# Normalization for Cubical Type Theory

Jonathan Sterling Carlo Angiuli
Carnegie Mellon University

CT 20$\rightarrow$21
August 2021

# □TT: cubical type theory

**□TT** is a variant of Homotopy Type Theory based on an interval:

- a new sort $\boxed{\Gamma \vdash \mathbb{I}}$ and context extension $\Gamma, i : \mathbb{I} \longrightarrow \Gamma$
- with endpoints $\Gamma \vdash 0, 1 : \mathbb{I}$
- and potentially further structure: $r \sqcup s, r \sqcap s, \sim r$ [Coh+17]

Has a "standard model" that's Quillen-equivalent to homotopy types,[1] therefore usable for synthetic homotopy theory.

Computer scientists like **□TT** because it has stronger computational properties than HoTT while retaining its good semantic properties (function extensionality, univalence, effective quotients).

[1] Paper forthcoming by Awodey, Cavallo, Coquand, Riehl, Sattler

# Computation in □TT

Canonicity is one such computational property. Let $\mathcal{C}_{\Box}$ be the syntactic category of **□TT**.

Theorem (Cubical canonicity [Hub18; AFH18])

*If* $M : \mathbb{I}^n \longrightarrow \mathsf{nat}$ *is an n-cube of natural numbers in* $\mathcal{C}_{\Box}$*, then there exists a numeral* $m \in \mathbb{N}$ *such that*

$$\begin{tikzcd} \mathbb{I}^n \arrow[r, "M"] \arrow[d] & \mathsf{nat} \\ \mathbf{1} \arrow[ur, dashed, "S^m(Z)"'] & \end{tikzcd}$$

Therefore **□TT** can be used as a programming language [Ang+21], and we have multiple implementations, *e.g.* Cubical Agda, `redtt`, `cooltt` [VMA19; Red18; Red20].

# Syntactic decidability and injectivity

Canonicity was a surprisingly difficult "dry run". Real goal of **□TT** was to show:

1. **Decidability of equality.** *It is effectively decidable whether two morphisms* $\Delta \longrightarrow \Gamma : \mathcal{C}_{\Box}$ *are (strictly) equal or unequal.*
2. **Injectivity of type constructors.** *If* $\Pi(A, B) \equiv \Pi(A', B')$*, then* $(A, B) \equiv (A', B')$*; and the same for dependent sums, etc.*

Decidability and injectivity are needed for implementing proof assistants (like Coq, Agda, Lean, *etc.*). Both are corollaries of **normalization**, which is much more complex to state.

# Normalization for □TT

**Idea of normalization: □TT** has a “standard” presentation by operations and equations. If we could present the theory *with only* operations and no equations, then both decidability and injectivity would be trivial. A *normalization argument* is:

1. **Normal forms:** define a family of sets $\nu : \mathbf{Nf}(C, A) \longrightarrow \mathrm{Hom}_{\mathcal{C}_{\square}}(C, A)$ such that $\mathbf{Nf}(C, A)$ satisfies decidability and injectivity by inspection.
2. **Normalization:** prove that each $\nu$ is an isomorphism.

Both steps require creativity.

# What are normal forms?

Insights due to Gentzen [Gen35] and Tait [Tai67][2] teach us that normal forms are in fact divided into two classes, **Ne** ("neutral") and **Nf** ("normal").

1. $\mathbf{Ne}(C, A)$ represents "eliminations": projections, counits, *etc.*
2. $\mathbf{Nf}(C, A)$ represents "introductions": universal maps, units; includes $\mathbf{Ne}(C, A)$ when $A$ lacks universal property.

**The reason it works:** represents morphisms by "maximally introduced" ($\eta$-long) and "minimally eliminated" ($\beta$-short) terms.

[2] and many others

# Normalization by gluing

Normalization results are proved by *Artin gluing* [AGV72; Wra74]; application of gluing to logic and type theory has a long history.[3]

**Idea:** embed the syntactic topos $\widehat{\mathcal{C}_{\square}}$ via *open immersion* into a topos $\mathcal{G}$ in which normal forms can be internalized. Then define a model of $\mathcal{C}_{\square}$ in $\mathcal{G}$ that restricts to the generic model under the open immersion.

$$\begin{tikzcd}
\widehat{\mathcal{C}_{\square}} \arrow[d, hook, "j", "\text{open immersion}"'] \arrow[dr, equal] & \\
\mathcal{G} \arrow[r, "g", "\text{computability model}"'] & \widehat{\mathcal{C}_{\square}}
\end{tikzcd}$$

---

[3]To name a few: Freyd [Fre78], Lambek and Scott [LS86], Crole [Cro93], Lafont [Laf88], Altenkirch, Hofmann, and Streicher [AHS95], Fiore [Fio02], and Coquand [Coq19].

# Synthetic Tait computability

Normalization is **not** an immediate consequence of the “obvious” computability model; normalization of type theory and $\lambda$-calculus is not abstract nonsense.

Artin gluing provides the correct setting in which to state normalization, but the normalization proof is a further construction.

Most easily phrased in the internal language of $\mathcal{G}$ using open and closed modalities [RSS20], called “synthetic Tait computability” [SH21; SA21; Ste21] by analogy with SDG, SDT, SAG, *etc*.

# Tait's Yoga for **ITT**

We must construct our model carefully, following *Tait's yoga*:

1. For each type $A$, normal/neutral forms internalized as objects $\mathbf{Nf}_A, \mathbf{Ne}_A : \mathcal{G}$ that restrict along $j$ to $\mathrm{y}(A)$.
2. Need morphisms $\mathbf{Ne}_A \to g^*\mathrm{y}(A) \to \mathrm{Nf}_\mathrm{A}$ ("reflection/reification") that are *vertical* in the gluing fibration $j^* : \mathcal{G} \longrightarrow \widehat{\mathcal{C}_\square}$.[4]

Resulting normalization map is automatically injective; prove surjective by induction on $\mathbf{Nf}_A$. $\square$

The above *almost* works for **□TT**.

[4]Hint: we only really want $g^*\mathrm{y}(A) \to \mathbf{Nf}_A$, since it assigns normal forms to terms in the model, but to close under exponentials we also need the map out of neutrals.

# Instability of cubical neutral forms

Neutral forms for **□TT** have new features not found in ITT, disrupting the classic gluing argument. Consider the *path* type:

$$\mathsf{path}_A(a, b) \cong \prod_{i:\mathbb{I}} \{x : A \mid (i = 0 \to x = a) \wedge (i = 1 \to x = b)\}$$

The evaluation map $\epsilon : \mathsf{path}_A(a, b) \times \mathbb{I} \longrightarrow A$ must be representable by a neutral form, but its restrictions to $0, 1 : \mathbb{I}$ may *not* be representable by a neutral form since $a, b$ need not be representable by neutrals!

# Instability of cubical neutral forms

Neutral forms for **□TT** have new features not found in ITT, disrupting the classic gluing argument. Consider the *path* type:

$$\mathrm{path}_A(a, b) \cong \prod_{i:\mathbb{I}} \{x : A \mid (i = 0 \to x = a) \land (i = 1 \to x = b)\}$$

The evaluation map $\epsilon : \mathrm{path}_A(a, b) \times \mathbb{I} \longrightarrow A$ must be representable by a neutral form, but its restrictions to $0, 1 : \mathbb{I}$ may *not* be representable by a neutral form since $a, b$ need not be representable by neutrals!

**Solution:** neutral forms $e$ are indexed in a *frontier* of instability $\partial(e)$ valued in $\Omega_{\mathcal{G}}$.

$$\partial(\epsilon(p, i)) :\equiv (i = 0 \lor i = 1)$$

Write $\mathbf{Ne}[\phi]_A$ for the "neutrals away from $\phi$", *i.e.* $\{e : \mathbf{Ne}_A \mid \partial(e) = \phi\}$. We will ensure that $\mathbf{Ne}[\top]_A \cong j_*\mathrm{y}(A)$.

# Stabilization of neutrals

$$\begin{tikzcd}
 & \mathbf{Ne}[\phi]_A \arrow[d] \\
 & j_* \mathrm{y}(A) \arrow[d] \\
\phi_* \phi^* g^* \mathrm{y}(A) \arrow[r] & \phi_* \phi^* j_* \mathrm{y}(A)
\end{tikzcd}$$

# Stabilization of neutrals

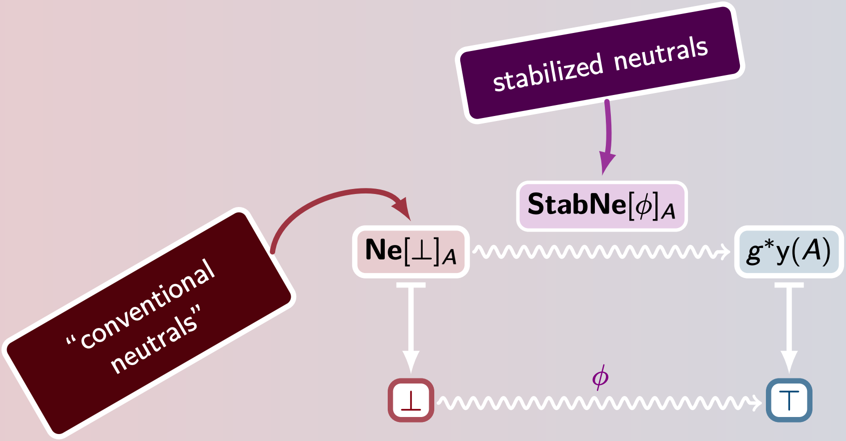
$$
\begin{tikzcd}
\mathbf{StabNe}[\phi]_A \arrow[r, dashed] \arrow[dd, dashed] \arrow[ddr, phantom, very near start, "\lrcorner"] & \mathbf{Ne}[\phi]_A \arrow[d] \\
 & j_*\mathrm{y}(A) \arrow[d] \\
\phi_*\phi^*g^*\mathrm{y}(A) \arrow[r] & \phi_*\phi^*j_*\mathrm{y}(A)
\end{tikzcd}
$$

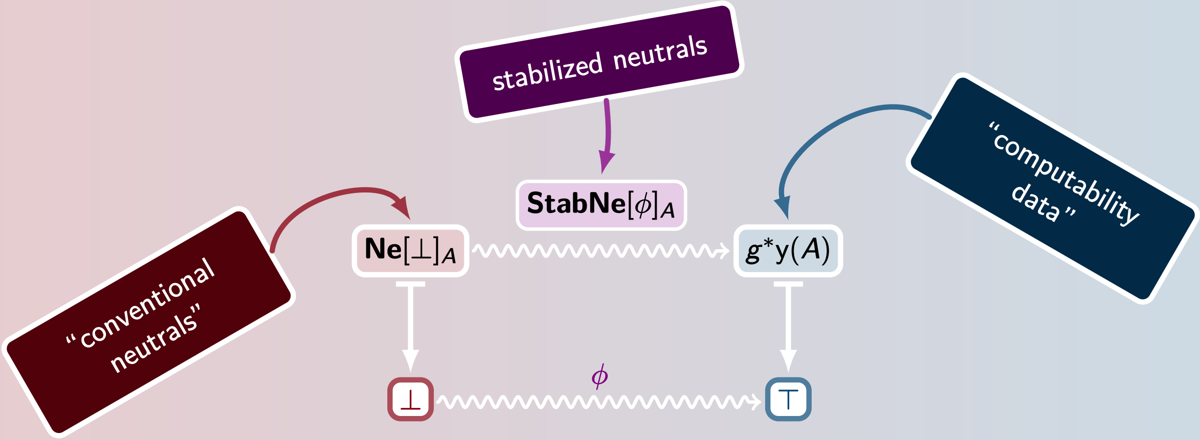
*Unstable neutrals are* ***glued together*** *with compatible computability data along their frontiers of instability.*

$$
\begin{tikzcd}
\mathbf{Ne}[\perp]_A \arrow[rr, squiggly, "\mathbf{StabNe}[\phi]_A"] \arrow[d, mapsto] & & g^*\mathrm{y}(A) \arrow[d, mapsto] \\
\perp \arrow[rr, squiggly, "\phi"] & & \top
\end{tikzcd}
$$

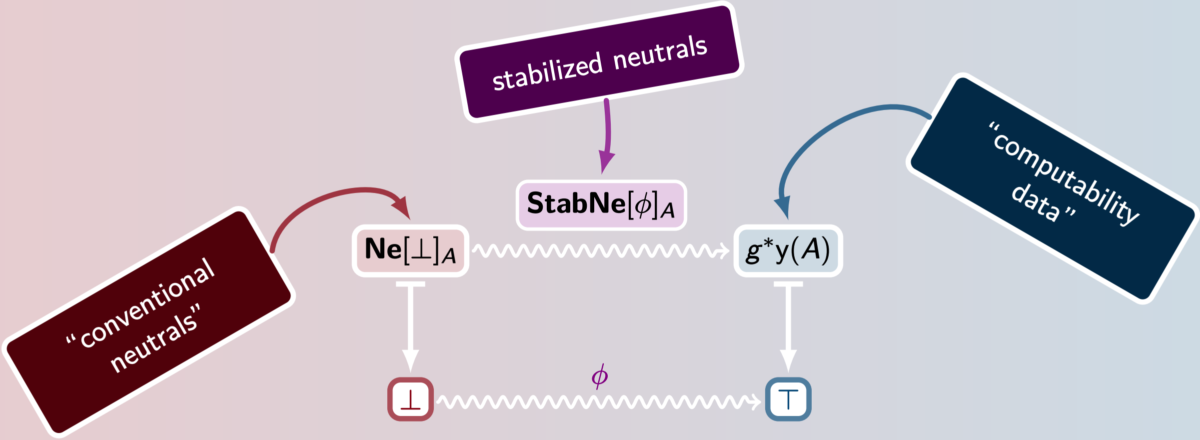
stabilized neutrals

$$\begin{tikzcd} \mathbf{Ne}[\perp]_A \arrow[rr, squiggly, "\mathbf{StabNe}[\phi]_A"] \arrow[d, mapsto] & & g^*\mathsf{y}(A) \arrow[d, mapsto] \\ \perp \arrow[rr, squiggly, "\phi"] & & \top \end{tikzcd}$$


stabilized neutrals
$\mathsf{StabNe}[\phi]_A$
"conventional neutrals"
$\mathsf{Ne}[\perp]_A$
$g^*\mathsf{y}(A)$
$\phi$
$\perp$
$\top$



stabilized neutrals
“computability data”
“conventional neutrals”
$\mathbf{StabNe}[\phi]_A$
$\mathbf{Ne}[\bot]_A$
$g^*\mathsf{y}(A)$
$\bot$
$\phi$
$\top$




**Stabilization interpolates between neutrals and computability data.**

# The stabilized Tait yoga

$$
\begin{tikzcd}
 & \phi_*\phi^* g^* \mathsf{y}(A) & \\
\mathbf{StabNe}[\phi]_A \arrow[ur] \arrow[r, "\uparrow^{\phi}_{A}"] \arrow[d] \arrow[dr] & g^* \mathsf{y}(A) \arrow[u] \arrow[r, "\downarrow_A"] \arrow[d] & \mathbf{Nf}_A \arrow[dl] \\
\mathbf{Ne}[\phi]_A \arrow[r] & j_* \mathsf{y}(A) &
\end{tikzcd}
$$

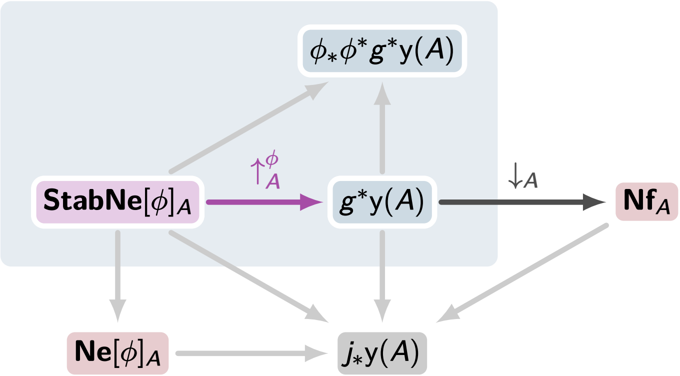
# The stabilized Tait yoga

$$
\begin{tikzcd}
 & \phi_*\phi^* g^* \mathrm{y}(A) & \\
\mathbf{StabNe}[\phi]_A \arrow[ur] \arrow[r, "\uparrow^{\phi}_{A}"] \arrow[d] \arrow[dr] & g^*\mathrm{y}(A) \arrow[u] \arrow[r, "\downarrow_A"] \arrow[d] & \mathbf{Nf}_A \arrow[dl] \\
\mathbf{Ne}[\phi]_A \arrow[r] & j_*\mathrm{y}(A) &
\end{tikzcd}
$$

# The stabilized Tait yoga

$$
\begin{tikzcd}
 & \phi_*\phi^* g^* \mathrm{y}(A) & \\
\mathbf{StabNe}[\phi]_A \arrow[ur] \arrow[r, "\uparrow^{\phi}_{A}"] \arrow[d] \arrow[dr] & g^* \mathrm{y}(A) \arrow[u] \arrow[r, "\downarrow_A"] \arrow[d] & \mathbf{Nf}_A \arrow[dl] \\
\mathbf{Ne}[\phi]_A \arrow[r] & j_* \mathrm{y}(A) &
\end{tikzcd}
$$

# The stabilized Tait yoga




**Theorem.** Every type is closed under the **stabilized** Tait yoga.

# Summary of results

For univalent $\square$TT with a countable cumulative hierarchy of univalent universes, we have proved the following results:

1. Every type and every term has a *unique* normal form.
2. Judgmental equality of types and terms is decidable.
3. Type constructors (*e.g.* $\Pi$) are injective.
4. Type checking is decidable (corollary of 1–3).

To learn more, see our LICS'21 paper and S.'s forthcoming dissertation (soon).


[SA21] Jonathan Sterling and Carlo Angiuli. "Normalization for Cubical Type Theory". In: *Proceedings of the 36th Annual ACM/IEEE Symposium on Logic in Computer Science*. To appear. New York, NY, USA: ACM, 2021. arXiv: 2101.11479 [cs.LO].

[Ste21] Jonathan Sterling. "First Steps in Synthetic Tait Computability". Forthcoming. PhD thesis. Carnegie Mellon University, 2021.

# References I


[AFH18] Carlo Angiuli, Kuen-Bang Hou (Favonia), and Robert Harper. “Cartesian Cubical Computational Type Theory: Constructive Reasoning with Paths and Equalities”. In: *27th EACSL Annual Conference on Computer Science Logic (CSL 2018)*. Ed. by Dan Ghica and Achim Jung. Vol. 119. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 6:1–6:17. ISBN: 978-3-95977-088-0. DOI: 10.4230/LIPIcs.CSL.2018.6. URL: http://drops.dagstuhl.de/opus/volltexte/2018/9673.

[AGV72] Michael Artin, Alexander Grothendieck, and Jean-Louis Verdier. *Théorie des topos et cohomologie étale des schémas*. Séminaire de Géométrie Algébrique du Bois-Marie 1963–1964 (SGA 4), Dirigé par M. Artin, A. Grothendieck, et J.-L. Verdier. Avec la collaboration de N. Bourbaki, P. Deligne et B. Saint-Donat, Lecture Notes in Mathematics, Vol. 269, 270, 305. Berlin: Springer-Verlag, 1972.

[AHS95] Thorsten Altenkirch, Martin Hofmann, and Thomas Streicher. “Categorical reconstruction of a reduction free normalization proof”. In: *Category Theory and Computer Science*. Ed. by David Pitt, David E. Rydeheard, and Peter Johnstone. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 182–199. ISBN: 978-3-540-44661-3.

[AK16] Thorsten Altenkirch and Ambrus Kaposi. “Normalisation by Evaluation for Dependent Types”. In: *1st International Conference on Formal Structures for Computation and Deduction (FSCD 2016)*. Ed. by Delia Kesner and Brigitte Pientka. Vol. 52. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, 6:1–6:16. ISBN: 978-3-95977-010-1. DOI: 10.4230/LIPIcs.FSCD.2016.6. URL: http://drops.dagstuhl.de/opus/volltexte/2016/5972.

# References II


[Ang+21] Carlo Angiuli et al. "Internalizing Representation Independence with Univalence". In: *Proceedings of the ACM on Programming Languages* 5.POPL (Jan. 2021). DOI: 10.1145/3434293.

[Coh+17] Cyril Cohen et al. "Cubical Type Theory: a constructive interpretation of the univalence axiom". In: *IfCoLog Journal of Logics and their Applications* 4.10 (Nov. 2017), pp. 3127–3169. URL: http://www.collegepublications.co.uk/journals/ifcolog/?00019.

[Coq19] Thierry Coquand. "Canonicity and normalization for dependent type theory". In: *Theoretical Computer Science* 777 (2019). In memory of Maurice Nivat, a founding father of Theoretical Computer Science - Part I, pp. 184–191. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2019.01.015. arXiv: 1810.09367 [cs.PL].

[Cro93] R. L. Crole. *Categories for Types*. Cambridge Mathematical Textbooks. New York: Cambridge University Press, 1993. ISBN: 978-0-521-45701-9.

[Fio02] Marcelo Fiore. "Semantic Analysis of Normalisation by Evaluation for Typed Lambda Calculus". In: *Proceedings of the 4th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming*. PPDP '02. Pittsburgh, PA, USA: ACM, 2002, pp. 26–37. ISBN: 1-58113-528-9. DOI: 10.1145/571157.571161.

[Fre78] Peter Freyd. "On proving that **1** is an indecomposable projective in various free categories". Unpublished manuscript. 1978.

[Gen35] Gerhard Gentzen. "Untersuchungen über das logische Schließen I". In: *Mathematische Zeitschrift* 39 (1935), pp. 176–210. URL: http://eudml.org/doc/168546.

# References III


[Hub18] Simon Huber. “Canonicity for Cubical Type Theory”. In: *Journal of Automated Reasoning* (June 13, 2018). ISSN: 1573-0670. DOI: 10.1007/s10817-018-9469-1.

[Laf88] Yves Lafont. “Logiques, catégories & machines : implantation de langages de programmation guidée par la logique catégorique”. PhD thesis. Université Paris 7, 1988.

[LS86] J. Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic*. New York, NY, USA: Cambridge University Press, 1986. ISBN: 0-521-24665-2.

[Red18] The RedPRL Development Team. redtt. 2018. URL: https://www.github.com/RedPRL/redtt.

[Red20] The RedPRL Development Team. cooltt. 2020. URL: https://www.github.com/RedPRL/cooltt.

[RSS20] Egbert Rijke, Michael Shulman, and Bas Spitters. “Modalities in homotopy type theory”. In: *Logical Methods in Computer Science* Volume 16, Issue 1 (Jan. 2020). DOI: 10.23638/LMCS-16(1:2)2020. arXiv: 1706.07526 [math.CT]. URL: https://lmcs.episciences.org/6015.

[SA21] Jonathan Sterling and Carlo Angiuli. “Normalization for Cubical Type Theory”. In: *Proceedings of the 36th Annual ACM/IEEE Symposium on Logic in Computer Science*. To appear. New York, NY, USA: ACM, 2021. arXiv: 2101.11479 [cs.LO].

[SH21] Jonathan Sterling and Robert Harper. “Logical Relations As Types: Proof-Relevant Parametricity for Program Modules”. In: *Journal of the ACM* (2021). To appear. arXiv: 2010.08599 [cs.PL].

[Ste21] Jonathan Sterling. “First Steps in Synthetic Tait Computability”. Forthcoming. PhD thesis. Carnegie Mellon University, 2021.

# References IV


[Str98] Thomas Streicher. “Categorical intuitions underlying semantic normalisation proofs”. In: *Preliminary Proceedings of the APPSEM Workshop on Normalisation by Evaluation*. Ed. by O. Danvy and P. Dybjer. Department of Computer Science, Aarhus University, 1998.

[Tai67] W. W. Tait. “Intensional Interpretations of Functionals of Finite Type I”. In: *The Journal of Symbolic Logic* 32.2 (1967), pp. 198–212. ISSN: 00224812. URL: http://www.jstor.org/stable/2271658.

[VMA19] Andrea Vezzosi, Anders Mörtberg, and Andreas Abel. “Cubical Agda: A Dependently Typed Programming Language with Univalence and Higher Inductive Types”. In: *Proceedings of the 24th ACM SIGPLAN International Conference on Functional Programming*. ICFP ’19. Boston, Massachusetts, USA: ACM, 2019. DOI: 10.1145/3341691.

[Wra74] Gavin Wraith. “Artin glueing”. In: *Journal of Pure and Applied Algebra* 4.3 (1974), pp. 345–348. ISSN: 0022-4049. DOI: 10.1016/0022-4049(74)90014-0.