Rewriting and Termination in Lambda Calculus

OPLSS June 2022

1 Outline

1. Rewriting Basics
2. Lambda Calculus
3. Confluence: Church-Rosser property, local confluence
4. Normalization: strong normalization, normalization
5. Strategies: Left-outermost strategy, perpetual strategies
6. Simple types in lambda calculus: strong normalization
7. Intersection types in lambda calculus: complete characterizations of normalizations

2 Strategies and Combinators Continued - Cleaning up the Morning

In the early explorations of combinators the question was asked whether the S combinator by itself could give rise to infinite reductions. The following example was constructed to resolve this question:

Example 1. *Let $A \equiv SSS$. Then AAA has an infinite head reduction.*

A related combinator, which is much more useful in practice and perhaps the most famous of them all is the Y combinator:

$$Y \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)) \tag{1}$$

This has the property of being a fixpoint combinator, which we've seen before in Gibbon's lectures earlier this week.

2.1 Normalization strategies

A strategy $\rightarrow_n \subseteq \rightarrow$ is normalizing if for every $M \in \bigwedge$ which has a normal form M_n, we have that $M \twoheadrightarrow_n M_n$.

We know that there exists a normalization strategy by the following construction:

Theorem 1. *If M has a normal form M_{nf}, then $M \twoheadrightarrow_l M_{nf}$.*

Proof.

$$\begin{aligned} M \text{ has NF } M_{\text{nf}} &\implies M \twoheadrightarrow M_{\text{nf}} \\ &\implies M \twoheadrightarrow_s M_{\text{nf}} \\ &\implies M \twoheadrightarrow_h P \twoheadrightarrow_i M_{\text{nf}} \end{aligned}$$

Head reductions are always leftmost and the internal reductions are performed from left to right. $\square$

2.2 Simply Typed Lambda Calculus

The simply typed lambda calculus is motivated mainly by two features of untyped lambda calculus that makes writing programs in it quite difficult. The first is that the untyped lambda calculus has terms without a normal form, such as Ω. Second, it has no rules to enforce your constructions. For example, there are no guarantees that your nice Church numerals won't be fed something terrible like AAA. We want to restrict what kind of computations we can perform to some "safe" subset of them.

There are generally two approaches to equipping lambda calculus with type systems: Curry's approach, which implicitly assigns types to terms, and Church's approach, which does so explicitly instead.

The type system can be defined over the signature $(T, \rightarrow)$, where T is some countable set of variables:

$$\tau, \sigma ::= x \in T \mid \tau \rightarrow \sigma \tag{2}$$

where $\rightarrow$ is right associative.

We can define the typing rules of the type system inductively over the terms as well:

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma} \text{Ax}$$

$$\frac{\Gamma \vdash M : \sigma \rightarrow \tau \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau} \rightarrow_{elim}$$

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau} \rightarrow_{intro}$$

This type system is both sound and has both progress and preservation. Additionally, if M is typable in the type system, then M is strongly normalizing.

The simply typed lambda calculus is at the bottom of the "lambda cube" - it can be extended with recursion, dependent types, and more, which preserve type systems and maintain the property that M being typable implies that M is strongly normalizing.

3 Intersection Types

We now add two more constructs to our simple type grammar:

$$\tau, \sigma ::= \tau \in T \,|\, \tau \to \sigma \,|\, \sigma \cap \tau \,|\, \omega \tag{3}$$

Additionally, we to impose a pre-order over the types in order to support a notion of subtyping. The ordering is defined by the following eight axioms:

1. $\sigma \leq \sigma$
2. $\sigma \leq \tau, \tau \leq \rho \implies \sigma \leq \rho$
3. $\sigma \leq \omega$
4. $\sigma \cap \tau \leq \sigma$ and $\sigma \cap \tau \leq \tau$
5. $\sigma \leq \sigma \cap \sigma$
6. If $\sigma \leq \sigma'$ and $\tau \leq \tau'$, then $\sigma \cap \tau \leq \sigma' \cap \tau'$
7. If $\sigma \leq \sigma'$ and $\tau \leq \tau'$, then $\sigma \to \tau \leq \sigma' \to \tau'$
8. $(\sigma \to \rho) \cap (\sigma \to \tau) \leq (\sigma \to \tau \cap \rho)$

For notational purposes, we say that $\sigma \sim \tau$ if $\sigma \leq \tau$ and $\tau \leq \sigma$. This is just the symmetric closure of $\leq$.

The typing rules for this system extends the rules for the simply typed lambda calculus from earlier:

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma}\ \text{Ax}$$

$$\frac{\Gamma \vdash M : \sigma \to \tau \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}\ \to_{elim}$$

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau}\ \to_{intro}$$

$$\frac{\Gamma \vdash M : \tau \cap \sigma}{\Gamma \vdash M : \sigma}\ \cap_{elim2}$$

$$\frac{\Gamma \vdash M : \tau \cap \sigma}{\Gamma \vdash M : \tau}\ \cap_{elim1}$$

$$\frac{\Gamma \vdash M : \tau \qquad \Gamma \vdash M : \sigma}{\Gamma \vdash M : \sigma \cap \tau}\, \tau_{intro}$$

$$\frac{\Gamma \vdash M : \sigma \qquad \sigma \leq \tau}{\Gamma \vdash M : \tau}\, \tau_{intro}$$

Intersection types were developed in the 1980's by Coppo, Dezani, Pottinger, and Sallé. While the $\cap$ construct may resemble intuitionistic conjunction, the type $\sigma \to \tau \to \sigma \cap \tau$ is not inhabited in lambda calculus with intersection types, whereas the corresponding proposition in intuitionistic logic is provable.

However, in spite of this limitation, we do get something exciting: $\lambda x.xx$ is finally typable!

$$\dfrac{\dfrac{\dfrac{x : (\sigma \to \tau) \cap \sigma \vdash x : (\sigma \to \tau) \cap \sigma}{x : (\sigma \to \tau) \cap \sigma \vdash x : \sigma \to \tau}\, \cap_{elim} \qquad \dfrac{x : (\sigma \to \tau) \cap \sigma \vdash x : (\sigma \to \tau) \cap \sigma}{x : (\sigma \to \tau) \cap \sigma \vdash x : \sigma}\, \cap_{elim}}{x : (\sigma \to \tau) \cap \sigma \vdash xx : \tau}\, \to_{elim}}{\vdash \lambda x.xx : ((\sigma \to \tau) \cap \sigma) \to \tau}\, \to_{intro}$$

In fact, we finally get the other direction of an earlier implication:

Theorem 2. *A term M is typable if and only if M is strongly normalizing.*

Proof. Typability $\implies$ SN

- reducibility method
- arithmetic proof
- non-idempotent intersection types

SN $\implies$ Typability

- typability of normal forms
- head subject expansion, perpetual strategies

$\square$

Note that the halting problem implies that now typability is undecidable for this simply typed lambda calculus with intersection types.

While this extension to the type system does cause typing to be equivalent to strong normalization and thus undecidable, we do get the following nice theorems

Theorem 3. *M is normalizing if and only if M is typable, and in the judgement $\Gamma \vdash M : \sigma$, $\omega \notin \sigma$, $\omega \notin \Gamma$*

Theorem 4. *M is head normalizing if and only if M is typable, and in the judgement $\Gamma \vdash M : \sigma$, we have that $\sigma \not\sim \omega$*

Theorem 5. *M is unsolvable if and only if M is typable, and in the judgement $\Gamma \vdash M : \sigma$, we have that $\sigma \sim \omega$.*

Proof. The proof of all three of these theorems are very similar. The forward direction relies on the reducibility method from previous lectures, and the backwards direction relies on reasoning about the various reduction and expansion strategies that we have introduced. ☐

Unfortunately, we do not have proofs of the forward direction for uninterpreted types. This is an open problem!

3.1 Union Types

We can again extend the type system, this time with union types!

$$\tau, \sigma ::= \tau \in T \,|\, \tau \to \sigma \,|\, \sigma \cap \tau \,|\, \sigma \cup \tau \tag{4}$$

We once again need to corresponding introduction and elimination rules from this new construct:

$$\frac{\Gamma \vdash M : \tau}{\Gamma \vdash M : \tau \cup \sigma} \cup_{intro1}$$

$$\frac{\Gamma \vdash M : \sigma}{\Gamma \vdash M : \tau \cup \sigma} \cup_{intro2}$$

$$\frac{\Gamma \vdash P : \sigma \cup \tau \qquad \Gamma, x : \sigma \vdash M : \rho \qquad \Gamma, x : \tau \vdash M : \rho}{\Gamma \vdash M[x := P] : \rho} \cup_{elim}$$