

Logika i teoria typów

Wykład 9

14 grudnia 2020

Zdaniowa logika drugiego rzędu

Motywujący przykład

$$\text{Nat}(a) = \forall R(\forall b(R(b) \rightarrow R(sb)) \rightarrow R(0) \rightarrow R(a))$$

Motywujący przykład

$$\text{Nat}(a) = \forall R(\forall b(R(b) \rightarrow R(sb)) \rightarrow R(0) \rightarrow R(a))$$

Dowód formuły $\text{Nat}(0)$ można zapisać tak: $\lambda R \lambda X \lambda Y. Y$,
gdzie $X : \forall b(R(b) \rightarrow R(sb))$, $Y : R(0)$

Dowodem $\text{Nat}(1)$, czyli $\text{Nat}(s0)$, jest $\lambda R \lambda X \lambda Y. X0(Y)$
(po wytarciu zostaje $\lambda R \lambda X \lambda Y. X(Y)$).

Dowód formuły $\text{Nat}(s^n 0)$ to $\lambda R \lambda X \lambda Y. X^n(Y)$, i tak dalej.

Wycieranie zależności

Formuła

$$\text{Nat}(a) = \forall R(\forall b(R(b) \rightarrow R(sb)) \rightarrow R(0) \rightarrow R(a))$$

pozbawiona treści indywidualowej wygląda tak:

$$\omega = \forall r((r \rightarrow r) \rightarrow r \rightarrow r)$$

To jest polimorficzny typ liczebników Churcha $\mathbf{n} = \lambda fx. f^n x$.
Liczebnik \mathbf{n} jest wytarciem dowodu formuły $\text{Nat}(s^n(0))$.

Zdaniowa logika drugiego rzędu

Składnia:

- ▶ Zmienne zdaniowe p, q, r, \dots są formułami;
- ▶ Stała \perp jest formułą;
- ▶ Jeśli α i β są formułami,
to $\alpha \rightarrow \beta$, $\alpha \vee \beta$ i $\alpha \wedge \beta$ są formułami;
- ▶ Jeśli α jest formułą i p jest zmienną zdaniową,
to $\forall p \alpha$ i $\exists p \alpha$ są formułami.

Brouwer-Heyting-Kolmogorow

- ▶ A construction of $\forall p\varphi(p)$ is a method (function) transforming any proposition \mathbf{P} into a proof of $\varphi(\mathbf{P})$.
- ▶ A construction of $\exists p\varphi(p)$ consists of a proposition \mathbf{P} and a construction of $\varphi(\mathbf{P})$.

Przykłady

$$\forall r((p \rightarrow r) \rightarrow (q \rightarrow r)) \rightarrow q \rightarrow p$$

$$\forall p(q \vee (q \rightarrow p)) \leftrightarrow \neg q \vee q$$

$$\exists p. ((p \rightarrow q) \rightarrow p) \rightarrow r$$

$$\forall p(q \vee p) \rightarrow q \vee \forall p p$$

Naturalna dedukcja (implikacyjno-uniwersalna)

(Ax) $\Gamma, \varphi \vdash \varphi$

$$(\rightarrow I) \frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi}$$

$$(\rightarrow E) \frac{\Gamma \vdash \varphi \rightarrow \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi}$$

$$(\forall I) \frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall p \varphi} \quad (p \notin FVT(\Gamma))$$

$$(\forall E) \frac{\Gamma \vdash \forall p \varphi}{\Gamma \vdash \varphi[p := \vartheta]}$$

Dwa slogany

- ▶ *Full comprehension*: Propositional variables range over all formulas:

$$\exists p (\varphi \leftrightarrow p)$$

$$\forall p \varphi(p) \rightarrow \varphi(\psi)$$

Dwa slogany

- ▶ *Full comprehension*: Propositional variables range over all formulas:

$$\exists p (\varphi \leftrightarrow p)$$

$$\forall p \varphi(p) \rightarrow \varphi(\psi)$$

The meaning of p in $\forall p \varphi$ can be $\forall p \varphi$ itself.

Dwa slogany

- ▶ *Full comprehension*: Propositional variables range over all formulas:

$$\exists p (\varphi \leftrightarrow p)$$

$$\forall p \varphi(p) \rightarrow \varphi(\psi)$$

The meaning of p in $\forall p \varphi$ can be $\forall p \varphi$ itself.

- ▶ *Impredicativity*: The meaning of $\forall p \varphi$ is defined by a reference to a domain to which $\forall p \varphi$ itself belongs.

Semantyka Kripkego

Model $\mathcal{C} = \langle C, \leq, \{D_c \mid c \in C\} \rangle$.

Zbiory $D_c \subseteq \mathbf{P}(C)$ spełniają warunki:

- jeśli $c \leq d$, to $D_c \subseteq D_d$;
- jeśli $d \in X \in D_c$ oraz $d \leq d'$, to $d' \in X$.

Wartościowanie *dopuszczalne* dla φ w stanie c :

$$v(p) \in D_c, \text{ gdy } p \in \text{FV}(\varphi)$$

Wymuszanie

$$c, v \Vdash p \quad \equiv \quad c \in v(p);$$

$$c, v \not\Vdash \perp;$$

$$c, v \Vdash \varphi \vee \psi \quad \equiv \quad c, v \Vdash \varphi \text{ lub } c, v \Vdash \psi;$$

$$c, v \Vdash \varphi \wedge \psi \quad \equiv \quad c, v \Vdash \varphi \text{ oraz } c, v \Vdash \psi;$$

$$c, v \Vdash \varphi \rightarrow \psi \quad \equiv \quad \text{jeśli } c \leq c' \text{ i } c', v \Vdash \varphi, \text{ to } c', v \Vdash \psi;$$

$$c, v \Vdash \exists p \varphi \quad \equiv \quad c, v[p \mapsto X] \Vdash \varphi, \text{ dla pewnego } X \in D_c;$$

$$c, v \Vdash \forall p \varphi \quad \equiv \quad \text{jeśli } c \leq c' \text{ i } X \in D_{c'}, \text{ to } c', v[p \mapsto X] \Vdash \varphi.$$

Kompletny model

Znaczenie formuły φ w stanie c , przy wartościowaniu v :

$$\llbracket \varphi \rrbracket_c = \{c' \mid c \leq c' \text{ oraz } c', v \Vdash \varphi\}$$

Model jest *kompletny*, gdy zawsze $\llbracket \varphi \rrbracket_c \in D_c$

Fakt: Model \mathcal{C} jest kompletny wtw, gdy dla każdego φ :

$$\mathcal{C} \Vdash \exists p(p \leftrightarrow \varphi).$$

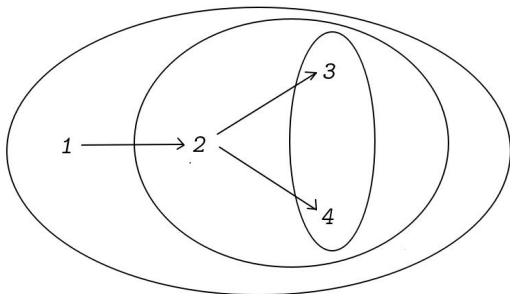
Pełność Kripkego

Piszemy $\Vdash \varphi$, gdy $\mathcal{C} \Vdash \varphi$, dla każdego **kompletnego** modelu \mathcal{C} .

Twierdzenie (Gabbay,Sobolew):

Warunki $\vdash \varphi$ i $\Vdash \varphi$ są równoważne.

$$\not\models \forall p (q \vee \neg p \vee \neg\neg p) \rightarrow q \vee \forall p (\neg p \vee \neg\neg p)$$



D_1 : zbiory jak na rysunku plus zbiór pusty. D_n : dowolne.
 Wartościowanie $v(q) = \{2, 3, 4\}$. Wtedy:

$$1, v \models \forall p (q \vee \neg p \vee \neg\neg p),$$

$$1, v \not\models q \quad \text{oraz} \quad 1, v \not\models \forall p (\neg p \vee \neg\neg p).$$

Semantyka Heytinga

W zupełnej algebrze Heytinga:

$$\llbracket \exists p \varphi \rrbracket_{\zeta} = \sup_{\mathbf{h} \in \mathcal{H}} \llbracket \varphi \rrbracket_{\zeta[p \mapsto \mathbf{h}]}$$

$$\llbracket \forall p \varphi \rrbracket_{\zeta} = \inf_{\mathbf{h} \in \mathcal{H}} \llbracket \varphi \rrbracket_{\zeta[p \mapsto \mathbf{h}]}$$

Semantyka Heytinga

W zupełnej algebrze Heytinga:

$$\llbracket \exists p \varphi \rrbracket_{\zeta} = \sup_{\mathbf{h} \in \mathcal{H}} \llbracket \varphi \rrbracket_{\zeta[p \mapsto \mathbf{h}]}$$

$$\llbracket \forall p \varphi \rrbracket_{\zeta} = \inf_{\mathbf{h} \in \mathcal{H}} \llbracket \varphi \rrbracket_{\zeta[p \mapsto \mathbf{h}]}$$

Fakt: *Jeśli* $\vdash \varphi$, *to* $\models \varphi$

Semantyka Heytinga

W zupełnej algebrze Heytinga:

$$\llbracket \exists p \varphi \rrbracket_{\zeta} = \sup_{\mathbf{h} \in \mathcal{H}} \llbracket \varphi \rrbracket_{\zeta[p \mapsto \mathbf{h}]}$$

$$\llbracket \forall p \varphi \rrbracket_{\zeta} = \inf_{\mathbf{h} \in \mathcal{H}} \llbracket \varphi \rrbracket_{\zeta[p \mapsto \mathbf{h}]}$$

Fakt: *Jeśli* $\vdash \varphi$, *to* $\models \varphi$

Hipoteza: *I na odwrot.*

Nierozstrzygalność

Twierdzenie (Löb, 1976, Gabbay, 1974, Sobolew, 1977)

*Intuicjonistyczna logika zdaniowa drugiego rzędu
jest nierozstrzygalna.*

Nierozstrzygalność

Twierdzenie (Löb, 1976, Gabbay, 1974, Sobolew, 1977)

*Intuicjonistyczna logika zdaniowa drugiego rzędu
jest nierozstrzygalna.*

Uwaga:

(0) Najprostszy znany dowód: Dudenhefner, Rehof, 2018
(zweryfikowany w Coqu)

Nierozstrzygalność

Twierdzenie (Löb, 1976, Gabbay, 1974, Sobolew, 1977)

*Intuicjonistyczna logika zdaniowa drugiego rzędu
jest nierozstrzygalna.*

Uwaga:

(0) Najprostszy znany dowód: Dudenhefner, Rehof, 2018
(zweryfikowany w Coqu)

(1) Wystarczy $\{\forall, \rightarrow\}$ lub $\{\exists, \rightarrow, \vee, \wedge, \neg\}$.

Nierozstrzygalność

Twierdzenie (Löb, 1976, Gabbay, 1974, Sobolew, 1977)

*Intuicjonistyczna logika zdaniowa drugiego rzędu
jest nierozstrzygalna.*

Uwaga:

(0) Najprostszy znany dowód: Dudenhefner, Rehof, 2018
(zweryfikowany w Coqu)

(1) Wystarczy $\{\forall, \rightarrow\}$ lub $\{\exists, \rightarrow, \vee, \wedge, \neg\}$.

(2) Fragment $\{\forall, \exists, \wedge, \neg\}$ jest rozstrzygalny.

Comparison with classical logic

- ▶ Validity/provability in classical propositional logic is complete in the complexity class co-NP .
- ▶ Provability in intuitionistic propositional logic is Pspace -complete.
- ▶ Validity/provability in second-order classical propositional logic (known as the *QBF problem*) is Pspace -complete.
- ▶ Provability in second-order intuitionistic propositional logic is *undecidable*.

Siła wyrazu: suma/alternatywa

$$x \in A \cup B \Leftrightarrow \forall P (A \subseteq P \rightarrow B \subseteq P \rightarrow x \in P),$$

Siła wyrazu: suma/alternatywa

$$x \in A \cup B \Leftrightarrow \forall P (A \subseteq P \rightarrow B \subseteq P \rightarrow x \in P),$$

$$A(x) \vee B(x) \Leftrightarrow \forall P (\forall y (A(y) \rightarrow P(y)) \rightarrow \forall y (B(y) \rightarrow P(y)) \rightarrow P(x)).$$

Siła wyrazu: suma/alternatywa

$$x \in A \cup B \Leftrightarrow \forall P(A \subseteq P \rightarrow B \subseteq P \rightarrow x \in P),$$

$$A(x) \vee B(x) \Leftrightarrow \forall P(\forall y(A(y) \rightarrow P(y)) \rightarrow \forall y(B(y) \rightarrow P(y)) \rightarrow P(x)).$$

$$A \vee B = \forall p((A \rightarrow p) \rightarrow (B \rightarrow p) \rightarrow p).$$

Siła wyrazu: iloczyn/koniunkcja

$$x \in A \cap B \Leftrightarrow \forall P(\forall z(z \in A \rightarrow z \in B \rightarrow z \in P) \rightarrow x \in P).$$

Siła wyrazu: iloczyn/koniunkcja

$$x \in A \cap B \Leftrightarrow \forall P(\forall z(z \in A \rightarrow z \in B \rightarrow z \in P) \rightarrow x \in P).$$

$$A \wedge B = \forall p((A \rightarrow B \rightarrow p) \rightarrow p).$$

Siła wyrazu: zbiór (typ) pusty czyli fałsz

$$x \in \emptyset \Leftrightarrow \forall P(x \in P).$$

Siła wyrazu: zbiór (typ) pusty czyli fałsz

$$x \in \emptyset \Leftrightarrow \forall P(x \in P).$$

$$\perp = \forall p p$$

Siła wyrazu: kwantyfikatory szczegółowy

$$x \in \bigcup_P S_P \Leftrightarrow \forall Q (\forall P (S_P \subseteq Q) \rightarrow x \in Q)$$

Siła wyrazu: kwantyfikator szczegółowy

$$x \in \bigcup_P S_P \Leftrightarrow \forall Q (\forall P (S_P \subseteq Q) \rightarrow x \in Q)$$

$$\exists p \sigma = \forall q (\forall p (\sigma \rightarrow q) \rightarrow q).$$

Poprawność

Dla tak określonych spójników spełnione są zwykłe reguły wnioskowania, w tym:

$$(\exists I) \frac{\Gamma \vdash \varphi[p := v]}{\Gamma \vdash \exists p \varphi}$$

$$(\exists E) \frac{\Gamma \vdash \exists p \varphi \quad \Gamma, \varphi \vdash \psi}{\Gamma \vdash \psi}$$

$$(p \notin FV(\Gamma, \psi))$$

Polimorficzny rachunek lambda

Morał:

Można się ograniczyć do języka z implikacją i kwantyfikatorem ogólnym.

W ten sposób otrzymujemy system **F** Girarda.

Polymorphic types

- ▶ Basic idea: $\forall p. p \rightarrow p$ is a type of a generic identity.

Polymorphic types

- ▶ Basic idea: $\forall p. p \rightarrow p$ is a type of a generic identity.
- ▶ Church style (explicit polymorphism):
 - ▶ Generic identity \mathbf{I} admits a type argument.
 - ▶ The application $\mathbf{I}\tau$ is of type $\tau \rightarrow \tau$.

Polymorphic types

- ▶ Basic idea: $\forall p. p \rightarrow p$ is a type of a generic identity.
- ▶ Church style (explicit polymorphism):
 - ▶ Generic identity I admits a type argument.
 - ▶ The application $I\tau$ is of type $\tau \rightarrow \tau$.
- ▶ Curry style (implicit polymorphism):
 - ▶ Generic identity I has all types $\tau \rightarrow \tau$ at once.

Girard's System **F** (Church style)

$$\Gamma(x : \varphi) \vdash x : \varphi$$

$$\frac{\Gamma(x : \varphi) \vdash M : \psi}{\Gamma \vdash \lambda x : \varphi M : \varphi \rightarrow \psi}$$

$$\frac{\Gamma \vdash M : \varphi \rightarrow \psi \quad \Gamma \vdash N : \varphi}{\Gamma \vdash MN : \psi}$$

$$\frac{\Gamma \vdash M : \varphi}{\Gamma \vdash \Lambda p M : \forall p \varphi} \quad (p \notin \text{FVT}(\Gamma))$$

$$\frac{\Gamma \vdash M : \forall p \varphi}{\Gamma \vdash M\vartheta : \varphi[p := \vartheta]}$$

Girard's System \mathbf{F} (Church style)

$$\Gamma(x : \varphi) \vdash x : \varphi$$

$$\frac{\Gamma(x : \varphi) \vdash M : \psi}{\Gamma \vdash \lambda x : \varphi M : \varphi \rightarrow \psi}$$

$$\frac{\Gamma \vdash M : \varphi \rightarrow \psi \quad \Gamma \vdash N : \varphi}{\Gamma \vdash MN : \psi}$$

$$\frac{\Gamma \vdash M : \varphi}{\Gamma \vdash \Lambda p M : \forall p \varphi} \quad (p \notin \text{FVT}(\Gamma))$$

$$\frac{\Gamma \vdash M : \forall p \varphi}{\Gamma \vdash M\vartheta : \varphi[p := \vartheta]}$$

A few examples

► Term $\Lambda q \lambda x^{\forall p(p \rightarrow p)}.x(q \rightarrow q)(xq)$

has type $\forall q(\forall p(p \rightarrow p) \rightarrow q \rightarrow q)$;

A few examples

▶ Term $\Lambda q \lambda x^{\forall p(p \rightarrow p)}.x(q \rightarrow q)(xq)$

has type $\forall q(\forall p(p \rightarrow p) \rightarrow q \rightarrow q)$;

▶ Term $\mathbf{2} = \Lambda p.\lambda f^{p \rightarrow p} \lambda x^p.f(fx)$

has type $\forall p((p \rightarrow p) \rightarrow (p \rightarrow p))$;

A few examples

- ▶ Term $\Lambda q \lambda x^{\forall p(p \rightarrow p)}.x(q \rightarrow q)(xq)$
has type $\forall q(\forall p(p \rightarrow p) \rightarrow q \rightarrow q)$;
- ▶ Term $\mathbf{2} = \Lambda p.\lambda f^{p \rightarrow p} \lambda x^p.f(fx)$
has type $\forall p((p \rightarrow p) \rightarrow (p \rightarrow p))$;
- ▶ Term $\lambda f^{\forall p(p \rightarrow q \rightarrow p)} \Lambda p \lambda x^p.f(q \rightarrow p)(fpx)$
has type $\forall p(p \rightarrow q \rightarrow p) \rightarrow \forall p(p \rightarrow q \rightarrow q \rightarrow p)$.

Reduction rules

Beta:

- ▶ $(\lambda x:\tau.M)N \Longrightarrow_{\beta} M[x := N]$;
- ▶ $(\Lambda\alpha.M)\tau \Longrightarrow_{\beta} M[\alpha := \tau]$,

Eta:

- ▶ $\lambda x:\tau.Mx \Longrightarrow_{\eta} M \quad (x \notin \text{FV}(M))$;
- ▶ $\Lambda p.Mp \Longrightarrow_{\eta} M \quad (p \notin \text{FVT}(M))$.

Subject reduction:

If $\Gamma \vdash M : \tau$ and $M \rightarrow_{\beta\eta} N$ then $\Gamma \vdash N : \tau$

System F w stylu Churcha

$$\Gamma(x : \varphi) \vdash x : \varphi$$

$$\frac{\Gamma(x : \varphi) \vdash M : \psi}{\Gamma \vdash \lambda x : \varphi M : \varphi \rightarrow \psi}$$

$$\frac{\Gamma \vdash M : \varphi \rightarrow \psi \quad \Gamma \vdash N : \varphi}{\Gamma \vdash MN : \psi}$$

$$\frac{\Gamma \vdash M : \varphi}{\Gamma \vdash \lambda p M : \forall p \varphi} \quad (p \notin \text{FVT}(\Gamma))$$

$$\frac{\Gamma \vdash M : \forall p \varphi}{\Gamma \vdash M\tau : \varphi[p := \tau]}$$

System F w stylu Curry'ego

$$\Gamma(x : \varphi) \vdash x : \varphi$$

$$\frac{\Gamma(x : \varphi) \vdash M : \psi}{\Gamma \vdash \lambda x \ M : \varphi \rightarrow \psi}$$

$$\frac{\Gamma \vdash M : \varphi \rightarrow \psi \quad \Gamma \vdash N : \varphi}{\Gamma \vdash MN : \psi}$$

$$\frac{\Gamma \vdash M : \varphi}{\Gamma \vdash M : \forall p \varphi} \quad (p \notin \text{FVT}(\Gamma))$$

$$\frac{\Gamma \vdash M : \forall p \varphi}{\Gamma \vdash M : \varphi[p := \tau]}$$

Wycieranie typów

$$|x| = x$$

$$|MN| = |M||N|$$

$$|\lambda x:\sigma. M| = \lambda x. |M|$$

$$|\Lambda p. M| = |M|$$

$$|M\tau| = |M|$$

Poprawność redukcji (Curry)

Twierdzenie:

Jeśli $\Gamma \vdash M : \tau$, oraz $M \rightarrow_{\beta} M'$ to $\Gamma \vdash M' : \tau$.

Poprawność redukcji (Curry)

Twierdzenie:

Jeśli $\Gamma \vdash M : \tau$, oraz $M \rightarrow_{\beta} M'$ to $\Gamma \vdash M' : \tau$.

Dowód: Pomijamy. (Nie jest oczywisty.)

Poprawność redukcji (Curry)

Twierdzenie:

Jeśli $\Gamma \vdash M : \tau$, oraz $M \rightarrow_{\beta} M'$ to $\Gamma \vdash M' : \tau$.

Dowód: Pomijamy. (Nie jest oczywisty.)

Uwaga: Powyższe nie zachodzi dla η -redukcji.

Nierozstrzygalność

Twierdzenie 1: (wniosek z tw. Löba) Problem inhabitacji:

Dany typ τ , czy istnieje term zamknięty M typu τ ?

jest nierozstrzygalny dla systemu **F**.

Nierozstrzygalność

Twierdzenie 1: (wniosek z tw. Löba) Problem inhabitacji:

Dany typ τ , czy istnieje term zamknięty M typu τ ?

jest nierozstrzygalny dla systemu **F**.

Twierdzenie 2: (J.B. Wells, 1993) Problem typowości:

Dany term M , czy istnieją takie Γ i τ , że $\Gamma \vdash M : \tau$?

i problem sprawdzenia typu:

Dane są Γ , M i τ . Czy $\Gamma \vdash M : \tau$?

są dla systemu **F** (Curry) nierozstrzygalne.

Nierozstrzygalność

Twierdzenie 1: (wniosek z tw. Löba) Problem inhabitacji:

Dany typ τ , czy istnieje term zamknięty M typu τ ?

jest nierozstrzygalny dla systemu **F**.

Twierdzenie 2: (J.B. Wells, 1993) Problem typowości:

Dany term M , czy istnieją takie Γ i τ , że $\Gamma \vdash M : \tau$?

i problem sprawdzenia typu:

Dane są Γ , M i τ . Czy $\Gamma \vdash M : \tau$?

są dla systemu **F** (Curry) nierozstrzygalne.

Najprostszy dowód: A. Dudenhefner, 2020
(zweryfikowany w Coqu)

Silna normalizacja

Twierdzenie (Jean-Yves Girard, 1972)

System \mathbf{F} ma własność silnej normalizacji.

Silna normalizacja

Twierdzenie (Jean-Yves Girard, 1972)

System F ma własność silnej normalizacji.

Dowód: Metoda „Candidats de reductibilit e”.
Wymaga kwantyfikowania zmiennych, które oznaczaj
rodziny zbiorw. To jest arytmetyka 3. rzdu!



Natural numbers

Numerals:

$$\mathbf{n} = \Lambda p \lambda f^{p \rightarrow p} \lambda x^p. f(f(\dots f(x)))$$

are of type

$$\omega = \forall p((p \rightarrow p) \rightarrow (p \rightarrow p)).$$

Natural numbers

Numerals:

$$\mathbf{n} = \Lambda p \lambda f^{p \rightarrow p} \lambda x^p. f(f(\dots f(x)))$$

are of type

$$\omega = \forall p((p \rightarrow p) \rightarrow (p \rightarrow p)).$$

Examples of representable functions:

- ▶ $\mathbf{Add} = \lambda mn. \Lambda p. \lambda fx. mpf(npfx);$

Natural numbers

Numerals:

$$\mathbf{n} = \Lambda p \lambda f^{p \rightarrow p} \lambda x^p. f(f(\dots f(x)))$$

are of type

$$\omega = \forall p((p \rightarrow p) \rightarrow (p \rightarrow p)).$$

Examples of representable functions:

- ▶ **Add** = $\lambda mn. \Lambda p. \lambda fx. mpf(npfx)$;
- ▶ **Mult** = $\lambda mn. \Lambda p. \lambda fx. mp(npf)x$;

Natural numbers

Numerals:

$$\mathbf{n} = \Lambda p \lambda f^{p \rightarrow p} \lambda x^p. f(f(\dots f(x)))$$

are of type

$$\omega = \forall p((p \rightarrow p) \rightarrow (p \rightarrow p)).$$

Examples of representable functions:

- ▶ **Add** = $\lambda mn. \Lambda p. \lambda fx. mpf(npfx)$;
- ▶ **Mult** = $\lambda mn. \Lambda p. \lambda fx. mp(npf)x$;
- ▶ **Exp** = $\lambda mn. \Lambda p. \lambda fx. m(p \rightarrow p)(np)fx$.

Słowa i listy

Słowa nad alfabetem dwuliterowym:

$$\forall p((p \rightarrow p) \rightarrow (p \rightarrow p) \rightarrow p \rightarrow p)$$

Słowo **baba** to term $\Lambda p \lambda a^{p \rightarrow p} \lambda b^{p \rightarrow p} \lambda x^p. b(a(b(a x)))$

Słowa i listy

Słowa nad alfabetem dwuliterowym:

$$\forall p((p \rightarrow p) \rightarrow (p \rightarrow p) \rightarrow p \rightarrow p)$$

Słowo *baba* to term $\Lambda p \lambda a^{p \rightarrow p} \lambda b^{p \rightarrow p} \lambda x^p. b(a(b(a x)))$

Listy liczb naturalnych:

$$\forall p((\omega \rightarrow p \rightarrow p) \rightarrow p \rightarrow p)$$

$\text{nil} = \Lambda p \lambda c^{\omega \rightarrow p \rightarrow p} \lambda x^p. x$ $n :: \ell = \Lambda p \lambda c^{\omega \rightarrow p \rightarrow p} \lambda x^p. cn(\ell pcx)$

Data types

Empty type: $\perp = \forall p p$

Ex falso quodlibet:

If $\Gamma \vdash M : \perp$ then $\Gamma \vdash M_{\tau} : \tau$.

Cartesian product

$$\sigma \wedge \tau = \forall p((\sigma \rightarrow \tau \rightarrow p) \rightarrow p)$$

Cartesian product

$$\sigma \wedge \tau = \forall p((\sigma \rightarrow \tau \rightarrow p) \rightarrow p)$$

Pairs and projections:

- ▶ Term $\langle M^\sigma, N^\tau \rangle = \Lambda p \lambda y^{\sigma \rightarrow \tau \rightarrow p}. yMN$ has type $\sigma \wedge \tau$.

Cartesian product

$$\sigma \wedge \tau = \forall p((\sigma \rightarrow \tau \rightarrow p) \rightarrow p)$$

Pairs and projections:

- ▶ Term $\langle M^\sigma, N^\tau \rangle = \Lambda p \lambda y^{\sigma \rightarrow \tau \rightarrow p}. y M N$ has type $\sigma \wedge \tau$.
- ▶ Term $\pi_1(M^{\sigma \wedge \tau}) = M_\sigma(\lambda x^\sigma \lambda y^\tau. x)$ has type σ .
- ▶ Term $\pi_2(M^{\sigma \wedge \tau}) = M_\tau(\lambda x^\sigma \lambda y^\tau. y)$ has type τ .

Cartesian product

$$\sigma \wedge \tau = \forall p((\sigma \rightarrow \tau \rightarrow p) \rightarrow p)$$

Pairs and projections:

- ▶ Term $\langle M^\sigma, N^\tau \rangle = \Lambda p \lambda y^{\sigma \rightarrow \tau \rightarrow p}. yMN$ has type $\sigma \wedge \tau$.
- ▶ Term $\pi_1(M^{\sigma \wedge \tau}) = M\sigma(\lambda x^\sigma \lambda y^\tau. x)$ has type σ .
- ▶ Term $\pi_2(M^{\sigma \wedge \tau}) = M\tau(\lambda x^\sigma \lambda y^\tau. y)$ has type τ .

Beta-reduction works: $\pi_1(\langle M, N \rangle) \rightarrow M$, because

$$(\Lambda p \lambda y^{\sigma \rightarrow \tau \rightarrow p}. yMN)\sigma(\lambda x^\sigma \lambda y^\tau. x) \rightarrow (\lambda x^\sigma \lambda y^\tau. x)MN \rightarrow M.$$

Cartesian product

$$\sigma \wedge \tau = \forall p((\sigma \rightarrow \tau \rightarrow p) \rightarrow p)$$

Pairs and projections:

- ▶ Term $\langle M^\sigma, N^\tau \rangle = \Lambda p \lambda y^{\sigma \rightarrow \tau \rightarrow p}. yMN$ has type $\sigma \wedge \tau$.
- ▶ Term $\pi_1(M^{\sigma \wedge \tau}) = M\sigma(\lambda x^\sigma \lambda y^\tau . x)$ has type σ .
- ▶ Term $\pi_2(M^{\sigma \wedge \tau}) = M\tau(\lambda x^\sigma \lambda y^\tau . y)$ has type τ .

Beta-reduction works: $\pi_1(\langle M, N \rangle) \rightarrow M$, because

$$(\Lambda p \lambda y^{\sigma \rightarrow \tau \rightarrow p}. yMN)\sigma(\lambda x^\sigma \lambda y^\tau . x) \rightarrow (\lambda x^\sigma \lambda y^\tau . x)MN \rightarrow M.$$

Eta-reduction does not work:

$$\langle \pi_1(M), \pi_2(M) \rangle = \Lambda p \lambda y. y(\pi_1(M))(\pi_2(M)) \not\rightarrow M$$

Coproduct

$$\sigma \vee \tau = \forall p((\sigma \rightarrow p) \rightarrow (\tau \rightarrow p) \rightarrow p)$$

Embeddings and cases:

- ▶ Term $\mathbf{inl}_{\sigma \vee \tau} M = \Lambda p \lambda u^{\sigma \rightarrow p} \lambda v^{\tau \rightarrow p}. uM$ has type $\sigma \vee \tau$.
- ▶ Term $\mathbf{inr}_{\sigma \vee \tau} M = \Lambda p \lambda u^{\sigma \rightarrow p} \lambda v^{\tau \rightarrow p}. vM$ has type $\sigma \vee \tau$.
- ▶ And $\mathbf{case } M \mathbf{ of } [x]P^\vartheta, [y]Q^\vartheta = M\vartheta(\lambda x^\sigma P)(\lambda y^\tau Q) : \vartheta$.

Coproduct

$$\sigma \vee \tau = \forall p((\sigma \rightarrow p) \rightarrow (\tau \rightarrow p) \rightarrow p)$$

Embeddings and cases:

- ▶ Term $\mathbf{inl}_{\sigma \vee \tau} M = \Lambda p \lambda u^{\sigma \rightarrow p} \lambda v^{\tau \rightarrow p}. uM$ has type $\sigma \vee \tau$.
- ▶ Term $\mathbf{inr}_{\sigma \vee \tau} M = \Lambda p \lambda u^{\sigma \rightarrow p} \lambda v^{\tau \rightarrow p}. vM$ has type $\sigma \vee \tau$.
- ▶ And $\mathbf{case } M \mathbf{ of } [x]P^\vartheta, [y]Q^\vartheta = M\vartheta(\lambda x^\sigma P)(\lambda y^\tau Q) : \vartheta$.

Beta-reduction:

$$\mathbf{case } \mathbf{in}_1(P) \mathbf{ of } [x]M, [y]N = (\mathbf{in}_1(P))\vartheta(\lambda x^\sigma M)(\lambda y^\tau N) = (\Lambda p \lambda u v. uP)\vartheta(\lambda x^\sigma M)(\lambda y^\tau N) \rightarrow (\lambda x^\sigma M)P \rightarrow M[x := P].$$

Abstract (encapsulated) type

Idea: Type $\exists p \sigma(p)$ is of shape $\sigma(\tau)$, where τ is unspecified (abstract) and not accessible to the user.

Example: An abstract pds object is of type

$$\exists p (p \times p \times (p \rightarrow p) \times (p \rightarrow \omega) \times (\omega \times p \rightarrow p))$$

Existential types

Type assignment:

$$\text{(pack)} \frac{\Gamma \vdash M : \sigma[p := \tau]}{\Gamma \vdash [M, \tau]_{\exists p. \sigma} : \exists p. \sigma}$$

$$\text{(unpack)} \frac{\Gamma \vdash M : \exists p. \sigma \quad \Gamma(x : \sigma) \vdash N : \rho \quad (p \notin FV(\Gamma, \rho))}{\Gamma \vdash \text{unpack } M \text{ as } [x, p] \text{ in } N : \rho}$$

Beta reduction:

$$\text{unpack } [M, \tau]_{\exists p. \sigma} \text{ as } [x, p] \text{ in } N \longrightarrow_{\beta} N[p := \tau][x := M].$$

Implementing existential quantifier

Definitions:

- ▶ $\exists p \sigma = \forall q (\forall p (\sigma \rightarrow q) \rightarrow q)$.

Implementing existential quantifier

Definitions:

- ▶ $\exists p \sigma = \forall q (\forall p (\sigma \rightarrow q) \rightarrow q)$.
- ▶ $[M^{\sigma[p:=\tau]}, \tau]_{\exists p. \sigma} = \Lambda q \lambda z^{\forall p (\sigma \rightarrow q)}. z \tau M$;

Implementing existential quantifier

Definitions:

- ▶ $\exists p \sigma = \forall q (\forall p (\sigma \rightarrow q) \rightarrow q)$.
- ▶ $[M^{\sigma[p:=\tau]}, \tau]_{\exists p. \sigma} = \Lambda q \lambda z^{\forall p (\sigma \rightarrow q)}. z \tau M$;
- ▶ **unpack** $M^{\exists p. \sigma}$ **as** $[x, p]$ **in** $N^{\rho} = M_{\rho} (\Lambda p \lambda x^{\sigma}. N)$.

Implementing existential quantifier

Definitions:

- ▶ $\exists p \sigma = \forall q (\forall p (\sigma \rightarrow q) \rightarrow q)$.
- ▶ $[M^{\sigma[p:=\tau]}, \tau]_{\exists p. \sigma} = \Lambda q \lambda z^{\forall p (\sigma \rightarrow q)}. z \tau M$;
- ▶ **unpack** $M^{\exists p. \sigma}$ **as** $[x, p]$ **in** $N^{\rho} = M_{\rho} (\Lambda p \lambda x^{\sigma}. N)$.

Correctness:

unpack $[M, \tau]$ **as** $[x, p]$ **in** N^{ρ}

Implementing existential quantifier

Definitions:

- ▶ $\exists p \sigma = \forall q (\forall p (\sigma \rightarrow q) \rightarrow q)$.
- ▶ $[M^{\sigma[p:=\tau]}, \tau]_{\exists p. \sigma} = \Lambda q \lambda z^{\forall p (\sigma \rightarrow q)}. z \tau M$;
- ▶ **unpack** $M^{\exists p. \sigma}$ **as** $[x, p]$ **in** $N^{\rho} = M_{\rho} (\Lambda p \lambda x^{\sigma}. N)$.

Correctness:

unpack $[M, \tau]$ **as** $[x, p]$ **in** N^{ρ}
 $= (\Lambda q \lambda z^{\forall p (\sigma \rightarrow q)}. z \tau M)_{\rho} (\Lambda p \lambda x^{\sigma}. N)$

Implementing existential quantifier

Definitions:

- ▶ $\exists p \sigma = \forall q (\forall p (\sigma \rightarrow q) \rightarrow q)$.
- ▶ $[M^{\sigma[p:=\tau]}, \tau]_{\exists p. \sigma} = \Lambda q \lambda z^{\forall p (\sigma \rightarrow q)}. z \tau M$;
- ▶ **unpack** $M^{\exists p. \sigma}$ **as** $[x, p]$ **in** $N^{\rho} = M_{\rho} (\Lambda p \lambda x^{\sigma}. N)$.

Correctness:

$$\begin{aligned} \text{unpack } [M, \tau] \text{ as } [x, p] \text{ in } N^{\rho} \\ &= (\Lambda q \lambda z^{\forall p (\sigma \rightarrow q)}. z \tau M)_{\rho} (\Lambda p \lambda x^{\sigma}. N) \\ &\rightarrow (\lambda z^{\forall p (\sigma \rightarrow \rho)}. z \tau M) (\Lambda p \lambda x^{\sigma}. N) \end{aligned}$$

Implementing existential quantifier

Definitions:

- ▶ $\exists p \sigma = \forall q (\forall p (\sigma \rightarrow q) \rightarrow q)$.
- ▶ $[M^{\sigma[p:=\tau]}, \tau]_{\exists p. \sigma} = \Lambda q \lambda z^{\forall p (\sigma \rightarrow q)}. z \tau M$;
- ▶ **unpack** $M^{\exists p. \sigma}$ **as** $[x, p]$ **in** $N^{\rho} = M \rho (\Lambda p \lambda x^{\sigma}. N)$.

Correctness:

$$\begin{aligned} \text{unpack } [M, \tau] \text{ as } [x, p] \text{ in } N^{\rho} & \\ &= (\Lambda q \lambda z^{\forall p (\sigma \rightarrow q)}. z \tau M) \rho (\Lambda p \lambda x^{\sigma}. N) \\ &\rightarrow (\lambda z^{\forall p (\sigma \rightarrow \rho)}. z \tau M) (\Lambda p \lambda x^{\sigma}. N) \\ &\rightarrow (\Lambda p \lambda x^{\sigma}. N) \tau M \end{aligned}$$

Implementing existential quantifier

Definitions:

- ▶ $\exists p \sigma = \forall q (\forall p (\sigma \rightarrow q) \rightarrow q)$.
- ▶ $[M^{\sigma[p:=\tau]}, \tau]_{\exists p. \sigma} = \Lambda q \lambda z^{\forall p (\sigma \rightarrow q)}. z \tau M$;
- ▶ **unpack** $M^{\exists p. \sigma}$ **as** $[x, p]$ **in** $N^{\rho} = M \rho (\Lambda p \lambda x^{\sigma}. N)$.

Correctness:

$$\begin{aligned} \text{unpack } [M, \tau] \text{ as } [x, p] \text{ in } N^{\rho} & \\ &= (\Lambda q \lambda z^{\forall p (\sigma \rightarrow q)}. z \tau M) \rho (\Lambda p \lambda x^{\sigma}. N) \\ &\rightarrow (\lambda z^{\forall p (\sigma \rightarrow \rho)}. z \tau M) (\Lambda p \lambda x^{\sigma}. N) \\ &\rightarrow (\Lambda p \lambda x^{\sigma}. N) \tau M \\ &\rightarrow N[p := \tau][x := M]. \end{aligned}$$