Lambda-Calculus and Type Theory

Part III

Ugo Dal Lago

Scuola Estiva AILA, Gargnano, August 2018
Section 1

Classical Logic and the $\lambda\mu$-Calculus
Propositional Classical Logic

- If we restrict our attention to the fragment with implication and \( \perp \), we get the system \( CPL_{\rightarrow, \perp} \), which is defined as follows:

\[
\begin{align*}
\Gamma, \varphi \vdash \varphi & \quad \text{AX} \\
\Gamma, \varphi \vdash \top & \quad \Gamma \vdash \varphi \\
\Gamma, \varphi \vdash \tau & \quad \Gamma \vdash \varphi \rightarrow \tau \\
\Gamma \vdash \varphi \rightarrow \tau & \quad \Gamma \vdash \tau \\
\end{align*}
\]

- Rule \( E\neg \) subsumes \( E\perp \). Moreover, the following two rules are equivalent to it:

\[
\begin{align*}
\Gamma \vdash \perp & \\
\Gamma \vdash \varphi & \\
\Gamma, \varphi \rightarrow \tau \vdash \varphi & \quad \Gamma, \varphi \rightarrow \tau \vdash \varphi
\end{align*}
\]

Theorem

Propositional classical logic is sound and complete for classical validity.
The $\lambda\mu$-Calculus

Terms of the Church-style $\lambda\mu$-calculus are the following ones:

\[ M ::= x \mid (MM) \mid (\lambda x : \varphi M) \mid ([a]M) \mid (\mu a : \neg \varphi M) \]

where:

- $a$ ranges over a set $\Xi$ of addresses;
- $\varphi$ ranges over $\text{CPL}_{\rightarrow, \bot}$ formulas.

The notions of $\alpha$-conversion and substitution can be generalized from the $\lambda$-calculus to the $\lambda\mu$-calculus, keeping in mind that:

- $\mu a : \neg \varphi M$ acts as a binder for $a$ in $M$.
- Only variables can be substituted for terms, while addresses are substituted for more complicated objects, to be defined later.
The $\lambda\mu$-Calculus

- Environments now assigns types to variables and *negated* types to addresses.
- Typing rules are as follows:

\[
\begin{align*}
\Gamma, x : \varphi & \vdash x : \varphi \\
\Gamma, x : \varphi & \vdash M : \tau \\
\Gamma & \vdash \lambda x M : \varphi \to \tau \\
\Gamma, a : \neg \varphi & \vdash M : \bot \\
\Gamma & \vdash \mu a : \neg \varphi M : \varphi \\
\Gamma, a : \neg \varphi & \vdash [a]M : \bot
\end{align*}
\]
The $\lambda\mu$-Calculus: Reduction

- $\lambda\mu$-contexts are defined as follows:

$$C ::= [\cdot] \mid CM \mid [a]C$$

- Given a $\lambda\mu$-context $C$ and a $\lambda\mu$-term $M$, the term $C[M]$ is defined as follows:

$$[\cdot][M] = M;$$
$$((CN)[M] = (C[M])N;$$
$$([a]C)[M] = [a](C[M]).$$

- The concept of free variable set is generalized to contexts as follows:

$$FV([\cdot]) = \emptyset;$$
$$FV(CM) = FV(C) \cup FV(M);$$
$$FV([a]C) = \{a\} \cup FV(C).$$
The $\lambda\mu$-Calculus: Reduction

For a $\lambda\mu$-context $C$, an address $a$, and a $\lambda\mu$-term $M$, we define the substitution $M[a := C]$ as follows:

\[
x[a := C] = x
\]
\[
(\lambda y : \varphi M)[a := C] = \lambda y : \varphi(M[a := C])
\]
\[
(MN)[a := C] = (M[a := C])(N[a := C])
\]
\[
(\mu b : \neg \varphi M)[a := C] = \mu b : \neg \varphi(M[a := C])
\]
\[
([a]M)[a := C] = C[M[a := C]]
\]
\[
([b]M)[a := C] = [b](M[a := C])
\]

where $y, b \notin FV(C)$ and $a \neq b$.

The relation $\rightarrow_\mu$ denotes the least compatible relation containing all the following pairs:

\[
(\lambda x : \varphi M)N \rightarrow_\mu M[x := N];
\]
\[
\mu a : \neg \varphi[a]M \rightarrow_\mu M \text{ if } a \notin FV(M);
\]
\[
[b](\mu a : \neg \varphi M) \rightarrow_\mu M[a := ([b][\cdot])] ; 
\]
\[
(\mu a : \neg (\varphi \rightarrow \tau)M)N \rightarrow_\mu \mu b : \neg \tau(M[a := ([b](\cdot)N)) \text{ if } a \neq b \notin FV(MN)
\]
Theorem (Subject Reduction Theorem)

If \( \Gamma \vdash M : \varphi \) and \( M \xrightarrow{\mu}^* N \), then \( \Gamma \vdash N : \varphi \).

Theorem (Strong Normalization)

The relation \( \xrightarrow{\mu} \) on \( \lambda\mu \)-terms is strongly normalizing.

Theorem (Confluence)

The relation \( \xrightarrow{\mu} \) on \( \lambda\mu \)-terms is weakly Church-Rosser, thus Church-Rosser.
Section 2

Gödel’s T and Arithmetic
It can be seen as an extension of the simply-typed λ-calculus natural numbers and a form of primitive recursion are directly available, rather than encoded.

Types are extended as follows:

\[ \varphi ::= \cdots \mid \text{int} \mid \text{unit} \]

Terms are extended as follows:

\[ M ::= \cdots \mid R \mid X \mid Z \mid * \]

where the new constants can be typed by the following rules:

\[ \Gamma \vdash R : \text{int} \rightarrow \varphi \rightarrow (\text{int} \rightarrow \varphi \rightarrow \varphi) \rightarrow \varphi \]

\[ \Gamma \vdash X : \text{int} \rightarrow \text{int} \]

\[ \Gamma \vdash Z : \text{int} \]

\[ \Gamma \vdash * : \text{unit} \]
Gödel’s T: Dynamics

- Two new reduction rules are necessary:

\[ R \, Z \, M \, N \rightarrow_{\beta} M; \]
\[ R \,(X\,L) \, M \, N \rightarrow_{\beta} N \,(L \,(R \, L \, M \, N)) \]

Theorem (Strong Normalization)

The relation \( \rightarrow_{\beta} \) on terms of Gödel’s T is strongly normalizing.

- A function \( f : \mathbb{N}^k \rightarrow \mathbb{N} \) is definable in T if there is a term \( M_f \) with \( \emptyset \vdash M_f : \text{int}^k \rightarrow \text{int} \) such that

\[ M_f \bar{n}_1 \cdots \bar{n}_k \rightarrow_{\beta} f(n_1, \ldots, n_k) \]

where \( \bar{n} \) is the encoding of \( n \) via \( Z \) and \( X \).

Theorem

The class of total functions on \( \mathbb{N} \) which are \( \lambda \)-definable in T coincides with those which are provably total in Peano’s arithmetic.
Section 3

Second-Order Logic and Polymorphism
Formulas are the expressions derived from the following grammar:

\[ \varphi ::= \bot \mid p \mid \varphi \to \varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \]

\[ \mid \exists p \varphi \mid \forall p \varphi \]

Usual concepts, like those of \( \alpha \)-conversion, free variables, and the like are generalized easily to this new setting.

Classically, the following two logical equivalences hold:

\[ \exists p \varphi \iff \varphi[p := \lnot \bot] \lor \varphi[p := \bot] \]

\[ \forall p \varphi \iff \varphi[p := \lnot \bot] \land \varphi[p := \bot] \]

where \( \lnot \varphi := \varphi \to \bot \).

Intuitionistically, the same does not hold.
Second-Order Propositional Logic

- Getting a natural deduction system for second-order propositional logic boils down to extending natural deduction for propositional logic with the following four rules:

\[
\frac{\Gamma \vdash \varphi[p := \tau]}{\Gamma \vdash \exists p \varphi} \quad I\exists \\
\quad \frac{\Gamma \vdash \exists p \varphi \quad \Gamma, \varphi \vdash \tau \quad p \notin FV(\Gamma, \tau)}{\Gamma \vdash \tau} \quad E\exists
\]

\[
\frac{\Gamma \vdash \varphi \quad p \notin FV(\Gamma)}{\Gamma \vdash \forall p \varphi} \quad I\forall \\
\quad \frac{\Gamma \vdash \forall p \varphi}{\Gamma \vdash \varphi[p := \tau]} \quad E\forall
\]

- Both Heyting and Kripke Semantics can be generalized to second-order quantifiers.

Theorem (Completeness)
\[\Gamma \vdash \varphi \text{ if and only if } \Gamma \models \varphi \text{ if and only if } \Gamma \vDash \varphi.\]
The Polymorphic $\lambda$-Calculus

- Types are taken as second-order propositional formulas including only the operators $\to$ and $\forall$.
- Terms are generated by the following grammar:

  \[
  M ::= x \mid \lambda x : \phi M \mid (MM) \mid \Lambda pM \mid M\phi
  \]

  $\Lambda pM$ binds $p$ in $M$.
- Typing rules are the following ones:

  \[
  \Gamma, x : \phi \vdash x : \phi \quad V
  \]

  \[
  \Gamma, x : \phi \vdash M : \tau \quad \lambda \\
  \frac{\Gamma \vdash \lambda x M : \phi \to \tau}{\Gamma \vdash \lambda xM : \phi \to \tau}
  \]

  \[
  \frac{\Gamma \vdash M : \phi \to \tau \quad \Gamma \vdash N : \phi}{\Gamma \vdash MN : \tau} \quad \@ 
  \]

  \[
  \frac{\Gamma \vdash M : \phi \quad p \notin FV(\Gamma)}{\Gamma \vdash \Lambda pM : \forall p\phi} \quad \Lambda \\
  \frac{\Gamma \vdash M : \forall p\phi}{\Gamma \vdash (M\tau) : \phi[p := \tau]} \quad \@_{\Lambda}
  \]
The Polymorphic λ-Calculus: Reduction

The relation $\rightarrow_\Lambda$ denotes the least compatible relation such that

$$(\lambda x : \varphi M)N \rightarrow_\Lambda M[x := N];$$

$$(\Lambda p M) \varphi \rightarrow_\Lambda M[p := \varphi].$$

Theorem (Subject Reduction Theorem)
If $\Gamma \vdash M : \varphi$ and $M \rightarrow^*_\Lambda N$, then $\Gamma \vdash N : \varphi$.

Theorem (Strong Normalization)
The relation $\rightarrow_\Lambda$ on λµ-terms is strongly normalizing.

Theorem (Confluence)
The relation $\rightarrow_\Lambda$ on λµ-terms is weakly Church-Rosser, thus Church Rosser.