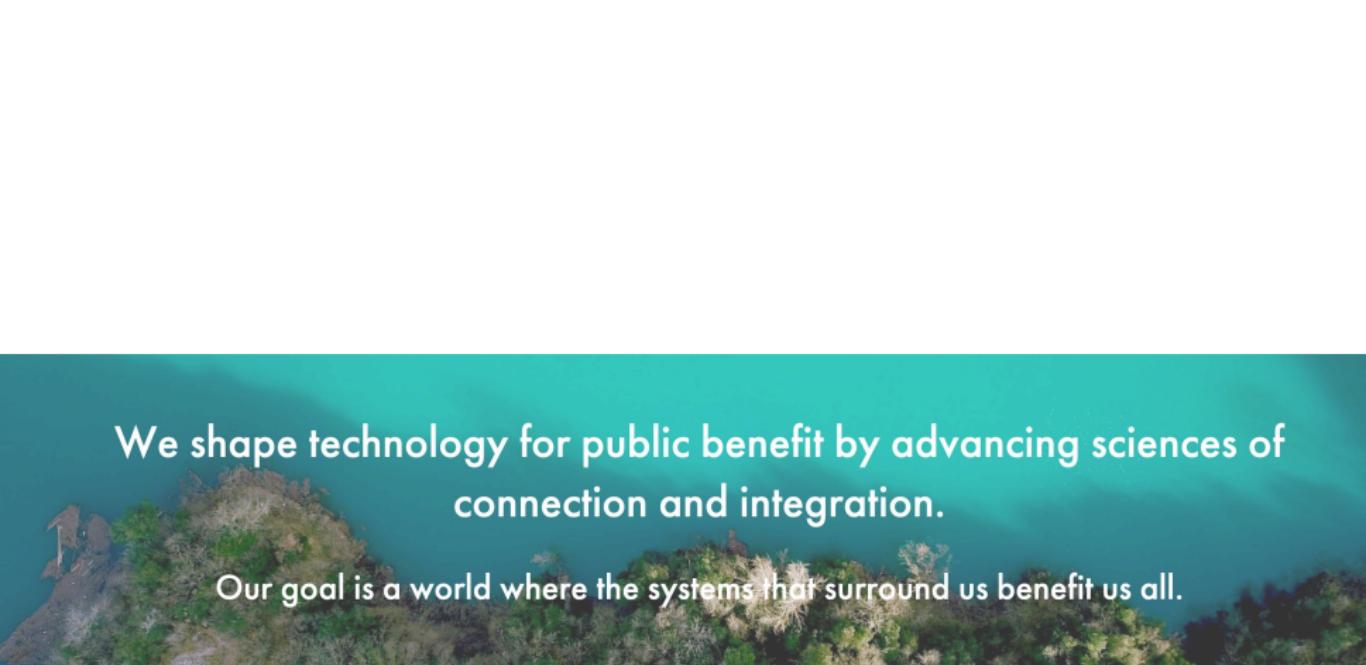


WARTIME PROOFS

& futuristic programs

Valeria de Paiva
UC Berkeley Logic Colloquium

October 2023



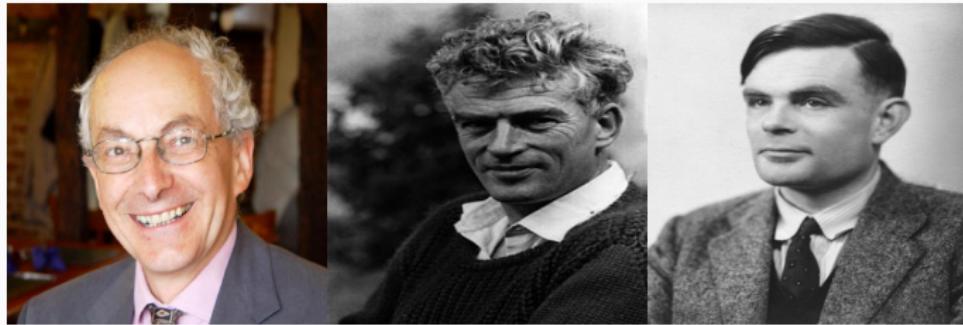
We shape technology for public benefit by advancing sciences of connection and integration.

Our goal is a world where the systems that surround us benefit us all.

Topos
Institute²



Journey



Background

■ Changing models is hard!

Revolution in the sky

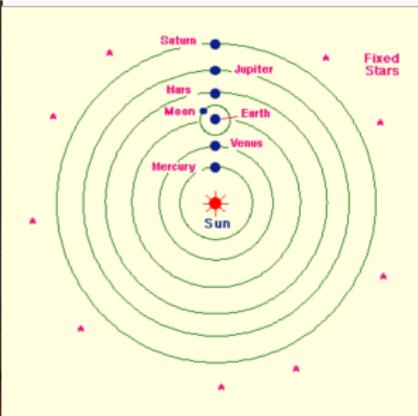
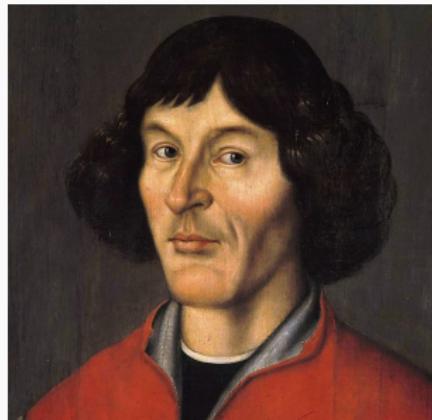
Revolution in Math

Revolution in Computation

[Dialectica Construction]



Copernicus *Celestial Spheres* 1543



- 1615 Galileo investigated
- 1620 Copernicus book 'corrected'
- 1632/3 Galileo book, under house arrest
- 1835 both books out of Index

Algebra & Proofs

A Revolution in Mathematics?

What Really Happened a Century Ago and Why It Matters Today

Frank Quinn (*Notices of the AMS*, Jan 2012)



*The bulk of mathematics today got crystallized in the last years of the 19th century, first years of the 20th century.
The shock is still being felt.*

Algebra & Proofs

[...] a fundamental shift occurred in mathematics from about 1880 to 1940—the consideration of a wide variety of mathematical “structures”—groups, fields, lattices, etc.—satisfying some axioms. This approach is so common now that it is almost superfluous to mention it explicitly, but it represented a major conceptual shift in answering the question: What is mathematics?

The axiomatization of Linear Algebra, Moore, Historia Mathematica, 1995.

Math NOT about numbers!
It is about structures, connections and proofs!

Algebra & Proofs



Bourbaki in 1938

The axiomatization of algebra was begun by Dedekind and Hilbert. It was then completed in the years following 1920 by Artin, Noether and their colleagues at Göttingen.

Bourbaki, Elements of the History of Mathematics, 1960.

Algebra: not solving 8th grade equations

Revolution in Computation

Curry-Howard Correspondence

1908 – 1932 – 1969 – now?

+ Curry-Howard Correspondence



1963



Lambda-calculus



1965



Cartesian
Closed
Categories

Intuitionistic
Propositional
Logic



What is this?



- a fundamental result connecting Logics, Programming Languages and Categories
- Each one of the arrows connects two different fields
- Original Curry-Howard ties logic and type theory. Category Theory is a late addition
- More in Wadler's *Proofs as Propositions*

How did it come about?



- Mathematics in turmoil because of paradoxes in set theory.
- **Hilbert's program** to provide secure foundations for all mathematics
- How? Formalization!
- Base math on *finitistic methods*
- goal: Prove *consistency of Arithmetic*

Hilbert's Wish List

- **Consistent:** no contradiction can be obtained in the formalism
- **Complete:** all true math statements can be proven
- **Conservative:** any result about “real objects” obtained using “ideal objects” (such as uncountable sets) can be proved without ideal objects.
- **Decidable:** an algorithm for deciding the truth or falsity of any math statement.

Failure of Hilbert's Program

Gödel's Incompleteness Theorems (1931)



Hilbert's program impossible, if interpreted in the most direct way.

THEN

- use more powerful methods, Gentzen
- Proof Theory, to know what can be proved with what

■ War Time Proofs



To prove the consistency of Arithmetic G. Gentzen (Hilbert's assistant) invented his systems of

NATURAL DEDUCTION

SEQUENT CALCULUS (1934)

These are the main proof systems used nowadays by provers (humans and machines)

■ War Time Proofs



- Gödel (1933, 1942, 1958)
- Liberalized version of Hilbert's program – justify classical systems in terms of notions as intuitively clear as possible.
- Computable (or primitive recursive) functionals of finite type (System T), using the Dialectica Interpretation.

Programs?



- Alonzo Church: lambda calculus (1932) a term for each machine computable function
- Haskell Curry: combinators and Combinatory Logic (1930) (also Schoenfinkel 1908)
- Church Thesis: λ -definability, recursive functions, Turing machines, all equivalent

Curry-Howard for Implication

Natural deduction rules for implication
(without λ -terms)

$$\frac{\begin{array}{c} [A] \\ \vdots \\ A \rightarrow B \quad A \\ \hline B \end{array}}{\begin{array}{c} \vdots \pi \\ \vdots \\ B \\ \hline A \rightarrow B \end{array}}$$

Curry-Howard for Implication

Natural deduction rules for implication (with λ -terms)

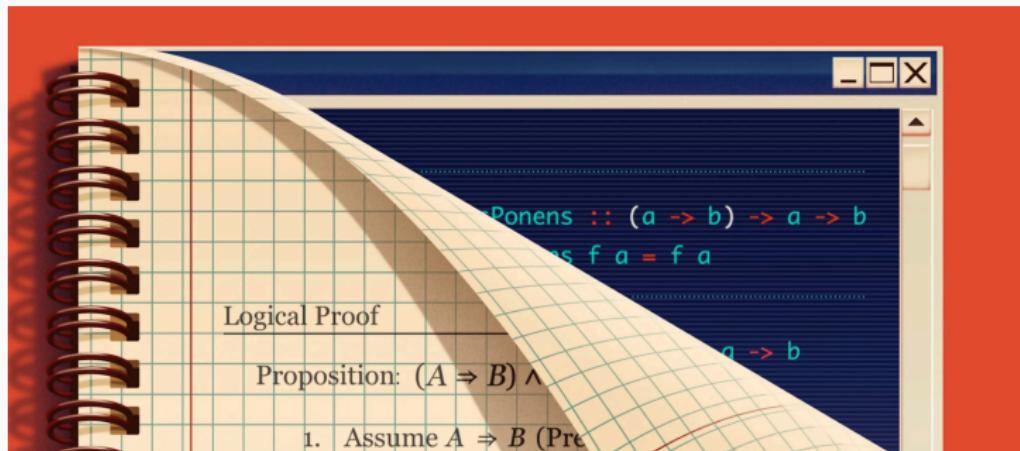
$$\frac{\begin{array}{c} [x:A] \\ \vdots \\ M:A \rightarrow B \quad N:A \\ \hline M(N):B \end{array}}{\lambda x.M:A \rightarrow B}$$

function application

$$\frac{\begin{array}{c} M:B \\ \vdots \\ \vdots \pi \\ \vdots \\ M:B \end{array}}{\lambda x.M:A \rightarrow B}$$

abstraction

Proofs as programs



Proofs as programs

- Lambda calculus as **universal programming language**
- Effects, parallel programming, distributed computing, others are active research
- How much can we extend it?
- A cornucopia of new logics/program constructs based on the correspondence between proofs and programs.

Category Theory



- Types: formulae/objects in a category
- Terms/programs: proofs/morphisms in **appropriate** category
- Logical constructors: categorical constructions
- Most important: Reduction is proof normalization (Tait)

Logics

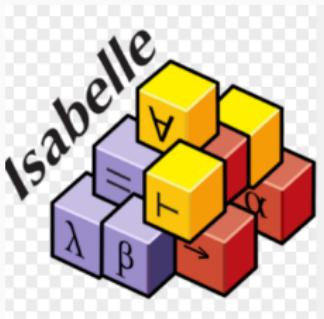
- Intuitionistic Logic
- System F
- Dependent type theory
- Linear Logics
- Modal logics
- Classical logic, etc

Why Categories?

- Model derivations/proofs, not whether theorems are true or not
- Why is it good? Modeling derivations useful in linguistics, functional programming, compilers..
- Why is it important? Solve the problem where it's easier and trasnport solution
- Also CS as new important problems to solve with our favorite tools.

Why so little impact on maths or logic?

Many Curry-Howard Correspondences



many more!

Curry-Howard Correspondence



Linear
Lambda-
Calculus

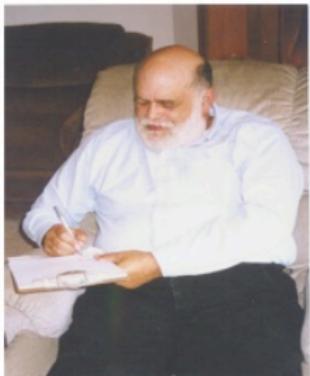


Linear
Categories

Linear
Logic



Modal (S4) Curry-Howard Correspondence



Modal S4
Lambda-
Calculus



CCC+monoidal
comonad

Constructive S4
Modal Logic



Lambek calculus Curry-Howard Correspondence



2-sided Linear
Lambda-
Calculus



Monoidal bi-
closed
Categories

Lambek
Calculus

Gödel Dialectica

- **Goal** Prove HA consistent. How?
- **Idea:** Translate every formula A of HA to

$$A^D = \exists u \forall x A_D$$

where A_D quantifier-free.

- Translation defined by clauses on the connectives.
- ‘Easy’ to prove the theorem desired, but hard to see **why** it works.

Gödel Dialectica

Theorem (Gödel 1958): if HA proves A , then System T proves quantifier-free $A_D(t, x)$, where x are functionals of finite type, and t a suitable sequence of terms (not containing x).

Proof by induction on length of derivations,
Troelstra 1973.

How intuitive are the functionals of finite type?

An internal **categorical model of Gödel's Dialectica interpretation** in my phd thesis.

Categorical Dialectica

Given C with finite limits, build a new category $\mathfrak{Dial}(C)$, with objects $A = (U, X, \alpha)$ where α is a subobject of $U \times X$ in C ; this object represents the formula

$$\exists u \forall x \alpha(u, x).$$

A map from $\exists u \forall x \alpha(u, x)$ to $\exists v \forall y \beta(v, y)$ can be thought of as a pair $(f : U \rightarrow V, F : U \times Y \rightarrow X)$ of terms/maps, subject to the entailment condition

$$\alpha(u, F(u, y)) \vdash \beta(f(u), y).$$

Surprise! A model of Linear Logic, instead of Constructive Logic

Dialectica categories

- Justifies Linear Logic in terms of Gödel's proof-theoretic tool. and conversely.
- Keep the differences that Girard wanted to make.
- Justifies Harper's Trinitarism, connections to programming and using CT as syntax guidance.
- Loads of applications, lenses, games, automata, etc.

Dialectica categories timeline

- 1940 Gödel lecture at Yale
- 1958 published in Dialectica
- 1988 first categorical interpretation
- 2008 fibrational generalization (Biering)
- 2011 modern version (Hofstra)
- 2018 dependent type theory (von Glehm, Moss)

Recent work with Trotta and Spadetto where the assumptions in Gödel's argument (hacks?) are used (2022, 2023)

Applications

- Concurrency theory, Petri nets and others (1990's)
- linear functional programming (2000's)
- partial compilers (Budiu and Plotkin, 2013)
- Lenses, BX-transformations, *Lenses for Philosophers*, Hedges 2017
- Automated Differentiation, Pedrot and Kerjean 2022?

Conclusions

- Changing models is hard!
- Underappreciated (categorical) Curry-Howard correspondence
- Important for interdisciplinary work: Math, Logic and Programming
- One example: Dialectica categories, Gödel fibrations and doctrines, rediscovered over and over
- Plenty of other applications to develop

Thanks!

Some References

-  K. Gödel, *Über eine bisher noch nicht benützte erweiterung des finiten standpunktes*, In *Dialectica*, 12(3-4):280–287.
(Translation in Gödel's Collected Works)
-  J.-Y. Girard, *Linear Logic*, TCS (1988).
-  P. Wadler, *Types as Propositions*, Communications of the ACM, pp 75–84, (2015).
-  V. de Paiva, *The Dialectica Categories*, In Proc of Categories in Computer Science and Logic, Boulder, CO, 1987.
-  D. Trotta, M. Spadetto, V. de Paiva, *The Gödel Fibration*, MFCS, 2021.
-  D. Trotta, M. Spadetto, V. de Paiva, *Dialectica Logical Principles: not only rules*, JLC 2022
-  D. Trotta, M. Spadetto, V. de Paiva, *Dialectica Principles via Gödel Doctrines*, TCS, 2023.

Extra slides 1

- why fibrations and doctrines?
- First-order is of course more expressive than propositional logic,
- Much tighter correspondence between the logic and the category theory
- Using Skolem and Gödel doctrines/fibrations
- See the 'Dialectica logical principles' paper JLC 2022

Extra slides 2 - (Gist)

- Given a doctrine P , when is there a doctrine P' such that $\mathfrak{Dial}(P') \cong P$?
- When such doctrine P' exists, how do we find it?

Extra slides 2 - (Gist)

- Given a doctrine P , when is there a doctrine P' such that $\text{Dial}(P') \cong P$? Such a P' exists precisely when P is a **Gödel doctrine**
- When such doctrine P' exists, how do we find it?

Extra slides 2 - (Gist)

- Given a doctrine P , when is there a doctrine P' such that $\text{Dial}(P') \cong P$? Such a P' exists precisely when P is a **Gödel doctrine**
- When such doctrine P' exists, how do we find it? P' is given by the **quantifier-free elements** of the Gödel doctrine P

Extra slides 3

- Quantifiers since Lawvere's work (1968) but how to say quantifier-free predicate in categorical logic?
- existential (and universal-free) objects in doctrines
- Via existential splitting predicates, existential-free predicates, doctrines P with enough existential-free predicates and Gödel doctrines. See Trotta, Spadetto and de Paiva (2023), Dialectica principles via Gödel doctrines, TCS, 2023.