We close this chapter with Dana Scott's construction of a CCωR in mathematical practice: The D_∞-models.

Definition 26: A poset $(P,\leq)$ is *complete* if $(P,\leq)$ - considered as a posetal category - has $(\omega$-$)$ filtered colimits (i.e. finitely filtered unions) and an initial object $\emptyset$.
The *Scott topology* $\mathcal{O}\subseteq \mathcal{P}(P)$ on a complete poset (cpo in short) $(P,\leq)$ consists of the upwards closed sets $X\subseteq P$ which are dense for filtered diagrams: $X\in\mathcal{O}$, $F\subseteq P$ filtered s.th. $\bigcup F\in X$, then $F\cap X\neq\emptyset$.
The category CPO consists of complete posets and continuous maps for the resp. Scott-topologies.

Exercise 27: 1. A cpo $(P,\leq)$ equipped with $\mathcal{O}$ is a T_0-space, which generally is not T_1.

2. Given cpo's $(P_i,\leq)$, for a function $f: P_1\to P_2$ TFAE.

- $f:(P_1,\mathcal{O}_1)\to(P_2,\mathcal{O}_2)$ is continuous.
- f preserves filtered colimits.

3. Continuous maps $(P_1,\mathcal{O}_1)\to(P_2,\mathcal{O}_2)$ are order-preserving.

Proposition 28: The category CPO is cartesian closed.

Proof: Straight-forward. In fact, the category of (small) accessible categories is cartesian closed, and the acc. structure on the full subcategory CPO is induced. $\square$

Proposition 29: The category CPO has ω-sequential limits.

Proof: Given a functor $P_\bullet : \mathbb{N}^{op} \to CPO$ via a sequence

$$\cdots \to P_n \xrightarrow{f_n} \cdots \to P_3 \xrightarrow{f_2} P_2 \xrightarrow{f_1} P_1 ,$$

the limit $P_\infty := \lim_{n<\infty} P_n$ is given by the standard construction

$P_\infty := \{ (p_n)_{n<\infty} \in \prod_{n<\infty} P_n \mid f_{n+1}(p_{n+1}) = p_n \text{ f.a. } n \geq 0 \}$ with componentwise order.

It is straight-forward to compute that $P_\infty \in CPO$. $\square$

Theorem 30: (Scott) Every $P \in CPO$ can be embedded into a $P_\infty \in CPO$ s.th. $P \cong P_\infty^{P_\infty} \in CPO$. In particular, every cpo gives rise to an extensional λ-model.

Note that every $(P, \leq) \in CPO$ has enough points since $CPO(*, (P, \leq)) \cong P$, and two arrows $P \overset{f,g}{\longrightarrow} P$ in CPO coincide iff $\forall p \in P: f(p) = g(p)$.

This theorem is most popular for the complete lattice $P = \mathcal{P}(\mathbb{N})$.

To prove the theorem, we make the following definitions and observations.

Lemma 31: 1. Let $P \underset{R}{\overset{L}{\underset{\perp}{\leftrightarrows}}} Q$ be a reflective localization of cpo's, i.e. L, R are continuous maps, $L \circ R = 1_P$, $R \circ L \leq 1_Q$. Then conjugation with L and R yields a reflective localization $P^P \underset{R_* \circ L^*}{\overset{L_* \circ R^*}{\underset{\perp}{\leftrightarrows}}} Q^Q$ in CPO.

2. Every cpo P gives rise to a refl. localization $P \underset{\ulcorner \pi_1 \urcorner =: i_0}{\overset{p_0}{\underset{\perp}{\leftrightarrows}}} P^P$.

Proof: Exercise. $\square$

Aim: We'd like to find a fixed point of the assignment $P \mapsto P^P =: P_1$

$\rightsquigarrow$ Consider the diagram $P_\bullet = \mathbb{N}^{op} \to CPO$ obtained from Lemma 29:

$$\begin{tikzcd} P \arrow[r, shift right, hook, "i_0"'] \arrow[r, phantom, "\perp"] & P_1 \arrow[l, shift right, "p_0"'] \arrow[r, shift right, hook, "i_1"'] \arrow[r, phantom, "\perp"] & P_2 \arrow[l, shift right, "p_1"'] \arrow[r, shift right, hook, "i_2"'] \arrow[r, phantom, "\perp"] & \cdots \arrow[l, shift right, "p_2"'] \arrow[r, shift right, hook, "i_{n-1}"'] \arrow[r, phantom, "\perp"] & P_n \arrow[l, shift right, "p_{n-1}"'] \arrow[r, shift right, hook, "i_n"'] \arrow[r, phantom, "\perp"] & \cdots \arrow[l, shift right, "p_n"'] & \lim P_n =: P_\infty \end{tikzcd}$$,

for $P_{n+1} := P_n^{P_n}$.

Definition 32: For $P \in CPO$, and $0 \leq n \leq m < \infty$ we obtain compound maps

- $i_{nm} : P_n \to P_m$ defined as $i_{nm} := i_{m-1} \circ \dots \circ i_n$ if $m > n$, and $i_{nn} = id_{P_n}$ else.
- $p_{mn} : P_m \to P_n$ " " $p_{mn} := p_n \circ \dots \circ p_{m-1}$ if $m > n$, and $p_{nn} = id_{P_n}$ else.

Furthermore, for $n \in \mathbb{N}$ we get

- $p_{\infty n} : P_\infty \to P_n$ the n-th projection,
- $i_{n\infty} : P_n \to P_\infty$ via $x \mapsto (p_{n0}(x), \dots, p_{n\,n-1}(x), x, i_{n\,n+1}(x), \dots) \in P_\infty$.

Lemma 33: For all $0 \leq n \leq m \leq \infty$, the pair $P_n \underset{i_{nm}}{\overset{p_{mn}}{\leftrightarrows}} P_m$ is a reflective localization.

Proof: Refl. localizations are closed under composition, so the statement holds for $0 \leq n \leq m < \infty$. For $m = \infty$, we clearly obtain a retract. Furthermore, $i_{n\infty}(p_{\infty n}(\vec{x})) = i_{n\infty}(x_n) \leq \vec{x}$, since the order is defined componentwise, and each $p_n \dashv i_n$ is a refl. localization. $\square$

Exercise 34: 1. The cocone $\mathbb{N}\cup\{\infty\} \to \mathbf{CPO}$ given by

$$P_0 \xhookrightarrow{i_0} P_1 \xhookrightarrow{i_1} \cdots \xhookrightarrow{i_n} P_n \xhookrightarrow{i_{n+1}} \cdots \hookrightarrow P_\infty$$

is a colimit cocone.

2. If $f_n : P \to Q$ for $n<\infty$ is a family of continuous maps in CPO s.th. $\forall n \geq 0 : f_n \leq f_{n+1}$, then
$$f_\omega : P \to Q, \quad p \mapsto \bigvee_{n<\infty} f_n(p)$$
is a cont. map as well.

Proof of Theorem 30: The pair $P_\infty \underset{F}{\overset{G}{\rightleftarrows}} P_\infty^{P_\infty}$ given by

- $F(q)(r) := \bigvee_{n<\omega} i_{n\infty}(q_{n+1}(r_n))$
- $G(f) := \bigvee_{n<\omega} i_{n+1\,\infty}(\underbrace{p_{\infty n} \circ f \circ i_{n\infty}}_{\in P_n^{P_n} = P_{n+1}})$

is easily computed to be a pair of continuous maps, since they both are compositions of elementary operations in the cartesian closed category CPO. Furthermore, Exercise 34.2 applies to F, since for all $n \geq 0$,

$$\begin{aligned} - \; i_{n\infty}(q_{n+1}(r_n)) &= i_{n\infty}((p_{n+1}(q_{n+2}))(p_n(r_{n+1}))) \\ &= i_{n\infty}(p_n \circ q_{n+2} \circ i_n(p_n(r_{n+1}))) \\ &= i_{n\infty}(p_n \circ q_{n+2}(\overbrace{i_n \circ p_n}^{\leq 1_{P_{n+1}}}(r_{n+1}))) \\ &\leq i_{n\infty}(p_n(q_{n+2}(r_{n+1}))) \leq i_{n+1\,\infty}(q_{n+2}(r_{n+1})). \end{aligned}$$

since $i_{n\infty} = i_{n+1\,\infty} \circ i_n$ and $i_n \circ p_n \leq 1_{P_{n+1}}$

Similarly for G. The maps F and G are mutually inverse by a series of elementary computations (see Barendregt's book, 18.2.7–18.2.16). □

Chapter II: SIMPLY TYPED λ-CALCULI

Oddities like the Fixed Point Theorems or the divergence of $\beta(\eta)$-reductions in the untyped λ-calculus are removed by stratifying λ-terms via a formal and intrinsic domain+codomain assignment. An introduction of according sorts defines simply typed λ-calculi.

Idea: * If t is a term of a type B, and x is a variable of type A, then $\lambda x.t$ is of type "$A \to B$".

* If f is of type $A \to B$ and a is of type A, then $App(f,a)$ is of type B.

Conditionals of this form effectively prohibit "self-referential" terms such as $App(x,x)$, but still allow intuitive constructions such as $I_A :\equiv \lambda x.x$ for all types A, and variables x of type A.

⇝ We therefore introduce a pair of calculi of type constructors and type equalities, on top of which we then will build the typed term-calculi of term-constructors and term equalities.

Here, we are confronted with various choices.

First, concerning the calculus of type constructors, we certainly want to allow consideration of some constant types A, and the compound type $A \to B$ whenever A, B are types. But we have seen that the λ-calculus allows for the internal constructions of all sorts of other mathematical compound structures such as pairs, numerals, Boolean truth values, ...

One may therefore consider to introduce according types $A \times B$ of finite tuples, $\mathbb{N}$ of numerals, Ω of truth values, ...

For the sake of simplicity, we will add product types, and mention the management of the others peripherically at the end only.

Notation 1: Consider the language $\mathcal{L}_{Ty}^{\mathbb{T}} := \{ \to, \times, 1, (,) \} \cup \mathbb{T}$ where $\mathbb{T}$ is an arbitrary set of constant type symbols.

Definition 2: The *calculus of type constructors* over $\mathcal{L}_{Ty}^{\mathbb{T}}$ is given by the following rules.

1. (Cnst.-Intro.) $\dfrac{}{T}$ for all $T \in \mathbb{T}$
2. (1-Intro.) $\dfrac{}{1}$
3. ($\to$-Intro.) $\dfrac{A \quad B}{(A \to B)}$
4. ($\times$-Intro.) $\dfrac{A \quad B}{(A \times B)}$

"Introduction-rules"

The associated product is denoted by $Ty^{\mathbb{T}}$. Brackets will be omitted if not necessary.

The calculus of *type formulas* over the language $\{Ty^{\mathbb{T}}, :\equiv_{Ty}\}$ is given by the rule

$$\dfrac{}{A :\equiv_{Ty} B} \quad \text{for } A, B \in Ty.$$

Its product is $\Phi_{Ty}^{\mathbb{T}}$. A theory of types is a subset $T_{Ty} \subseteq \Phi_{Ty}^{\mathbb{T}}$.

The according structural rules over $(\Phi^{T}_{T_\gamma})^\infty$ are the following. The meta-variable Γ ranges over $(\Phi^{T}_{T_\gamma})^\infty$.

1. $\dfrac{\Gamma \vdash_{T_\gamma} A :\equiv_{T_\gamma} A' \qquad \Gamma \vdash_{T_\gamma} B :\equiv_{T_\gamma} B'}{\Gamma \vdash_{T_\gamma} A \times B :\equiv_{T_\gamma} A' \times B'}$

2. $\dfrac{\Gamma \vdash_{T_\gamma} A :\equiv_{T_\gamma} A' \qquad \Gamma \vdash_{T_\gamma} B :\equiv_{T_\gamma} B'}{\Gamma \vdash_{T_\gamma} A \to B :\equiv_{T_\gamma} A' \to B'}$

3. $\dfrac{}{\Gamma \vdash_{T_\gamma} A :\equiv_{T_\gamma} A}$

4. $\dfrac{\Gamma \vdash_{T_\gamma} A :\equiv_{T_\gamma} B}{\Gamma \vdash_{T_\gamma} B :\equiv_{T_\gamma} A}$

5. $\dfrac{\Gamma \vdash_{T_\gamma} A :\equiv_{T_\gamma} B \qquad \Gamma \vdash_{T_\gamma} B :\equiv_{T_\gamma} C}{\Gamma \vdash_{T_\gamma} A :\equiv_{T_\gamma} C}$

(Type-Congruence rules)

6. $\dfrac{}{\Gamma, A :\equiv_{T_\gamma} B \vdash_{T_\gamma} A :\equiv_{T_\gamma} B}$ (Monotonicity)

7. $\dfrac{\Gamma \vdash_{T_\gamma} A :\equiv_{T_\gamma} B}{\Gamma, C :\equiv_{T_\gamma} D \vdash_{T_\gamma} A :\equiv_{T_\gamma} B}$ (Weakening)

Let $\vdash_{T_\gamma}$ denote the product. For a theory $\mathcal{T}_{T_\gamma} \subseteq \Phi^{T}_{T_\gamma}$, let

$$\overline{\mathcal{T}}_{T_\gamma} := \{ A :\equiv_{T_\gamma} B \in \Phi^{T}_{T_\gamma} \mid \exists\, \Gamma \subseteq \mathcal{T}_{T_\gamma} \text{ finite} : \Gamma \vdash_{T_\gamma} A :\equiv_{T_\gamma} B \}.$$

Second, to construct terms, there are two major conventions:

a. The Church-version: Type-assignment of terms is intrinsic, i.e. terms are introduced as terms of a given type from the get-go. This applies in particular to variables. Thus, for every type A, there is a distinguished countably infinite set Var_A of variables of type A.

b. The Curry-de Bruijn version: Terms are introduced as such in a calculus of pre-terms; type-assignments are imposed as part of the logical rules afterwards. In particular, there is only one set Var of variables, and type-assignment of variables is a relative/local judgement formulated with respect to contexts of variable declarations.

In this chapter, we will adopt the Church-version, both for the sake of versatility - as the dependently typed case will be formulated as in $\mathcal{J}$. - as well as for the fact that Lambek's original work was performed in this context.

Notation 3: Let $\mathbb{T}$ be a set of constant type symbols and $\mathbb{C} := \coprod_{A:Ty^{\mathbb{T}}} C_A$ be a set of constant term symbols. Consider the language

$$\mathcal{L}^{\mathbb{T},\mathbb{C}}_{Tm} := \{:;App;(,);\lambda;.;[;];,;pr_1;pr_2,*\} \amalg \mathbb{C} \amalg \coprod_{A:Ty^{\mathbb{T}}} Var_A \amalg Ty^{\mathbb{T}}.$$

Definition 4: The *simply typed λ-term calculus* wrt. some theory $\mathcal{T} \subseteq \Phi^{\mathbb{T}}_{Ty}$ is given by the following rules over $\mathcal{L}^{\mathbb{T},\mathbb{C}}_{Tm}$. The meta-variables A, B range over $Ty^{\mathbb{T}}$.

1. $\dfrac{}{x:A}$ for all $x \in Var_A$ 2. $\dfrac{}{c:A}$ for all $c \in C_A$.

3. $\dfrac{b:B}{\lambda x.b : A \to B}$ for $x \in Var_A$ (λ-Abstraction / $\to$-Construction)

4. $\dfrac{f:A\to B \quad a:A}{App(f,a):B}$ ($\to$-Elimination) 5. $\dfrac{a:A \quad b:B}{[a,b]:A\times B}$ ($\times$-Construction)

6. $\dfrac{c:A\times B}{pr_1(c):A}$, $\dfrac{c:A\times B}{pr_2(c):B}$ ($\times$-Elimination)

7. $\dfrac{}{*:1}$ (1-Construction)

8. $\dfrac{a:A}{a:B}$ whenever $\Gamma\vdash_{T_y} A :\equiv_{T_y} B$ (Term-Conversion)

The product of this calculus is the disjoint union $\coprod_{A:T_y^{\Pi}/\Gamma} \Lambda_A^{\alpha}$ of well-formed terms of type A for $A:T_y^{\Pi}$, i.e.

$$\Lambda_A = \{ t\in (\mathcal{L}_{Tm}^{\Pi,\alpha})^{\infty} \mid \text{There is a derivation } \vec{\omega} \text{ in the } \lambda\text{-term calculus s.th. } \frac{\omega_1 \cdots \omega_n}{t:A} \}$$

Example 5: * $I_A := \lambda x.x : A \to A$ for all $A:T_y^{\Pi}$, $x\in Var_A$

$\overset{?}{=} \lambda y.y$ for $y\in Var_A$?

* $K_{A,B} := \lambda x.\lambda y.x : A \to (B\to A)$ f.a. $A,B\in T_y^{\Pi}$, $x\in Var_A$, $y\in Var_B$.

$\overset{?}{\parallel}$

* $\lambda x.pr_1(x) : A\times B \to A$ for $A,B\in T_y^{\Pi}$, $x\in Var_{A\times B}$.

Definition 6: The set $FV(t)$ of free variables of a term $t\in\Lambda_A$ is defined recursively in analogy to the untyped case (Definition II.1.5), i.e. the only variable-binding operator is λ-abstraction.
Accordingly, a term t is closed if $FV(t)=\emptyset$.

Definition 7: The *substitution* $s[t/x]$ of a term $t \in \Lambda_A$ in a term $s \in \Lambda_B$ for a variable $x \in \mathrm{Var}_A$ is defined again by recursion, just as in the untyped case (Definition II.1.6), only additionally respecting type-assignments conditionally. All steps are straight-forward except λ-abstraction, which is handled like in the untyped case.

Definition 8: The calculus of *typed λ-term formulas* over the language $\left(\coprod_{A : \mathrm{Ty}^{\mathbb{T}}} \Lambda_A^{\mathfrak{C}} \right) \amalg \{ :\equiv_{\mathrm{Tm}} \}$ is given by the rule

$$\frac{}{s :\equiv_{\mathrm{Tm}} t} \quad \text{f.a. } s,t \in \Lambda_A^{\mathfrak{C}},\ A : \mathrm{Ty}^{\mathbb{T}}.$$

Its product will be denoted by $\phi_{\mathrm{Tm}}^{\mathbb{T},\mathfrak{C}}$.

Remark 9: Lambek in his original work (see [LS86]) considers λ-term formulas in context of possibly over-determined contexts of variable declarations:

$$\frac{}{(X,\ s :\equiv_{\mathrm{Tm}} t)} \quad \text{f.a. } s,t \in \Lambda_A^{\mathfrak{C}},\ A : \mathrm{Ty}^{\mathbb{T}},\ \mathrm{FV}(s) \cup \mathrm{FV}(t) \subseteq X \in \Big(\coprod_{A : \mathrm{Ty}^{\mathbb{T}}/\tau} \mathrm{Var}_A \Big)^{\omega}.$$

While this may be worthwile for the study of theory-extensions, it is rather redundant for the development of basic theory.
Furthermore, the calculus of types in LS86, Chapter 10 is not a formal calculus, but rather a family of sets underlying the formal calculus of terms.