Categorical Logic in Computer Science

Valeria de Paiva

Cuil, Inc.
Menlo Park, CA

Sonoma, March 4th, 2009

Outline

- What is categorical logic?
- How did it come about?
- Applications to computer science
- My example: Dialectica categories
- Conclusions & Prospects

caveats: very personal viewpoint, pub chat?

Categorical Logic:
mathematical ideas in new settings

- Which kind of math?
- Algebra to Theory of Categories
- Categories to Categorical Logic
- Categorical Logic to Computer Science
- Which kind of computer science?

Algebra to Category Theory

- (Abstract) Algebra: area of mathematics that studies algebraic structures, such as **groups**, **rings**, **fields**, **modules**, **vector spaces**, and **algebras**.
- Abstract (instead of elementary) algebra: turn of the 20th century, van der Waerden textbook 'Modern Algebra', 1930.
- Sources: linear algebra, permutation groups, diophantine equations,...

check wikipedia

Category Theory

- Eilenberg & MacLane (1945): "General Theory of Natural Equivalences."
- more than a convenient language? diagram chasing methodology/technology
- Milestones: Eilenberg & Steenrod (1952), algebraic topology; Cartan & Eilenberg (1956), homological algebra; Grothendieck's (1957) "Sur quelques points d'algebre homologique' and more; Kan (1958) adjoint functors for homotopy theory.
- Freyd, Tierney, Lambek, mostly Lawvere (1960's) pervasiveness of adjoint functors, **categorical logic**

Category Theory

Basic idea: there's an underlying unity of mathematical concepts/theories. More important than the mathematical concepts themselves is how they relate to each other.

Topological spaces come with continuous maps, while vector spaces come with linear transformations, for example.

Morphisms, ie how structures transform into others is the way to organize the mathematical edifice.

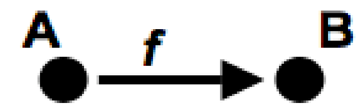
Detractors call CT abstract Nonsense

The language of CT is well-accepted in all branches of Math, the praxis and the philosophy less so.

Categories: a picture

A category consists of:

- a class of *objects*
- a class of *morphisms* ("arrows")



- for each morphism, *f*, one object as the *domain* of *f* and one object as the *codomain* of *f*.
- for each object, *A*, an *identity morphism* which has domain *A* and codomain *A*. ("ID_A")

- for each pair of morphisms f:A→B and g:B→C, (i.e. cod(f)=dom(g)), a *composite morphism*, $g \circ f$: A→C

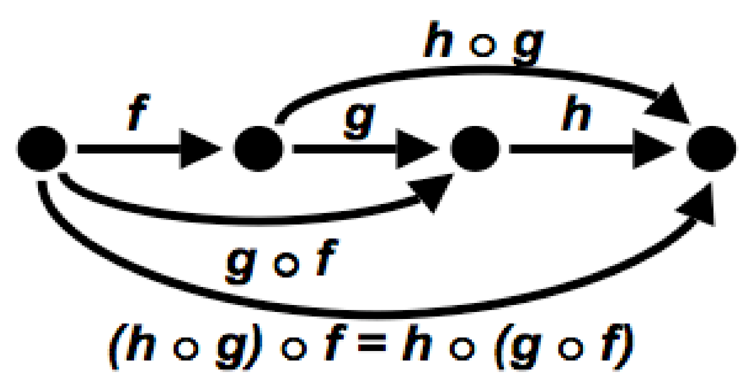
With these rules:

- *Identity composition:* For each morphism *f*:A→B,

  $f \circ ID_A = f$ and $ID_B \circ f = f$

- *Associativity:* For each set of morphisms f:A→B, g:B→C, h:C→D,

  $(h \circ g) \circ f = h \circ (g \circ f)$



Category Theory

- Category: a collection of objects and of morphisms, satisfying obvious laws
- Functors: the natural notion of morphism between categories
- Natural transformations: the natural notion of morphisms between functors
- Constructors: products, sums, limits, duals....
- Adjunctions: an abstract version of equality
- How does this relates to logic?

Categorical Logic?

- (Mathematical) Logic usually divided in: Model theory, Proof theory, Recursion theory and Set theory
- Category theory used for Model theory and Proof theory, both are **categorical semantics**
- This talk: categorical logic means categorical **proof** theory
- Proof theory is Hilbert's brainchild: 'proofs' as objects of mathematical study
- Categorical logic Lawvere's seminal idea:

logic=algebra=geometry

Categorical Proof Theory

- A change in paradigm: model theory about whether theorems are true or not. (**categorical** model theory considers truthfulness in different categories)
- Proof theory about whether different proofs can be considered 'the same'
- Categorical proof theory models proofs, using categories, so that you can talk about sameness of proofs using mathematical tools
- Based on the Curry-Howard correspondence

Extended Curry-Howard correspondence

- Timeline: 30's Curry analogy, 1969 Howard preprint, 1981 H.B. Haskell volume, Lawvere thesis, Lambek&Scott book 1986
- Basic idea: types are formulae/objects in **appropriate** category, terms are proofs/morphisms in same category,logical constructors are appropriate categorical constructions. Mostly: reduction is proof normalization
- Transfer results/tools from logic to CT to CS

Illustration: Curry-Howard for Implication

Natural deduction rules for implication (without λ-terms):

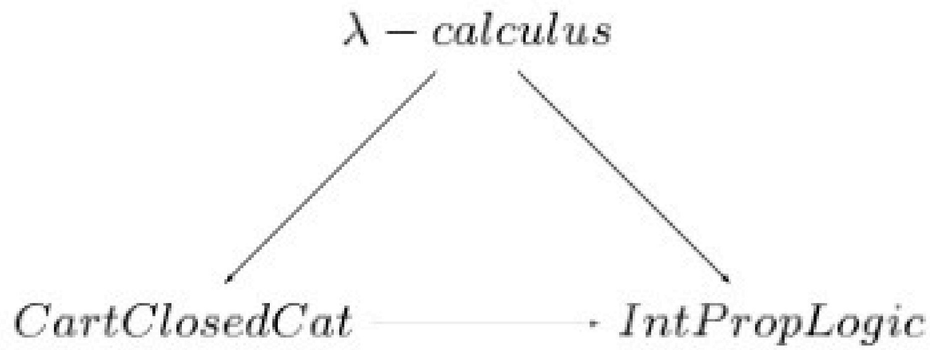
$$\frac{A \to B \qquad A}{B} \to_E \qquad\qquad \frac{\begin{array}{c}[A] \\ \vdots \\ B\end{array}}{A \to B} \to_I$$

Natural deduction rules for implication (with λ-terms):

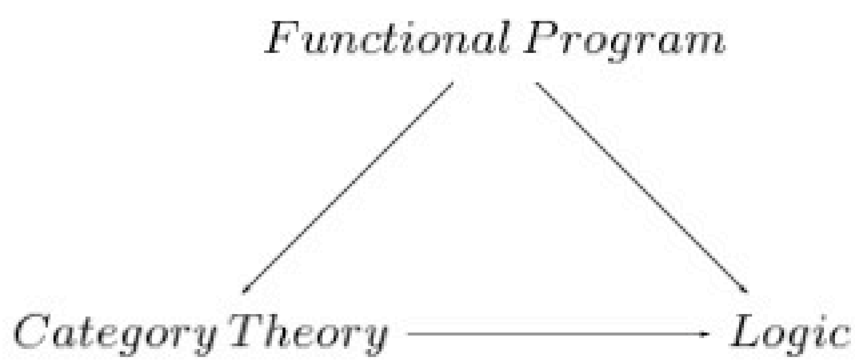
$$\frac{M : A \to B \qquad N : A}{M(N) : B} \to_E \qquad\qquad \frac{\begin{array}{c}[x : A] \\ \vdots \\ M : B\end{array}}{\lambda x.M(x) : A \to B} \to_I$$

Function application   *λ-abstraction*

Categorical Semantics: a picture



Framework connecting



lots of important stuff elided

Categorical Proof Theory

models derivations/proofs, not whether theorems are true or not

Proofs definitely first-class citizens

How?Uses extended Curry-Howard isomorphism

Why is it good? Modeling derivations useful in linguistics, functional programming, compilers..

Why is it important? Widespread use of logic/ algebra in CS means new important problems to solve with our favorite tools, as well as jobs for mathematicians & logicians.

Why so little impact on math itself?

Successes/Opportunities of Categorical Semantics

+Models for the untyped lambda-calculus

+Typed programming language & polymorphism

+Dependent Type Theory

+Operational Semantics & Full abstraction results

+Game Semantics

-Proof theory of Classical Logic/Modal logics

-Effect full computation, mobile/grid computing, etc

-Practical impact

-Acceptance of methodology in math

Outline

- What is categorical logic?
- How did it come about?
- Applications to computer science
- My work: Dialectica categories
- Prospects & Conclusions

caveats: very personal viewpoint, pub chat?

Applications of categories in CS

- semantics/design of programming languages:

  Lisp, ML, O'CamL, **Haskell**, (Lua?), etc.
- Compilers, proving optimizations in code, ...
- categorical semantics of specification (work on Institutions,)

  security, (CMU, Microsoft, etc)

  concurrency…
- Modeling databases (sketches?)
- (at large) functional programming, language design (Type hackery – SPJ)
- Formal methods and high assurance
- interactive theorem proving (HOL, Isabelle, Coq, PVS, Twelf, HOLlight..)

Projects, Forums: a small sample

- CLiCS: Categorical Logic in Computer Science, I and II from Oct 89-Dec95, report: http://www.disi.unige.it/person/MoggiE/PROJECTS/clics/
- APPSEM I and II: Apr 1998-Mar 2002 and Jan03-Dec05
- LINEAR: May 1998-Apr2002
- CMU/Penn Manifest Security http://www.cis.upenn.edu/~plclub/ms/
- Grey project:
- LOGOSPHERE: digital libraries/mathematical knowledge

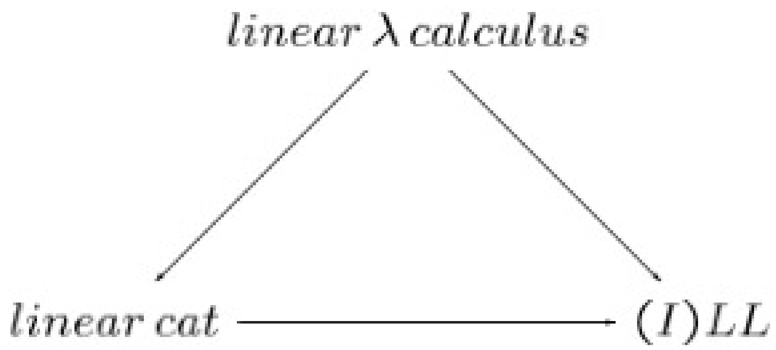
Outline

- What is categorical logic?
- How did it come about?
- Applications to computer science
- My example: Dialectica categories
- Conclusions & Prospects

The same picture...

Categorical Semantics of (Intuitionistic) Linear Logic

Want linear version of Extended Curry-Howard Isomorphism



- Logic intuitionistic

- Already have Intuitionistic Linear Logic (ILL),
  — must find other two sides of triangle...

...different logic

- Linear Logic, a proof theoretic logic described by Jean-Yves Girard in 1986.
- Basic idea: assumptions cannot be discarded or duplicated. They must be used exactly once —just like dollar bills...
- Other approaches to accounting for logical resources. Great win of Linear Logic:
- Account for resources when you want to, otherwise fall back on traditional logic,
- A=>B iff !A –o B

My example: Dialectica categories

Nice version of Curry Howard correspondence for Intuitionistic LINEAR LOGIC

- Why is it nice?
- Godel's Dialectica interpretation (1958)

Gödel's **result:** an interpretation of intuitionistic arithmetic HA in a quantifier-free theory of functionals of finite type T
idea: translate every formula A of HA to $A^D = \exists u \forall x . A_D$ where A_D is quantifier-free.
use: If HA proves A then T proves $A_D(t, y)$ where y is a string of variables for functionals of finite type, t a suitable sequence of terms not containing y.

Dialectica categories

- What? Famous tool in proof theory, the Dialectica interpretation shows that one can translate classical arithmetic formulae into quantifier-free ones — with care.
- Liberalized Hilbert programme
- Would like to understand exactly why the interpretation works..and how it relates to other semi-constructive ones like realizability

Dialectica categories

- Goal: to be as constructive as possible while being able to interpret all of classical arithmetic
- An internal categorical version of the Dialectica, modelling the main connective, implication

If $A^D = \exists u \forall x. A_D$ and $B^D = \exists v \forall y. B_D$

$$(A \Rightarrow B)^D = \exists V(u) \exists X(u,y) \forall u \forall y [A_D(u, X(u,y)) \Rightarrow B_D(V(u), y)]$$

Dialectica categories

Assume C is cartesian closed category

DC objects are relations between U and X
monics $A \overset{\alpha}{\hookrightarrow} U \times X$ written as $(U \overset{\alpha}{\nleftarrow} X)$.

DC maps are pairs of maps of C
$f\colon U \to V$, $F\colon U \times Y \to X$
making a certain pullback condition hold.
Condition in Sets: $u\alpha F(u,y) \Rightarrow f(u)\beta y$.

What is the point?

- First, the construction ends up as a model of Linear Logic, instead of constructive logic.Then it allows us to see where the assumptions in Godel's argument are used
- It justifies linear logic in terms of more traditional logic and conversely explains the more traditional work in terms of a 'modern' (linear, resource conscious) decomposition of concerns.
- Theorems(87/89): Dialectica categories provide models of linear logic as well as an internal representation of the dialectica interpretation. Modeling the exponential ! Is hard, first model to do it.

Dialectica categories: 20 years later

- It was pretty: produces models of Intuitionistic and clasical linear logic and special connectives that allow us to get back to usual logic
- Extended it in a number of directions—a robust proof can be pushed in many ways.
- Got used in CS as a model of Petri nets(2 phds), it has a non-commutative version for Lambek calculus (linguistics), it has been used as a model of state and even of quantum groups.
- Also used for generic models (with Schalk04)

20 years later

- 2008: ressurgence on connections between different functional interpretations (Oliva, Gernest and Trifonov),via Shirahata's work (TAC vol 17).
- Extension to higher-order logic via topos theory, Bodil Biering phd thesis (2008),

  The Copenhagen interpreatation

Conclusions

- Working in interdisciplinary areas is hard, but rewarding
- The frontier between logic, computing and categories is a fun place to be
- Mathematics teaches you a way of thinking, more than specific theorems
- Fall in love with your ideas and enjoy talking to many about them..

Thank you!

Prospects?

- Still want to see more work on Hopf algebras and dialectica categories: quantum mechanics anyone?
- Same about state modeling and separation logic: a couple of ideas not written anywhere
- Also about super power games and relationships between functional interpretations

Some references

Godel's Collected Works, eds Feferman and Dawson

Categories for the Working Mathematician Saunders MacLane

The Curry-Howard Correspondence ed de Groote, Gallier's paper

Proofs and Types – Jean-Yves Girard

http://www.cs.bham.ac.uk/~vdp/publications/papers.html