Categories with Families
Unityped, Simply Typed, Dependentely Typed

Peter Dybjer
Chalmers tekniska högskola

joint work with
Simon Castellan, Imperial College
Pierre Clairambault, ENS Lyon

TYPES 2019
Oslo, 11-14 June
Lambek and Scott:

A quotation from their book *Introduction to higher order categorical logic*, 1986:

*We also claim that intuitionistic type theories and toposes are closely related, in as much as there is a pair of adjoint functors between their respective categories. This is worked out in Part II. The relationship between Martin-Löf type theories and locally cartesian closed categories was established too recently (by Robert Seely) to be treated here.*
Categories and dependent type theory

P. Clairambault, PD, TLCA 2011, MSCS 2014:
- Two biequivalences of 2-categories

```
\[
\begin{array}{ccc}
\text{FL} & \xrightarrow{L} & \text{Cwf}_{\text{dem}}^{N_1, \Sigma, I_{\text{ext}}} \\
\text{C} & \text{Cwf}_{\text{dem}}^{N_1, \Sigma, I_{\text{ext}}} & \text{LCC} \\
\text{L} & \text{C} & \text{C}
\end{array}
\]
```

Democracy means there exists type $\overline{\Gamma}$ and $d_{\Gamma} : \Gamma \simeq 1.\overline{\Gamma}$.

S. Castellain, P. Clairambault, PD, TLCA 2015, LMCS 2017:
- Equality of arrows in the bifree $\text{LCC}$ is undecidable.
- (Construction of free cwfs with extra structure and bifree lcccs.)
Lambek and Scott

1. Categorical preliminaries
2. Simply typed $\lambda$-calculus and cccs
3. Untyped $\lambda$-calculus and C-monoids
4. Intuitionistic type theory and toposes
How to rewrite Lambek and Scott?

Instead use cwf as central notion on the *syntactic* side:

1. Categorical preliminaries (add lcccs, bicategories, . . .)
2. Untyped cwfs (ucwfs) with \( \lambda \)-structure
   instead of untyped \( \lambda \)-calculus
3. Simply typed cwfs (scwfs) supporting \( \to, \times, \ldots \)
   instead of simply typed \( \lambda \)-calculus
4. Cwfs supporting \( \Pi, \Sigma, I, \ldots \)
   instead of Martin-Löf type theory

(See arXiv:1904.00827 [cs.LO], 44 pages.)
Simply typed and unityped cwfs

**Cwfs:** has presheaf of types

**Scwfs:** has set of types (presheaf is constant)

**Ucwfs:** has one (constant) type

Context comprehension simplifies for scwfs and ucwfs.
Ucwfs

Three equivalences of 1-categories (structure strictly preserved)

- with cartesian operads:

\[
\text{CartOperad} \leftrightarrow \text{Ucwf}_{\text{ctx}}
\]

- with Lawvere theories:

\[
\text{LawTh} \leftrightarrow \text{Ucwf}_{\text{ctx}}
\]

- with Lawvere theories of type $\lambda\beta\eta$ (Obułowicz 1977)

\[
\text{LawTh}^{\lambda,\beta,\eta} \leftrightarrow \text{Ucwf}_{\text{ctx}}^{\lambda,\beta,\eta}
\]

In contextual ucwfs the objects of $\mathcal{C} \cong \mathbb{N}$. 
Two Lambek-Scott-style equivalences of categories

- Two equivalences of 1-categories (structure strictly preserved)

\[
\begin{align*}
\text{CCs} & \xrightarrow{L} \text{Scwf}^{\times}_{\text{ctx}} \\
\text{Scwf}^{\times}_{\text{ctx}} & \xleftarrow{C} \text{CCs}
\end{align*}
\]

\[
\begin{align*}
\text{CCCs} & \xrightarrow{L} \text{Scwf}^{\times,\to}_{\text{ctx}} \\
\text{Scwf}^{\times,\to}_{\text{ctx}} & \xleftarrow{C} \text{CCCs}
\end{align*}
\]

- Products and exponentials in CCs and CCCs are given as structure which is preserved strictly.
- Scwfs are contextual, that is, contexts are lists of types.
Two biequivalences of 2-categories

Products and exponentials in **CCs** and **CCC**s are given as a property which is preserved up to isomorphism.

Pseudo scwf-morphisms preserve structure up to isomorphism.

Scwfs are *democratic*, that is, context $\Gamma$ is *represented* as type $\overline{\Gamma}$, such that $\Gamma \simeq 1.\overline{\Gamma}$. 
Initial ucwfs and scwfs with extra structure

For example: the untyped $\lambda\beta\eta$-calculus defined as the initial object in $\text{Ucwf}^{\lambda,\beta,\eta}$. Two instances:

- Turn the rules for $\lambda\beta\eta$-ucwfs into an inductive family: yields well-scoped, name-free version of the untyped $\lambda\sigma$-calculus.
- Construct the $\lambda\beta\eta$-ucwf of (one of the) "usual" definitions of the $\lambda$-calculus.

An "abstract syntax perspective" of logical systems.
Contextuality

A cwf is **contextual** iff there is a length function

\[ l : C_0 \to \mathbb{N} \]

such that

- \( l(\Gamma) = 0 \) iff \( \Gamma = 1 \) and
- \( l(\Gamma) = n + 1 \) iff there are unique \( \Delta \) and \( A \) such that \( \Gamma = \Delta.A \) and \( l(\Delta) = n \).

Cf Cartmell’s 1978 **contextual categories**. Note that unlike the other parts of the definition of cwfs

- it does not correspond to an inference rule of dependent type theory;
- it is not expressed in the language of generalized algebraic theories;
- however, free cwfs are contextual.
Democracy

A cwf is **democratic** provided each context $\Gamma$ is represented by a type $\overline{\Gamma}$ in the sense that there is an isomorphism

$$d_{\Gamma} : \Gamma \cong 1.\overline{\Gamma}$$

- it does not correspond to an inference rule of dependent type theory;
- however, the free cwf is democratic;
- democracy can actually be expressed in the language of generalized algebraic theories.