Logika i teoria typów

Wykład 6

16 listopada 2021

Tydzień temu: gry i twierdzenie o pełności

- Dla dowolnego osądu $\Gamma \vdash \tau$, albo $\exists$ros ma strategię albo $\forall$frodyta ma strategię.
- Strategia $\exists$rosa definiuje pewien dowód (lambda-term).
- Strategia $\forall$frodyty definiuje pewien kontrmodel.
- Zatem albo $\Gamma \vdash \tau$ albo $\Gamma \nvdash \tau$.

Jakie termy są strategiami $\exists$rosa?

W przypadku implikacyjnym możliwe strategie są takie:

- zmienne;
- $M[y := xN]$, gdzie M, N – strategie;
- $\lambda x : \beta.\, M$, gdzie M – strategia.

To są dokładnie postaci normalne beta:

- Abstrakcje: $\lambda x : \beta.\, M$;
- Eliminatory: $xN_1 \dots N_k$.

Typ eliminatora jest „końcówką" typu zmiennej czołowej.

Pełność i „normalizacja" w jednym

(Fragment implikacyjny)

Twierdzenie: *Dla dowolnego osądu $\Gamma \vdash \tau$:*

– albo istnieje normalny inhabitant $\Gamma \vdash M : \tau$,

– albo istnieje kontrmodel, tj. $\Gamma \nvdash \tau$.

Wniosek: *Jeśli osąd ma dowód, to ma dowód normalny.*

Inaczej:
Jeśli jakikolwiek term M ma typ τ w otoczeniu Γ,
to istnieje term tego typu w postaci normalnej.

Postaci normalne (fragment implikacyjny)

Konstruktory: $\lambda x : \alpha.\, N$

Dobre eliminatory: $xN_1 \dots N_k$

Poszukiwanie dowodu normalnego:

To prove τ, use an assumption with suffix τ.
Otherwise proof is a constructor.

Algorytm Ben-Yellesa

To answer $\Gamma \vdash ? : \alpha$, apply one of the following tactics:

- For $\alpha = \beta \to \gamma$, ask $\Gamma \cup \{x : \beta\} \vdash ? : \gamma$ (fresh x).

  (Solution $M = \lambda x{:}\beta.\, N^{\gamma}$.)

- Find $x : \beta_1 \to \dots \to \beta_k \to \alpha$ in Γ,
  then ask $\Gamma \vdash ? : \beta_i$, for all i. Success if $k = 0$.

  (Solution $M = xN_1^{\beta_1} \dots N_k^{\beta_k}$.)

W drugim przypadku można żądać aby α była zmienną.

Przypadek ogólny

Dowody normalne (definicja I)

Strategie $\exists$rosa w przypadku ogólnym:

- zmienne;
- $M[y := xN]$;
- $M[y := x\{i\}]$, gdzie $i = 1, 2$;
- $x[y.\, M; z.\, N]$;
- $x[\tau]$.
- $\lambda x : \beta.\, M$;
- $\langle M, N \rangle$;
- $\mathrm{in}_i(M)$;

Dowody normalne (definicja II)

- *Introductions:*
  - $\lambda x : \alpha.\, N^{\beta} : \alpha \to \beta$,
  - $\langle N_1^{\alpha}, N_2^{\beta} \rangle : \alpha \wedge \beta$,
  - $\mathrm{in}_i(N^{\alpha}) : \alpha \vee \beta$;
- *Proper eliminators (P):*
  - $x^{\alpha} : \alpha$,
  - $P^{\alpha \to \beta} N^{\alpha} : \beta$,
  - $P^{\alpha_1 \wedge \alpha_2}\{i\} : \alpha_i$;
- *Improper eliminators:*
  $P^{\perp}[\varphi] : \varphi$,
  $P^{\alpha \vee \gamma}[u^{\alpha}.\, M^{\beta}; v^{\gamma}.\, N^{\beta}] : \beta$

Ważne: Eliminator normalny ma zmienną czołową.

Poprawność (do domu)

Te dwie definicje są równoważne.
Dowód wykorzystuje taki techniczny lemat:

Lemat: *Jeśli M jest dowodem normalnym (w sensie definicji II), a P jest eliminatorem właściwym, to także $M[x := P]$ jest dowodem normalnym.*

Pożytek z normalizacji: niesprzeczność

Wniosek: *Intuicjonistyczny rachunek zdań jest niesprzeczny: formuła $\perp$ nie ma dowodu.*

Dowód: Wtedy istniałby dowód normalny, czyli taki term M w postaci normalnej, że $\varnothing \vdash M : \perp$. Jak on może wyglądać?

- Nie może być konstruktorem, bo nie ma konstruktora dla $\perp$ (nie ma reguły $(W\perp)$);
- Nie może być właściwym eliminatorem postaci $xE_1 \ldots E_m$, bo nie ma zmiennych.
- Zatem nie ma też eliminatorów postaci $P[\ldots]$, bo do tego potrzebny właściwy eliminator P.

Dowody normalne (definicja III?)

To powinny być postaci normalne
ze względu na odpowiednie redukcje.
Beta-redukcje?

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

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

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

Beta-redukcje to za mało

Term w postaci beta-normalnej może np. wyglądać tak:

$$z^{\tau \vee \sigma}[x^{\tau}.\, P^{\alpha \to \beta};\, y^{\sigma}.\, Q^{\alpha \to \beta}]^{\alpha \to \beta} M^{\alpha} : \beta$$

Jak nad tym zapanować?

Wprowadzić dodatkowe redukcje porządkujące (*permutacje*).

Permutacje

Kłopotliwa sytuacja:

Brzydka eliminacja, a potem znowu eliminacja, czyli:

$M[x.\, Q;\, y.\, R]E \qquad M[\sigma]E.$

Eliminator E nie ma dostępu do „wnętrza" termu.

Permutacje:

$M[x.\, Q;\, y.\, R]E \;\Rightarrow M[x.\, QE;\, y.\, RE]$

$M[\sigma]E^{\tau} \;\Rightarrow M[\tau]$

gdzie E oznacza N, $\{i\}$, $[\psi]$, $[u.\, N_1;\, v.\, N_2]$.

Permutacje dla $\perp$

- $M^{\perp}[\perp][\varphi] \;\Rightarrow\; M^{\perp}[\varphi]$;
- $M^{\perp}[\varphi \to \psi]N \;\Rightarrow\; M^{\perp}[\psi]$;
- $M^{\perp}[\varphi_1 \wedge \varphi_2]\{i\} \;\Rightarrow\; M^{\perp}[\varphi_i]$;
- $M^{\perp}[\sigma \vee \tau][u.\, R^{\rho};\, v.\, S^{\rho}] \;\Rightarrow\; M^{\perp}[\rho]$.

Schemat: $M[\sigma]E^{\tau} \Rightarrow M[\tau]$

Permutacje dla alternatywy

- $M[x.\, P;\, y.\, Q][\varphi] \;\Rightarrow\; M[x.\, P[\varphi];\, y.\, Q[\varphi]]$;
- $M[x.\, P;\, y.\, Q]N \;\Rightarrow\; M[x.\, PN;\, y.\, QN]$;
- $M[x.\, P;\, y.\, Q]\{i\} \;\Rightarrow\; M[x.\, P\{i\};\, y.\, Q]\{i\}]$;
- $M[x.\, P;\, y.\, Q][u.\, R;\, v.\, S] \;\Rightarrow$
  $M[x.\, P[u.\, R;\, v.\, S];\, y.\, Q[u.\, R;\, v.\, S]]$.

Schemat: $M[x.\, P^{\tau};\, y.\, Q^{\tau}]E^{\sigma} \Rightarrow M[x.P^{\tau}E^{\sigma}, y.Q^{\tau}E^{\sigma}]$

Dobra wiadomość

Postaci normalne ze względu na beta i permutacje, to dokładnie dowody normalne:

Konstruktory: $\lambda x : \alpha.\, N, \quad \langle N_1, N_2 \rangle, \quad \mathrm{in}_i(N);$

Ładne eliminatory: $x, \quad PN, \quad P\{i\};$

Brzydkie eliminatory: $P^{\bot}[\varphi] : \varphi, \quad P^{\alpha\vee\gamma}[u^{\alpha}.\, M^{\beta}; v^{\gamma}.\, N^{\beta}] : \beta$

Eliminatory właściwe

- *Proper eliminators (P):*
  - $x^{\alpha} : \alpha$,
  - $P^{\alpha\to\beta} N^{\alpha} : \beta$,
  - $P^{\alpha_1\wedge\alpha_2}\{i\} : \alpha_i$;

Taki eliminator właściwy ma zawsze postać $xE_1E_2 \ldots E_n$, gdzie E_i to albo termy, albo rzutowania $\{1\}$, $\{2\}$.

Typ eliminatora właściwego jest "końcówką" typu zmiennej.

Suffixes $S(\tau)$ of a type τ

- $\tau \in S(\tau)$;
- If $\alpha \to \beta \in S(\tau)$ then $\beta \in S(\tau)$;
- If $\alpha \wedge \beta \in S(\tau)$ then $\alpha, \beta \in S(\tau)$;

Lemma:

If $\Gamma, x : \tau \vdash P : \sigma$, and P is a proper eliminator beginning with x, then $\sigma \in S(\tau)$.

Przykład niewłaściwej eliminacji

W otoczeniu:

$x : (q \wedge p) \vee r, \quad y : r \to (r \to p) \vee q, \quad z : q \to p$

term $x[u.\, u\{2\};\ v.\, yv[w_1.\, w_1 v; w_2.\, zw_2]]$ ma typ p.

Typy zmiennych: $u : q \wedge p, \quad v : r, \quad w_1 : r \to p, \quad w_2 : q.$

Example application:

Disjunction property: *If $\Gamma \vdash \alpha \vee \beta$, and $\vee$ does not occur in Γ, then $\Gamma \vdash \alpha$ or $\Gamma \vdash \beta$.*

Proof: Show that if $\Gamma \vdash M : \alpha \vee \beta$, and M is normal, then either $\Gamma \vdash \bot$ or M is a constructor **in**.

Types of proper eliminators contain no $\vee$, so M is either an introduction (OK) or an improper eliminator.

If $M = M'[\alpha \vee \beta]$, then $\Gamma \vdash M' : \bot$ (OK).

Otherwise $M = M'[\ldots]$ where M' is a proper eliminator of a disjunction type – excluded as above.

Wnioski:

1. Prawo wyłączonego środka $p \vee \neg p$ nie ma dowodu.
2. Prawo De Morgana $\neg(p \wedge q) \to \neg p \vee \neg q$ też nie ma.

Separation theorem (Mordchaj Wajsberg, 1938)

Twierdzenie: *Spójniki intuicjonistyczne nie są wzajemnie definiowalne.*

Disjunction: *There is no α without $\vee$ such that $p \vee q \leftrightarrow \alpha$.*

Proof:

If $\alpha \vdash p \vee q$ then $\alpha \vdash p$ or $\alpha \vdash q$. But $p \vee q \nvdash p$ and $p \vee q \nvdash q$.

Separation theorem

Implication: *There is no α without $\to$ such that $p \to q \vdash \alpha$.*

Proof: Induction.

If $\alpha = \beta \wedge \gamma$ then $p \to q \vdash \beta$.

If $\alpha = \beta \vee \gamma$ then either $p \to q \vdash \beta$ or $p \to q \vdash \gamma$.

Separation theorem

Conjunction: *If* $\alpha \vdash p \wedge q$, *and* $\wedge$ *is not in* α, *then* $\alpha \vdash \bot$.

Proof: If $\alpha = \beta \vee \gamma$ then $\beta \vdash \bot$ and $\gamma \vdash \bot$ by induction.

Assume that $\alpha = \beta \to \gamma$.

Then also $\beta \to \bot \vdash p$.
Normal inhabitant must be $(x^{\beta \to \bot} M^{\beta})[p]$.
Hence $\beta \to \bot \vdash \bot$.

On the other hand, also $\gamma \vdash p \wedge q$, so $\gamma \vdash \bot$, by induction.

Thus we have $\beta \to \gamma \vdash \beta \to \bot$,
and therefore $\beta \to \gamma \vdash \bot$.

Poszukiwanie dowodu normalnego:

Dowód formuły τ może być:

- Konstruktorem (o ile τ nie jest atomem);
- Eliminatorem o zmiennej czołowej typu σ:
  - Właściwym, gdy τ jest „sufiksem" typu σ;
  - Niewłaściwym, gdy σ ma „sufiks" postaci $\alpha \vee \beta$ lub $\bot$.

Drobne usprawnienie: Można stosować drugi przypadek tylko wtedy, gdy τ jest atomem lub alternatywą.

To jest *algorytm Wajsberga-Bezaniszwilego*.

Drobne usprawnienie: długa postać normalna

- Term $N : \alpha \to \beta$ można zamienić na $\lambda x^{\alpha}.\, Nx$.
- Term $N : \alpha \wedge \beta$ można zamienić na $\langle N\{1\}, N\{2\}\rangle$.

Uwaga: To nie jest całkiem oczywiste, bo taka zamiana może „popsuć" postać normalną np., gdy

$$N = y^{\gamma \vee \delta}[u^{\gamma}.\, \lambda z.\, P; v^{\delta} \lambda z.\, Q] : \alpha \to \beta$$

Algorytm Wajsberga

To answer $\Gamma \vdash \alpha$?, apply one of the following tactics:

- For $\alpha = \beta \to \gamma$, ask if $\Gamma, \beta \vdash \gamma$?
- For $\alpha = \beta \wedge \gamma$, ask if $\Gamma \vdash \beta$ AND $\Gamma \vdash \gamma$.
- For $\alpha = \alpha_1 \vee \alpha_2$, GUESS $i \in \{1, 2\}$, and ask if $\Gamma \vdash \alpha_i$.
- GUESS $\beta_1 \to \cdots \to \beta_k \to A$ from Γ, where $A = \alpha$ or $A = \bot$; then ask FOR ALL i, if $\Gamma \vdash \beta_i$. (Sukces, gdy $k = 0$.)
- GUESS $\beta_1 \to \cdots \to \beta_k \to \beta \vee \gamma$ from Γ, and ask FOR ALL i, if $\Gamma \vdash \beta_i$. Also ask if $\Gamma, \beta \vdash \alpha$ AND $\Gamma, \gamma \vdash \alpha$.
- GUESS $\beta_1 \to \cdots \to \beta_k \to \beta \wedge \gamma$ from Γ, and ask FOR ALL i, if $\Gamma \vdash \beta_i$. Also ask if $\Gamma, \beta, \gamma \vdash \alpha$.

Alternacja

Algorytm Wajsberga rozgałęzia się:

- Egzystencjalnie (niedeterministycznie), bo można wybrać różne taktyki i różne zmienne;
- Uniwersalnie (rekurencyjnie), bo trzeba niezależnie rozwiązywać różne zadania.

To się nazywa *alternacja*.

Terminacja

W każdej gałęzi obliczenia:

- Używa się tylko podformuł zadania początkowego.
- Otoczenie Γ nigdy nie maleje.
- Można ignorować powtarzające się założenia (utożsamiać zmienne tego samego typu).

Morał: Po wielomianowej liczbie kroków zadanie musi się powtórzyć. Złożoność tego algorytmu jest wielomianowa w czasie alternującym.

Zatem problem decyzyjny dla intuicjonistycznego rachunku zdań jest w klasie PSPACE.

Algorytm Wajsberga: operacje

Zadanie ma postać $\Gamma \vdash \alpha$.
Jakie operacje na nim wykonujemy?

- For $\alpha = \beta \to \gamma$, ask if $\Gamma, \beta \vdash \gamma$?
  Zmiana celu plus dodanie nowego założenia. Wymaga celu odpowiedniej postaci.
- For $\alpha = \beta \wedge \gamma$, ask if $\Gamma \vdash \beta$ AND $\Gamma \vdash \gamma$.
  Rekurencyjne wywołanie dla dwóch nowych celów (rozgałęzienie uniwersalne). Wymaga odp. celu.
- For $\alpha = \alpha_1 \vee \alpha_2$, GUESS $i \in \{1, 2\}$, and ask if $\Gamma \vdash \alpha_i$.
  Egzystencjalny wybór ze zmianą celu.

Algorytm Wajsberga: operacje

Zadanie ma postać $\Gamma \vdash \alpha$.
Jakie operacje na nim wykonujemy?

- GUESS $\beta_1 \to \cdots \to \beta_k \to A$ from Γ, where $A = \alpha$ or $A = \bot$; then ask if $\Gamma \vdash \beta_i$, FOR ALL i.
  Wybór egzystencjalny (niedeterministyczny), potem rozgałęzienie rekurencyjne (uniwersalne) ze zmianą celów. Wymaga dostępności odp. założenia.
- GUESS $\beta_1 \to \cdots \to \beta_k \to \beta \vee \gamma$ from Γ, and ask if $\Gamma \vdash \beta_i$, FOR ALL i. Also ask if $\Gamma, \beta \vdash \alpha$ AND $\Gamma, \gamma \vdash \alpha$.
  Rozgałęzienie egzystencjalne i uniwersalne jw. W niektórych gałęziach dodanie założenia.
- GUESS $\beta_1 \to \cdots \to \beta_k \to \beta \wedge \gamma$ from Γ, and ask FOR ALL i, if $\Gamma \vdash \beta_i$. Also ask if $\Gamma, \beta, \gamma \vdash \alpha$.
  Podobnie.

Algorytm Wajsberga: operacje

Zadanie ma postać $\Gamma \vdash \alpha$.
Jakie operacje na nim wykonujemy?

- Dodanie założenia;
- Zmiana celu;
- Sprawdzenie dostępności założenia;
- Sprawdzenie postaci celu;
- Rozgałęzienie egzystencjalne;
- Rozgałęzienie uniwersalne.

Algorytm Wajsberga jako automat

- Cel dowodowy to stan maszyny;
- Otoczenie Γ to zawartość pamięci.
- Operacje:
  - Dodanie założenia = podniesienie flagi;
  - Zmiana celu = zmiana stanu;
  - Sprawdzenie dostępności założenia = sprawdzenie flagi;
  - Sprawdzenie postaci celu = sprawdzenie stanu maszyny;
  - Rozgałęzienie egzystencjalne;
  - Rozgałęzienie uniwersalne.

Uwaga: (1) nie można opuścić flagi (usunąć danej z pamięci);
(2) nie można sprawdzić, że flaga jest opuszczona.

Monotonic automata

Monotonic automaton: $\mathcal{M} = \langle Q, R, f, \mathcal{I} \rangle$,

- Q is a finite set of states, $f \in Q$ the final state.
- R is a finite set of registers;
- $\mathcal{I}$ is a finite set of instructions of the form:
  (1) q : `check` S_1; `set` S_2; `jmp` p,
  (2) q : `jmp` p_1 `and` p_2,
  where $q, p, p_1, p_2 \in Q$ and $S_1, S_2 \subseteq R$.

Monotonic automata

Configuration: $\langle q, S \rangle$, where $q \in Q$ and $S \subseteq R$.

Transitions:

- for $I = q$: `check` S_1; `set` S_2; `jmp` p:
  $\langle q, S \rangle \to_I \langle p, S \cup S_2 \rangle$, provided $S_1 \subseteq S$;
  If $\langle p, S \cup S_2 \rangle$ accepting then $\langle q, S \rangle$ accepting
- for $I = q$: `jmp` p_1 `and` p_2:
  $\langle q, S \rangle \to_I \langle p_1, S \rangle$ **and** $\langle q, S \rangle \to_I \langle p_2, S \rangle$
  If $\langle p_1, S \rangle$ and $\langle p_2, S \rangle$ accepting then so is $\langle q, S \rangle$.
- konfiguracje $\langle f, S \rangle$ są akceptujące.

Proofs into programs

Twierdzenie: *Problem dowodzenia w intuicjonistycznym rachunku zdań sprowadza się (w pamięci logarytmicznej) do problemu stopu dla automatów monotonicznych.*

Dowód: Dla danej formuły φ konstruujemy automat:

- Stany = możliwe cele (podformuły formuły φ) i jeden stan końcowy;
- Rejestry = możliwe założenia, czyli też podformuły.
- Instrukcje = jak w algorytmie Wajsberga.

Trzeba jeszcze dodać przejścia do stanu końcowego z pozycji wygrywających ∃rosa.

Monotonic automata

Lemma

The halting problem for monotonic automata
Is a given configuration accepting?
is PSPACE*-complete.*

Cześć łatwa: Problem stopu rozstrzygamy w pamięci wielomianowej, bo zbiór rejestrów tylko rośnie.

Część trudna: Monotonic encoding of an ATM

Alternating time $p(n)$.

Registers: $stan_s^t$, lit_{ia}^t, poz_i^t,

Initial configuration:

$C_0 = \langle loop_1^0, \{lit_{1a_1}^0, ..., lit_{na_n}^0, lit_{n+1B}^0, ..., lit_{p(n)B}^0, stan_0^0, poz_1^0\} \rangle$.

represents initial ID of TM for input $w = a_1...a_n$.

Later:

$C = \langle loop_1^t, S \rangle$ encodes the first t steps of TM computation,

e.g., $lit_{6a}^5 \in S$ iff after 5 steps, the 6th cell contains a.

Ciąg dalszy nastąpi...