

# Lecture 5: Set Theoretic Semantics for STLC

Lecturer: Max S. New

Scribe: Jesse Slater

September 10th, 2025

## 1 Review of STLC

In STLC, we have four basic notions. Types, contexts, terms, and, equations (or proofs of equations).

$A$  Type

$\Gamma$  Contexts

$\Gamma \vdash M : A$  Terms

$\Gamma \vdash M = N : A$  Equations

Let's talk about semantics of STLC just like we were talking about semantics of IPL. We can think about STLC from the perspective of logic in two different ways. On the one hand, we can think about types as propositions and terms as proofs. This is just like IPL. It's the Curry Howard correspondance. We will call this interpretation 1. On the other hand, we can view equations as propositions and derivations as proofs in a logic. We will call this interpretation 2. For each of these views of logics, we can ask the same questions we asked about IPL.

First, are they consistent (i.e. nontrivial)? If they aren't consistent, then they aren't useful for reasoning about truth.

### 1.1 Interpretation 1 Consistency

We can prove consistency by showing a type for which there are no terms. We can use the empty type. We need to show there is no way to form a term of the empty type under the empty context (i.e. there is no closed term of the empty type). In other words, the following is impossible:

$\cdot \vdash M : 0$

**Theorem 1.** *STLC viewed as a logic through interpretation 1 is consistent.*

*Proof.* We interpret terms STLC as proofs in IPL. IPL is consistent so STLC is consistent.  $\square$

## 1.2 Interpretation 2 Consistency

Is the equational theory of  $\text{STLC}(\emptyset)$  consistent? We fix the empty signature because if not, we could always choose a signature which is inconsistent.

**Theorem 2.** *STLC viewed as a logic through interpretation 2 is consistent.*

To prove this, we will pick a pair of terms of the same type and show that it is impossible to prove they are equal. We choose  $\sigma_1() : 1 + 1$  and  $\sigma_2() : 1 + 1$ . We need to show that there is no way to derive  $\Gamma \vdash \sigma_1() = \sigma_2() : 1 + 1$ . This will be our task for today. Our approach will be similar to our proof of IPL's consistency. We will have an interpretation of the propositions of our logic as facts, and we will show the fact which this proposition corresponds to is false.

We will have a translation for all of lambda calculus. We will provide interpretations of types, contexts, and terms. The equations will be interpreted as equalities between the meaning of terms.

This is the most intuitive understanding of STLC's meaning. Types are sets. Contexts are products of sets. Terms are functions between sets. Equations are truths about terms. This is a logic view of the meaning of notation.

## 2 Set theoretic semantics for STLC

We are going to define denotational set theoretic semantics for STLC. We will be general for non-empty signatures.

Recall, a signature has three parts:

$\Sigma_0$  base types

$\Sigma_1$  function symbols

$\Sigma_2$  axioms

Function symbols are allowed to include complex types, not just base types. Thus, we need interpretations of complex types as sets before we can define interpretations of function symbols. We will define an interpretation of base types, then complex types, then function symbols, and then axioms.

## 2.1 Interpretation of types and contexts

We will define a denotation function which maps types and contexts to sets, given an interpretation  $i : \Sigma_0 \rightarrow \text{Set}$ .

$$\frac{A \text{ type}}{\llbracket A \rrbracket \text{ Set}} \qquad \frac{\Gamma \text{ context}}{\llbracket \Gamma \rrbracket \text{ Set}}$$

The denotation function is defined as follows:

$$\begin{aligned} \llbracket X \rrbracket &:= i(X) \\ \llbracket 1 \rrbracket &:= \{*\} \\ \llbracket 0 \rrbracket &:= \{\} \\ \llbracket A \times B \rrbracket &:= \llbracket A \rrbracket \times \llbracket B \rrbracket \text{ (not getting into how to interpret product set theoretically)} \\ \llbracket A + B \rrbracket &:= \llbracket A \rrbracket \uplus \llbracket B \rrbracket = \{1\} \times \llbracket A \rrbracket \cup \{2\} \times \llbracket B \rrbracket \\ \llbracket A \Rightarrow B \rrbracket &:= \llbracket B \rrbracket^{\llbracket A \rrbracket} \\ \llbracket x_1 : A_1, \dots, x_n : A_n \rrbracket &:= \prod_{i \in [n]} \llbracket A_i \rrbracket \end{aligned}$$

We call an element of this  $\tilde{\gamma}$ , a semantic substitution. It provides values for all free variables in the context.

## 2.2 Interpretation of functions

Now we can define an interpretation for functions. A function symbol interpretation  $i_1$  extends  $i_0$ .

For each function symbol  $f :$

$$f : (A_1 \dots A_n) \rightarrow B \mapsto i_1(f) : \llbracket x_1 : A_1, \dots, x_n : A_n \rrbracket \rightarrow \llbracket B \rrbracket$$

## 2.3 Interpretation of terms

We define an interpretation for every well typed term  $\Gamma \vdash M : A$ .

$$\llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket$$

We define this recursively by cases on the shape of  $M$ .

Variable:

$$\llbracket x \rrbracket : \llbracket \Delta, x : A_i, \Gamma \rrbracket \rightarrow \llbracket A_i \rrbracket$$

$$\llbracket x \rrbracket(\tilde{\gamma}) := \tilde{\gamma}.i$$

Function:

$$\llbracket f(M_1 \dots M_n) \rrbracket(\tilde{\gamma}) := i(f)(\llbracket M_1 \rrbracket(\tilde{\gamma}) \dots \llbracket M_n \rrbracket(\tilde{\gamma}))$$

Product:

$$\llbracket (M_1, M_2) \rrbracket(\tilde{\gamma}) := (\llbracket M_1 \rrbracket(\tilde{\gamma}), \llbracket M_2 \rrbracket(\tilde{\gamma}))$$

Left projection:

$$\llbracket \pi_1(M) \rrbracket(\tilde{\gamma}) := \pi_1(\llbracket M \rrbracket(\tilde{\gamma}))$$

Right projection:

$$\llbracket \pi_2(M) \rrbracket(\tilde{\gamma}) := \pi_2(\llbracket M \rrbracket(\tilde{\gamma}))$$

Unit introduction:

$$\llbracket () \rrbracket(\tilde{\gamma}) := *$$

Empty elimination:

$$\llbracket \text{case } M : 0 \rrbracket(\tilde{\gamma}) := !$$

Left tagging:

$$\llbracket \sigma_1(M) \rrbracket(\tilde{\gamma}) := (1, \llbracket M \rrbracket(\tilde{\gamma}))$$

Right tagging:

$$\llbracket \sigma_2(M) \rrbracket(\tilde{\gamma}) := (2, \llbracket M \rrbracket(\tilde{\gamma}))$$

Cases:

$$\llbracket \text{case } M \{ \sigma_1(x).N_1 \mid \sigma_2(y).N_2 \} \rrbracket(\tilde{\gamma}) := \begin{cases} \llbracket N_1 \rrbracket(\tilde{\gamma}, \tilde{x}) & \text{if } \llbracket M \rrbracket(\tilde{\gamma}) = (1, \tilde{x}) \\ \llbracket N_2 \rrbracket(\tilde{\gamma}, \tilde{y}) & \text{if } \llbracket M \rrbracket(\tilde{\gamma}) = (2, \tilde{y}) \end{cases}$$

Lambda:

$$\llbracket \lambda x.M \rrbracket(\tilde{\gamma}) := \tilde{x} \mapsto \llbracket M \rrbracket(\tilde{\gamma}, \tilde{x})$$

Application:

$$\llbracket M N \rrbracket(\tilde{\gamma}) := \llbracket M \rrbracket(\tilde{\gamma})(\llbracket N \rrbracket(\tilde{\gamma}))$$

To prove completeness, we need to prove that the axioms of our equational theory correspond to true facts about equality of functions and elements of sets. We still need to define interpretation for axioms.

## 2.4 Interpretation of axioms

For an axiom interpretation  $i_3$ ,  $i_1$  and  $i_2$  are a valid interpretation of the axioms when  $M = N \in \Sigma_2 \Rightarrow \llbracket M \rrbracket = \llbracket N \rrbracket$ . (soundness of equational theory in set theoretic model)

Reflexivity, transitivity, and symmetry rules are trivial because equality has these properties.

Congruence rules:

$$\frac{M = M' \quad N = N'}{(M, N) = (M', N')}$$

We have  $\llbracket M \rrbracket = \llbracket M' \rrbracket$  and  $\llbracket N \rrbracket = \llbracket N' \rrbracket$ . We need to show  $\llbracket (M, N) \rrbracket = \llbracket (M', N') \rrbracket$ . By definition  $\llbracket (M, N) \rrbracket(\tilde{\gamma}) = (\llbracket M \rrbracket(\tilde{\gamma}), \llbracket N \rrbracket(\tilde{\gamma}))$  and  $\llbracket (M', N') \rrbracket(\tilde{\gamma}) = (\llbracket M' \rrbracket(\tilde{\gamma}), \llbracket N' \rrbracket(\tilde{\gamma}))$ . Thus, they are equal by assumption.

In general, all the congruence rules are trivial because everything in our model respects the congruence of equality. The only exception is congruence of substitution, which is more interesting.

Congruence of substitution:

$$\frac{M = M' \quad N = N'}{M[N/x] = M'[N'/x]}$$

Since we don't interpret substitution as a set theoretic operation. To prove it, we need the compositionality lemma.

**Lemma 1.**  $\llbracket M[N/x] \rrbracket(\tilde{\gamma}) = \llbracket M \rrbracket(\tilde{\gamma}, \llbracket N \rrbracket(\tilde{\gamma}))$

This proof is tedious. It requires first proving the weakening lemma.

**Lemma 2.**  $\llbracket M \rrbracket^{\Gamma, A} = \llbracket M' \rrbracket^{\Gamma', A} \circ \pi^{\Gamma', \Gamma}$

We will skip both proofs because they are boring. Try them if you like induction.

Functions:

$$(\lambda x. M)N = M[N/x]$$

Need to show the two sides semantics are the same.

$$\llbracket \lambda x. M \rrbracket(\tilde{\gamma}) = \llbracket M \rrbracket(\tilde{\gamma}, \llbracket N \rrbracket(\tilde{\gamma}))$$

Finish by compositionality lemma.

$$(\lambda x. Mx) = M$$

Need to show the two sides semantics are the same.

$$\llbracket (\lambda x. Mx)N \rrbracket(\tilde{\gamma})(\tilde{x}) = \llbracket M \rrbracket^{\Gamma, x:A}(\tilde{\gamma}, \tilde{x})(\tilde{x}) = \llbracket M \rrbracket^{\Gamma}(\tilde{\gamma})(\tilde{x}) \text{ by weakening lemma.}$$

All the other cases are easier than function, so we won't do them here.

Now we can return to our question. Is  $1 = 0$ ? Assume we have a term Assume we have a term  $\cdot \vdash \sigma_1() = \sigma_2() : 1 + 1$ . Then the semantics of each side applied to elements is equal.  $\llbracket \sigma_1() \rrbracket(*) = \llbracket \sigma_2() \rrbracket(*) \Rightarrow (1, *) = (2, *)$ . But this is false, so no such term can exist. This proves soundness.

There are questions which set theoretic semantics can't answer. For example, canonicity. That is, any closed term of type Bool is either true or false. And if we can prove canonicity computable, then we get an evaluator which is proven correct with respect to the equational theory.

What about completeness? Are all functions in the semantics definable by lambda terms (functional completeness)? If the denotations of two terms are equal, are the terms provably equal in the equational theory (equational completeness)? It turns out both are true for the empty signature but not for non-trivial signatures.

### 3 Category Theory

We want a complete semantics for STLC. In the same way that we showed biheyting pre-algebras are complete for IPL, we want to come up with a semantics which are general enough for STLC. By breaking semantics into reusable pieces, we can avoid repeating big inductions over variations on STLC, and instead consider which semantic components it has.

The correct generalization of a biheyting pre-algebra is a bi-cartesian closed category.

**Definition 1.** *A category  $\mathcal{C}$  consists of*

1.  $\mathcal{C}_0$  a set of "objects"
2.  $\forall x, y \in \mathcal{C}_0, \mathcal{C}_1(x, y)$  set of "arrows"
3.  $\forall x \in \mathcal{C}_0, id_x \in \mathcal{C}_1(x, x)$  identity element
4.  $\forall x, y, z \in \mathcal{C}_0, f \in \mathcal{C}_1(x, y), g \in \mathcal{C}_1(y, z), g \circ f \in \mathcal{C}_1(x, z)$  composition
5.  $g \circ id_x = g, id_y \circ g = g, (h \circ g) \circ f = h \circ (g \circ f)$

#### 3.1 Example Categories

1. Any preorder forms a category where objects are elements in  $P$  and morphisms exist between related elements.
2. Sets. Objects are sets. Morphisms are functions between sets.
3. STLC where types are objects and terms with single variables are morphisms. Composition is substitution.
4. STLC where types are objects and closed terms of function types are morphisms. Composition is lambda composition.