Computational Logic

Interactive Theorem Proving

Week 3

Cezary Kaliszyk

March 22, 2013

Summary

So far

Proof Assistants
HOL Light

- $\lambda_{\rightarrow}$, STT
- à la Church, à la Curry
- Type Assignment
- Curry-Howard Isomorphism and example derivations

Today

- Gentzen Style Natural Deduction
- Principal Types
- TCP, TSP, TIP
- Lambda Cube

Gentzen style natural deduction

assumption

$$\frac{\vdots}{A} \quad \rightarrow \quad [A]^{\mathrm{H}}$$

conjunction introduction

$$\frac{\vdots}{A \wedge B} \quad \rightarrow \quad \frac{\begin{array}{cc}\vdots & \vdots \\ A & B\end{array}}{A \wedge B}\,\wedge\mathrm{i}$$

Gentzen style natural deduction

conjunction elimination left

$$\begin{array}{c}\vdots \\ A\end{array} \quad \rightarrow \quad \dfrac{\begin{array}{c}\vdots \\ A \wedge B\end{array}}{A}\ \wedge e_1$$

conjunction elimination right

$$\begin{array}{c}\vdots \\ B\end{array} \quad \rightarrow \quad \dfrac{\begin{array}{c}\vdots \\ A \wedge B\end{array}}{B}\ \wedge e_2$$

Gentzen style natural deduction

disjunction introduction left

$$\frac{\vdots}{A \vee B} \quad \rightarrow \quad \frac{\dfrac{\vdots}{A}}{A \vee B} \vee \mathrm{i}_1$$

disjunction introduction right

$$\frac{\vdots}{A \vee B} \quad \rightarrow \quad \frac{\dfrac{\vdots}{B}}{A \vee B} \vee \mathrm{i}_2$$

Gentzen style natural deduction

disjunction elimination

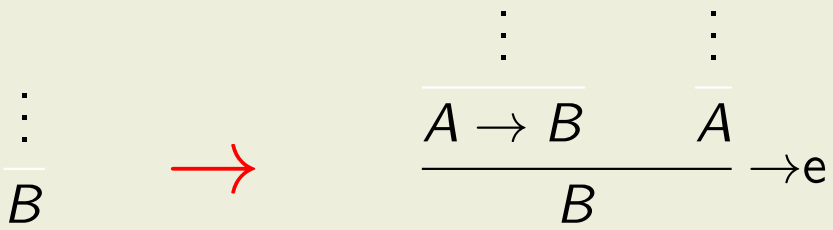
$$\dfrac{\vdots}{C} \quad \rightarrow \quad \dfrac{\begin{array}{ccc} \vdots & [A]^{\mathtt{H1}} & [B]^{\mathtt{H2}} \\ \vdots & \vdots & \vdots \\ A \vee B & C & C \end{array}}{C} \vee\mathsf{e}\,[\mathtt{H1},\mathtt{H2}]$$

implication introduction

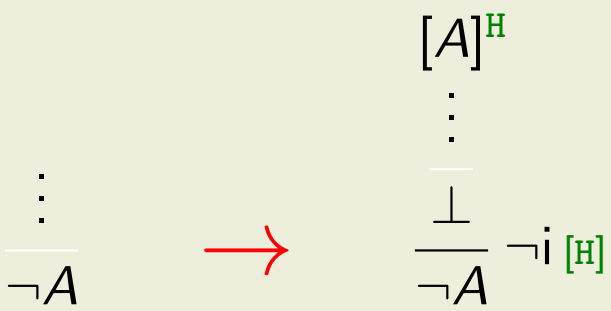
$$\dfrac{\vdots}{A \rightarrow B} \quad \rightarrow \quad \dfrac{\begin{array}{c} [A]^{\mathtt{H}} \\ \vdots \\ B \end{array}}{A \rightarrow B} \rightarrow\mathsf{i}\,[\mathtt{H}]$$

Gentzen style natural deduction

implication elimination

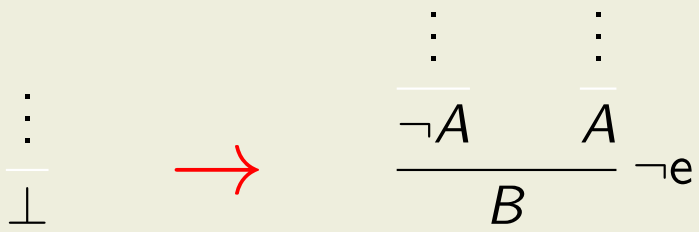


negation introduction



Gentzen style natural deduction

negation elimination



bottom elimination

Gentzen style natural deduction

universal introduction

$$\dfrac{\vdots}{\forall x\, A} \quad \longrightarrow \quad \dfrac{\dfrac{\vdots}{A[y/x]}}{\forall x\, A}\,\forall \mathrm{i}$$

universal elimination

$$\dfrac{\vdots}{A[t/x]} \quad \longrightarrow \quad \dfrac{\dfrac{\vdots}{\forall x\, A}}{A[t/x]}\,\forall \mathrm{e}$$

Gentzen style natural deduction

existential introduction

$$\frac{\vdots}{\exists x\, A} \quad \rightarrow \quad \frac{\begin{array}{c}\vdots\\ A[t/x]\end{array}}{\exists x\, A}\,\exists\mathrm{i}$$

existential elimination

$$\frac{\vdots}{B} \quad \rightarrow \quad \frac{\exists x\, A \qquad \begin{array}{c}[A[y/x]]^{\mathtt{H}}\\ \vdots\\ B\end{array}}{B}\,\exists\mathrm{e}\,[\mathtt{H}]$$

Example Derivation

$$\dfrac{\dfrac{[\exists x(P(x) \vee \neg Q(a))]^{H1} \qquad \dfrac{[P(b) \vee \neg Q(a)]^{H3} \qquad \dfrac{\dfrac{[P(b)]^{H4}}{\exists x\, P(x)}\,\exists i \qquad \dfrac{\dfrac{[\neg Q(a)]^{H5} \quad [Q(a)]^{H2}}{\bot}\,\neg e}{\exists x\, P(x)}\,\bot e}{\exists x\, P(x)}\,\vee e\ [H4,H5]}{\exists x\, P(x)}\,\exists e\ [H3]}{Q(a) \to \exists x\, P(x)}\,\to i\ [H2]}{\exists x(P(x) \vee \neg Q(a)) \to Q(a) \to \exists x\, P(x)}\,\to i\ [H1]$$

Corresponding Box-style Proof

1		$\exists x(P(x) \lor \neg Q(a))$	assumption
2		$Q(a)$	assumption
3	b	$P(b) \lor \neg Q(a)$	assumption
4		$P(b)$	assumption
5		$\exists x\, P(x)$	$\exists$i 4
6		$\neg Q(a)$	assumption
7		$\bot$	$\neg$e 6,2
8		$\exists x\, P(x)$	$\bot$e 7
9		$\exists x\, P(x)$	$\lor$e 3,4—5,6—8
10		$\exists x\, P(x)$	$\exists$e 1,3—9
11		$Q(a) \to \exists x\, P(x)$	$\to$i 2—10
12		$\exists x(P(x) \lor \neg Q(a)) \to Q(a) \to \exists x\, P(x)$	$\to$i 1—11

Definitions

Definition: type substitution

A map from type variables to types

Definition: unifier

For two given types σ and τ their unifier, is such a type substitution s that $s(\sigma) = s(\tau)$

Definition: mgu (most general unifier)

Given types σ and τ their mgu is a type substitution s, such that:

- $s(\sigma) = s(\tau)$
- $\forall t.t(\sigma) = t(\tau) \rightarrow \exists r.t = r \circ s$

Definitions

Definition: type substitution

A map from type variables to types

Definition: unifier

For two given types σ and τ their unifier, is such a type substitution s that $s(\sigma) = s(\tau)$

Definition: mgu (most general unifier)

Given types σ and τ their mgu is a type substitution s, such that:

- $s(\sigma) = s(\tau)$
- $\forall t.t(\sigma) = t(\tau) \rightarrow \exists r.t = r \circ s$

The above notions generalize to lists of types

Algorithm computing mgu

In $\lambda_{\to}$ only one function

Input: $\sigma_1, \ldots, \sigma_n$, output: mgu or "not unifiable".

$$\frac{E_1; g(\tau_1, \ldots, \tau_n) \approx g(\tau'_1, \ldots, \tau'_n); E_2}{E_1; \tau_1 \approx \tau'_1; \ldots; \tau_n \approx \tau'_n; E_2}\, d_1$$

$$\frac{E_1; \tau_1 \to \tau_2 \approx \tau'_1 \to \tau'_2; E_2}{E_1; \tau_1 \approx \tau'_1; \tau_2 \approx \tau'_2; E_2}\, d_2$$

$$\frac{E_1; \alpha \approx \tau; E_2 \quad \alpha \notin V(\tau)}{(E_1; E_2)\{\alpha/\tau\}}\, v_1$$

$$\frac{E_1; \tau \approx \alpha; E_2 \quad \alpha \notin V(\tau)}{(E_1; E_2)\{\alpha/\tau\}}\, v_2$$

$$\frac{E_1; \tau \approx \tau; E_2}{E_1; E_2}\, t$$

Principal Types

Definition: Principal type

σ is a principal type for an untyped λ-term M if:

- $M : \sigma$ in STT à la Curry
- $\forall \tau, M : \tau \rightarrow \exists s.\tau = s(\sigma)$

Principal Types: example

$$\lambda x^{\alpha}.\lambda y^{\beta}.y^{\beta}(\lambda z^{\gamma}.y^{\beta}x^{\alpha})$$

1. Assign type variables to all variables: $x : \alpha, y : \beta, z : \gamma$.
2. Assign type variables to all applicative subterms: $y\,x : \delta$, $y(\lambda z.y\,x) : \epsilon$.
3. Generate equations between types, necessary for the term to be typable:
$$\beta = \alpha \to \delta \quad \beta = (\gamma \to \delta) \to \epsilon$$
4. Find a most general unifier that solves the above equations:
$$\alpha := \gamma \to \delta, \beta := (\gamma \to \delta) \to \epsilon, \delta := \epsilon$$
5. The principal type of $\lambda x.\lambda y.y(\lambda z.xy)$ is now:
$$(\gamma \to \epsilon) \to ((\gamma \to \epsilon) \to \epsilon) \to \epsilon$$

Typical questions in Type Theory

TCP (type checking problem)

$M : \sigma$?

TSP (type synthesis problem)

M :?

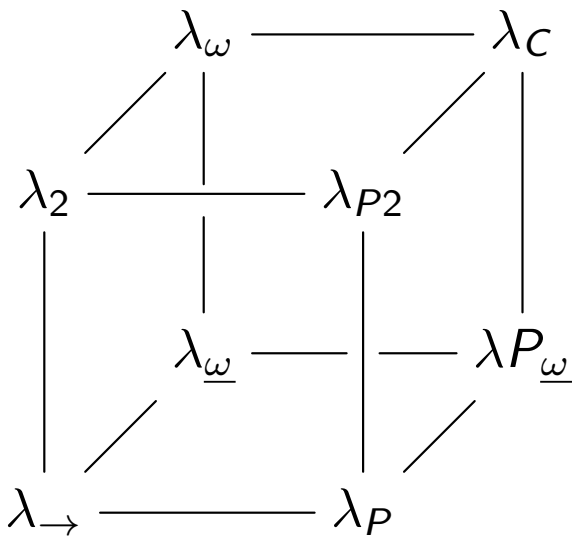
TIP (type inhabitation problem)

? : σ (by a closed term)

- For $\lambda_{\rightarrow}$ all are decidable
  - both in Curry and in Church style
- TCP and TSP are usually equivalent
  - application typing rule is to blame
- TCP and TSP quickly become undecidable in Curry style
  - TIP corresponds to provability in some logic

Lambda cube (Barendregt, 1991)

propositional logic	$\longleftrightarrow$	$\lambda_{\rightarrow}$
predicate logic	$\longleftrightarrow$	λ_P (dependent types)
2nd order propositional logic	$\longleftrightarrow$	λ_2, System F (2nd order typed λ-calc)
		$\lambda_{\underline{\omega}}$ (type operators)



Summary

Today

- Gentzen Style Natural Deduction
- Principal Types
- TCP, TSP, TIP
- Lambda Cube

Next time

- Polymorphism
- HOL Light subgoal package