Logika i teoria typów

Wykład 3

19 października 2022

Tydzień temu: rozszerzony rachunek λ

$$\frac{\Gamma \vdash M : \bot}{\Gamma \vdash M[\sigma] : \sigma}$$

Write also $\varepsilon_\sigma(M)$ for $M[\sigma]$.

$$\frac{\Gamma \vdash M : \alpha \quad \Gamma \vdash N : \beta}{\Gamma \vdash \langle M, N\rangle : \alpha \wedge \beta} \qquad \frac{\Gamma \vdash M : \alpha_1 \wedge \alpha_2}{\Gamma \vdash M\{i\} : \alpha_i}$$

Koniunkcja to iloczyn kartezjański. Fałsz to typ pusty.

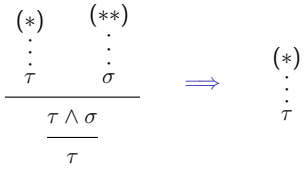
Alternatywa to suma prosta

$$\frac{\Gamma \vdash M : \alpha_i}{\Gamma \vdash \mathsf{in}_i(M) : \alpha_1 \vee \alpha_2}$$

$$\frac{\Gamma \vdash M : \alpha \vee \beta \quad \Gamma, u{:}\alpha \vdash R : \tau \quad \Gamma, v{:}\beta \vdash Q : \tau}{\Gamma \vdash M[u.R, v.Q] : \tau}$$

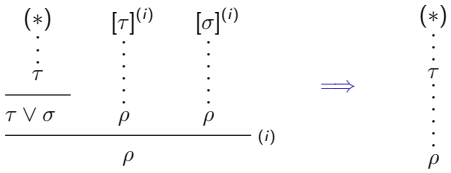
Write also case M of $[u]R$ or $[v]Q$ for $M[u.R, v.Q]$
inl, inr for in_1, in_2.

Normalizacja dowodu: koniunkcja



$\langle M^\tau, N^\sigma\rangle\{1\} \implies M : \tau$

Normalizacja dowodu: alternatywa



$\mathsf{in}_1(P^\tau)[x^\tau.\, M^\rho, y^\sigma.\, N^\rho] \implies M[x := P] : \rho$

Beta-redukcja

Beta-redeks to eliminacja spójnika zastosowana bezpośrednio po jego wprowadzeniu. *Beta-redukcja* to „upraszcza":

$(\to)$ $(\lambda x^\tau\, M^\sigma)N^\tau \implies M[x := N] : \sigma.$

$(\wedge)$ $\langle M^\tau, N^\sigma\rangle\{1\} \implies M : \tau, \qquad \langle M^\tau, N^\sigma\rangle\{2\} \implies N : \sigma.$

$(\vee)$ $\mathsf{in}_1(P^\tau)[x^\tau.\, M^\rho, y^\sigma.\, N^\rho] \implies M[x := P] : \rho$
$\mathsf{in}_2(Q^\sigma)[x^\tau.\, M^\rho, y^\sigma.\, N^\rho] \implies N[y := Q] : \rho.$

Curry-Howard Isomorphism

A propositional formula α is an intuitionistic theorem iff there exists a closed term of type α (type α is nonempty).

"Propositions-as-Types"

- Formula = type = specification.
- Proof = program = implementation.
- Proof normalization = computation.

Przykłady

$x : \neg p \vdash \lambda y^{p\wedge q}.\, x(y\{1\}) : \neg(p \wedge q)$;

$x : p \to \neg q, y : \neg p \to \neg q \vdash \lambda z^q.\, y(\lambda u^p.\, xuz)z : \neg q$;

$x : \neg p \wedge \neg q \vdash \lambda y^{p\vee q}.\, y[u^p.\, x\{1\}u, v^q.\, x\{2\}v] : \neg(p \vee q)$;

$x : \neg(p \vee q) \vdash \langle \lambda y^p.\, x(\mathsf{in}_1(y)), \lambda z^q.\, x(\mathsf{in}_2(z))\rangle : \neg p \wedge \neg q$;

$x : \neg p \vee \neg q \vdash \lambda y^{p\wedge q}.\, x[u^{\neg p}.\, u(y\{1\}), v^{\neg q}.\, v(y\{2\})] : \neg(p \wedge q)$.

Przykłady

$\vdash \lambda x^{\alpha}\, \mathsf{in}_1(x) : \alpha \to \alpha \vee \beta$;

$\vdash \lambda x^{\beta}\, \mathsf{in}_2(x) : \beta \to \alpha \vee \beta$;

$\lambda y^{\alpha\to\gamma} z^{\beta\to\gamma} x^{\alpha\vee\beta}.\, x[u.\, yu, v.\, zv] :$
$(\alpha \to \gamma) \to (\beta \to \gamma) \to (\alpha \vee \beta) \to \gamma$

$y{:}\alpha\to\gamma, z{:}\beta\to\gamma \vdash \lambda x^{\alpha\vee\beta}.\, x[u.\, yu, v.\, zv] : (\alpha\vee\beta) \to \gamma$;

$\vdash \lambda x^{\alpha\wedge\beta}\, x\{1\} : \alpha \wedge \beta \to \alpha$;

$\vdash \lambda x^{\alpha\wedge\beta}\, x\{2\} : \alpha \wedge \beta \to \beta$;

$y : \gamma \to \alpha, z : \gamma \to \beta \vdash \lambda x^{\gamma}\, \langle yx, zx\rangle : \gamma \to \alpha \wedge \beta$.

Hilbert-style proofs

Many axioms – few rules

Assume a (recursively enumerable) set of axioms.

Definition: A *proof* of ψ from Γ is a sequence of formulas $\psi_1, \psi_2, \ldots, \psi_n$, such that $\psi_n = \psi$ and, for all $i = 1, \ldots, n$,

- either ψ_i is an axiom, or $\psi_i \in \Gamma$, or
- there are $j, \ell < i$ such that $\psi_j = \psi_\ell \to \psi_i$ (ψ_i is obtained from ψ_j and ψ_ℓ using *modus ponens*).

Possible axiom schemes for implicational IPC

All formulas of the following form are axioms:

(K) $\varphi \to \psi \to \varphi$;
(S) $(\varphi \to \psi \to \vartheta) \to (\varphi \to \psi) \to \varphi \to \vartheta$.

Add the scheme $((\varphi \to \psi) \to \varphi) \to \varphi$ for classical logic.

Example proof

1. $(\varphi \to (\psi \to \varphi) \to \varphi) \to (\varphi \to \psi \to \varphi) \to \varphi \to \varphi$;
2. $\varphi \to (\psi \to \varphi) \to \varphi$;
3. $(\varphi \to \psi \to \varphi) \to \varphi \to \varphi$ (detach 2 from 1);
4. $\varphi \to \psi \to \varphi$;
5. $\varphi \to \varphi$ (detach 4 from 3).

Curry-Howard: proofs are typed combinators.

A fundamental property: the deduction theorem

Theorem: $\Gamma, \varphi \vdash_H \psi$ iff $\Gamma \vdash_H \varphi \to \psi$.

Proof: ($\Leftarrow$) Immediate use of modus ponens.

($\Rightarrow$) Induction wrt proofs of ψ from Γ, φ.

Case 1: If $\psi \in \Gamma$, or ψ is an axiom, we detach ψ from the axiom $\psi \to \varphi \to \psi$.

Case 1$^+$: One can do the same if φ not used in proof.

Case 2: If $\psi = \varphi$ we use the example proof of $\varphi \to \varphi$.

Case 3: If ψ obtained from α and $\alpha \to \psi$, apply induction. There are proofs of $\varphi \to \alpha$ and $\varphi \to (\alpha \to \psi)$ from Γ. Detach those from axiom (S).

Deduction theorem $\Leftrightarrow$ combinatory abstraction

$\Gamma, x : \varphi \vdash M : \psi$ iff $\Gamma \vdash \lambda^* x.\, M : \varphi \to \psi$.

Case 1: If $\psi \in \Gamma$, or ψ is an axiom, we detach ψ from the axiom $\psi \to \varphi \to \psi$.

If M is a constant or a variable in Γ, then $\lambda^* x.\, M = \mathsf{K}M$.

Case 1$^+$: One can do the same if $x \notin FV(M)$.

Case 2: If $\psi = \varphi$ we use the example proof of $\varphi \to \varphi$.

If $M = x$, then $\lambda^* x.\, x = \mathsf{I} = \mathsf{SKK}$.

Deduction theorem $\Leftrightarrow$ combinatory abstraction

$\Gamma, x : \varphi \vdash M : \psi$ iff $\Gamma \vdash \lambda^* x.\, M : \varphi \to \psi$.

Case 3: If ψ obtained from α and $\alpha \to \psi$, apply induction. There are proofs of $\varphi \to \alpha$ and $\varphi \to (\alpha \to \psi)$ from Γ. Detach those from axiom (S).

If $M = P^{\alpha\to\psi} Q^{\alpha}$, then $\lambda^* x.\, M = \mathsf{S}(\lambda^* x.\, P)(\lambda^* x.\, Q)$.

Jak trudny jest rachunek zdań (dygresja)

Problem wnioskowania w rachunku zdań:
Czy dana formuła (implikacyjnego) rachunku zdań jest konsekwencją danych schematów aksjomatów?

Twierdzenie (Linial, Post, 1949):
Problem wnioskowania w rachunku zdań jest nierozstrzygalny.

Automat dwulicznikowy: $\mathcal{A} = \langle Q, q_0, q_f, \delta \rangle$

- Q – skończony zbiór stanów;
- q_0 – stan początkowy;
- q_f – stan końcowy;
- δ – funkcja przejścia, $\mathrm{Dom}(f) = Q - \{q_f\}$;
- $\delta(q)$ jest jednej z postaci ($i = 1, 2$):
  - „$c_i := c_i + 1$; goto p";
  - „$c_i := c_i - 1$; goto p";
  - „if $c_i = 0$ then goto p else goto r".

Automat dwulicznikowy

Konfiguracja automatu: trójka $\mathcal{C} = \langle q, n_1, n_2 \rangle$.

Zmiana konfiguracji: $\mathcal{C} \to \mathcal{C}'$, gdzie $\mathcal{C}'$ zależy od $\delta(q)$:

- Jeśli $\delta(q) =$ „$c_1 := c_1 + 1$; goto p",
  to $\mathcal{C}' = \langle p, n_1 + 1, n_2 \rangle$;
- Jeśli $\delta(q) =$ „$c_1 := c_1 - 1$; goto p", oraz $n_1 > 0$,
  to $\mathcal{C}' = \langle p, n_1 - 1, n_2 \rangle$ (wpp. nieokreślone);
- Jeśli $\delta(q) =$ „if $c_1 = 0$ then goto p else goto r", to
  - $\mathcal{C}' = \langle p, n_1, n_2 \rangle$, w przypadku $n_1 = 0$;
  - $\mathcal{C}' = \langle r, n_1, n_2 \rangle$, w przypadku $n_1 \neq 0$.
- Dla instrukcji dotyczących c_2 – analogicznie.

Automat dwulicznikowy

Automat *akceptuje* konfigurację $\mathcal{C}$, jeżeli istnieje ciąg przejść

$$\mathcal{C} \to \mathcal{C}_1 \to \mathcal{C}_2 \to \cdots \to \mathcal{C}_n$$

do konfiguracji ze stanem końcowym.

Problem stopu:
Dany automat $\mathcal{A}$ i konfiguracja $\mathcal{C}$, czy $\mathcal{A}$ akceptuje $\mathcal{C}$?

Twierdzenie 1: *Problem stopu jest nierozstrzygalny.*

Twierdzenie 2: *Istnieje taki automat $\mathcal{A}$, że problem stopu jest nierozstrzygalny przy ustalonym $\mathcal{A}$.*

Definicje pomocnicze

Below we assume $n > 1$:

$$\tau_n(\alpha) = \alpha^{n-1} \to \alpha = \underbrace{\alpha \to \cdots \to \alpha}_{n-1} \to \alpha.$$
$$\sigma(\alpha) = (\alpha \to \alpha) \to \alpha$$

Fakt: *Typy $\tau_n(\alpha)$ dla $n \neq m$ są „nieunifikowalne":*
dla każdych α, β i każdego podstawienia S zachodzi

$$S(\tau_n(\alpha)) \neq S(\tau_m(\beta)).$$

Fakt: *Typ $\sigma(\alpha)$* jest nieunifikowalny ze wszystkimi typami postaci $\tau_n(\beta)$: $S(\sigma(\alpha)) \neq S(\tau_n(\beta))$, dla wszystkich S, α, β.

Konwencja: Stany automatu to liczby od 4 do F, przy czym $q_0 = 4$ i $q_f = F$.

Kodowanie konfiguracji

Kodem liczby n jest dowolna formuła postaci:

$$\tau_2(\alpha_1) \to \cdots \to \tau_2(\alpha_n) \to \tau_3(\beta),$$

czyli postaci

$$(\alpha_1 \to \alpha_1) \to \cdots \to (\alpha_n \to \alpha_n) \to (\beta \to \beta \to \beta).$$

Kodem stanu q jest dowolna formuła postaci $\tau_q(\alpha)$.

Kodem $\mathcal{C} = \langle q, n_1, n_2 \rangle$ jest dowolna formuła postaci

$$kod(q) \to kod(n_1) \to kod(n_2) \to \sigma(\xi).$$

Rachunek zdań określony przez automat

Dla każdej instrukcji $\delta(q)$ przyjmujemy jeden lub dwa schematy aksjomatów:

- Dla $\delta(q) =$ „$c_1 := c_1 + 1$; goto p":

$[\tau_p(\alpha) \to (\tau_2(\varepsilon) \to \beta) \to \gamma \to \sigma(\xi)] \to [\tau_q(\alpha) \to \beta \to \gamma \to \sigma(\xi)]$;

- Dla $\delta(q) =$ „$c_1 := c_1 - 1$; goto p":

$[\tau_p(\alpha) \to \beta \to \gamma \to \sigma(\xi)] \to [\tau_q(\alpha) \to (\tau_2(\varepsilon) \to \beta) \to \gamma \to \sigma(\xi)]$;

- Dla $\delta(q) =$ „if $c_1 = 0$ then goto p else goto r":

$[\tau_p(\alpha) \to \tau_3(\beta) \to \gamma \to \sigma(\xi)] \to [\tau_q(\alpha) \to \tau_3(\beta) \to \gamma \to \sigma(\xi)]$,

$[\tau_r(\alpha) \to (\tau_2(\varepsilon) \to \beta) \to \gamma \to \sigma(\xi)] \to [\tau_q(\alpha) \to (\tau_2(\varepsilon) \to \beta) \to \gamma \to \sigma(\xi)]$.

- Dla instrukcji dotyczących c_2 – analogicznie.
- Na koniec jeszcze schemat $\tau_F(\alpha) \to \beta \to \gamma \to \sigma(\xi)$.

Główny lemat

Dla dowolnej konfiguracji $\mathcal{C} = \langle q, n_1, n_2 \rangle$ i dowolnej formuły φ, która jest kodem $\mathcal{C}$, następujące warunki są równoważne:

- *Automat $\mathcal{A}$ akceptuje konfigurację $\mathcal{C}$;*
- *Formuła φ ma dowód w rachunku zdań automatu $\mathcal{A}$.*

Dowód: Indukcja:

($\Downarrow$) ze względu na długość obliczenia;
($\Uparrow$) ze względu na długość dowodu.

Dowód ($\Downarrow$) – krok bazowy

Jeśli $\mathcal{A}$ akceptuje konfigurację $\mathcal{C} = \langle q, n_1, n_2\rangle$, oraz ψ_1 i ψ_2 są kodami n_1 i n_2, to $\tau_q(\alpha) \to \psi_1 \to \psi_2 \to \sigma(\xi)$ ma dowód.

Dowód jest ze względu na długość obliczenia.
Krok bazowy to przypadek konfiguracji końcowej $\mathcal{C} = \langle F, n_1, n_2\rangle$.

Na to mamy aksjomat $\tau_F(\alpha) \to \beta \to \gamma \to \sigma(\xi)$.

Dowód ($\Downarrow$) – przykładowy krok indukcji

Jeśli $\mathcal{A}$ akceptuje konfigurację $\mathcal{C} = \langle q, n_1, n_2\rangle$, oraz ψ_1 i ψ_2 są kodami n_1 i n_2, to $\tau_q(\alpha) \to \psi_1 \to \psi_2 \to \sigma(\xi)$ ma dowód.

Niech $\mathcal{C} \to \mathcal{C}' = \langle p, n_1, n_2 + 1\rangle$ wskutek wykonania instrukcji $\delta(q) =$ „$c_2 := c_2 + 1$; goto p". Wtedy mamy taki aksjomat:

$[\tau_p(\alpha)\to\psi_1\to(\tau_2(\varepsilon)\to\psi_2)\to\sigma(\xi)] \to [\tau_q(\alpha)\to\psi_1\to\psi_2\to\sigma(\xi)]$.

Formuła $\tau_p(\alpha)\to\psi_1\to(\tau_2(\varepsilon)\to\psi_2)\to\sigma(\xi)$ jest kodem $\mathcal{C}'$, więc ma dowód z założenia indukcyjnego.

Zatem $\tau_q(\alpha) \to \psi_1 \to \psi_2 \to \sigma(\xi)$ otrzymamy przez odrywanie.

Dowód ($\Uparrow$) – krok bazowy

Jeśli ψ_1 i ψ_2 są kodami n_1 i n_2, oraz $\tau_q(\alpha) \to \psi_1 \to \psi_2 \to \sigma(\xi)$ ma dowód, to $\mathcal{A}$ akceptuje konfigurację $\mathcal{C} = \langle q, n_1, n_2\rangle$.

Dowód jest ze względu na długość dowodu,
zatem krok bazowy, to przypadek, gdy formuła $\tau_q(\alpha) \to \psi_1 \to \psi_2 \to \sigma(\xi)$ jest aksjomatem.

Kod konfiguracji jest aksjomatem tylko wtedy,
gdy jest to konfiguracja końcowa...

Na pewno?

Trzeba pokazać, że inne aksjomaty nie są unifikowalne z żadnym kodem konfiguracji.

Dowód ($\Uparrow$) – przykładowy krok indukcji

Jeśli ψ_1 i ψ_2 są kodami n_1 i n_2, oraz $\tau_q(\alpha) \to \psi_1 \to \psi_2 \to \sigma(\xi)$ ma dowód, to $\mathcal{A}$ akceptuje konfigurację $\mathcal{C} = \langle q, n_1, n_2\rangle$.

Dowód formuły $\tau_q(\alpha) \to \psi_1 \to \psi_2 \to \sigma(\xi)$ polega na zastosowaniu modus ponens do aksjomatu, np.

$[\tau_p(\alpha)\to\tau_3(\beta)\to\gamma\to\sigma(\xi)] \to [\tau_q(\alpha)\to\tau_3(\beta)\to\gamma\to\sigma(\xi)]$,

Wtedy „oczywiście" $\psi_1 = \tau_3(\beta)$ i $\psi_2 = \gamma$, czyli test na zero jest poprawny. Zatem $\mathcal{C} \to \mathcal{C}' = \langle p, n_1, n_2\rangle$. Formuła $[\tau_p(\alpha)\to\tau_3(\beta)\to\gamma\to\sigma(\xi)]$ ma dowód (krótszy) i jest kodem $\mathcal{C}'$. Z założenia indukcyjnego automat akceptuje $\mathcal{C}'$ a więc i $\mathcal{C}$.

Uwaga: Z powodu nieunifikowalności nie da się uzyskać kodu konfiguracji przez kilkakrotne odrywanie od aksjomatu (4 przypadki do sprawdzenia ręcznie).

Various implicational axioms...

B : $(\beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma$;

B′ : $(\alpha \to \beta) \to (\beta \to \gamma) \to \alpha \to \gamma$;

C : $(\alpha \to \beta \to \gamma) \to \beta \to \alpha \to \gamma$;

W : $(\alpha \to \alpha \to \beta) \to \alpha \to \beta$.

... and logics:

SBCI – relevant logic (every assumption must be used, cf. λI).

BCK – affine logic (every assumption used at most once).

BCI – BCI-logic (every assumption used exactly once).

BB'IW – ticket entailment.

(Logika intuicjonistyczna to **SK**, klasyczna to **SK**+Peirce.)

Złożoność logik implikacyjnych

SBCI – relevant logic (every assumption must be used, cf. λI).
– 2-Exptime-zupełna (Schmitz, 2015).

BCK – affine logic (every assumption used at most once).
– NP-zupełna (łatwe)

BCI – BCI-logic (every assumption used exactly once).
– NP-zupełna (Kanowicz, 1994, Buszkowski 2008).

BB'IW – ticket entailment
– rozstrzygalne (Padovani; Bimbó/Dunn, 2011).

SK – intuicjonistyczna
– Pspace-zupełna (Statman, 1979)

SK+Peirce – klasyczna
– co-NP-zupełna (Stålmarck, 1989).