

Seminarium: Programowanie w teorii typów

Teoria typów

Wojciech Jedynek, Paweł Wieczorek

Instytut Informatyki Uniwersytetu Wrocławskiego

3 października 2011

Plan

- Przypomnimy sobie ideę konstruktywizmu oraz izomorfizmu Curry'ego-Howard'a.
- Zapoznamy się z podstawami teorii typów Martina-Löf'a:
 - ▶ Zapoznamy się z sądami w tym systemie.
 - ▶ Sformułujemy podstawowe typy jak liczby naturalne czy *typy wyliczeniowe*.
 - ▶ Przebrniemy przez formalne definicje podstawowych typów zależnych: Π -typ, Σ -typ
 - ▶ Zobaczymy jak teoria typów koduje logikę pierwszego rzędu.
 - ▶ Zapoznamy się z równością propozycyjną (typem identycznościowym).

Matematyka konstruktywna

- powstały na początku poprzedniego wieku pogląd na temat fundamentów matematyki
- L.E.J.Brouwer, twórca ideologii
- empiryczna zawartość twierdzeń matematycznych
- co znaczy orzeczenie o istnieniu pewnego obiektu?
- odrzucenie dowodów przez sprowadzenie do sprzeczności
- odrzucenie idealistycznego podejścia do prawdziwości orzeczeń
- E. Bishop, konstruktywna analiza matematyczna

Książkowy przykład twierdzenia niekonstruktywnego

Twierdzenie

Istnieją takie dwie liczby niewymierne a oraz b , że a^b jest liczbą wymierną.

Dowód.

Orzeczenie, że $\sqrt{2}^{\sqrt{2}} \in \mathbf{Q}$ musi być prawdziwe lub musi być fałszywe.

- jeżeli jest prawdziwe to mamy szukane a oraz b
- jeżeli jest fałszywe to niech $a = \sqrt{2}^{\sqrt{2}}$ oraz $b = \sqrt{2}$, wtedy $a^b = 2$



Jedyne co wiemy, to to że muszą istnieć takie liczby.

Kolejny przykład.

Twierdzenie (klasycznie)

Jeżeli funkcja f jest ciągła na przedziale $[0, 1]$ oraz wartości funkcji na krańcach przedziału mają różne znaki to istnieje punkt w tym przedziale na którym funkcja się zeruje.

Twierdzenie (konstruktywnie)

Jeżeli funkcja f jest ciągła na przedziale $[0, 1]$ oraz wartości funkcji na krańcach przedziału mają różne znaki to dla każdego $\epsilon > 0$ istnieje punkt w tym przedziale na którym bezwzględna wartość funkcji jest mniejsza od ϵ .

Interpretacja Brouwer-Heyting-Kołmogorow

- $A \wedge B$ to konstrukcja składająca się z dwóch pod-konstrukcji
- $A \vee B$ to konstrukcja składająca się z lewej lub prawej pod-konstrukcji
- $A \rightarrow B$ to metoda przekształcająca konstrukcję B mając do dyspozycji A
- \perp absurd, konstrukcja której nie można zrealizować
- $\forall x.P(x)$ to metoda przekształcająca wartość a w konstrukcję $P(a)$
- $\exists x.P(x)$ to konstrukcja mająca składać się ze świadka a oraz z konstrukcji $P(a)$
- $\neg A$ to skrót od $A \rightarrow \perp$
- Czy przy tej interpretacji wszystkie klasyczne prawa mają sens?
 - ▶ $\exists x P(x) \vee \neg \exists x P(x)$
 - ▶ $\exists x P(x) \equiv \neg(\forall x \neg P(x))$
 - ▶ $A \vee B \equiv \neg(\neg A \wedge \neg B)$

System naturalnej dedukcji

- system dowodzenia
- posługujemy się sędami $\Gamma \vdash \varphi$
- dowód to wyprowadzenie o strukturze drzewa
 - ▶ korzeń - wniosek (sąd)
 - ▶ węzeł - reguła wnioskowania
 - ▶ liść - aksjomat
- reguły wprowadzania i eliminacji spójników logicznych

System naturalnej dedukcji

$$I_{\wedge} \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \quad E_{\wedge 1} \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}$$

$$I_{\vee 1} \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \quad E_{\vee} \frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}$$

$$I_{\rightarrow} \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \quad E_{\rightarrow} \frac{\Gamma \vdash A \rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}$$

$$AX \frac{}{\Gamma, A \vdash A}$$

Izomorfizm Curry-Howard (proposition as types)

- Martin-Löf, Curry, Howard, deBruijn i wiele innych
- typy oznaczają formuły
- otypowane termy oznaczają dowody dla swoich typów (formuł)
- izomorfizm pomiędzy wyprowadzeniami formuł w logice intuicjonistycznej a sędami w systemie typów
- realizacja BHK

$$\begin{array}{c} \text{I} \rightarrow \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x : A. t : A \rightarrow B} \quad \text{E} \rightarrow \frac{\Gamma \vdash f : A \rightarrow B \quad \Gamma \vdash x : A}{\Gamma \vdash f x : B} \end{array}$$

System typów dla λ -rachunku

$$I_{\wedge} \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash (M, N) : A \wedge B} \quad E_{\wedge_1} \frac{\Gamma \vdash M : A \wedge B}{\Gamma \vdash \text{fst } M : A}$$

$$I_{\vee_1} \frac{\Gamma \vdash M : A}{\Gamma \vdash \text{inl } M : A \vee B}$$

$$E_{\vee} \frac{\Gamma \vdash M : A \vee B \quad \Gamma, x : A \vdash P : C \quad \Gamma, x : B \vdash Q : C}{\Gamma \vdash \text{when } M (\lambda x. P) (\lambda x. Q) : C}$$

$$I_{\rightarrow} \frac{\Gamma, A \vdash M : B}{\Gamma \vdash \lambda x. M : A \rightarrow B} \quad E_{\rightarrow} \frac{\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash N : A}{\Gamma \vdash M N : B}$$

$$A_X \frac{}{\Gamma, x : A \vdash x : A}$$

Teoria typów: historia, idee, początki

- Teoria typów jako logika matematyczna - B. Russel

$$A = \{w \mid w \notin w\}$$

- λ -rachunek , funkcja jako pojęcie pierwotne - A.Church
- System typów, likwidacja paradoksu Kleene'go

$$K = \lambda x. \neg(x\ x)$$

$$(K\ K) = \neg(K\ K) = \neg\neg(K\ K) = \dots$$

Teoria typów Martina-Löf'a

- Per Martin-Löf, A Theory of Types, 1972
- Per Martin-Löf, Constructive Mathematics and Computer Programming, 1979
- Per Martin-Löf, **Intuitionistic Type Theory**, 1980
- Per Martin-Löf, Truth of a Proposition, Evidence of a Judgement, Validity of a Proof, 1985
- B. Nordström, K. Petersson, and Jan M. Smith, **Programming in Martin-Löf's Type Theory: An Introduction**, 1990

Wyrażenia jakimi się posługujemy

- język wyrażeń:
 - ▶ $e(e')$ - aplikacja ($e(e_1, \dots, e_n)$ skrócony zapis $e(e_1) \dots (e_n)$)
 - ▶ $(x)e$ - abstrakcja
 - ▶ stałe, np λ , Π , *apply*, 0, *succ*
- wyrażeniom przypisujemy arność:
 - ▶ x

Wyrażenia jakimi się posługujemy

- Pozwalamy na definiowanie nowych stałych jako makra, np

$$\textit{double} \equiv (n)(n + n)$$

$$\textit{double}(n) \equiv n + n$$

- Równość definicyjna
 - ▶ relacja równoważności, kongruencja
 - ▶ surowa równość na wyrażeniach
 - ▶ nic nie mówi o znaczeniu
 - ▶ elementy równe definicyjne traktujemy jako synonimy

Wzbogacony system typów o więcej sądów

- Teoria typów Martina Löf'a - system typów dla λ -rachunku, w którym możemy wydawać różne sądy:
 - ▶ A set - jest zbiorem
 - ▶ $a \in A$ - a jest elementem zbioru
 - ▶ $a =_A b \in A$ - a oraz b są równymi elementami w zbiorze A
 - ▶ $A = B$ - A oraz B są równymi zbiorami
- elementy zbiorów dzielimy na
 - ▶ kanoniczne - wartości (postać normalna)
 - ▶ niekanoniczne - obliczenia
- sformułowanie zbioru to
 - ▶ określenie kanonicznych elementów jakie ten zbiór zawiera
 - ▶ określenie co znaczy że dwa elementy są równe w tym zbiorze
 - ▶ określenie obliczeń

Sądy mają więcej interpretacji

- A set - jest zbiorem
 - ▶ $a \in A$ - a jest elementem zbioru
- A set - jest problemem, zagadnieniem, zadaniem
 - ▶ $a \in A$ - a jest rozwiązaniem problemu A
- A prop - jest formułą logiczną
 - ▶ $a \in A$ - a jest dowodem propozycji A
 - ▶ A true - umiemy zrealizować A , istnieje dowód A
- A set - jest specyfikacją
 - ▶ $a \in A$ - a jest programem spełniającym specyfikację A

Produkt indeksowanej rodziny zbiorów (w matematyce)

- matematyczna definicja

$$\prod_{x \in A} B_x = \left\{ f : A \rightarrow \bigcup_{x \in A} B_x \mid \forall x \in A. f(x) \in B_x \right\}$$

- zależność przeciwdziedziny od argumentu

$$\text{sort} \in \prod_{xs \in \text{Lists}} \{ys \in \text{Lists} \mid \text{perm}(ys, xs) \wedge \text{sorted}(ys)\}$$

$$\text{perm}(\text{sort}(xs_0), xs_0) \wedge \text{sorted}(\text{sort}(xs_0))$$

- możemy też wyrazić „zwykły” zbiór funkcji, jeżeli $B_x = B$ to

$$\prod_{x \in A} B_x = A \rightarrow B$$

Produkt indeksowanej rodziny zbiorów

$$\frac{A \text{ set} \quad B(x) \text{ set } [x \in A]}{(\prod x \in A) B(x) \text{ set}} \quad \frac{b(x) \in B(x) [x \in A]}{\lambda x. b \in (\prod x \in A) B(x) \text{ set}}$$

$$\frac{A = A' \quad B(x) = B'(x) [x \in A]}{(\prod x \in A) B(x) = (\prod x \in A') B'(x)}$$

$$\frac{b(x) = b'(x) \in B(x) [x \in A]}{\lambda x. b = \lambda x. b' \in (\prod x \in A) B(x) \text{ set}}$$

Produkt indeksowanej rodziny zbiorów

$$\frac{f \in (\prod_{x \in A} B(x)) \quad a \in A}{\text{apply}(f, a) \in B(a)} \qquad \frac{b(x) \in B(x) [x \in A] \quad a \in A}{\text{apply}(\lambda x. b, a) = b(a) \in B(a)}$$

$$\frac{a = a' \in A \quad f = f' \in (\prod_{x \in A} B(x))}{\text{apply}(f, a) = \text{apply}(f', a') \in B(a)}$$

Produkt indeksowanej rodziny zbiorów

$$(\forall x \in A)B(x) \equiv (\prod x \in A)B(x)$$

$$\frac{A \text{ set} \quad B(x) \text{ set } [x \in A]}{(\prod x \in A)B(x) \text{ set}} \Rightarrow \frac{A \text{ set} \quad B(x) \text{ prop } [x \in A]}{(\forall x \in A)B(x) \text{ prop}}$$

$$\frac{b(x) \in B(x) [x \in A]}{\lambda x. b \in (\prod x \in A)B(x) \text{ set}} \Rightarrow \frac{B(x) \text{ true } [x \in A]}{(\forall x \in A)B(x) \text{ true}}$$

$$\frac{f \in (\prod x \in A)B(x) \quad a \in A}{\text{apply}(f, a) \in B(a)} \Rightarrow \frac{(\forall x \in A)B(x) \text{ true} \quad a \in A}{B(a) \text{ true}}$$

Produkt indeksowanej rodziny zbiorów

- Załóżmy że $f \in (\prod X \in A)(\prod y \in B)P(x, y)$, zaprogramujmy *flip* f

$$\frac{f \in (\prod X \in A)(\prod y \in B)P(x, y) \quad x \in A [x \in A]}{\underbrace{\text{apply}(f, x) \in (\prod y \in B)P(x, y) [x \in A]}_D}$$

$$\frac{\frac{\frac{\overbrace{\text{apply}(f, x) \in (\prod y \in B)P(x, y) [x \in A]}^D \quad y \in B [y \in B]}{\text{apply}(\text{apply}(f, x), y) \in P(x, y) [y \in B, x \in A]}}{\lambda x. \text{apply}(\text{apply}(f, x), y) \in (\prod x \in B)P(x, y) [y \in B]}}{\lambda y. \lambda x. \text{apply}(\text{apply}(f, x), y) \in (\prod y \in B)(\prod x \in A)P(x, y)}$$

$$\frac{(\forall x \in A)(\forall y \in B)P(x, y) \text{ true}}{(\forall y \in B)(\forall x \in A)P(x, y) \text{ true}}$$

Produkt indeksowanej rodziny zbiorów

- Jeżeli x nie występuje w B to

$$A \rightarrow B \equiv (\prod_{x \in A} B)$$

$$\begin{array}{c} \frac{A \text{ set} \quad B \text{ set } [x \in A]}{A \rightarrow B \text{ set}} \qquad \frac{b(x) \in B [x \in A]}{\lambda x. b \in A \rightarrow B} \\[2ex] \frac{A \text{ prop} \quad B \text{ prop } [A \text{ true}]}{A \rightarrow B \text{ prop}} \qquad \frac{B \text{ true } [A \text{ true}]}{A \rightarrow B \text{ true}} \end{array}$$

Suma rozłączna indeksowanej rodziny zbiorów

$$\frac{A \text{ set} \quad B(x) \text{ set } [x \in A]}{(\sum x \in A) B(x) \text{ set}}$$

$$\frac{a \in A \quad B(x) \text{ set } [x \in A] \quad p \in B(a)}{\langle a, p \rangle \in (\sum x \in A) B(x)}$$

$$\frac{A = A' \quad B(x) = B'(x) [x \in A]}{(\sum x \in A) B(x) = (\sum x \in A') B'(x)}$$

$$\frac{a = a' \in A \quad b = b' \in B(a)}{\langle a, b \rangle = \langle a', b' \rangle \in (\sum x \in A) B(x) \text{ set}}$$

Suma rozłączna indeksowanej rodziny zbiorów

$$\frac{c \in (\Sigma x \in A)B(x) \quad C(v) \text{ set } [v \in (\Sigma x \in A)B(x)] \quad d(x, y) \in C(\langle x, y \rangle) [x \in A, y \in B(a)]}{\text{split}(c, d) \in C(c)}$$

$$\frac{a \in A \quad b \in B(a) \quad C(v) \text{ set } [v \in (\Sigma x \in A)B(x)] \quad d(x, y) \in C(\langle x, y \rangle) [x \in A, y \in B(a)]}{\text{split}(\langle a, b \rangle, d) = d(a, b) \in C(\langle a, b \rangle)}$$

$$\frac{c = c' \in (\Sigma x \in A)B(x) \quad C(v) \text{ set } [v \in (\Sigma x \in A)B(x)] \quad d(x, y) = d'(x, y) \in C(\langle x, y \rangle) [x \in A, y \in B(a)]}{\text{split}(c, d) = \text{split}(c', d') \in C(c)}$$

Suma rozłączna indeksowanej rodziny zbiorów

$$(\exists x \in A) B(x) \equiv (\Sigma x \in A) B(x)$$

$$\frac{A \text{ set} \quad B(x) \text{ set } [x \in A]}{(\Sigma x \in A) B(x) \text{ set}} \Rightarrow \frac{A \text{ set} \quad B(x) \text{ prop } [x \in A]}{(\exists x \in A) B(x) \text{ prop}}$$

$$\frac{a \in A \quad p \in B(a)}{\langle a, p \rangle \in (\Sigma x \in A) B(x)} \Rightarrow \frac{a \in A \quad B(a) \text{ true}}{(\exists x \in A) B(x) \text{ true}}$$

$$\frac{c \in (\Sigma x \in A) B(x) \quad C(v) \text{ set } [v \in (\Sigma x \in A) B(x)] \quad d(x, y) \in C(\langle x, y \rangle) [x \in A, y \in B(a)]}{\text{split}(c, d) \in C(c)}$$

$$\frac{(\exists x \in A) B(x) \text{ true} \quad \begin{array}{c} \Downarrow \\ C \text{ prop} \end{array} \quad C \text{ true } [x \in A, B(a) \text{ true}]}{C \text{ true}}$$

Suma rozłączna indeksowanej rodziny zbiorów

- Jeżeli x nie występuje w B to

$$A \times B \equiv A \wedge B \equiv (\Sigma x \in A) B$$

$$\frac{A \text{ set} \quad B \text{ set } [x \in A]}{A \times B \text{ set}}$$

$$\frac{a \in A \quad b \in B}{\langle a, b \rangle \in A \times B}$$

$$\frac{A \text{ prop} \quad B \text{ prop } [A \text{ true}]}{A \wedge B \text{ prop}}$$

$$\frac{A \text{ true} \quad B \text{ true}}{A \wedge B \text{ true}}$$

Równość propozycyjna

- Nie mamy odpowiednika atomowej formuły $a = b$
- $a = b \in C$ to sąd, nie propozycja.
- Czego oczekujemy od równości propozycyjnej?
 - ▶ By był zamieszkany wtedy i tylko wtedy gdy formuła atomowa $a = b$ jest prawdziwa.
 - ▶ Abyśmy mogli prowadzić wnioskowanie bazujące na równościach.

Równość propozycyjna

$$\frac{A \text{ set} \quad a \in A \quad b \in A}{[a =_A b] \text{ set}}$$

$$\frac{a \in A}{id(a) \in [a =_A a]}$$

$$\frac{A = A' \quad a = a' \in A \quad b = b' \in A}{[a =_A b] = [a' =_{A'} b']}$$

$$\frac{a = a' \in A}{id(a) = id(a') \in [a =_A a]}$$

Równość propozycyjna

$$\frac{\begin{array}{l} a \in A \quad b \in A \quad c \in [a =_A b] \\ C(x, y, z) \text{ set } [x \in A, y \in A, z \in [x =_A y]] \\ d(x) \in C(x, x, id(x)) [x \in A] \end{array}}{idpeel(c, d) \in C(a, b, c)}$$

$$\frac{\begin{array}{l} a \in A \\ C(x, y, z) \text{ set } [x \in A, y \in A, z \in [x =_A y]] \\ d(x) \in C(x, x, id(x)) [x \in A] \end{array}}{idpeel(id(a), d) = d(a) \in C(a, a, id(a))}$$

- Zbyt silna eliminacja:

$$\frac{[a =_A b] \text{ true}}{a = b \in A}$$