimes B \mid A \wp B$ where $i \in \mathbf{N}$

(a) MLL Formulas

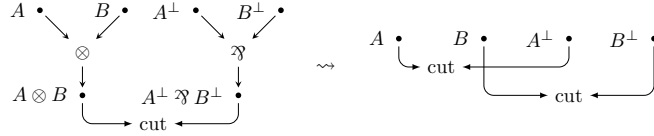
$$\frac{}{\vdash A, A^\perp} \quad \frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} \quad \frac{\vdash \Gamma, A \quad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta} \quad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \wp B}$$



(b) Sequent calculus rules



(c) Proof-structures



(d) Cut-elimination reductions: axiomatic cut (left), multiplicative cut (right)

■ **Figure 1** Syntax of Multiplicative Linear Logic (MLL)

3.1 Multiplicative Linear Logic

331

332 Multiplicative linear logic (MLL) is a fragment of linear logic [18] restricted to the tensor \otimes
 333 and par \wp connectives. The set \mathcal{F}_{MLL} of MLL formulas is defined by the grammar of the
 334 figure 1a. Linear negation $(\cdot)^\perp$ is extended to formulas by involution and de Morgan laws:
 335 $X^{\perp\perp} = X$, $(A \otimes B)^\perp = A^\perp \otimes B^\perp$, and $(A \wp B)^\perp = A^\perp \wp B^\perp$.

336 Proofs of MLL can be written in the traditional sequent calculus fashion [15, 16] using the
 337 set of rules shown in figure 1b. In his seminal paper [18], Girard introduced an alternative
 338 syntax, akin to *natural deduction*, based on a graph-theoretic representation of proofs. In this
 339 syntax, one considers the notion of proof-structure, i.e. an directed hypergraph with vertices
 340 labelled by formulas and constructed from hyperedges⁵ labelled within $\{ax, cut, \otimes, \wp\}$ and
 341 satisfying the arities and labelling constraints shown in figure 1c. A proof-structure also
 342 satisfies the additional constraint that each vertex must be (1) the target of exactly one
 343 hyperedge, and (2) the source of at most one hyperedge. When needed, a proof-structure
 344 will be defined as a 6-tuple $(V, E, s, t, \ell_V, \ell_E)$, where (V, E, s, t) is a directed hypergraph (see
 345 Definition 31) and $\ell_V : V \rightarrow \mathcal{F}_{\text{MLL}}$, $\ell_E : E \rightarrow \{\otimes, \wp, ax, cut\}$ are labelling maps.

346 The cut-elimination procedure, which is defined in the natural way for MLL sequent
 347 calculus, becomes a graph-rewriting system on proof-structures, defined by the two rewriting
 348 rules (figure 1d). The following definition explains how sequent calculus proofs can be
 349 represented as proof-structures.

350 ► **Definition 37** (Translation of MLL sequent calculus). *We define a translation $\llbracket \cdot \rrbracket$ from MLL
 351 sequent calculus derivations to proof-structures. The proof-structures of the domain of $\llbracket \cdot \rrbracket$ are
 352 called the proof-nets.*



353

⁵ For practical purposes, the source edges are ordered, and we will talk about the "left" and "right" sources since there never are more than two; illustrations implicitly represent the left (resp. right) source on the left (resp. right).

XX:10 Stellar Resolution: Multiplicatives

$$\begin{array}{ccc}
\begin{array}{c} \pi \\ \vdots \\ \vdash \Gamma, A, B \\ \hline \vdash \Gamma, A \wp B \end{array} & \rightsquigarrow [\cdot] & \begin{array}{c} \boxed{[\pi]} \\ \swarrow \quad \searrow \\ A \quad B \\ \searrow \quad \swarrow \\ \wp \\ \downarrow \\ A \wp B \end{array} \\
354 & &
\end{array}
\qquad
\begin{array}{ccc}
\begin{array}{c} \pi_1 \quad \pi_2 \\ \vdots \quad \vdots \\ \vdash \Gamma, A \quad \vdash \Delta, B \\ \hline \vdash \Gamma, \Delta, A \otimes B \end{array} & \rightsquigarrow [\cdot] & \begin{array}{c} \boxed{[\pi_1]} \quad \boxed{[\pi_2]} \\ \swarrow \quad \searrow \quad \swarrow \quad \searrow \\ \Gamma \quad A \quad B \quad \Delta \\ \searrow \quad \swarrow \\ \otimes \\ \downarrow \\ A \otimes B \end{array} \\
355 & &
\end{array}$$

355 Note that this translation is not surjective, and some proof-structures do not represent
356 sequent calculus proofs. This is tackled by the *correctness criterion* which characterises those
357 proof-structures that do translate sequent calculus proofs through topological properties.
358 This is discussed in the next section but for the time being we define the notion of proof-net.

359 ► **Definition 38** (Proof-nets). A proof-net is a proof-structure \mathcal{S} such that there exists a
360 MLL sequent calculus proof π with $\mathcal{S} = \llbracket \pi \rrbracket$.

361 3.2 Reconstruction of proof-structures and their dynamics

362 Let us note that proof-structures can be defined inductively. A proof-structure with only
363 one hyperedge is necessarily an axiom with two conclusions. Then a proof-structure with n
364 hyperedges is either built from the union of two proof-structures with respectively k and
365 $n - k$ hyperedges, or from a proof-structure with $n - 1$ hyperedges extended by either a \otimes , \wp ,
366 or *cut* hyperedge on two of its conclusions. In the following we use this inductive definition
367 to define the *address* of occurrences of atoms in a proof-structure.

368 ► **Definition 39.** A vertex v is above another vertex u in a proof-structure if there exists a
369 directed path from v to u going through only \otimes and \wp hyperedges.

370 ► **Definition 40** (Address). We now consider a signature in which there exists at least two
371 unary function symbols \mathbf{r}, \mathbf{l} and unary functions symbols p_A for all occurrences⁶ of MLL
372 formulas A . We define the partial address $\mathbf{pAddr}_{\mathcal{S}}(d, x)$ of an occurrence of atom d in a MLL
373 proof-structure \mathcal{S} , with respect to the variable x , inductively:

- 374 ■ $\mathbf{pAddr}_{\mathcal{S}}(d, x) = x$ when \mathcal{S} consists only of an axiom hyperedge;
- 375 ■ $\mathbf{pAddr}_{\mathcal{S}}(d, x) = \mathbf{pAddr}_{\mathcal{S}_i}(d, x)$ if \mathcal{S} is the union of two smaller proof-structures $\mathcal{S}_1, \mathcal{S}_2$ and
376 d appears in \mathcal{S}_i ;
- 377 ■ $\mathbf{pAddr}_{\mathcal{S}}(d, x) = \mathbf{l}(\mathbf{pAddr}_{\mathcal{S}'}(d, x))$ (resp. $\mathbf{pAddr}_{\mathcal{S}}(d, x) = \mathbf{r}(\mathbf{pAddr}_{\mathcal{S}'}(d, x))$) if \mathcal{S} is obtained
378 from \mathcal{S}' by adding a \otimes or \wp hyperedge e , and if d is above the left source (resp. the right
379 source) of e .
- 380 ■ $\mathbf{pAddr}_{\mathcal{S}}(d, x) = \mathbf{pAddr}_{\mathcal{S}'}(d, x)$ otherwise.

381 The partial address of d is defined with respect to either a conclusion of the structure or the
382 source of a cut hyperedge, which is uniquely defined as the occurrence of formula c such that
383 d is above c and c is not source of either a \otimes or a \wp ; the address of d is then defined as the
384 term $\mathbf{Addr}_{\mathcal{S}}(d, x) = p_c(\mathbf{pAddr}_{\mathcal{S}}(d, x))$.

385 ► **Notation 41.** Let \mathcal{S} be a proof-structure. We write $\mathbf{Ax}(\mathcal{S})$ (resp. $\mathbf{Cut}(\mathcal{S})$) the set of axioms
386 (resp. cut) hyperedges in \mathcal{S} . Given $e \in \mathbf{Ax}(\mathcal{S})$ ($e \in \mathbf{Cut}(\mathcal{S})$), we write C_e^l and C_e^r the left and
387 right conclusions (resp. sources) of e respectively.

388 ► **Definition 42** (Vehicle). The vehicle of proof-structure \mathcal{S} is the constellation $\mathcal{S}^{\mathbf{ax}}$ defined
389 by: $|\mathcal{S}^{\mathbf{ax}}| = \mathbf{Ax}(\mathcal{S})$ and $\alpha_{\mathcal{S}^{\mathbf{ax}}}(e) = [\mathbf{Addr}_{\mathcal{S}}(C_e^l), \mathbf{Addr}_{\mathcal{S}}(C_e^r)]$.

⁶ The set of formulas \mathcal{F} is countable, and there are only finite numbers of occurrences of a given formula in a given proof-structure, hence the set $\mathcal{F} \times \mathbf{N}$ suffices and is still countable.

390 ► **Definition 43** (Colouration of constellation). *The colouration $\pm c.\Sigma$ of Σ with a colour $c \in \mathcal{C}$*
 391 *and a polarity $\pm \in \{+, -\}$ changes all the rays of its stars by the following functions ϕ_c^\pm*
 392 *defined by: $\phi_c^\pm(+d.t) = \pm c.t$ $\phi_c^\pm(-d.t) = \pm c.t$ $\phi_c^\pm(t) = \pm c.t$.*

393 ► **Notation 44.** Let Σ_1, Σ_2 be two constellations. We write $\Sigma_1 \cdot \Sigma_2$ their multiset union, i.e.
 394 the coproduct $\rho_{\Sigma_1} + \rho_{\Sigma_2} : |\Sigma_1| + |\Sigma_2| \rightarrow \text{Stars}(\mathbb{S})$.

395 ► **Definition 45** (Translation of proof-structures). *The translation of a proof-structure \mathcal{S} into*
 396 *a constellation is defined as $\mathcal{S}^\star = (\mathcal{S}^{\text{ax}}, \mathcal{S}^{\text{cut}})$ where \mathcal{S}^{cut} is the constellation defined by*
 397 *$|\mathcal{S}^{\text{cut}}| = \text{Cut}(\mathcal{S})$ and $\alpha_{\mathcal{S}^{\text{cut}}}(e) = [p_{C_e^l}(x), p_{C_e^r}(x)]$. Execution is extended to the translation of*
 398 *proof-structures: $\text{Ex}(\mathcal{S}^\star) = \text{Ex}(+c.\Sigma_{\text{ax}} \cdot -c.\Sigma_{\text{cut}})$.*

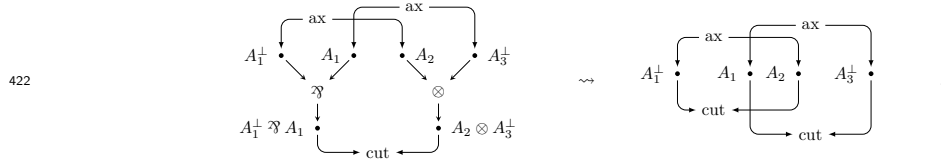
399 ► **Lemma 46.** *Let \mathcal{R} be a MLL proof-structure reducing in one step to another proof-structure*
 400 *\mathcal{S} . $\text{Ex}(\mathcal{R}^\star) = \text{Ex}(\mathcal{S}^\star)$.*

401 **Proof.** Since each stars are unique by definition of **Addr** the connexions are never am-
 402 biguous in the proof. We have two cases of reduction. If we have a cut on an ax-
 403 iom, the corresponding execution will be $\text{Ex}(\mathcal{R}^\star) = \text{Ex}(\Sigma_{\mathcal{R}} + [+c.p_{A_1}(t), +c.p_{A_2}(x)] +$
 404 $[-c.p_{A_2}(x), -c.p_{A_3}(x)])$. The cut $[-c.p_{A_2}(x), -c.p_{A_3}(x)]$ connects $[+c.p_{A_1}(t), +c.p_{A_2}(x)]$
 405 and another axiom $[+c.p_{A_3}(u), +c.p_{A_4}(v)]$. We can connect them all to produce the star
 406 $\theta[+c.p_{A_1}(t), +c.p_{A_4}(v)]$ where $\theta = \{x \mapsto u\}$. We identified A_1 and A_3 two occurrences of A .
 407 This is exactly how \mathcal{S} is translated thus $\text{Ex}(\mathcal{R}^\star) = \text{Ex}(\mathcal{S}^\star)$. If we have a \wp/\otimes cut, the cor-
 408 responding executions are $\text{Ex}(\mathcal{R}^\star) = \text{Ex}(\Sigma_{\mathcal{R}} + [-c.p_{A \otimes B}(x), -c.p_{A \wp B^+}(x)])$ and $\text{Ex}(\mathcal{S}^\star) =$
 409 $\text{Ex}(\Sigma_{\mathcal{S}} + [-c.p_A(x), -c.p_{A^+}(x)] + [-c.p_B(x), -c.p_{B^+}(x)])$. The \wp and \otimes vertices have both
 410 two premises translated into rays $+c.p_{A \wp B^+}(\mathbf{1} \cdot u)$, $+c.p_{A^+ \wp B^+}(\mathbf{r} \cdot u')$, $+c.p_{A \otimes B}(\mathbf{1} \cdot t)$ and
 411 $+c.p_{A \otimes B}(\mathbf{r} \cdot t')$ of axiom stars in $\Sigma_{\mathcal{R}}$. In order to construct a diagram, the cut has to be
 412 duplicated twice to saturate these rays. Since a \wp and \otimes vertex disappear after the reduc-
 413 tion, the previous axioms are relocalised into $+c.p_{A^+}(u)$, $+c.p_{B^+}(u')$, $+c.p_A(t)$ and $+c.p_B(t')$
 414 in $\Sigma_{\mathcal{S}}$. The cuts $[-c.p_A(x), -c.p_{A^+}(x)]$ and $[-c.p_B(x), -c.p_{B^+}(x)]$ connect these rays in
 415 the same way as in $\Sigma_{\mathcal{R}}$. Since $\Sigma_{\mathcal{S}}$ is $\Sigma_{\mathcal{R}}$ with a relocalisation of axioms they both have
 416 the same free rays. Moreover, all connexions with cuts does not involve free rays, hence
 417 $\text{Ex}(\mathcal{R}^\star) = \text{Ex}(\mathcal{S}^\star)$. ◀

418 ► **Theorem 47** (Dynamics). *For a proof-net \mathcal{R} of normal form \mathcal{S} , we have $\text{Ex}(\mathcal{R}^\star) = \mathcal{S}^\star$.*

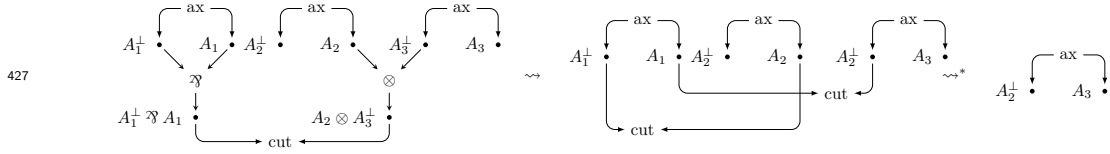
419 **Proof.** This result is a consequence of lemma 46 by induction of the number of cut-elimination
 420 steps from \mathcal{R} to \mathcal{S} , as well as the fact that $\text{Ex}(\mathcal{S}^\star) = \mathcal{S}^\star$ since \mathcal{S} does not contain cuts. ◀

421 ► **Example 48.** Take the following reduction $\mathcal{S} \rightsquigarrow \mathcal{S}'$ of proof-structure:



423 then $\text{Ex}(\mathcal{S}^\star) = \text{Ex}([+c.p_{A_1^+ \wp A_1}(\mathbf{1} \cdot x), +c.p_{A_2 \otimes A_3^+}(\mathbf{r} \cdot x)] + [-c.p_{A_1^+ \wp A_1}(x), -c.p_{A_2 \otimes A_3^+}(x)] +$
 424 $[+c.p_{A_2 \otimes A_2^+}(\mathbf{1} \cdot x), +c.p_{A_1^+ \wp A_1}(\mathbf{r} \cdot x)])$ and $\text{Ex}(\mathcal{S}^\star) = \emptyset$. This is equal to $\text{Ex}(\mathcal{S}'^\star)$ as the only
 425 correct saturated diagram for \mathcal{S}^\star (made by duplicating the star representing the cut) has no
 426 free rays. If we look at the following reduction $\mathcal{S} \rightsquigarrow^* \mathcal{S}'$ instead:

XX:12 Stellar Resolution: Multiplicatives



428 Then $\text{Ex}(\mathcal{S}^\star) = [+c.p_{A_1^\perp \mathfrak{A} A_1}(\mathbf{1} \cdot x), +c.p_{A_1^\perp \mathfrak{A} A_1}(\mathbf{r} \cdot x)] + [+c.p_{A_2^\perp}(x), +c.p_{A_2 \otimes A_3^\perp}(\mathbf{1} \cdot x)] +$
 429 $[+c.p_{A_2 \otimes A_3^\perp}(\mathbf{r} \cdot x), +c.p_{A_3}(x)] + [-c.p_{A_1^\perp \mathfrak{A} A_1}(x), -c.p_{A_2 \otimes A_3^\perp}(x)]$. Computing the executions,
 430 one obtains $\text{Ex}(\mathcal{S}^\star) = [+c.p_{A_2^\perp}(x), +c.p_{A_3}(x)] = \mathcal{S}'^\star$.

4 Interpreting the logical content of MLL

4.1 Correctness of proof-structures

433 As mentioned above, proof-structures are more permissive than sequent calculus proofs.
 434 In other words, some proof-structures do not represent proofs, and the syntax of MLL is
 435 therefore restricted to proof-nets, i.e. proof-structures that do represent sequent calculus
 436 proofs. A beautiful result of Girard, analysed by many subsequent works [10, 9, 37, 41, 11], is
 437 that those proof-structures that are proof-nets can be characterised by a topological property
 438 called a *correctness criterion*. While Girard's original criterion, called the long-trip criterion
 439 [18], is about the set of walks in a proof-structure, we will here work with Danos and Regnier's
 440 simplified criterion [10].

441 ► **Notation 49.** Given a proof-structure $\mathcal{S} = (V, E, s, t, \ell_V, \ell_E)$, we write $\mathfrak{A}(\mathcal{S})$ the subset
 442 $P \subseteq E$ of \mathfrak{A} -labelled edges, i.e. $\mathfrak{A}(\mathcal{S}) = \{e \in E \mid \ell_E(e) = \mathfrak{A}\}$.

443 We now define *correction graphs*, which are the undirected hypergraphs obtained by
 444 removing one source of each \mathfrak{A} -labelled edge. The Danos-Regnier criterion then states that a
 445 proof-structure is a proof-net if and only if all correction graphs are trees.

446 ► **Definition 50 (Correction Graph).** Let $\mathcal{S} = (V, E, s, t, \ell_V, \ell_E)$ be a proof-structure. A
 447 *switching* is a map $\sigma : \mathfrak{A}(\mathcal{S}) \rightarrow \{l, r\}$. The *correction hypergraph* \mathcal{S}_σ is the undirected
 448 hypergraph (V, E, s') induced by the switching σ is defined by letting $s'(e) = \{v\} \cup t(e)$ where
 449 v is the left (resp. right) source of v in \mathcal{S} when $e \in \mathfrak{A}(\mathcal{S})$ and $\sigma(e) = l$ (resp. $\sigma(e) = r$), and
 450 $s'(e) = s(e) \cup t(e)$ for $e \notin \mathfrak{A}(\mathcal{S})$.

451 ► **Theorem 51 (Danos-Regnier correctness criterion [10]).** A proof-structure \mathcal{S} is a proof-net
 452 if and only if \mathcal{S}_σ is a tree for all switching σ .

453 ► **Remark 52.** Each correction graphs can be defined as the union of two graphs: one which
 454 comes from the axioms and is uniquely defined by the proof-structure, and one which is
 455 obtained from edges that are not axioms and is dependent on the switching. This point of
 456 view allows for an *interactive* formulation of the correctness criterion in which the set of
 457 axioms is *tested* against graphs corresponding to switchings [42].

4.2 Reconstruction of correctness

459 We have already seen in the previous section how constellations can represent proofs. We
 460 now explain how to define tests to allow for an interactive, internal, representation of the
 461 correctness criterion. This is done by translating the Danos-Regnier criterion within the
 462 framework of stellar resolution.

463 We now use two colours c (computation) and t (testing). A vehicle will be coloured with
 464 the colour c when we want its execution by connecting it with cuts and it will be coloured
 465 with the colour t when being subject to tests against ordeals.

466 ▶ **Definition 53** (Ordeal). Let \mathcal{S} be a MLL proof-structure and σ one of its switchings. The
 467 ordeal \mathcal{S}_σ^\star associated to \mathcal{S}_σ is the constellation obtained by translating all the vertices of \mathcal{S}_σ^\star
 468 in the following way:

- 469 ■ $(C_e^d)^\star = [-t.\text{Addr}_S(C_e^d), +c.q_{C_e^d}(x)]$ for $e \in \text{Ax}(\mathcal{S})$,
- 470 ■ $(A \wp_L B)^\star = [-c.q_A(x)] + [-c.q_B(x), +c.q_{A\wp B}(x)]$,
- 471 ■ $(A \wp_R B)^\star = [-c.q_A(x), +c.q_{A\wp B}(x)] + [-c.q_B(x)]$,
- 472 ■ $(A \otimes B)^\star = [-c.q_A(x), -c.q_B(x), +c.q_{A\otimes B}(x)]$,
- 473 ■ We add $[-c.q_A(x), p_A(x)]$ for each conclusion A

474 We define $q_A(t)$ as a shortcut for $p_A(\mathbf{g} \cdot t)$ with \mathbf{g} a constant only used for that definition.

475 ▶ **Theorem 54** (Stellar correctness criterion). A proof-structure \mathcal{S} is a proof-net if and only if
 476 for all switching σ , we have $\text{Ex}(+t.\mathcal{S}^{\text{ax}} \cdot -c.\mathcal{S}^{\text{cut}} \cdot \mathcal{S}_\sigma^\star) = [p_{A_1}(x), \dots, p_{A_n}(x)]$ where A_1, \dots, A_n
 477 are the conclusions of \mathcal{S} .

478 **Proof (Sketch).** (\Rightarrow) One can observe that an ordeal \mathcal{S}_σ^\star perfectly reproduces the structure of
 479 $\text{Sub} = (V^{\mathcal{S}_\sigma}, E^{\mathcal{S}_\sigma} - \text{Ax}(\mathcal{S}), s')$ which is always a forest (because it is made of the syntactic tree
 480 of A_1, \dots, A_n) and so is $(-c.\mathcal{S}^{\text{cut}} \cdot \mathcal{S}_\sigma^\star)[c, t]$. We can construct a saturated diagram of $-c.\mathcal{S}^{\text{cut}} \cdot$
 481 \mathcal{S}_σ^\star by connecting all its stars and by cancelling the free rays not corresponding to conclusions
 482 thanks to the cuts in $-c.\mathcal{S}^{\text{cut}}$. This connexion corresponds to the contraction of a forest and we
 483 end up with the constellation $\Sigma_{\text{ord}} = \sigma_{A_1} + \dots + \sigma_{A_n}$ where $p_{A_i}(x), -t.B_1(t_1), \dots, -t.B_m(t_m) \in$
 484 σ_{A_i} for each B_1, \dots, B_m subformulas of A_i which are conclusions of axioms. If \mathcal{S} is a proof-net
 485 then \mathcal{S}_σ must be tree. In this case, $(+t.\mathcal{S}^{\text{ax}} \cdot \Sigma_{\text{ord}})[c, t]$ must be acyclic (otherwise, we would
 486 have a cycle in \mathcal{S}_σ). Since all matchings are exact (i.e produce equations $t \doteq t$), we have a
 487 unique correct saturated diagram with free rays p_{A_1}, \dots, p_{A_n} . (\Leftarrow) If $\text{Ex}(+t.\mathcal{S}^{\text{ax}} \cdot -c.\mathcal{S}^{\text{cut}} \cdot \mathcal{S}_\sigma^\star) =$
 488 $[p_{A_1}(x), \dots, p_{A_n}(x)]$, it means that $(+t.\mathcal{S}^{\text{ax}} \cdot -c.\mathcal{S}^{\text{cut}} \cdot \mathcal{S}_\sigma^\star)[c, t]$ is acyclic (otherwise we would
 489 have infinitely many correct saturated diagrams because only axioms cause cycles). Since
 490 the ordeal together with the translation of axioms are designed to reproduce the structure
 491 of \mathcal{S}_σ , then \mathcal{S}_σ must be acyclic. Since the normalisation produces only one unique star, \mathcal{S}_σ
 492 must be connected. Therefore, \mathcal{S}_σ is a tree and \mathcal{S} is a proof-net. ◀

493 ▶ **Corollary 55** (Corollary of the proof of Theorem 54). All correction graphs of a proof-structure
 494 \mathcal{S} are acyclic if and only if $+t.\mathcal{S}^{\text{ax}} \cdot -c.\mathcal{S}^{\text{cut}} \cdot \mathcal{S}_\sigma^\star$ is strongly normalising.

495 ▶ **Example 56.** We have two proof-structures for a chosen switching together with their
 496 corresponding ordeal⁷:

$$\begin{array}{c}
 \begin{array}{c}
 \text{ax} \quad \text{ax} \\
 \begin{array}{c}
 A \quad B \quad A^\perp \\
 \diagdown \quad \diagup \quad \diagdown \\
 \otimes \quad \wp_L \\
 A \otimes B \quad A^\perp \wp B^\perp
 \end{array}
 \end{array}
 \quad B^\perp
 \end{array}
 \begin{array}{l}
 \left[\frac{-t.p_{A\otimes B}(\mathbf{1} \cdot x)}{+c.q_A(x)} + \frac{-t.p_{A\wp B}(\mathbf{1} \cdot x)}{+c.q_B(x)} + \frac{-t.p_{A\otimes B}(\mathbf{r} \cdot x)}{+c.q_{A^\perp}(x)} + \frac{-t.p_{A\wp B}(\mathbf{r} \cdot x)}{+c.q_{B^\perp}(x)} \right] + \\
 \left[\frac{-c.q_A(x)}{+c.q_{A\otimes B}(x)} \quad \frac{-c.q_B(x)}{+c.q_{A\otimes B}(x)} \right] + \left[\frac{-c.q_{A^\perp}(x)}{+c.q_{A^\perp \wp B^\perp}(x)} \right] + \\
 \left[\frac{-c.q_{A\otimes B}(x)}{p_{A\otimes B}(x)} \right] + \left[\frac{-c.q_{A^\perp \wp B^\perp}(x)}{p_{A^\perp \wp B^\perp}(x)} \right]
 \end{array}$$

$$\begin{array}{c}
 \begin{array}{c}
 \text{ax} \\
 \begin{array}{c}
 A \quad A^\perp \\
 \diagdown \quad \diagup \\
 \otimes \\
 A \otimes A^\perp
 \end{array}
 \end{array}
 \end{array}
 \begin{array}{l}
 \left[\frac{-t.p_{A\otimes A^\perp}(\mathbf{1} \cdot x)}{+c.q_A(x)} \right] + \left[\frac{-t.p_{A\otimes A^\perp}(\mathbf{r} \cdot x)}{+c.q_{A^\perp}(x)} \right] + \left[\frac{-c.q_A(x)}{+c.q_{A\otimes A^\perp}(x)} \quad \frac{-c.q_{A^\perp}(x)}{+c.q_{A\otimes A^\perp}(x)} \right] + \\
 \left[\frac{-c.q_{A\otimes A^\perp}(x)}{p_{A\otimes A^\perp}(x)} \right]
 \end{array}$$

499 The first one will be connected with the axioms $[+t.p_{A\otimes B}(\mathbf{1} \cdot x), +t.p_{A^\perp \wp B^\perp}(\mathbf{1} \cdot x)] +$
 500 $[+t.p_{A\otimes B}(\mathbf{r} \cdot x), +t.p_{A^\perp \wp B^\perp}(\mathbf{r} \cdot x)]$. By connecting all stars of the ordeal, we will form a
 501 unique correct tree-like saturated diagram normalising into the conclusion stars $[p_{A\otimes B}(x)] +$

⁷ The purely esthetical use of the fractional notation is used to ease the reading of the stars.

XX:14 Stellar Resolution: Multiplicatives

502 $[p_{A^\perp \wp B^\perp}(x)]$ since they are the only free rays. From the second ordeal, when connected
 503 to $[+t.p_{A \otimes A^\perp}(1 \cdot x), +t.p_{A \otimes A^\perp}(r \cdot x)]$, we can construct infinitely many correct saturated
 504 diagrams because of the cycle. So the vehicle does not satisfy the stellar correctness criterion.

4.3 Reconstruction of formulas

506 We here follow standard realisability constructions for linear logic [24, 33, 50]. Note that we
 507 explicit the *trefoil property* [52] instead of the special case that is usually called *adjunction*.

508 ► **Definition 57** (Orthogonality). *We say that two constellations Σ_1, Σ_2 are orthogonal w.r.t.*
 509 *a set of colours $\mathcal{A} \subseteq \mathcal{C}$, written $\Sigma_1 \perp_{\mathcal{A}} \Sigma_2$, when $\text{Ex}_{\mathcal{A}}(\Sigma_1 \cdot \Sigma_2)$ is strongly normalising. The*
 510 *orthogonal of a set of constellations is defined by $\mathbf{A}^{\perp_{\mathcal{A}}} = \{\Sigma \mid \forall \Sigma' \in \mathbf{A}, \Sigma \perp_{\mathcal{A}} \Sigma'\}$.*

511 ► **Notation 58.** Let (P, \preceq) be a partially ordered set, and $A \subseteq P$. We write $\text{prefix}(A)$ the
 512 set of *prefixes in A* i.e $\text{prefix}(A) = \{a \in A \mid \forall b \in A, \neg(b \preceq a)\}$.

513 ► **Definition 59** (Order on rays). *We define the following partial order on the set of rays:*
 514 *given r, r' two rays, $r \preceq r'$ if and only there exists a substitution θ such that $\theta r = r'$.*

515 We leave the verification that this defines a partial order to the reader. More intuitively,
 516 we have $r \preceq r'$ when r is less specialized (thus more general) than r' . For instance,
 517 $\pm f(x) \preceq \pm f(g(y))$ because for $\theta = \{x \mapsto g(y)\}$, we have $\theta(\pm f(x)) = \pm f(g(y))$.

518 ► **Definition 60** (Location). *We define:*

- 519 ■ *the location $\# \sigma$ of a star $\rho_\sigma : |\sigma| \rightarrow \text{rays}(\mathbb{S})$ as the set $\text{prefix}(\{\rho_\sigma(s) \mid s \in |\sigma|\})$;*
- 520 ■ *the location $\# \Sigma$ of a constellation $\alpha_\Sigma : |\Sigma| \rightarrow \text{Stars}(\mathbb{S})$ as the set $\text{prefix}(\cup_{\sigma \in |\Sigma|} \# \alpha_\Sigma(\sigma))$;*
- 521 ■ *the location $\# \mathbf{A}$ of a set \mathbf{A} of constellations as the set $\text{prefix}(\cup_{\Sigma \in \mathbf{A}} \# \Sigma)$.*

522 ► **Definition 61** (Conduct). *A set of constellation \mathbf{A} is a conduct w.r.t. a set of colours*
 523 *$\mathcal{A} \subseteq \mathcal{C}$ if there exists a set of constellation \mathbf{B} such that $\mathbf{A} = \mathbf{B}^{\perp_{\mathcal{A}}}$.*

524 ► **Proposition 62** (Biorthogonal closure). *A set of constellations \mathbf{A} is a conduct w.r.t. a set*
 525 *of colours $\mathcal{A} \subseteq \mathcal{C}$ if and only if $\mathbf{A} = (\mathbf{A}^{\perp_{\mathcal{A}}})^{\perp_{\mathcal{A}}}$.*

526 ► **Definition 63** (Intersection up to unification). *Let R and Q be sets of rays. We define their*
 527 *intersection up to unification as the set:*

$$528 \quad R \pitchfork Q = \text{prefix}(\{m \in \text{rays}(\mathbb{S}) \mid \exists r \in R, q \in Q, r \preceq m \text{ and } q \preceq m\}).$$

529 *We say that R and Q are disjoint when $R \pitchfork Q = \emptyset$; by extension, we say that two sets of*
 530 *constellations \mathbf{A}, \mathbf{B} are disjoint when $\# \mathbf{A} \pitchfork \# \mathbf{B} = \emptyset$.*

531 ► **Definition 64** (Tensor). *Let \mathbf{A}, \mathbf{B} be disjoint conducts. We define their tensor by*

$$532 \quad \mathbf{A} \otimes_{\mathcal{A}} \mathbf{B} = (\{\Sigma_1 \cdot \Sigma_2 \mid \Sigma_1 \in \mathbf{A}, \Sigma_2 \in \mathbf{B}\})^{\perp_{\mathcal{A}}}.$$

533 ► **Proposition 65** (Associativity/commutativity). *Given $\mathbf{A}, \mathbf{B}, \mathbf{C}$ pairwise disjoint conducts*
 534 *w.r.t. a set of colours $\mathcal{A} \subseteq \mathcal{C}$, we have $\mathbf{A} \otimes_{\mathcal{A}} \mathbf{B} = \mathbf{B} \otimes_{\mathcal{A}} \mathbf{A}$ and $\mathbf{A} \otimes_{\mathcal{A}} (\mathbf{B} \otimes_{\mathcal{A}} \mathbf{C}) = (\mathbf{A} \otimes_{\mathcal{A}} \mathbf{B}) \otimes_{\mathcal{A}} \mathbf{C}$.*

535 ► **Definition 66** (Par and linear implication). *Let \mathbf{A}, \mathbf{B} be conducts w.r.t. a set of colours*
 536 *$\mathcal{A} \subseteq \mathcal{C}$. We define: $\mathbf{A} \wp_{\mathcal{A}} \mathbf{B} = (\mathbf{A}^{\perp_{\mathcal{A}}} \otimes_{\mathcal{A}} \mathbf{B}^{\perp_{\mathcal{A}}})^{\perp_{\mathcal{A}}}$ and $\mathbf{A} \multimap_{\mathcal{A}} \mathbf{B} = \mathbf{A}^{\perp_{\mathcal{A}}} \wp_{\mathcal{A}} \mathbf{B}$.*

537 ► **Theorem 67** (Associativity of execution). *Choose a set of colours $\mathcal{A} \subseteq \mathcal{C}$. For constellations*
 538 *$\Sigma_1, \Sigma_2, \Sigma_3$ such that $\# \Sigma_1 \pitchfork \# \Sigma_2 \pitchfork \# \Sigma_3 = \emptyset$, we have*

$$539 \quad \text{Ex}_{\mathcal{A}}(\Sigma_1 \cdot \text{Ex}_{\mathcal{A}}(\Sigma_2 \cdot \Sigma_3)) = \text{Ex}_{\mathcal{A}}(\text{Ex}_{\mathcal{A}}(\Sigma_1 \cdot \Sigma_2) \cdot \Sigma_3)$$

540 **Proof.** Since all constellations have disjoint locations, their stars cannot be connected
 541 together and we have $\text{Sat}_{\mathcal{A},*}(\Sigma_2 \cdot \Sigma_3) = \text{Sat}_{\mathcal{A},*}(\Sigma_2) \uplus \text{Sat}_{\mathcal{A},*}(\Sigma_3)$. These diagrams can't
 542 be connected to the ones of Σ_1 which has its own saturated diagrams. Hence, $\text{Sat}_{\mathcal{A},*}(\Sigma_1 \cdot$
 543 $\text{Ex}_{\mathcal{A}}(\Sigma_2 \cdot \Sigma_3)) = \text{Sat}_{\mathcal{A},*}(\Sigma_1) \uplus \text{Sat}_{\mathcal{A},*}(\Sigma_2) \uplus \text{Sat}_{\mathcal{A},*}(\Sigma_3)$. With the same reasoning, we obtain
 544 $\text{Sat}_{\mathcal{A},*}(\Sigma_1 \cdot \text{Ex}_{\mathcal{A}}(\Sigma_2 \cdot \Sigma_3)) = \text{Sat}_{\mathcal{A},*}(\text{Ex}_{\mathcal{A}}(\Sigma_1 \cdot \Sigma_2) \cdot \Sigma_3)$. Therefore, $\text{Ex}_{\mathcal{A}}(\Sigma_1 \cdot \text{Ex}_{\mathcal{A}}(\Sigma_2 \cdot \Sigma_3)) =$
 545 $\text{Ex}_{\mathcal{A}}(\text{Ex}_{\mathcal{A}}(\Sigma_1 \cdot \Sigma_2) \cdot \Sigma_3)$. \blacktriangleleft

546 **► Theorem 68 (Trefoil Property).** *Choose a set of colours $\mathcal{A} \subseteq \mathcal{C}$. For all constellations*
 547 $\Sigma_1, \Sigma_2, \Sigma_3$ *s.t. $\# \Sigma_1 \pitchfork \# \Sigma_2 \pitchfork \# \Sigma_3 = \emptyset$:*

548 $(\Sigma_2 \perp_{\mathcal{A}} \Sigma_3 \text{ and } \Sigma_1 \perp_{\mathcal{A}} \text{Ex}_{\mathcal{A}}(\Sigma_2 \cdot \Sigma_3))$ *if and only if $(\Sigma_1 \perp_{\mathcal{A}} \Sigma_2 \text{ and } \text{Ex}_{\mathcal{A}}(\Sigma_1 \cdot \Sigma_2) \perp_{\mathcal{A}} \Sigma_3)$*

549 **► Theorem 69 (Alternative linear disjunction).** *Let \mathbf{A}, \mathbf{B} be two conducts w.r.t. a set of*
 550 *colours $\mathcal{A} \subseteq \mathcal{C}$. Then: $\mathbf{A} \multimap_{\mathcal{A}} \mathbf{B} = \{\Sigma_f \mid \forall \Sigma_a \in \mathbf{A}, \text{Ex}_{\mathcal{A}}(\Sigma_f \cdot \Sigma_a) \in \mathbf{B} \text{ and } \Sigma_f \perp_{\mathcal{A}} \Sigma_a\}$.*

551 As described in several work by the second author [50, 51, 52], the trefoil property and the
 552 associativity of execution ensure that one can define a $*$ -autonomous category with conducts
 553 as objects and vehicles as morphisms. Due to the lack of space, we chose to omit this result
 554 which do not require new proof techniques and involves lots of bureaucratic definitions to
 555 deal with locations. We instead prove full soundness and completeness results.

556 4.4 Truth and Soundness

557 We will now define the interpretation of MLL formulas, which depends on a *basis of*
 558 *interpretation*, and then prove full soundness and full completeness for MLL+MIX, that is
 559 the system obtained by adding to MLL sequent calculus the MIX rule, corresponding to the
 560 axiom scheme $A \wp B \multimap A \otimes B$. It is known that the correctness criterion for the MIX rule
 561 consists in taking the Danos-Regnier correction graphs and checking for acyclicity (but not
 562 connectedness). The proof of Theorem 54 shows how the orthogonality of a vehicle w.r.t.
 563 the ordeal coincides with MLL+MIX correction, since the constellation $+t.\mathcal{S}^{\text{ax}} \cdot -c.\mathcal{S}^{\text{cut}} \cdot \mathcal{S}^{\star}$
 564 is strongly normalisable if and only if the structure \mathcal{S} is acyclic. This is the key ingredient in
 565 the proof of full completeness. We are still working on obtaining similar results for MLL
 566 without MIX, using the results of Section 4.2.

567 We chose to fix the set of colours used, and omit the subscript that appeared in the
 568 constructions of the previous section. We also use a notion of *localised* formulas, following
 569 previous work of the second author [50, 51, 52]: it is defined using the same grammar as MLL
 570 formulas, except that variables are of the form $X_i(j)$, where j is a term (here constructed from
 571 unary symbols \mathbf{r}, \mathbf{l} and p_A for all occurrences of formulas A) used to distinguish occurrences,
 572 and one expect each occurrence to appear at most once in a formula.

573 A *basis of interpretation* is a function Φ associating to each integer $i \in \mathbf{N}$ a conduct
 574 $\Phi(i)$ in such a way that the conducts $(\Phi(i))_{i \in \mathbf{N}}$ are pairwise disjoint. We moreover fix a
 575 bijection $\varphi : \text{Addr}_x(\mathcal{S}) \times \mathbf{N} \rightarrow \mathbf{N}$, where $\text{Addr}_x(\mathcal{S})$ is the set of addresses $\text{Addr}_{\mathcal{S}}(_, x)$ – i.e.
 576 the countable set of all terms of the form $p_A(t(x))$ where A ranges over conclusions of \mathcal{S} and
 577 t is a term constructed from the unary symbols \mathbf{l} and \mathbf{r} . While the interpretation depends
 578 on the choice of φ , we will not indicate it in the notation for the sake of clarity.

579 In the next definition, we use the substitutions θ_r and θ_l which are defined as the identity
 580 for all variables except for x , and are defined respectively by $\theta_r(x) = \mathbf{r}(x)$ and $\theta_l(x) = \mathbf{l}(x)$.

581 **► Definition 70 (Interpretation of formulas in proof-structures).** *Given Φ a basis of interpret-*
 582 *ation, and A a MLL formula occurrence identified by a unique unary function symbol p_A (cf.*
 583 *Definition 40). We define the interpretation $I_{\Phi}(A, t)$ along Φ and a term t inductively:*

XX:16 Stellar Resolution: Multiplicatives

584 ■ $I_{\Phi}(A, t) = \Phi(\varphi(t, i))$ when $A = X_i$;

585 ■ $I_{\Phi}(A, t) = \Phi(\varphi(t, i))^{\perp}$ when $A = X_i^{\perp}$;

586 ■ $I_{\Phi}(A \otimes B, t) = I_{\Phi}(A, u) \otimes I_{\Phi}(B, v)$ where $u = \theta_l(t)$ and $v = \theta_r(t)$;

587 ■ $I_{\Phi}(A \wp B, t) = I_{\Phi}(A, u) \wp I_{\Phi}(B, v)$ where $u = \theta_l(t)$ and $v = \theta_r(t)$.

588 We then define $I_{\Phi}(A)$ as $I_{\Phi}(A, p_A(x))$. We extend the interpretation to sequents by letting

589 $I_{\Phi}(\vdash A_1, \dots, A_n) = I_{\Phi}(A_1) \wp \dots \wp I_{\Phi}(A_n)$.

590 ► **Definition 71** (Proof-like constellations). A constellation Σ is proof-like w.r.t. a set of
591 addresses $\mathfrak{A} = \{l_1, \dots, l_k\}$ if $\sharp\Sigma = \mathfrak{A}$ and Σ consists of binary stars only.

592 The proof of the following result is a simple induction, combined with Theorem 47 to
593 ensure that $\text{Ex}(\mathcal{S}^{\star})$ is proof-like w.r.t. the set $\sharp\mathcal{S}$ of addresses of conclusions of axioms in \mathcal{S} .

594 ► **Theorem 72** (Full soundness). Let \mathcal{S} be a MLL+MIX proof-net of conclusion $\vdash \Gamma$ and Φ a
595 basis of interpretation. We have $\text{Ex}(\mathcal{S}^{\star}) \in I_{\Phi}(\vdash \Gamma)$, and $\text{Ex}(\mathcal{S}^{\star})$ is proof-like w.r.t. $\sharp\mathcal{S}$.

596 We now consider syntax trees of formulas as incomplete proof-structures, where axioms
597 are missing. We can extend the notion of switching to those *pre-proof-structures*, and define
598 their ordeal (as ordeals are defined without considering axioms hyperedges⁸). This is extended
599 to sequents and used in the next lemma: given a sequent $\vdash \Gamma$ one can consider *switchings* σ
600 of $\vdash \Gamma$ and ordeals $(\vdash \Gamma)_{\sigma}^{\star}$. We also define $\sharp\Gamma$ as the set of addresses of occurrences of atoms
601 of Γ seen as a pre-proof-structure.

602 ► **Lemma 73**. Let Φ be a basis of interpretation, $\vdash \Gamma$ a sequent, and σ a switching of $\vdash \Gamma$.
603 Then $\text{Ex}_{\{c\}}((\vdash \Gamma)_{\sigma}^{\star}) \in (I_{\Phi}(\vdash \Gamma))^{\perp}$.

604 **Proof sketch.** The proof is done by induction.

605 ■ If $\vdash \Gamma$ has only formulas X_i or X_i^{\perp} , then there is a single switching (there are no \wp), and
606 $\text{Ex}_{\{c\}}((\vdash \Gamma)_{\sigma}^{\star}) = \sum[-t.p_{X_i}, p_{X_i}]$. Since an element $\Sigma \in I_{\Phi}(\vdash \Gamma)$ is necessarily strongly
607 normalisable and this implies that $\Sigma \cdot \text{Ex}_{\{c\}}((\vdash \Gamma)_{\sigma}^{\star})$ is strongly normalisable, this shows
608 the result.

609 ■ If $\vdash \Gamma = \vdash \Delta, A \wp B$, then a switching σ of $\vdash \Gamma$ is a switching $\bar{\sigma}$ of $\vdash \Delta, A, B$ extended to the
610 additional \wp connective linking A and B . It should be clear that $\text{Ex}_{\{c\}}((\vdash \Delta, A \wp B)_{\sigma}^{\star}) =$
611 $\text{Ex}_{\{c\}}((\vdash \Delta, A, B)_{\bar{\sigma}}^{\star})$. This shows the result, since $I_{\Phi}(\vdash \Delta, A \wp B) = I_{\Phi}(\vdash \Delta, A, B)$.

612 ■ If $\vdash \Gamma = \vdash \Delta, A \otimes B$, a switching of $\vdash \Gamma$ is a switching of $\vdash \Delta, A, B$, and $\text{Ex}_{\{c\}}((\vdash$
613 $\Delta, A \otimes B)_{\sigma}^{\star})$ can be defined from $\text{Ex}_{\{c\}}((\vdash \Delta, A, B)_{\sigma}^{\star})$ by colouring the terms starting
614 by p_A and p_B with a fresh colour $+u$ to obtain a constellation Θ and considering
615 $\text{Ex}_{\{u\}}(\Theta \cdot [-u.p_A(x), -u.p_B(x), p_{A \otimes B}(x)])$. Moreover, one can show that $I_{\Phi}(\vdash \Delta, A \otimes B)$
616 is generated (in the sense of bi-orthogonal closure) by a set of constellations E in which
617 no star connects locations of A with locations of B . This shows the result since this
618 implies that $\text{Ex}_{\{c\}}((\vdash \Delta, A \otimes B)_{\sigma}^{\star}) \in E^{\perp}$ and it is known that $E^{\perp} = E^{\perp\perp}$ in general. ◀

619 ► **Theorem 74** (Full completeness). If a constellation $\Sigma \in I_{\Phi}(\vdash \Gamma)$ is proof-like w.r.t. $\sharp\Gamma$,
620 there exists a MLL+MIX proof-net \mathcal{S} of conclusion $\vdash \Gamma$ such that $\Sigma = \mathcal{S}^{\text{ax}}$.

621 **Proof.** A proof-like constellation $\Sigma \in I_{\Phi}(\vdash \Gamma)$ w.r.t to $\sharp\Gamma$ can always be considered as the
622 interpretation of a proof-structure with only axioms; we can then construct a proof-structure
623 \mathcal{S} by considering the union of the latter with the syntax forest of $\vdash \Gamma$. Since Σ belongs

⁸ We adapt the first case of Definition 53 and introduce the stars for atoms, i.e. for vertices that are not the target of an hyperedge.

624 to $I_{\Phi}(\vdash \Gamma)$, and for all switching σ of $\vdash \Gamma$ (equivalently, of \mathcal{S}) the ordeal $(\vdash \Gamma)_{\sigma}^* = \mathcal{S}_{\sigma}^*$ is
 625 orthogonal to Σ , Corollary 55 shows that \mathcal{S} is acyclic, i.e. satisfies the correctness criterion
 626 for MLL+MIX. ◀

627 5 Perspectives and future works

628 While we have shown here how to reconstruct the multiplicative fragment of linear logic,
 629 an interpretation of exponential connectives should also be possible using stellar resolution.
 630 Extension to the MELL fragment will be particularly interesting since it allows for the
 631 interpretation of System F [17] and pure λ -calculus [9, 46]. In fact, Girard's first article on
 632 Transcendental Syntax [29] sketches some reconstruction of the exponentials, but relies on
 633 constructions from later articles.

634 To interpret the additive connectives $\oplus, \&$, a way to exclude or force some choices in
 635 the construction of diagrams. For this purpose, Girard third article on Transcendental
 636 Syntax [27] mentions some involved coherence relations between stars that is not completely
 637 satisfying. This idea was already properly developed in Seiller's PhD thesis [51] in the
 638 setting of interaction graphs; an improved and extended account can be found in a recent
 639 article by Nguyen and Seiller [43]. We can expect to build on the latter to interpret additive
 640 connectives.

641 Last, but not the least, the third article on Transcendental Syntax [28] suggests to
 642 interpret the terms of first-order logic as multiplicative propositions and the equality as
 643 the linear equivalence. This extension to first-order logic is the initial motivation behind
 644 the present work, and the authors expect to provide a formal account of these ideas. This
 645 would provide computational content for first-order logic in the sense of the Curry-Howard
 646 correspondence, something new and fascinating that would open numerous applications.

647 References

- 648 1 Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for pcf.
 649 *Information and Computation*, 163(2):409–470, 2000.
- 650 2 Clément Aubert and Thomas Seiller. Characterizing co-nl by a group action. *Mathematical*
 651 *Structures in Computer Science*, 26(4):606–638, 2016.
- 652 3 Clément Aubert and Thomas Seiller. Logarithmic space and permutations. *Information and*
 653 *Computation*, 248:2–21, 2016.
- 654 4 Marc Bagnol. *On the resolution semiring*. PhD thesis, Aix-Marseille Université, 2014.
- 655 5 Patrick Baillot and Marco Pedicini. Elementary complexity and geometry of interaction.
 656 *Fundamenta Informaticae*, 45(1-2):1–31, 2001.
- 657 6 Robert Berger. *The undecidability of the domino problem*. Number 66. American Mathematical
 658 Soc., 1966.
- 659 7 Alonzo Church. On the concept of a random sequence. *Bulletin of the American Mathematical*
 660 *Society*, 46(2):130–135, 1940.
- 661 8 Haskell B Curry. Functionality in combinatory logic. *Proceedings of the National Academy of*
 662 *Sciences of the United States of America*, 20(11):584, 1934.
- 663 9 Vincent Danos. *La Logique Linéaire appliquée à l'étude de divers processus de normalisation*
 664 *(principalement du Lambda-calcul)*. PhD thesis, Paris 7, 1990.
- 665 10 Vincent Danos and Laurent Regnier. The structure of multiplicatives. *Archive for Mathematical*
 666 *logic*, 28(3):181–203, 1989.
- 667 11 Paulin Jacobé De Naurois and Virgile Mogbil. Correctness of linear logic proof structures is
 668 nl-complete. *Theoretical Computer Science*, 412(20):1941–1957, 2011.
- 669 12 Yannis Dimopoulos and Alberto Torres. Graph theoretical structures in logic programs and
 670 default theories. *Theoretical Computer Science*, 170(1-2):209–244, 1996.

- 671 13 Thomas Eiter, Giovambattista Ianni, and Thomas Krennwallner. Answer set programming: A
672 primer. In *Reasoning Web International Summer School*, pages 40–110. Springer, 2009.
- 673 14 Michael Gelfond. Answer sets. *Foundations of Artificial Intelligence*, 3:285–316, 2008.
- 674 15 Gerhard Gentzen. Untersuchungen über das logische schließen. i. *Mathematische Zeitschrift*,
675 39(1):176–210, 1935.
- 676 16 Gerhard Gentzen. Untersuchungen über das logische schließen. ii. *Mathematische Zeitschrift*,
677 39(1):405–431, 1935.
- 678 17 Jean-Yves Girard. *Interprétation fonctionnelle et élimination des coupures de l'arithmétique*
679 *d'ordre supérieur*. PhD thesis, Éditeur inconnu, 1972.
- 680 18 Jean-Yves Girard. Linear logic. *Theoretical computer science*, 50(1):1–101, 1987.
- 681 19 Jean-Yves Girard. Geometry of interaction 2: Deadlock-free algorithms. In *International*
682 *Conference on Computer Logic*, pages 76–93. Springer, 1988.
- 683 20 Jean-Yves Girard. Normal functors, power series and λ -calculus. *Annals of pure and applied*
684 *logic*, 37(2):129–177, 1988.
- 685 21 Jean-Yves Girard. Geometry of interaction 1: Interpretation of system f. In *Studies in Logic*
686 *and the Foundations of Mathematics*, volume 127, pages 221–260. Elsevier, 1989.
- 687 22 Jean-Yves Girard. Towards a geometry of interaction. *Contemporary Mathematics*, 92(69-108):6,
688 1989.
- 689 23 Jean-Yves Girard. Geometry of interaction III: accommodating the additives. *London*
690 *Mathematical Society Lecture Note Series*, pages 329–389, 1995.
- 691 24 Jean-Yves Girard. Locus solum: From the rules of logic to the logic of rules. *Mathematical*
692 *structures in computer science*, 11(3):301, 2001.
- 693 25 Jean-Yves Girard. Geometry of interaction V: logic in the hyperfinite factor. *Theoretical*
694 *Computer Science*, 412(20):1860–1883, 2011.
- 695 26 Jean-Yves Girard. Three lightings of logic (invited talk). In *Computer Science Logic 2013*
696 *(CSL 2013)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.
- 697 27 Jean-Yves Girard. Transcendental syntax II: Non-deterministic case. *Logical Methods in*
698 *Computer Science*, to appear, 2016.
- 699 28 Jean-Yves Girard. Transcendental syntax III: equality. *Logical Methods in Computer Science*,
700 2016.
- 701 29 Jean-Yves Girard. Transcendental syntax I: deterministic case. *Mathematical Structures in*
702 *Computer Science*, 27(5):827–849, 2017.
- 703 30 Jacques Herbrand. *Recherches sur la théorie de la démonstration*. PhD thesis, Université de
704 Paris, 1930.
- 705 31 William A Howard. The formulae-as-types notion of construction. *To HB Curry: essays on*
706 *combinatory logic, lambda calculus and formalism*, 44:479–490, 1980.
- 707 32 J Martin E Hyland and C-HL Ong. On full abstraction for pcf: I, ii, and iii. *Information and*
708 *computation*, 163(2), 2000.
- 709 33 Martin Hyland and Andrea Schalk. Glueing and orthogonality for models of linear logic.
710 *Theoretical computer science*, 294(1-2):183–231, 2003.
- 711 34 Neil Immerman. Relational queries computable in polynomial time. *Information and Control*,
712 68(1):86 – 104, 1986. doi:[https://doi.org/10.1016/S0019-9958\(86\)80029-8](https://doi.org/10.1016/S0019-9958(86)80029-8).
- 713 35 Neil Immerman. *Descriptive complexity*. Springer Science & Business Media, 2012.
- 714 36 Robert Kowalski. Predicate logic as programming language. In *IFIP congress*, volume 74,
715 pages 569–544, 1974.
- 716 37 Yves Lafont. From proof nets to interaction nets. *London Mathematical Society Lecture Note*
717 *Series*, pages 225–248, 1995.
- 718 38 J-L Lassez, Michael J Maher, and Kim Marriott. Unification revisited. In *Foundations of logic*
719 *and functional programming*, pages 67–113. Springer, 1988.
- 720 39 Jorge Lobo, Arcot Rajasekar, and Jack Minker. Semantics of horn and disjunctive logic
721 programs. *Theoretical Computer Science*, 86(1):93–106, 1991.

- 722 40 Alberto Martelli and Ugo Montanari. An efficient unification algorithm. *ACM Transactions*
723 *on Programming Languages and Systems (TOPLAS)*, 4(2):258–282, 1982.
- 724 41 Andrzej S Murawski and C-HL Ong. Dominator trees and fast verification of proof nets.
725 In *Proceedings Fifteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No.*
726 *99CB36332)*, pages 181–191. IEEE, 2000.
- 727 42 Alberto Naibo, Mattia Petrolo, and Thomas Seiller. On the computational meaning of axioms.
728 In *Epistemology, Knowledge and the Impact of Interaction*, pages 141–184. Springer, 2016.
- 729 43 Lê Thành Dũng Nguyen and Thomas Seiller. Coherent interaction graphs. *arXiv preprint*
730 *arXiv:1904.06849*, 2019.
- 731 44 Manh Thang Nguyen, Jürgen Giesl, Peter Schneider-Kamp, and Danny De Schreye. Termination
732 analysis of logic programs based on dependency graphs. In *International Symposium on*
733 *Logic-based Program Synthesis and Transformation*, pages 8–22. Springer, 2007.
- 734 45 Matthew J Patitz. An introduction to tile-based self-assembly and a survey of recent results.
735 *Natural Computing*, 13(2):195–224, 2014.
- 736 46 Laurent Regnier. *Lambda-calcul et réseaux*. PhD thesis, Paris 7, 1992.
- 737 47 Colin Riba. Strong normalization as safe interaction. In *22nd Annual IEEE Symposium on*
738 *Logic in Computer Science (LICS 2007)*, pages 13–22. IEEE, 2007.
- 739 48 John Alan Robinson et al. A machine-oriented logic based on the resolution principle. *Journal*
740 *of the ACM*, 12(1):23–41, 1965.
- 741 49 Nadrian C Seeman. Nucleic acid junctions and lattices. *Journal of theoretical biology*, 99(2):237–
742 247, 1982.
- 743 50 Thomas Seiller. Interaction graphs: multiplicatives. *Annals of Pure and Applied Logic*,
744 163(12):1808–1837, 2012.
- 745 51 Thomas Seiller. *Logique dans le facteur hyperfini: géométrie de l’interaction et complexité*.
746 PhD thesis, Aix-Marseille Université, 2012.
- 747 52 Thomas Seiller. Interaction graphs: additives. *Annals of Pure and Applied Logic*, 167(2):95–154,
748 2016.
- 749 53 Thomas Seiller. Interaction graphs: Full linear logic. In *2016 31st Annual ACM/IEEE*
750 *Symposium on Logic in Computer Science (LICS)*, pages 1–10. IEEE, 2016.
- 751 54 Thomas Seiller. Interaction graphs: Graphings. *Annals of Pure and Applied Logic*, 168(2):278–
752 320, 2017.
- 753 55 Thomas Seiller. Interaction graphs: Non-deterministic automata. *ACM Transactions on*
754 *Computational Logic (TOCL)*, 19(3):1–24, 2018.
- 755 56 Thomas Seiller. Interaction graphs: Exponentials. *Logical Methods in Computer Science*, 15,
756 2019.
- 757 57 Thomas Seiller. Probabilistic complexity classes through semantics. *arXiv preprint*
758 *arXiv:2002.00009*, 2020.
- 759 58 Alan Mathison Turing. On computable numbers, with an application to the entscheidungs-
760 problem. *J. of Math*, 58(345-363):5, 1936.
- 761 59 Moshe Y Vardi. The complexity of relational query languages. In *Proceedings of the fourteenth*
762 *annual ACM symposium on Theory of computing*, pages 137–146, 1982.
- 763 60 Hao Wang. Proving theorems by pattern recognition —II. *Bell system technical journal*,
764 40(1):1–41, 1961.
- 765 61 Erik Winfree. *Algorithmic self-assembly of DNA*. PhD thesis, Citeseer, 1998.
- 766 62 Damien Woods, David Doty, Cameron Myhrvold, Joy Hui, Felix Zhou, Peng Yin, and Erik
767 Winfree. Diverse and robust molecular algorithms using reprogrammable dna self-assembly.
768 *Nature*, 567(7748):366–372, 2019. doi:10.1038/s41586-019-1014-9.