Typed λ-Calculus and Logic

April 11, 2001

CS 131: Programming Languages

Pure Simply-Typed λ-Calculus

- Syntax

```
M,N ::=  x              variables
       |  λx:t.M        functions
       |  M N           applications

t,u ::=  α1 | α2 | ...  type variables
       |  t -> u        function types
```

One-step β-Reduction

- The relation $\rightarrow_\beta$ is defined by:

$$\frac{}{(\lambda x{:}t.M)N \rightarrow_\beta M[x \rightarrow N]}$$

$$\frac{M \rightarrow_\beta M'}{M\ N \rightarrow_\beta M'\ N} \qquad \frac{N \rightarrow_\beta N'}{M\ N \rightarrow_\beta M\ N'}$$

$$\frac{M \rightarrow_\beta M'}{\lambda x{:}t.M \rightarrow_\beta \lambda x{:}t.M'}$$

Extension: Adding Pairs

- Syntax as in NQSML

```
M,N ::=  ...
       |  <M,N>          pairs
       |  fst M | snd M  projections
t,u ::=  ...
       |  t * u          pair types
```

$$\frac{}{\texttt{fst <M,N>} \rightarrow \texttt{M}} \qquad \frac{}{\texttt{snd <M,N>} \rightarrow \texttt{N}}$$

Summary: Static Semantics

$$\frac{}{\Gamma, x{:}t, \Gamma' \vdash x : t}$$

$$\frac{\Gamma, x{:}t \vdash M : u}{\Gamma \vdash (\lambda x{:}t.M) : t \to u} \qquad \frac{\Gamma \vdash M : t \to u \quad \Gamma \vdash N : t}{\Gamma \vdash M\ N : u}$$

$$\frac{\Gamma \vdash M : t \quad \Gamma \vdash N : u}{\Gamma \vdash \langle M, N \rangle : t * u}$$

$$\frac{\Gamma \vdash M : t * u}{\Gamma \vdash \mathrm{fst}\ M : t} \qquad \frac{\Gamma \vdash M : t * u}{\Gamma \vdash \mathrm{snd}\ M : u}$$

Erasing All but the Types

$$\frac{}{\Gamma, t, \Gamma' \vdash t}$$

$$\frac{\Gamma, t \vdash u}{\Gamma \vdash t \to u} \qquad \frac{\Gamma \vdash t \to u \quad \Gamma \vdash t}{\Gamma \vdash u}$$

$$\frac{\Gamma \vdash t \quad \Gamma \vdash u}{\Gamma \vdash t * u}$$

$$\frac{\Gamma \vdash t * u}{\Gamma \vdash t} \qquad \frac{\Gamma \vdash t * u}{\Gamma \vdash u}$$

Rules for Propositional Logic

$$\frac{}{\Gamma, p, \Gamma' \vdash p}$$

$$\frac{\Gamma, p \vdash q}{\Gamma \vdash p \Rightarrow q} \qquad \frac{\Gamma \vdash p \Rightarrow q \quad \Gamma \vdash p}{\Gamma \vdash q}$$

$$\frac{\Gamma \vdash p \quad \Gamma \vdash q}{\Gamma \vdash p \wedge q}$$

$$\frac{\Gamma \vdash p \wedge q}{\Gamma \vdash p} \qquad \frac{\Gamma \vdash p \wedge q}{\Gamma \vdash q}$$

Curry-Howard Isomorphism

- a.k.a. "Proofs as programs", "Propositions as types"
- Every type corresponds to a logical proposition
  - Type variables correspond to propositional variables
  - Function types correspond to implications
  - Pair types correspond to conjunctions
- A proposition is provable if and only if there is a term of the corresponding type
  - Such types are said to be *inhabited*.
  - Typed **λ**-terms are encodings of proofs.

Examples

1. Show that

   $\vdash p \Rightarrow (p \wedge p)$

   by finding a term of type

   α -> (α * α)

2. Show that

   $\vdash (p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \wedge q) \Rightarrow r))$

   by finding a term of type

   (α->(β->γ)) -> ((α*β)->γ)

Extensions

- The `true` proposition corresponds to any non-empty type
  - E.g., `unit`.
- The `false` proposition corresponds to an empty type.
  - Usually called `void`.
  - Encode $\neg p$ as $(p \Rightarrow$`false`$)$.
- Second-order predicate calculus: polymorphic types
  - Second-order = quantifying over *propositions*.
  - E.g., $\forall\alpha.\ \alpha \rightarrow (\alpha * \alpha)$ vs. $\forall p.\ p \rightarrow (p \wedge p)$
- Disjunctions: sum types
- Propositional logic: dependent types
- Modal logic: run-time code generation
- Linear logic: linear types

Intuitionism

- Generally, λ-calculi correspond to *constructive* or *intuitionistic* logics.
  - E.g., no terms of type

    $\forall\alpha.\ \alpha$ + (α->void) *($\forall p.\ p$ or $\neg p$)*

    $\forall\alpha.$ ((α->void)->void) -> α *($\forall p.\ \neg\neg p \Rightarrow p$)*

    $\forall\alpha.\forall\beta.$(($\alpha$->$\beta$)->$\alpha$) -> α *(Pierce's Law)*

- However, there has been some work on extending the isomorphism to classical logics
  - Corresponds to calculi with `callcc`-like operators

Proof Normalization

- Reductions as proof "simplifications".

$$\dfrac{\dfrac{\Gamma \vdash M : t \quad \Gamma \vdash N : u}{\Gamma \vdash \langle M,N\rangle : t*u}}{\Gamma \vdash \mathtt{fst}\ \langle M,N\rangle : t} \qquad \Gamma \vdash M : t$$

$$\dfrac{\dfrac{\Gamma \vdash t \quad \Gamma \vdash u}{\Gamma \vdash t \wedge u}}{\Gamma \vdash t} \qquad \Gamma \vdash t$$

Hilbert System for Implication

Axiom Schema

$$\frac{}{p \Rightarrow (q \Rightarrow p)}$$

$$\frac{}{(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))}$$

Modus Ponens

$$\frac{p \Rightarrow q \quad p}{q}$$

Summary

λ-Calculus		Logic
Type	⟷	Proposition
Term (program)	⟷	Proof
Reduction	⟷	Proof Normalization