Denotational
5. Semantics.

① Interpreting the lambda calculus denotationally is not easy. It ~~has been~~ was one of the longstanding open problems in the 1960's, until Scott solved it using the techniques from the domain theory (which Scott himself developed partly for this purpose).

② To understand why it was an open problem, let's try to interpret expressions in the lambda calculus (denotationally) using sets and functions. This trial will fail as we explain shortly and will show the challenge of handling the fact that in the lambda calculus, functions can act on themselves.

The first thing that we should do (in this trial) is to find an (appropriate) space S (which in this case is a set) for the meanings of the expressions. Let's suppose that we somehow managed to find such S. Then, S should include all functions on S that may be denoted by expressions in the lambda calculus. ~~Suppose that~~ Let's make a slightly stronger but nicer assumption that

$$[S \to S] \subseteq S$$

(set of all functions on S.)

We will now show that S is the singleton set. This is because the assumption implies that every function f on S has a fixed point, which can happen only if S is a singleton set. ~~How~~ does a function $f \in [S \to S]$ have ? (what fixed point)
Here is an answer for the question: Let p be a function on S s.t. $p(x) = f(x(x))$ for all $x \in [S \to S] \subseteq S$. Such p exists. Then $p(p) = f(p(p))$. So, $p(p)$ is a fixed point of f.
We have just shown that $[S \to S] \not\subseteq S$ for any set S that contains more than one element.

This shows that we cannot get a decent denotational semantics using sets and functions.

③ What should we do? We need to use the domain theory and the categorical tools, in particular general categorical fixed-point theorem and the category Dom^{EP}, which consists of domains and a particular kind of strict continuous functions called embeddings. If you are curious about these, look at the notes on "5. Famous Example of the Fixed Point Theorem" (6 Nov 2018). Using these tools, we can find domains D_1, D_2, D_3 and V_2, V_3 s.t.

① $D_1 \cong [D_1 \to_c D_1]$ (Also denoted by D_∞ in other textbooks.)

Isomorphism between domains.

(continuous functions. We often omit c but here we wrote it explicitly to emphasise the fact that we are considering continuous functions only here.)

② $D_2 \cong [D_2 \to_c D_2]_\perp$ $V_2 \stackrel{def}{=} [D_2 \to_c D_2]$.

$D_3 \stackrel{def}{=} (V_3)_\perp$ ③ $V_3 \cong [V_3 \to_c (V_3)_\perp]$

④ Note that D_1, D_2, V_3 are solutions of slightly different recursive domain equations. Why do we consider three such equations, instead of one? It is because the contraction relation, the normal-order evaluation and the eager evaluation provide three different meanings to the expressions in the lambda calculus, and these equations capture these differences. D_1 is for the contraction relation, D_2 and V_2 for the normal-order evaluation, and D_3 and V_3 for the eager evaluation.