Lecture Notes
LMFI Master 2
Proofs and Programs: Part 2
Denotational Semantics

LMFI Students   Gabriele Vanoni

February 2, 2026

Disclaimer: these notes have been taken by students and lightly revised by the teacher. They are not intended to substitute textbooks or papers. They are mostly intended to give an account of what has been done in class, with the main definitions. They are not complete, and probably not even sound. We apologize for mistakes, errors, typos, etc.

1 The λ-Calculus

Given a countable set of variables $\mathcal{V}$, *terms* and *contexts* are defined by the following grammar:

$$\begin{array}{rlll} \textsc{Terms} & t,u & ::= & x \in \mathcal{V} \mid \lambda x.t \mid tu \\ \textsc{Contexts} & C & ::= & \langle\,\rangle \mid \lambda x.C \mid Ct \mid tC \end{array}$$

Free and *bound variables* are defined as usual: $\lambda x.t$ binds x in t. Terms are considered modulo α-equivalence. Capture-avoiding (meta-level) *substitution* of u for all the free occurrences of x in t is written $t\{x/u\}$. Contexts are just λ-terms containing exactly one occurrence of a special symbol, the hole $\langle\,\rangle$, intuitively standing for a removed subterm. We furthermore consider the top level β-rule as follows:

$$(\lambda x.t)u \mapsto t\{x := u\}$$

and β-reduction as its *closure* by evaluation contexts:

$$\frac{t \mapsto u}{C\langle t\rangle \to C\langle u\rangle}\ \beta$$

2 Simple types

We want to interpret terms in a mathematical world. We want to define a map which interprets λ-terms as $[\![\cdot]\!] : \Lambda \to D$ where Λ is the set of lambda terms and D is a domain of of interpretation. We would like to interpret certain lambda terms as actual set-theoretic functions. For example, in a term tu, we would like to interpret $[\![t]\!] \in D \to D$ and $[\![u]\!] \in D$, but also $[\![t]\!] \in D$. However, no set satisfies $D = D^D$, for example by a cardinality argument $|D^D| \geq 2^{|D|} > |D|$. We introduce types to solve this problem.

$$\textsc{Types} \quad \mathbb{T} \ni A, B \quad ::= \quad o \mid A \to B$$

$$\frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash tu : A}\ \text{T-@} \qquad \frac{}{\Gamma, x : A \vdash x : A}\ \text{T-Var} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\ \text{T-}\lambda$$

Simple types have strong normalization, but weak expressivity, for example, if we use Church numerals, we can only encode extended polynomials (polynomials and if-then-else).

We can still define semantics for them, as follows.

Definition 2.1 (STλC Model). A *model* of STλC is a family of sets $\mathcal{M} = \{M_A\}_{A\in\mathbb{T}}$ and an interpretation map $[\![\cdot]\!]$ such that for each A we have $[\![\pi \triangleright \Gamma \vdash t : A]\!]_\rho \in M_A$, where $\rho : \mathcal{V} \to \bigcup_A M_A$ respects $\Gamma = x_1 : A_1, \ldots, x_n : A_n$, i.e. $\rho(x_i) \in M_A$ for each $1 \leq i \leq n$.

Definition 2.2 (Set-Theoretic Model). Given a set O which interprets the base type o, we define:

$$\begin{aligned}
M_o &:= O \\
M_{A\to B} &:= B^A \\
[\![\pi \triangleright x : A \vdash x : A]\!]_\rho &:= \rho(x) \\
[\![\pi \triangleright \Gamma \vdash \lambda x.t : A \to B]\!]_\rho &:= \begin{cases} f : M_A \to M_B \\ f(a) = [\![\pi' \triangleright \Gamma, x : A \vdash t : B]\!]_{\rho[x\leftarrow a]:=\rho\cup\{(x,a)\}} \end{cases} \\
[\![\pi \triangleright \Gamma \vdash tu : A]\!] &:= [\![\pi' \triangleright \Gamma \vdash t : B \to A]\!]_\rho([\![\pi'' \triangleright \Gamma \vdash u : B]\!]_\rho)
\end{aligned}$$

Example 2.3. $[\![\pi\triangleright \vdash \lambda x.x : A \to A]\!]_\rho = f$ where $f : M_A \to M_A$ is the function defined by $f(a) = [\![x]\!]_{\rho[x\leftarrow a]} = a$, which is indeed the set-theoretic identity function.

Example 2.4. If we have

$$A, B ::= \quad \mathsf{Bool} \mid A \to B$$

and we take $O := \{T, F\}$, we recover the standard higher order functions on booleans.

Since we are giving meanings to terms, we would like to described what happens if one term β-reduces to another. In fact one could prove that the interpretation we gave is invariant under β-reduction. Types are also preserved by reduction:

Theorem 2.5 (Subject Reduction). *If $t \to u$ and $\Gamma \vdash t : A$, then $\Gamma \vdash u : A$.*

Theorem 2.6. *If $t \to u$ then $[\![t]\!]_\rho = [\![u]\!]_\rho$, for all ρ.*

Proof. By induction on evaluation contexts, using the following substitution lemma to handle the base case.

Lemma 2.7 (Substitution Lemma). *For all ρ, $[\![t]\!]_{\rho[x\leftarrow[\![u]\!]_\rho]} = [\![t\{x/u\}]\!]_\rho$*

□

3 Syntactical λ-Models

The goal of this section is to construct models of λ-calculus that somehow express interpretation in a mathematical way (e.g. set-theoretic functions). There are many definitions possible including the syntactical and the categorical definitions which are roughly equivalent modulo some choice in the axioms which are not settled yet in the community. We will use the following syntactical definition.

Definition 3.1 (Applicative structure / magma). An applicative structure (or magma in algebra) is a tuple $(S, \cdot)$ where S is a set and $\cdot : S \to S \to S$ is a binary operation on S.

Definition 3.2 (λ-model). A λ-model is a tuple $(D, \cdot, [\![\cdot]\!]_{(\cdot)})$ where $(D, \cdot)$ is an applicative structure and $[\![\cdot]\!]_{(\cdot)} : \Lambda \to (\mathrm{Var} \to D) \to D$ is an interpretation function such that the following hold:

1. $[\![x]\!]_\rho = \rho(x)$.
2. $[\![tu]\!]_\rho = [\![t]\!]_\rho \cdot [\![u]\!]_\rho$

3. $[\![\lambda x.t]\!]_\rho \cdot d = [\![t]\!]_{\rho[x \leftarrow d]}$

4. $[\![t]\!]_\rho = [\![t]\!]_{\rho'}$ if $\forall x \in \mathrm{FV}(t)$, $\rho(x) = \rho'(x)$. In other words, the interpretation of a term depends only on the interpretation of the free variables in the environment.

5. $(\forall d \in D, [\![t]\!]_{\rho[x \leftarrow d]} = [\![u]\!]_{\rho[x \leftarrow d]}) \implies [\![\lambda x.t]\!]_\rho = [\![\lambda x.u]\!]_\rho$

This definition is due to Hindley and Longo in 1980.

The first thing to highlight is that the invariance under β-reduction still follows from this definition:

Lemma 3.3. $[\![t\{x \leftarrow u\}]\!]_\rho = [\![t]\!]_{\rho[x \leftarrow [\![u]\!]_\rho]}$

Proof. By induction on t.

Case $t = x$

$$[\![x\{x \leftarrow u\}]\!]_\rho = [\![u]\!]_\rho = [\![x]\!]_{\rho[x \leftarrow [\![u]\!]_\rho]}$$

Case $t = y \neq x$ We can assume by α-renaming, that x does not occur free in u. Then, we can write the following:

$$[\![y\{x \leftarrow u\}]\!]_\rho = [\![y]\!]_\rho = [\![y]\!]_{\rho[x \leftarrow [\![u]\!]_\rho}$$

Case $t = \lambda y.r$ By the induction hypothesis, we have $[\![r\{x \leftarrow u\}]\!]_\rho = [\![r]\!]_{\rho[x \leftarrow [\![u]\!]_\rho]}$. We can also assume that x does not occur free in u so that:

$$[\![r\{x \leftarrow u\}]\!]_\rho = [\![r\{x \leftarrow u\}]\!]_{\rho[x \leftarrow [\![u]\!]_\rho]}$$

This is for any ρ, in particular, for any $d \in D$, we have:

$$\begin{aligned} [\![r\{x \leftarrow u\}]\!]_{\rho[x \leftarrow [\![u]\!]_\rho][y \leftarrow d]} &= [\![r]\!]_{\rho[x \leftarrow [\![u]\!]_\rho][y \leftarrow d]} \\ [\![\lambda y.r\{x \leftarrow u\}]\!]_{\rho[x \leftarrow [\![u]\!]_\rho]} &= [\![\lambda y.r]\!]_{\rho[x \leftarrow [\![u]\!]_\rho]} \\ [\![(\lambda y.r)\{x \leftarrow u\}]\!]_{\rho[x \leftarrow [\![u]\!]_\rho]} &= [\![\lambda y.r]\!]_{\rho[x \leftarrow [\![u]\!]_\rho]} \\ [\![(\lambda y.r)\{x \leftarrow u\}]\!]_\rho &= [\![\lambda y.r]\!]_{\rho[x \leftarrow [\![u]\!]_\rho]} \end{aligned}$$

Case $t = rs$

$$\begin{aligned} [\![rs\{x \leftarrow u\}]\!]_\rho &= [\![r\{x \leftarrow u\}s\{x \leftarrow u\}]\!]_\rho \\ &= [\![r\{x \leftarrow u\}]\!]_\rho [\![s\{x \leftarrow u\}]\!]_\rho \\ &= [\![r]\!]_{\rho[x \leftarrow [\![u]\!]_\rho]} [\![s]\!]_{\rho[x \leftarrow [\![u]\!]_\rho]} \\ &= [\![rs]\!]_{\rho[x \leftarrow [\![u]\!]_\rho]} \end{aligned}$$

□

Proposition 3.4. *If $t \to u$ then $[\![t]\!]_\rho = [\![u]\!]_\rho$, for all ρ.*

Proof. By induction on contexts with the special base case of

$$\begin{aligned} [\![(\lambda x.t)u]\!]_\rho &= [\![\lambda x.t]\!]_\rho [\![u]\!]_\rho \\ &= [\![t]\!]_{\rho[x \leftarrow [\![u]\!]_\rho]} \\ &= [\![t\{x \leftarrow u\}]\!]_\rho \end{aligned}$$

where the last step uses the substitution lemma above. □

Furthermore, one can prove also that interpretation is closed under evaluation contexts:

Proposition 3.5. *For any context* C, $[\![t]\!]_\rho = [\![u]\!]_\rho \implies [\![C\langle t\rangle]\!]_\rho = [\![C\langle u\rangle]\!]_\rho$

Proof. By induction on contexts

Case $C = \langle\rangle$ $\quad C\langle t\rangle = t$ so this follows by hypothesis.

Case $C = C'r$

$$\begin{aligned}[\![C'\langle t\rangle r]\!]_\rho &= [\![C'\langle t\rangle]\!]_\rho \cdot [\![r]\!]_\rho \\ &= [\![C'\langle u\rangle]\!]_\rho \cdot [\![r]\!]_\rho \\ &= [\![C'\langle u\rangle r]\!]_\rho\end{aligned}$$

Case $C = rC'$

$$\begin{aligned}[\![rC'\langle t\rangle]\!]_\rho &= [\![r]\!]_\rho \cdot [\![C'\langle t\rangle]\!]_\rho \\ &= [\![r]\!]_\rho \cdot [\![C'\langle u\rangle]\!]_\rho \\ &= [\![rC'\langle u\rangle]\!]_\rho\end{aligned}$$

Case $C = \lambda x.C'$ $\quad$ This follows by property 5 and the induction hypothesis.

□

The simplest model one can think of is the (purely syntactical) *term model*:

Definition 3.6 (Term model)**.** We define here the **term model** of λ-calculus, which interprets terms as themselves, modulo β-equivalence. Formally, it's the tuple $(D, \cdot, [\![\cdot]\!]_{(\cdot)})$ such that:

- $D = \{[t] \mid t \text{ is a } \lambda\text{-term}\}$ where $[t] = \{u \mid t =_\beta u\}$ is the equivalence class of t modulo β-equivalence.
- $[t] \cdot [u] = [tu]$
- $[\![t]\!]_\rho = [t\{x_1 \leftarrow u_1, \cdots, x_n \leftarrow u_n\}]$ with
  - For all i, $\rho(x_i) = [u_i]$
  - $\mathrm{FV}(t) = \{x_1, \cdots, x_n\}$.

Proposition 3.7. *The term model is a λ-model.*

Proof. We need to check the properties 1 through 5.

1. If $\rho(x) = [u]$, then $[\![x]\!]_\rho = [x\{x \leftarrow u\}] = [u] = \rho(x)$

2. If $\mathrm{FV}(t) = \{x_1, \cdots, x_n\}$, and $\mathrm{FV}(u) = \{y_1, \cdots, y_m\}$ with $\rho(x_i) = [u_i]$ and $\rho(y_i) = [v_i]$ for every appropriate i, then,

$$\begin{aligned}[\![t]\!]_\rho \cdot [\![u]\!]_\rho &= [t\{x_1 \leftarrow u_1, \cdots, x_n \leftarrow u_n\}u\{y_1 \leftarrow v_1, \cdots, y_m \leftarrow v_m\}] \\ &= [(tu)\{x_1 \leftarrow u_1, \cdots, x_n \leftarrow u_n, y_1 \leftarrow v_1, \cdots, y_m \leftarrow v_m\}] \\ &= [\![tu]\!]_\rho\end{aligned}$$

3. $\forall d \in D$, $\exists u \in \Lambda$, such that $d = [u]$.

$$\begin{aligned}[\![\lambda x.t]\!]_\rho \cdot d &= [\![\lambda x.t]\!]_\rho \cdot [u] \\ &= [\lambda x.t\{x_1 \leftarrow u_1, \cdots, x_n \leftarrow u_n\}u] \\ &= [\![t]\!]_{\rho[x \leftarrow u]}\end{aligned}$$

4. Let $\mathrm{FV}(t) = \{x_1, \cdots, x_n\}$, and ρ, ρ' be two environments such that $\forall i, \rho(x_i) = \rho'(x_i) = [u_i]$. Then:

$$\begin{aligned} [\![t]\!]_\rho &= [t\{x_1 \leftarrow u_1, \cdots, x_n \leftarrow u_n\}] \\ [\![t]\!]_\rho &= [\![t]\!]_{\rho'} \end{aligned}$$

5. □

3.1 Engeler's model

The term model we have seen so far interprets terms as equivalence classes of terms, so it does not fully capture the notion of interpretation in a 'mathematical way'. Other models can do this better, and one of those models is Engeler's model, which is an instance of a graph model.

Definition 3.8 (Engeler's model)**.** $\mathbb{A}$ is defined as the smallest set such that:

- $a \in \mathbb{A}$
- if $I \subseteq_f A$ and $A \in \mathbb{A}$, then $(I, A) \in \mathbb{A}$.

Engeler's model is then defined by:

- $D = \mathcal{P}(\mathbb{A})$.
- $d \cdot e = \{A : \exists I \subseteq_f e, (I, A) \in d\}$ [1].
- $[\![x]\!]_\rho = \rho(x)$
- $[\![\lambda x.t]\!]_\rho = \{(I, A) : A \in [\![t]\!]_{\rho[x:=I]}\}$
- $[\![tu]\!]_\rho = [\![t]\!]_\rho \cdot [\![u]\!]_\rho$.

Theorem 3.9. *Engeler's model is a λ-model.*

4 Strict Intersection Types

$$\begin{array}{rrcl} \textsc{Types} & A & ::= & a \mid I \to A \\ \textsc{Intersections} & I & ::= & \{A_1, \ldots, A_n\} \quad n \geq 0 \\ \textsc{Generic Types} & G & ::= & A \mid I \end{array}$$

$$\frac{A \in I}{\Gamma, x : I \vdash x : A}\ \text{T-VAR} \qquad \frac{[\Gamma \vdash t : A_i]_{i \in F}}{\Gamma \vdash t : \{A_i\}_{i \in F}}\ \text{T-MANY}$$

$$\frac{\Gamma, x : I \vdash t : A}{\Gamma \vdash \lambda x.t : I \to A}\ \text{T-}\lambda \qquad \frac{\Gamma \vdash t : I \to A \quad \Gamma \vdash u : I}{\Gamma \vdash tu : A}\ \text{T-@}$$

The system is syntax-directed, meaning that given a judgment $\Gamma \vdash t : G$, there is a unique way to have possibly obtained the judgment which is clear from the syntax. If $G = I$, the last rule must have been T-MANY, otherwise, if $t = \lambda x.t'$, the last rule must have been T-λ, if $t = x$, the last rule must have been T-VAR, and finally, if $t = uv$, the last rule must have been T-@.

Remark 4.1. After defining intersecting types one can slightly modify the interpretation in Def 3.8 as follows:

$$[\![t]\!]_\rho = \{A \in \mathbb{A} \mid \Gamma \vdash t : A \text{ if } \rho \vDash \Gamma\}$$

where $\rho \vDash \Gamma$ if and only if $\Gamma(x) \subseteq \rho(x)$ for each $x \in \mathcal{V}$. One can prove that the definition here and the one above are equivalent.

[1] This mimics the application rule on types T-@.

Remark 4.2. Given a proof (type inference) π, we indicate that π ends with the judgment J by writing $\pi \triangleright J$.

Intersection types also provide weakening, that is, if we have $\pi \triangleright \Gamma \vdash t : A$, then we can also form $\pi \triangleright \Gamma \uplus \Delta \vdash t : A$ by simply 'inserting' Δ everywhere in the proof. This can be easily proved by induction on the structure of π.

Theorem 4.3 (Subject reduction (SR))**.** *If* $\Gamma \vdash t : A$ *and* $t \to u$*, then* $\Gamma \vdash u : A$*.*

Proof. By induction on evaluation contexts, using the following substitution lemma to handle the base case. □

Lemma 4.4 (Substitution lemma)**.** *If* $\Gamma, x : I \vdash t : G$ *and* $\Gamma \vdash u : I$*, then* $\Gamma \vdash t\{x \leftarrow u\} : G$*.*

Proof. The proof goes by induction on type derivations $\pi \triangleright \Gamma \vdash t : A$. □

Theorem 4.5 (Subject expansion (SE))**.** *If* $\Gamma \vdash u : A$ *and* $t \to u$*, then* $\Gamma \vdash t : A$*.*

Lemma 4.6 (Antisubstitution lemma)**.** *If* $\Gamma \vdash t\{x \leftarrow u\} : G$*, then, there exists an intersection type* I *such that* $\Gamma, x : I \vdash t : G$ *and* $\Gamma \vdash u : I$*.*

Proof. Again by induction on $\pi \triangleright \Gamma \vdash t\{x \leftarrow u\}$ with one interesting case, that of $t = y \neq x$. In this case, we can type y, but not x. The solution is to choose $I = \emptyset$. So given

$$\frac{A \in J}{\Gamma, y : J \vdash y : A}\ \text{T-Var}$$

We get $I = \emptyset$, $\Gamma \vdash u : \emptyset$, and $\Gamma, x : \emptyset, y : J \vdash y : A$. □

The key idea behind this and other uses of the empty type is that the empty type is the type of terms that are erased during evaluation. For example, we can type $(\lambda y.x)\Omega$ without having any type for Ω, by giving $\Omega : \emptyset$, and erasing it during the reduction $(\lambda y.x)\Omega \to x$.

Proposition 4.7. *If* t *is in head-normal-form (HNF), then there exists* Γ, A, π *such that* $\pi \triangleright \Gamma \vdash t : A$*. In other words, any HNF is typable with some type in some environment.*

Recall: t is in HNF when it is of the form $\lambda x_1 \ldots \lambda x_n.yt_1 \ldots t_m$ with y free or one of x_i for $1 \leq i \leq n$ and $n, m \geq 0$ (possibly also 0).

Proof. Since $t_1, \cdots, t_m$ may not actually be typable, the idea is to give them types $\emptyset$ and give y the type a. If y is free, add $y : a$ to the environment and you get:

$$y : a \vdash t : \emptyset^n \to a.$$

If $y = x_i$, then keep the environment empty and give to t the type:

$$\vdash t : \emptyset^{i-1} \to (\emptyset^m \to a) \to \emptyset^{n-i} \to a.$$

Note, the exponent here does not mean a product type, but rather what would be the currified version of that type. So $f : A^3 \to B$ actually means $f : A \to (A \to (A \to B))$. □

Theorem 4.8 (Completeness theorem)**.** *If HN*(t) *(*t *is head-normalising), then,* $\exists \pi \triangleright \Gamma \vdash t : A$*.*

Proof. If HN(t), then there exists h in HNF such that $t \to h$. So there is Γ, A, π, such that $\pi \triangleright \Gamma \vdash h : A$. By induction on the length of the reduction and by the 1-step subject expansion, we get that $\Gamma \vdash t : A$. □

The goal now is to prove the converse, namely, $\Gamma \vdash t : A \implies \text{HN}(t)$. This is done via techniques of the theme of reducibility (Tait / Girard), realisability, and logical relations.

Definition 4.9 ($\vDash$ – logical relation – semantic entailment)**.**

(i) $\vDash t : a$ iff $\mathrm{HN}(t)$.

(ii) $\vDash t : I \to A$ iff for each u such that $\Gamma \vDash u$, we have $\vDash tu : A$

(iii) $\vDash t : \{A_1, \cdots, A_n\}$ iff $\forall i, \vDash t : A_i$.

We can extend this by Γ on the left of $\vDash$ as follows: $\{x_1 : I_1, \cdots, x_n : I_n\} \vDash t : G$ if and only if, for all u_i such that $\vDash u_i : I_i$, we have $\vDash t\{x_i \leftarrow u_1, \cdots, x_n \leftarrow u_n\} : G$

Goal: $\vdash t : A \implies \vDash t : A \implies \mathrm{HN}(t)$.

Lemma 4.10 (Neutral terms)**.** *For all G, $\vDash xt_1 \cdots t_n : G$.*

Proof. By induction on G.

- For $G = a$, $xt_1 \cdots t_n$ is in HNF; therefore, $\vDash xt_1 \cdots t_n : a$.
- For $G = I \to A$, let u be such that $\vDash u : I$, then consider $xt_1 \cdots t_n u$. Because the induction is on the type, not on the number n of terms, we can apply the induction hypothesis with $A \leq I \to A$ and obtain that $\vDash xt_1 \cdots t_n u : A$; therefore, $xt_1 \cdots t_n : I \to A$.
- For $G = \{A_1, \cdots, A_m\}$, by induction hypothesis, we have $\vDash xt_1 \cdots t_n : A_i$ for all $i : 1 \to m$; therefore, $\vDash t : G$.

□

Proposition 4.11. $\vDash t : A \implies$ *$HN(t)$.*

Proof. By induction on A.

- $\vDash t : a \implies \mathrm{HN}(t)$ by definition.
- $\vDash t : I \to A \implies \forall u$ such that $\vDash u : I, \mathrm{HN}(tu)$. Pick u to be any neutral term u_0 (see lemma above), and you get $\mathrm{HN}(tu_0)$; therefore, $\mathrm{HN}(t)$.

□

Remark 4.12. We used here the standard fact that $\mathrm{HN}(tu) \implies \mathrm{HN}(u)$ without proof.

Lemma 4.13. *If $t \to u$ and $\vDash u : A$, then $\vDash t : A$.*

Proof. By induction A.

- $\vDash u : a \implies \mathrm{HN}(u) \implies \mathrm{HN}(t) \implies \vDash t : a$.
- $\vDash u : I \to A$ implies that for all v such that $\vDash v : I$, we have $\vDash uv : A$. However, $tv \to uv$; therefore, by induction hypothesis, $\vDash tv : A$. Finally, this was for any v with $\vDash v : I$, so, $\vDash t : I \to A$.

□

Lemma 4.14 (Fundamental lemma)**.** $\Gamma \vdash t : G \implies \Gamma \vDash t : G$

Proof. By induction on $\pi \triangleright \Gamma \vdash t : G$. □

Proposition 4.15. *If $\Gamma \vDash t : A$ for some Γ, then $\vDash t : A$.*

Proof. x_i are neutral terms, so $\vDash x_i : I_i$. Applying the definition of $\vDash$ extended to Γ with $u_i = x_i$, we have $\vDash t\{x_1 \leftarrow x_1, \cdots, x_n \leftarrow x_n\} : A$, so $\vDash t : A$. □

This last proposition allows us to conclude that $\Gamma \vdash t : G$ implies $\Gamma \vDash t : G$ which implies $\vDash t : G$, finally implying $\mathrm{HN}(t)$.