Computational Logic

Interactive Theorem Proving

Week 4

Cezary Kaliszyk

April 12, 2013

Summary

So far

Proof Assistants, HOL Light, $\lambda_{\rightarrow}$

- Gentzen Style Natural Deduction
- Principal Types
- TCP, TSP, TIP
- HOL-Light subgoal package

Today

- Properties of $\lambda_{\rightarrow}$
- BHK interpretation and λ-cube again
- Dependent types
- λ_P

Properties of $\lambda_{\to}$

- Uniqueness of Types

$$\text{If } \Gamma \vdash M : \sigma \text{ and } \Gamma \vdash M : \tau \text{ , then } \sigma = \tau.$$

- Subject Reduction

$$\text{If } \Gamma \vdash M : \sigma \text{ and } M \to_{\beta\eta} N \text{ , then } \Gamma \vdash N : \sigma.$$

- Strong Normalization

$$\text{If } \Gamma \vdash M : \sigma, \text{ then all } \beta\eta\text{-reductions from } M \text{ terminate.}$$

- Substitution Property

$$\text{If } \Gamma, x : \tau, \Delta \vdash M : \sigma, \Gamma \vdash P : \tau, \text{ then } \Gamma, \Delta \vdash M[x := P] : \sigma.$$

- Thinning

$$\text{If } \Gamma \vdash M : \sigma \text{ and } \Gamma \subset \Delta, \text{ then } \Delta \vdash M : \sigma.$$

- Strengthening

$$\text{If } \Gamma, x : \tau \vdash M : \sigma \text{ and } x \notin FV(M), \text{ then } \Gamma \vdash M : \sigma.$$

Intuitionistic Logic

Drawbacks of classical logic

- There are $x \notin \mathbb{Q}$ and $y \notin \mathbb{Q}$ st. $x^y \in \mathbb{Q}$.
  - Proof: by cases $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$
- There are seven 7s in a row in the decimal representation of π.

Brouwer, beginning of 20th century

Intuitionistic logic developed later around 1930

- $A \to \neg\neg A$ has an intuitionistic interpretation
- but $\neg\neg A \to A$ does not

Easier correspondence to $\lambda_?$-calculi

Constructive proofs have computational content

Brouwer-Heyting-Kolmogorov interpretation

Proof of $A \to B$

Function that maps proofs of A to proofs B

Proof of $A \wedge B$

Pair of proofs of A and B

Proof of $A \vee B$

Either a proof of A or a proof of B

Proof of $\forall x.P(x)$

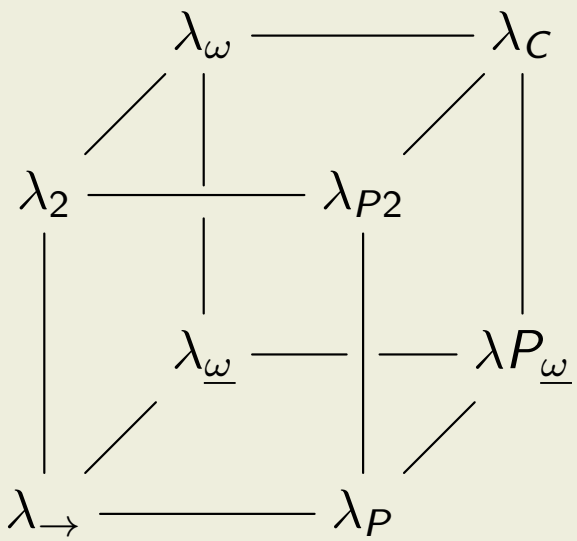
Function that maps an object x to a proof of $P(x)$

Proof of $\bot$

Does not exist.Negation of A turns a proof of A into a nonexistant object

Lambda cube (again)

Four kinds of dependencies



- terms on terms (already in $\lambda_\to$)
- dependent types (λ_P)
- polymorphism (λ_2)
- terms depend on types
- Combining the three is hard! (Girard's paradox)

Dependent types vs Polymorphism vs CoC

Printf

What type does it have?

Bit-strings of length n

- Type of bit-strings: $bs : \mathbb{N} \to \star$
- Bit-string made of zeros: $0_{bs} : (\forall n : \mathbb{N}) bs(n)$
- $\mathbb{R}^{\mathbb{N}}$

Vectors (later polymorphic)

- Type of hd?

Constructive Division

$$a/b/\!\!/P \qquad \approx \qquad \frac{a}{b \neq 0} \quad .$$

Intuition behind λ_P

functions from A to B

$$A \to B$$

dependent functions from A to B

$$\Pi x : A.B$$

- Also called: dependent product
- Type of B can now depend on the argument x
- arrow type becomes a special case of dependent product

Three kinds of judgements

Kind formation judgements

$$\Gamma \vdash k : \Box$$

Kinding judgements

$$\Gamma \vdash \varphi : k$$

Typing judgements

$$\Gamma \vdash M : \tau$$

The meaning of $k : \Box$ is that k is a well-formed kind.

Syntax of λ_P

- variables

$$x, y, z, \ldots$$

- abstraction

$$\lambda x : M.N$$

- function application

$$MN$$

- dependent product

$$\Pi x : M.N \text{ (sometimes } \forall x : M.N)$$

- two sorts

$$\star, \Box$$

Abbreviations

If x is not free in k

We write $\tau \Rightarrow k$ instead of $(\Pi x : \tau)k$.

If x is not free is σ

We write $\tau \rightarrow \sigma$ instead of $(\forall x : \tau)\sigma$.

β-reduction in λ_P

- Like in $\lambda_\to$

$$(\lambda x : \tau . M) N \to_\beta M[x := N]$$

- Under lambda and application
- In the type of a λ-expression or Π-expression
- In a type application or under a Π

Rules of λ_P (1/3)

Axiom rule

$$\frac{}{\vdash \star : \Box}$$

Variable rule

$$\frac{\Gamma \vdash A : \{\star, \Box\}}{\Gamma, x : A \vdash x : A}$$

Weakening rule

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash C : \{\star, \Box\}}{\Gamma, x : C \vdash A : B}$$

Rules of λ_P (2/3)

Dependent product rule

$$\frac{\Gamma \vdash A : \star \qquad \Gamma, x : A \vdash B : \{\star, \Box\}}{\Gamma \vdash \Pi x : A.\, B : \{\star, \Box\}}$$

Abstraction rule

$$\frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash \Pi x : A.\, B : \{\star, \Box\}}{\Gamma \vdash \lambda x : A.\, M : \Pi x : A.\, B}$$

Application Rule

$$\frac{\Gamma \vdash M : \Pi x : A.\, B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x := N]}$$

Rules of λ_P (3/3)

Conversion Rule

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : \{\star, \Box\}}{\Gamma \vdash A : B'} \quad \text{where } B =_\beta B'$$

Summary

Today

- Properties of $\lambda_{\rightarrow}$
- BHK interpretation and λ-cube again
- Dependent types
- λ_P

Next time

- Polymorphism
- More advanced features in HOL-Light